

# SPARK

## SPARK - Remote Firing System

Bachelor thesis - June 2020

University of South-Eastern Norway  
Faculty of Technology, Natural Sciences  
and Maritime Sciences

Martin Brunæs

Stian Jørgensen

Sirajuddin Asjad

Bjørn-Ivar Bekkevold

Kristian Alfheim

Anne Synnøve Brendøy

## CENSORED VERSION

This is an unclassified version of the original document (612 pages). All confidential and sensitive information has been censored. The redacted information is only available to certain individuals.

## **Abstract**

A Remote Firing System (RFS) is a tool employed by military and law-enforcement personnel who seek to initiate the firing of ordnance remotely from a safe distance. Today, the solution consists of a receiver, which is capable of initiating the necessary energy transfer to facilitate the initiation, and a transmitter, which controls the receiver. As a result of the encumbrance and impracticality of such a solution in the context of a modern and inherently complex setting, there has emerged a desire to innovate this system in order to develop a RFS that is not only of minimal weight and high durability, but is also easily and securely operated under duress. Furthermore, in an effort to better aid the operator of such a system in making well-informed decisions in the spur of the moment, a software solution to that effect is discussed. The objective of this thesis is to provide a brief overview of what the modern environment is like in terms of warfare, how a RFS fits into this environment, what the problems are with today's RFS and our identified objectives to remedy these problems, and how a multi-disciplinary engineering team has attacked the problem. Ultimately, the end-goal is to provide a feasible proof of concept, and supply some of the theoretical knowledge needed for further expansion of the system to eventually fully realize the solution.

## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
<b>2</b>	<b>Project Management</b>	<b>17</b>
2.1	Project Plan . . . . .	19
2.2	Project Model . . . . .	20
2.3	Project Environment . . . . .	21
2.4	Project Tools . . . . .	25
2.4.1	Jira Project Management . . . . .	25
2.4.2	Communication Platform . . . . .	26
2.4.3	Time Tracking: Clockify . . . . .	27
2.4.4	File Storage: Google Drive . . . . .	28
2.5	Risk Assessment . . . . .	29
<b>3</b>	<b>The Problem</b>	<b>31</b>
3.1	Problem Definition and Objectives . . . . .	33
3.2	Requirements Specification . . . . .	38
3.3	Validation Process . . . . .	39
3.4	Verification Process . . . . .	39
<b>4</b>	<b>The Proposal</b>	<b>40</b>
4.1	Software Design . . . . .	43
4.1.1	Evolution of Software Proposal . . . . .	43
4.1.2	Proposed Conceptual Software Model . . . . .	46
4.1.2.1	Functionality of the Proposed Software Solution . . . . .	47
4.1.2.2	Software Abstractions from Sequence Diagrams . . . . .	48
4.1.2.3	Software Architectural Model . . . . .	51
4.1.3	Elaborating Software Functionality in Preparation for Implementation	52
4.1.3.1	Expanding the Base Use Case Model . . . . .	53
4.1.3.2	Expanding Excerpts of the Proposed Software Architectural Model . . . . .	65
4.1.3.3	Example of the Section of the Final Software Architectural Model . . . . .	72
4.1.4	Paving the Way Towards Implementation . . . . .	73
4.1.4.1	Ad Hoc Networking . . . . .	74
4.1.4.2	Fault Tolerance . . . . .	83
4.1.5	Military Communication and Information Systems . . . . .	85
4.1.5.1	Cloud Computing in Military Applications . . . . .	85

4.1.5.2	Implementing Cloud Computing in Military Operations . . .	87
4.1.5.3	NATO Generic Vehicle Architecture (NGVA) . . . . .	88
4.2	Physical Design . . . . .	93
4.2.1	Design Challenges . . . . .	93
4.2.2	Casing Design . . . . .	94
4.2.2.1	Material and Production Methods . . . . .	95
4.2.2.2	IP, Atex and Test Standards . . . . .	96
4.2.2.3	Testing of Material and Casing . . . . .	99
4.3	Electrical Design . . . . .	101
4.3.1	The Electrical System . . . . .	101
4.3.2	Plasma Ignition System . . . . .	104
4.3.3	Step Up . . . . .	104
4.3.4	Electrical safety barriers . . . . .	108
4.3.5	Ethernet . . . . .	113
4.3.6	Hardware Paring and Authentication . . . . .	115
4.3.7	Battery status system . . . . .	117
<b>5</b>	<b>Implementation</b>	<b>121</b>
5.1	Software Components . . . . .	124
5.2	Software Application Development . . . . .	125
5.2.1	Qt Framework . . . . .	125
5.2.2	Initial Development Phase . . . . .	126
5.2.3	Networking Protocol . . . . .	129
5.2.4	Verification of Network Communication . . . . .	137
5.2.5	Near Field Communication (NFC) . . . . .	139
5.2.6	Software Safety Barriers . . . . .	144
5.2.7	Software Security Concerns . . . . .	154
5.2.8	IEC Functional Safety and IEC 61508 . . . . .	155
5.2.9	Android Application, Implementation Overview . . . . .	157
5.2.10	Communication Controls . . . . .	159
5.2.10.1	TacticalOperationsHandler . . . . .	159
5.2.10.2	ConnectionManager . . . . .	160
5.2.10.3	PacketManager . . . . .	160
5.2.11	Device Data . . . . .	161
5.2.11.1	ReceiverHandler & SenderHandler . . . . .	161
5.2.12	IP Interface . . . . .	162
5.2.13	Functional Behavior . . . . .	165
5.2.14	Map & Waypoint Functionality . . . . .	169
5.3	Electrical Design . . . . .	173



5.3.1	MCU . . . . .	173
5.3.2	Communication . . . . .	175
5.3.2.1	NFC Module . . . . .	175
5.3.2.2	Ethernet . . . . .	177
5.3.2.3	Wi-Fi . . . . .	179
5.3.3	Plasma Igniter System . . . . .	182
5.3.3.1	350V Step Up . . . . .	182
5.3.3.2	355V Capacitive Discharge Circuit . . . . .	185
5.3.4	Sensors . . . . .	193
5.3.4.1	Accelerometer . . . . .	193
5.3.4.2	GPS . . . . .	194
5.3.4.3	Battery capacity estimation . . . . .	196
5.3.4.4	Watchdog . . . . .	198
5.3.5	Power Supply . . . . .	199
5.3.6	PCB Design . . . . .	202
5.3.7	System integration . . . . .	206
5.4	Final Mechanical Implementation . . . . .	207
5.4.0.1	Nanotube enhanced laminates . . . . .	207
5.4.0.2	Casing V3.1, 3D Printed . . . . .	208
5.4.0.3	Casing V3.2, Nano Laminate Assembly . . . . .	208
5.4.0.4	Moisture Inside the Casing . . . . .	211
5.5	Early Stages of Prototyping . . . . .	212
5.5.1	Prototype V1 . . . . .	212
5.5.2	Prototype V2 . . . . .	214
5.5.2.1	Casing V2 . . . . .	220
<b>6</b>	<b>Conclusions</b>	<b>223</b>
6.0.1	Thesis Summary . . . . .	224
6.0.2	Evaluation . . . . .	225
6.0.3	Future Work . . . . .	226
6.0.4	Closing Thoughts . . . . .	227
<b>7</b>	<b>References</b>	<b>228</b>
<b>8</b>	<b>Appendices</b>	<b>238</b>
8.1	List of Requirements . . . . .	239
8.2	FMEA V7.0 . . . . .	319
8.3	Test reports . . . . .	333
8.4	Economy attachments . . . . .	352
8.5	Prototype 1.0 code . . . . .	360

8.6	Prototype 2.0 code . . . . .	362
8.7	Software Architectural Model . . . . .	366
8.8	Receiver IP Simulation . . . . .	369
8.9	Sequence Diagram Pairing . . . . .	372
8.10	Sequence Diagram Tactical Operations . . . . .	373
8.11	Sequence Diagram Heartbeat . . . . .	374
8.12	Material Test Report (ILSS and Tensile Properties) . . . . .	375
8.13	Material Test Results of Assembly . . . . .	432
8.14	Bill of Material . . . . .	434
8.15	Gantt diagram . . . . .	437
8.16	SPARK PCB Schematics . . . . .	440

## List of Figures

1	Our Jira board . . . . .	25
2	Our Slack channels . . . . .	26
3	Examples of time registration report in Clockify . . . . .	27
4	Folder Structure . . . . .	28
5	Stakeholder Context Diagram . . . . .	35
6	System Context Diagram . . . . .	36
7	Receiver Context Diagram . . . . .	37
8	Structure of Requirements . . . . .	38
9	Current radio communication system . . . . .	43
10	Proposed communication system . . . . .	44
11	Block Diagram - Peer to Peer Network . . . . .	45
12	Initial Use Case Model . . . . .	47
13	Sequence Diagrams for Figure 12 . . . . .	50
14	Configure Context Use Case Diagram . . . . .	53
15	Collect Data Use Case Diagram . . . . .	54
16	Authorize Action Use Case Diagram . . . . .	55
17	Archive Data Use Case Diagram . . . . .	56
18	SD: Send Heartbeat (receiver-side) . . . . .	57
19	SD: Send Component Status . . . . .	58
20	SD: Create Team . . . . .	59
21	SD: Plan Mission . . . . .	60
22	SD: Set Access . . . . .	61
23	Expansion of Make Decision Use Case . . . . .	65
24	Arm Receiver Sequence Diagram . . . . .	67
25	Disarm Receiver Sequence Diagram . . . . .	67
26	Detonate Receiver Sequence Diagram . . . . .	68
27	Set Timer Sequence Diagram . . . . .	68
28	Abort Detonation Sequence Diagram . . . . .	69
29	Forward Control Sequence Diagram . . . . .	69
30	Request Control Sequence Diagram . . . . .	70
31	Warn Operator Sequence Diagram . . . . .	70
32	Create Waypoint Sequence Diagram . . . . .	71
33	Name Waypoint Sequence Diagram . . . . .	71
34	Component Based Architecture of Make Decision . . . . .	72
35	Many-to-Many Communication . . . . .	75
36	Multi-hop Routing Scheme . . . . .	76
37	P2P Group in Wifi Direct . . . . .	77

38	Bluetooth Scatternet (inspired by [126]) . . . . .	79
39	Transmitter / Receiver Network . . . . .	83
40	Receiver Failover . . . . .	84
41	Cloud Computing deployment models [37] . . . . .	87
42	NGVA integration compatibility levels [43] . . . . .	89
43	RFS integrated into a NGVA data network . . . . .	90
44	Abstract illustration of the NGVA data network . . . . .	90
45	Publish-subscribe pattern amongst two nodes . . . . .	91
46	IP standards. Picture from Nema Enclosures . . . . .	98
47	Tinius Olsen Company Super-L 300 . . . . .	100
48	PETG- and nano casing after test . . . . .	100
49	The Main Electrical System . . . . .	102
50	Booster in series . . . . .	104
51	Step up circuit . . . . .	105
52	Step up circuit simulation . . . . .	106
53	LT3757 from Analog Devices . . . . .	107
54	General discharge safety barriers . . . . .	108
55	Digital safety logic from MCU to discharge circuit. . . . .	110
56	Silicon controlled rectifier (SCR) thyristor [62] . . . . .	111
57	Stability verification using D-flip flops . . . . .	112
58	Typical ENC28J60-based interface [39] . . . . .	114
59	Wiring diagram for NFC system[80] . . . . .	115
60	Discharge curve of the CR123 from GP Batteries [66] . . . . .	118
61	LTSpice simulation of LM3914 main circuitry . . . . .	119
62	Software components (sender) . . . . .	124
63	Software components (receiver) . . . . .	124
64	Main view (sketch) . . . . .	126
65	Receiver overview (sketch) . . . . .	127
66	Detailed overview (sketch) . . . . .	127
67	Map view (sketch) . . . . .	128
68	Protocol Frame . . . . .	129
69	Header Fields . . . . .	130
70	Protocol Pairing Handshake . . . . .	133
71	Heartbeat Command . . . . .	134
72	Detonation Sequence . . . . .	134
73	Forward Heartbeat and Forward Waypoint . . . . .	135
74	Component Status and Situational Context . . . . .	135
75	Handover Transaction . . . . .	136
76	Packet Analysis in Wireshark . . . . .	137

77	Example of deadlock between two processes . . . . .	144
78	Firing Safety System . . . . .	145
79	Safety barriers for arming . . . . .	146
80	Capacitor charging subcomponent . . . . .	147
81	Safety barriers for detonation . . . . .	148
82	Watchdog timer operation . . . . .	149
83	Watchdog timeout operation . . . . .	150
84	Using watchdog timer to check the battery level . . . . .	151
85	Watchdog Control Register (custom-designed) . . . . .	152
86	Watchdog Control Configuration (custom-designed) . . . . .	152
87	Watchdog timer process . . . . .	153
88	Overview Android Application . . . . .	157
89	Controller Static Overview . . . . .	159
90	Handling of receiver and transmitter Data . . . . .	161
91	Server Static Overview . . . . .	163
92	IP Communication Object Interaction . . . . .	164
93	Flowcharts, Pairing Process and Incoming Data . . . . .	166
94	Flowchart Heartbeat Functionality . . . . .	167
95	Flowchart Tactical Operations . . . . .	168
96	Esri, OSM and Mapbox alternative map views . . . . .	170
97	Waypoint States Overlay . . . . .	171
98	SPI communication [53] . . . . .	173
99	Sequence diagram of a TCP socket initiation with W5500 [137] . . . . .	177
100	LLS between MCU and Wi-Fi subsystem . . . . .	180
101	Transistor based bidirectional LLS . . . . .	181
102	LTspice simulations with $5\mu F$ . . . . .	184
103	Discharge circuit that is implemented in the SPARK PCB . . . . .	185
104	Breakdown voltage variation with temperature . . . . .	189
105	Gate trigger current variation with temperature[125] . . . . .	190
106	Discharge curve with $20k\Omega$ in series with $R_{DS(ON)} = 17\Omega$ . . . . .	191
107	Gate trigger voltage variation with temperature[125] . . . . .	192
108	Schematic for battery measurement for SPARK PCB . . . . .	197
109	Watchdog Implementation . . . . .	198
110	Component Placement . . . . .	201
111	Component Placement . . . . .	202
112	Component Placement . . . . .	203
113	Component Placement . . . . .	204
114	Component Placement . . . . .	205
115	Online whiteboard for system integration . . . . .	206

116	Laminate in the making . . . . .	207
117	Laminate ready for hardening. . . . .	207
118	PETG Assembly Open . . . . .	208
119	PETG Assembly Closed . . . . .	208
120	Test assembly with plywood . . . . .	209
121	Plywood Assembly . . . . .	209
122	Laminate cutting . . . . .	209
123	Machine program . . . . .	209
124	Nano casing for testing in Super-L 300 . . . . .	210
125	Custom gaskets . . . . .	210
126	Post-curing assembly . . . . .	210
127	Prototype 1.0 physical architecture . . . . .	212
128	Prototype V1 . . . . .	213
129	Prototype V2 physical architecture . . . . .	214
130	Main view (landscape) . . . . .	215
131	Navigation sidebar (landscape) . . . . .	216
132	Receiver overview (landscape) . . . . .	216
133	Map view (landscape) . . . . .	217
134	Tactical Map View . . . . .	218
135	Bottom of device . . . . .	221
136	Lid with NFC space . . . . .	221
137	Battery- and board compartment partition . . . . .	221
138	Storage lid . . . . .	221
139	Closed assembly . . . . .	222
140	Open assembly . . . . .	222
141	Preliminary budget . . . . .	353
142	Purchase list for prototype v2 . . . . .	354
143	Pairing Static Process . . . . .	372
144	Tactical Operation Commands . . . . .	373
145	Processing a Heartbeat . . . . .	374
146	Test of Nano Assembly . . . . .	432
147	Test og PETG Assembly . . . . .	433

## List of Tables

2	Scrum roles . . . . .	20
3	Group members . . . . .	23
4	Project role description . . . . .	24
5	Step up design vs step up component . . . . .	107
6	Switch normal operation . . . . .	109
7	Different states based on different inputs . . . . .	109
8	CR123 battery specifications from different manufacturers . . . . .	117
9	NFC module supported standards . . . . .	175
10	Component criteria for 350V step up . . . . .	182
11	Component criteria for input step up . . . . .	183
12	System states and its corresponding signals . . . . .	187
13	Current Range derived from components data-sheets . . . . .	199
14	Component criteria for 3.3V supply . . . . .	200

## Glossary

EOD	Explosive Ordnance Disposal; a field in the military concerned with the disposal of explosive ordnance.
Receiver	The remote ignition device placed upon the explosives; the box unit itself.
Transmitter	The android phone used to transmit orders to the receiver.
IED	Improvised Explosive Device; a bomb constructed and deployed in ways other than in conventional military action.
Operator	A person who intentionally interacts with the system
Unauthorized user	The person(s) that is not intended to use the system such as, but not limited to; civilians and other military groups.
MCU	Microcontroller unit
NFC	Near field communication



# SPARK

## 1 Introduction

Contents

Warfare in the 21st century has grown to differ greatly from that of the past; not only because of the perpetually evolving military technology, but also - and perhaps more notably - because of the degree of responsibility expected from the lower echelon in military hierarchy in, for example, the area of decision-making in the field [69, p. 1-2]. Consequently, the leadership training in the lower echelon of military rank must be high; in “The Strategic Corporal: Leadership in the Three Block War” by General Krulak [60], he introduces the role of the “strategic corporal”, who are low-level unit leaders expected to be capable of taking independent action and making major decisions in the field. Part of this paradigm shift in modern warfare was perhaps first discussed in 1989 by William Lind, where he introduced the concept of fourth-generation warfare [130, chapter 3]. In particular, Chifu discusses four basic principles of the 4GW in [130, chapter 3], the two first of which aligns with what Krulak predicted back in 1999: small-scale missions and operations, and flexibility & autonomy within these missions and operations.

Indeed, Krulak (1999) states that “Success or failure will rest, increasingly, with the rifleman and with his ability to make the right decision at the right time at the point of contact.”. It is no longer an entirely binary choice of “do” or “do not”, as the degree of autonomy of the modern individual soldier in the field has expanded over the years – the context, decision, and repercussions of said decision could result in severe consequences [69, p. 1-2]. In other words, the modern soldier is faced with a number of difficulties that were not present in the early 20th century.

Alongside the role of the “strategic corporal” introduced by Krulak (1999), he also introduces the concept of the “three block war”, which is used to describe the nature of the challenges faced by a soldier on the modern battlefield. In it, he states that a modern soldier may be confronted by the entire spectrum of tactical challenges (full scale military action, peacekeeping, and humanitarian aid) in the span of a few hours, and within the space of three contiguous city blocks.

Analyzing Krulak’s predictions, Lovell identifies four key challenges of the modern soldier: the tactical environment is complex can change rapidly, necessitating flexibility from soldiers; decisions must be made rapidly and under extreme circumstances; junior leadership have a large degree of autonomy, which means hard choices must be made on a lower level; the media is everywhere, and wrong decisions are very visible to the world [69, p. 59].

A modern, more specific challenge faced by the modern soldier, which correlates to the four points above to various degrees, is that of IEDs (Improvised Explosive Devices). An improvised explosive device is described as follows [128, p. 10]:

*A device placed or fabricated in an improvised manner incorporating destructive, lethal, noxious, pyrotechnic, or incendiary chemicals and designed to destroy, incapacitate, harass, or distract. It may incorporate military stores, but is normally devised from nonmilitary components.*

According to [128, p. 2], the quantity and complexity of IEDs has resulted in a large increase in casualties across the board. These devices, in 2019, account for 42% of civilian deaths in Afghanistan [3, p. 6], and 45% of all U.S. deaths in operational war zones [74, p. 1]. Being able to effectively and safely dispose of such devices is of critical importance to preserve lives. It’s clear the ease-of-construction and effectiveness of these devices result in significant collateral damage, and facilitates the means by which inferior militant forces can combat forces of superior equipment and experience [33].

Alleviating the effect of IEDs is no easy task, as they are not constructed by conventional means under government oversight, and they adhere to no manufacturing standards [33] [128, p. 10]. Consequently, there is no “panacea” which eliminates the threat of IEDs entirely; instead, EOD (Explosive Ordnance Disposal) units have to carry a variety of equipment in an attempt to best combat the threat, ranging from remote-controlled robots to **remote firing systems**, and more [33]. This equipment is in addition to the cumbersome suit of “armor” donned by EOD technicians, which itself weighs around 35kg or more [27].

A **Remote Firing System** (RFS) is a system employed to safely and remotely initiate the firing of a payload (e.g. an explosive compound). Such a system will make use of a receiver to facilitate the energy transference to initiate the reaction with the payload, and a stand-alone transmitter to generate and transmit signals to control the receiver [15]. For example, special forces may use such a system to gain entry through a locked door (known as breaching); by connecting a receiver to an explosive compound, the special forces operators are able to take cover away from the door, and safely detonate it from a distance. In this context, “safely” refers not only to the distance between the operator and the payload, but also in terms of only initiating upon the operator’s explicit decision to do so. [15].

The rest of the thesis is structured as follows: Chapter 2 addresses our project management, including who we are, what our roles are in relation to the project, how we use Scrum to facilitate daily workflow, how we use tools such as Jira and Slack to help us adhere to the project model, and how we address and handle risk using FMEA.

Chapter 3 discusses the problems with today’s remote firing systems, including issues such as the size and weight of the system, its reliance on a stand-alone transmitter, and its lack of software solution. We discuss our understanding of these problems, from which we present the overarching objectives of the project. Furthermore, how we handle our requirements by dividing them into customer, system, and technical requirements are addressed, and how we intend to perform our validation & verification process.

Chapter 4 presents our multidisciplinary proposal to solving the problem described in the previous section. Here, software proposes an architectural model, and brings focus to one functionality in particular, Make Decision, to elaborate deeper on. Electrical proposes a custom-built PCB constructed to not only match (and in some cases exceed) the functionality of existing remote firing systems, but do so at a reduced size and weight. Mechanical proposes a casing for the receiver unit, discussing topics such as material choice, ingress protection standards, and user friendliness/manageability.

Chapter 5 details the implementation of of the Spark remote firing system, based on the ideas and information provided in the previous section.

Finally, in Chapter 6, the gist of the entirety of our findings and solution is presented, alongside deliberations as to how the project could evolve in the future. The thesis concludes with some closing thoughts in relation to the project overall.

# SPARK

## 2 Project Management

### Contents

2.1	Project Plan . . . . .	19
2.2	Project Model . . . . .	20
2.3	Project Environment . . . . .	21
2.4	Project Tools . . . . .	25
2.4.1	Jira Project Management . . . . .	25
2.4.2	Communication Platform . . . . .	26
2.4.3	Time Tracking: Clockify . . . . .	27
2.4.4	File Storage: Google Drive . . . . .	28
2.5	Risk Assessment . . . . .	29

### **Chapter Introduction**

This chapter of the bachelor thesis for the Spark remote firing system addresses the various means and tools employed in facilitating project workflow. Our work environment and methods are addressed, including an overarching project plan, alongside how we use Scrum to facilitate this plan and daily workflow. Furthermore, how we distribute the roles of Scrum is detailed, alongside our own custom roles (such as documentation responsible, team leader, and so on). The tools we use in the project are also presented, such as Jira, Slack, Clockify and Google Drive.

## 2.1 Project Plan

The goal of this project is to create a proof-of-concept remote firing system for Spectac and write an academic thesis for USN. The project will be managed by us, the students with supervision from USN and Spectac.

Our main stakeholders are Spectac, USN and the operator of the system. Spectac has expressed a desire to see a proof-of-concept to determine whether it could be a feasible product for their customers. Consequently, we want to make it as appealing as possible, and plan to employ creative solutions to showcase our system. We consider it of great importance to collaborate closely with Spectac and USN to create a synergistic effect to deliver the best desired performance for each stakeholder. To be able to full fill both Spectac and USN requirements we have decided that we want to focus on creating a proof-of-concept and a realizable solution. The proof-of-concept is for Spectac to get product that can be used to demonstrate the system capabilities and gather user and customer feedback. The realizable solution will have a much more realistic implementation and will have a much higher demand for quality technical solutions.

In Fig. 8.15 you can see the Gantt chart of our project. For us the Gantt chart is a way to visualize the project plan, but we also realize that we are planning with a great amount of uncertainty. The absolutes of the project such as presentations and delivery dates are displayed as milestones.

We have decided to use Scrum as our project model because it matches our project plan, more about that can be found in subsection 2.2. The tasks planned for each sprint is only an estimate and because of the nature of Scrum we expect the estimated tasks will change throughout the project. We have decided to set different milestones for both the project and the product to ensure we please our stakeholders. In our project plan we emphasize the need for a user and customer validation plan in beginning of the project. More on validation will be covered in subsection 3.3.

The milestones of the project are absolutes such as presentations and other requirements from USN and a user validation test. For every Scrum sprint we want to have a potentially shippable product to showcase to our customer and possibly users.

## 2.2 Project Model

The Scrum Project Model is an agile and incremental framework that is flexible and encourages changes over the course of the project, facilitating quick adaption to altered requirements or new ideas. The incremental part of Scrum is the "potentially shippable product" that should be ready at the end of each sprint. We have chosen to use Scrum because of the overall management structure it provides. Of particular note is the way Scrum handles the routines before and after a sprint, giving us a chance to review our work, and discuss how to approach the next sprint.

A sprint is a time interval selected in advance where the project group commits to completing some of the requirements for the product; in our case, we use 2 week sprints to give us ample room to change our minds, make mistakes, and quickly begin a new iteration. Before the development team starts a sprint they have to prioritize the requirements together with the customer and determine which requirements to work on in the upcoming sprint. Then they evaluate how much work they can take on in the sprint to come. This gives us a better understanding of the requirements, what the customer wants, a deadline for when it should be done, and tangible progress at the end of the sprint.

After a sprint, we set aside time to go through what the team has accomplished. This is to be done in collaboration with the customer and will provide the team with crucial and valuable feedback. After the evaluation of the work, the team reflects on the process that have been completed. This will include how tasks were organized and completed, and other team related issues. The team then recognize the issues that needs to be further addressed, and attempts to adjust them in the next sprint. This gives us the opportunity to customize the way we work and increase our efficiency.

Scrum gives us the flexibility to organize our work and at the same time provides us with some guidelines that will help us to stay on track during the project. The way Scrum is built and organized concur with the way we visualized our workflow in the project.

### Project Model Roles

The roles in Scrum is the development team, the product owner and the Scrum master. More details on Scrum can be found at [127].



Product owner	Lasse Hertel, Spectac.
Scrum master	Stian Jørgensen
Development team	Anne Synnøve Brendøy, Bjørn Ivar Bekkevold, Kristian Alfheim, Martin Brunæs, Sirajuddin Asjad, Stian Jørgensen.



Table 2: Scrum roles



## 2.3 Project Environment

### Group Composition

Information	Project role and description
 <p><b>Martin Brunæs</b> Electrical engineer martin@spark-rfs.no +47 93218283</p>	<ul style="list-style-type: none"> <li>- Team leader</li> <li>- Economics responsible</li> </ul> <p>Interested in technology and innovation. I find it exciting to think about new ways to use existing technology to create new products. I attained a certificate in ICT Service Operations prior to attending the university. A part from being a nerd, I enjoy skiing and gymnastics on my spare time.</p>
 <p><b>Stian Jørgensen</b> Electrical engineer stian@spark-rfs.no +47 45275588</p>	<ul style="list-style-type: none"> <li>- Scrum master</li> </ul> <p>Interested in technology and outdoor life. Before my studies I worked as a mechanic for 5 years. I like to learn new things and figure out how things work so started on my degree to find new and existing challenges.</p>

 <p><b>Sirajuddin Asjad</b> Computer engineer sira@spark-rfs.no +47 99207543</p>	<ul style="list-style-type: none"> <li>- Software lead</li> <li>- Requirements responsible</li> <li>- Marketing responsible</li> </ul> <p>Software developer, crazy tech lover and an endless learner. I'm experienced in C and C++ programming, and I like working with embedded development and hardware design. I am very curious and interested in working with both application-level and low-level programming, especially when it comes to projects involving embedded solutions.</p>
 <p><b>Kristian Alfheim</b> Computer engineer kristian@spark-rfs.no +47 48057193</p>	<ul style="list-style-type: none"> <li>- Documentation responsible</li> </ul> <p>From my youth I have always been interested in computers, which later evolved into an interest for technology and programming, and subsequently my choice of degree. Also enjoys reading about history, which would have been my second choice of a degree. Certificate in ICT Service Operations.</p>



 <p><b>Bjørn-Ivar Bekkevold</b> Computer engineer bjorn@spark-rfs.no +47 47453167</p>	<p>- Test and verification responsible</p> <p>Interested in Technology, Computers and Software. I Work part time as an Analyst in the field of Cyber Security. On my free time i enjoy reading books, hiking, work on hobby projects and spend time with friends and my family.</p>
 <p><b>Anne Synnøve Brendøy</b> Mechanical engineer anne@spark-rfs.no +47 41041953</p>	<p>- Risks responsible</p> <p>I'm 42 years old, single mother with two beautiful boys who now has reached the age of 7 and 9. Besides my boys, my hobbies are indoor climbing, Via Ferrata, kayaking and fixing cars.</p>

Table 3: Group members

### Project Role Description

The list below contains some short descriptions of our different project roles.

Project role	Description
Team leader	<ul style="list-style-type: none"> <li>- Administrative tasks.</li> <li>- Facilitates team collaboration.</li> <li>- Point of contact for USN and Spectac.</li> </ul>
Economics responsible	<ul style="list-style-type: none"> <li>- Maintains budget.</li> <li>- Invoice handling.</li> <li>- Purchasing.</li> </ul>
Test and verification responsible	<ul style="list-style-type: none"> <li>- Ensure test protocol is followed.</li> <li>- Monitor and ensure traceability.</li> <li>- Responsible for proper documentation of tests.</li> </ul>
Scrum master	<ul style="list-style-type: none"> <li>- Facilitates Scrum events.</li> <li>- Keeps the development team focused on the tasks at hand.</li> <li>- Educates the development team on Scrum.</li> </ul>
Requirements responsible	<ul style="list-style-type: none"> <li>- Ensures quality and updates of requirements.</li> <li>- Oversees creation of new requirements.</li> <li>- Ensure traceability between requirements.</li> </ul>
Marketing responsible	<ul style="list-style-type: none"> <li>- Responsible for marketing the project.</li> <li>- Responsible for social media.</li> <li>- Responsible for merchandise.</li> </ul>
Documentation responsible	<ul style="list-style-type: none"> <li>- Ensures quality of documentation.</li> <li>- Ensures consistency across the thesis.</li> <li>- Oversees and approves revisions.</li> </ul>
Risks responsible	<ul style="list-style-type: none"> <li>- Continuously monitor risks within the project.</li> <li>- Identify and add risks, and pair them with the right requirements and tests.</li> <li>- Make sure that the other disciplines identify and add risks.</li> </ul>

Table 4: Project role description

## 2.4 Project Tools

### 2.4.1 Jira Project Management

To keep track of our progress we have chosen to use Jira. This is an online tool that is customized for project groups that use Scrum or Kanban, and gives each group member easy access to our product backlog, sprint backlog, burndown charts and more. Figure 1 shows our board and current sprint. With everything consolidated into one place we can easily keep track of our progress. Most of the features in Jira are automatic and intuitive, which saves us from many frustrating and time consuming problems. For more information on Jira, see [10].

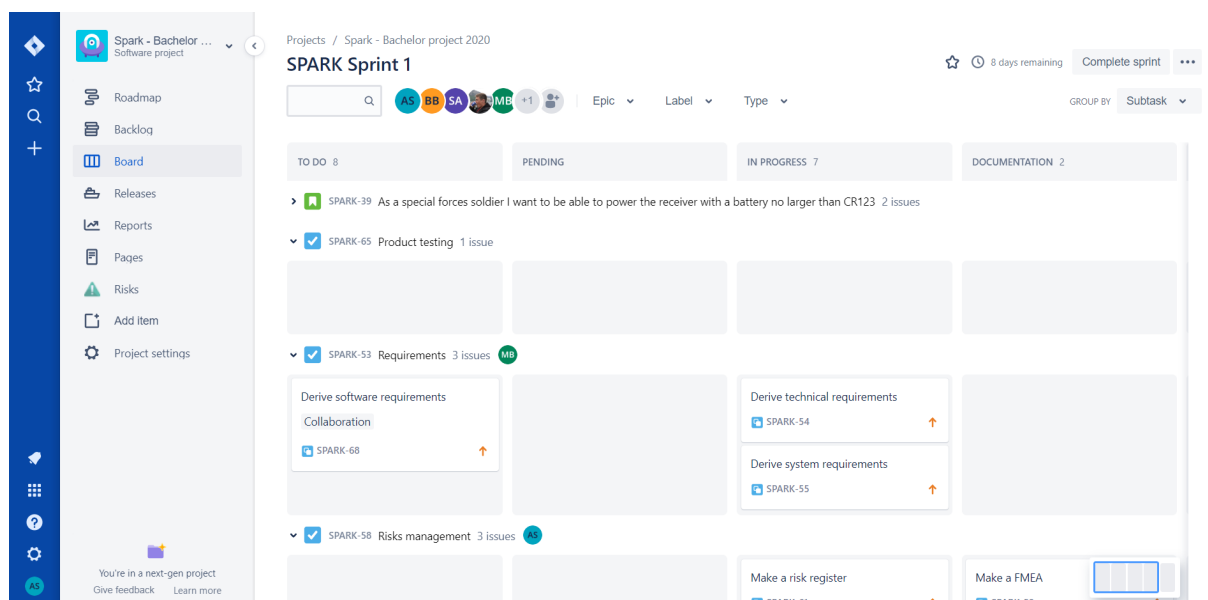


Figure 1: Our Jira board

### 2.4.2 Communication Platform

In order to minimize miscommunication and ensure there is history from our conversations, we have chosen to use Slack. This allows us create a private workspace for not only our group, but also our internal and external advisors, and divide the workspace into different channels for different topics, as shown in Figure 2. Since Slack also is ideal for sharing documents we have no need for another communication platform, which will make it easier for us along the way.

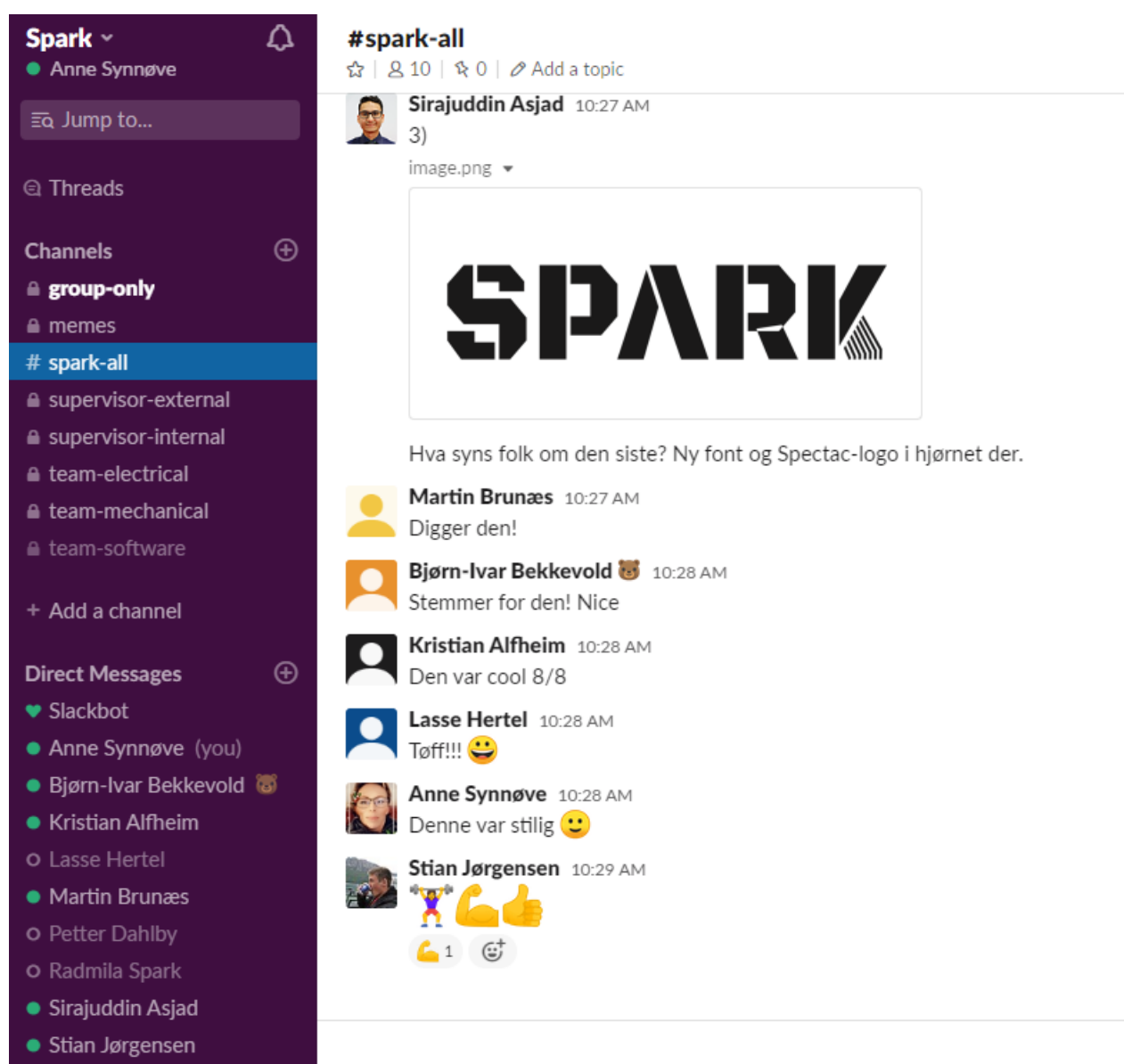


Figure 2: Our Slack channels

### 2.4.3 Time Tracking: Clockify

We have chosen to use Clockify to register time spent on this project. The reason for this is that Clockify is compatible with both Android and Apple, and have a mobile app so that we always can have Clockify available. We can also easily produce PDF reports for each team member and the entire team together.

All hours spent on this project by 24.05.2020 are listet in appendix

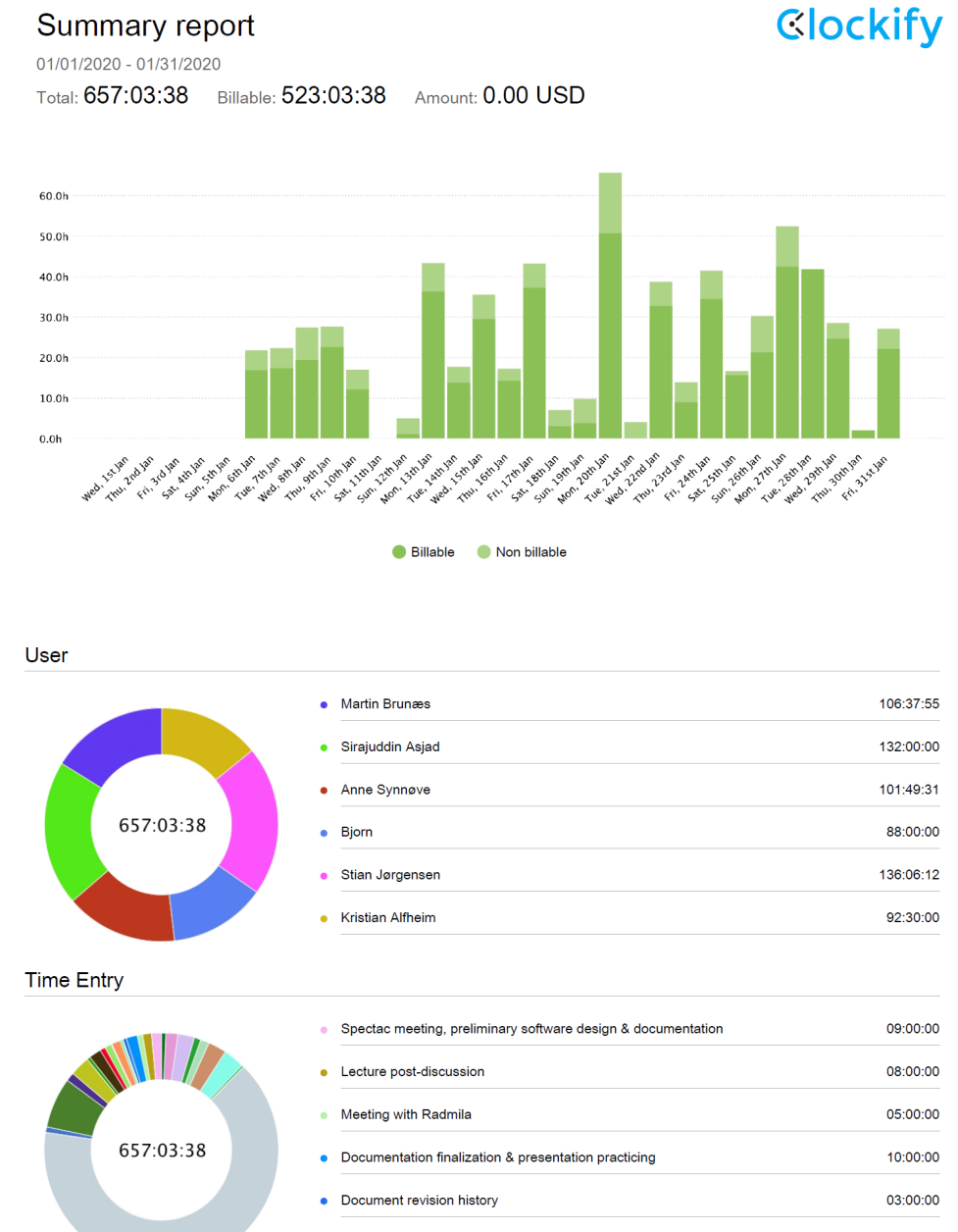


Figure 3: Examples of time registration report in Clockify

## 2. PROJECT MANAGEMENT

### 2.4.4 File Storage: Google Drive

In an effort to better organize our project, we have carefully structured our file storage to appropriately reflect our project model. The top layer of the structure is divided in four: "1 - Project Progress", which contains work that directly contributes to the progress of the project; "2 - Administrative", which contains administrative documents such as USN documents and timesheets; "3 - Meetings", which contains all meeting meeting notes; and "4 - Miscellaneous", which contains all other documents that do not fit under the three aforementioned folders. 4 illustrates the folder structure, where all purple boxes are on the same level.

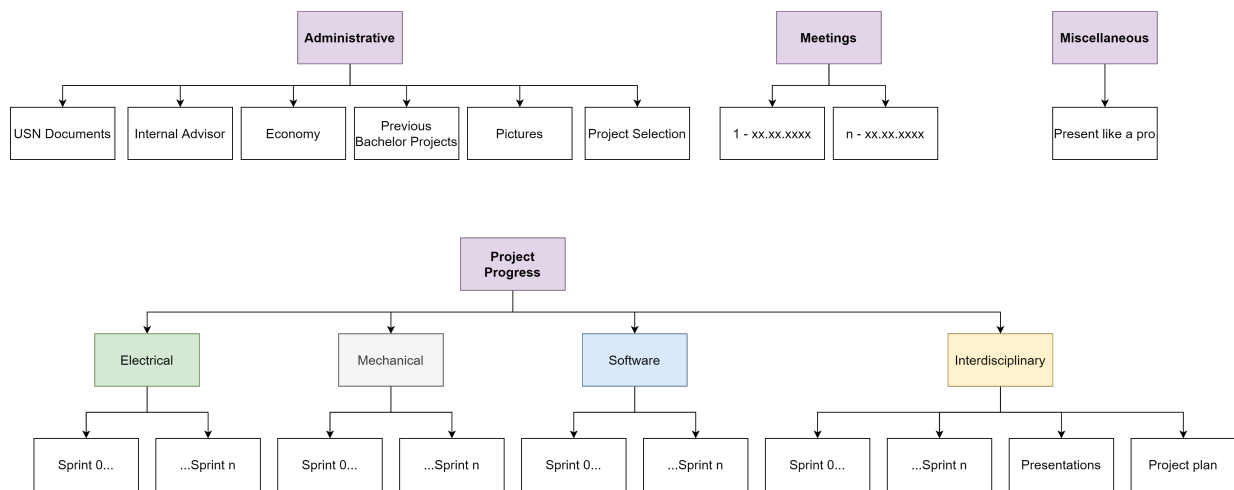


Figure 4: Folder Structure

Of particular note is the structure of the first top folder, which is again divided into folders for each discipline: software, mechanical, and electrical, as well as an interdisciplinary folder for items such as risk and overall requirements. The inside of these folders reflect our SCRUM project model, wherein the content is divided into sprints, which ensures we maintain a history and visible evolution of our work.



## 2.5 Risk Assessment

Every member of a development team or employees in a workplace has the responsibility to make sure that health, safety and welfare to any end user are taken into consideration with any product. Risk assessment tools are an aid that enables every worker to view potential risk and how to reduce, eliminate or avoid them completely. Together with a set of testing specification we can search and identify risks, and make sure that the preventive measures we have taken according to the risk at hand are sufficient.

Risk assessment are also to take care of the welfare of the company who are developing the upcoming product. If the product is a failure, the company may end up with a red bottom line, which could affect the welfare of the employees in the division responsible. More about risk assessment can be read here [90].

For this project a Failure Mode and Effect Analysis (FMEA) is chosen for the risk assessment process. The foundation of the layout in this thesis is from Visual Paradigm [96].

The risk assessment is an ongoing process that needs to be reviewed in every sprint, allowing the development team to catch any unforeseen risks or risks that are no longer relevant for the project. Every risk has its own ID and are connected to one or more requirements and tests. The requirement- and test ID should be plotted in a FMEA chart so that the development team can assure adequate traceability throughout the project.

The risk assessment will contain technical-, environmental-, external- and internal risks.

- **Technical risk** are connected with the function or design of the device.
- **Environmental risk** will have an impact on the environment, whether it is disposal at the end of our product life-cycle or the residue that will be left behind after usage.
- **External Risk** are connected to government, market and regulations.
- **Internal risk** are team collaboration, customer satisfaction, service, cost and quality.

In the 7th sprint there are now a version 7.0 of the FMEA for the risk assessment process. The FMEA can be read in its entirety in the appendices, 8.2.

We have chosen to remove the occurrence section from the FMEA. This because if any of our risk are likely to happen we do not have a viable product and no proof-of-concept.

The ISO-14000 standard states that life-cycle assessment (LCA) is:

*... a systematic set of procedures for compiling and examining the inputs and outputs of materials and energy, and the associated environmental impacts or burdens directly attributable to the functioning of a product, process, or service system throughout its entire life cycle.*

The life-cycle assessment is another ongoing process in a project. From the beginning, in preliminary design, to the very end, which is disposal or recycling.

In LCA the things we have to consider are [58]:

- Extraction of natural resources
- Raw material processing
- Manufacturing process
- Transportation and distribution
- Usage and maintenance
- Disposal and recycling

In this project the development team will strive to continuously consider the impact the choices they make in material, manufacturing and usage, will have on the environment. This is to make the engineering as green as possible without degrading any of the customer- or system requirements.

# SPARK

## 3 The Problem

### Contents

3.1	Problem Definition and Objectives . . . . .	33
3.2	Requirements Specification . . . . .	38
3.3	Validation Process . . . . .	39
3.4	Verification Process . . . . .	39

## Chapter Introduction

This chapter of the bachelor thesis for the Spark remote firing system concerns the problem definition, the objectives, and the requirements for our project. In it, we discuss the three identified main problems with remote firing systems today: its weight and size, its usage of a stand-alone transmitter, and its lack of a software solution. Furthermore, the five primary objectives of the project are addressed, derived from the aforementioned problems: 1) reduce its size and weight, 2) replace the transmitter with a mobile device, 3) develop a software solution, 4) develop training functionality, and 5) transform the RFS into a digitalized system. Subsequently, the requirements process is addressed by elaborating on how we organize our requirements by customer-, system-, and technical requirements. Finally, we detail how we intend to verify and validate these requirements, and ensure traceability. It is noted that all our requirements are in the appendix.

### 3.1 Problem Definition and Objectives

To ensure a 21st century soldier makes well-informed, confident decisions in the field, it's critical to equip them with equipment that can facilitate the ability to aid their decision-making in a variety of demanding circumstances. Furthermore, it's critical that such a soldier has received ample training in this equipment, so they may operate it with ease when confronted with a stressful situation in the field.

One such piece of equipment, touched briefly upon in the introduction, is that of the **remote firing system**. There exists a number of such systems on the market currently, some of which are commercially available and often used in the context of fireworks, and some that are used in a military context. Specifically, those used in a military context today present a number of issues that could stand to be innovated and modernized to ease the job of the soldiers whose job it is to protect.

It was mentioned in the introduction that EOD units have to carry a wide variety of equipment to combat IEDs, including, at times, a RFS. As it stands, the system used today can be cumbersome, and according to the trials performed in [27], simply the act of wearing the protective suit and performing physical activities can be enough to render the participants exhausted over the course of less than an hour. This holds no less true for EOD technicians who do not don such a suit, however; special forces also carry a significant amount of equipment. Consequently, a reduction in size and weight of the existing system to lessen their burden, and thereby allow them to bring more equipment to make them better prepared for their mission, is of great importance. In [15], they concur with this focus, and emphasise the importance of solving this particular problem: *“The remote initiation equipment needs to be as small in volume and as lightweight as possible.”*

In the case of Norwegian military, they carry the aforementioned transmitter in addition to a phone they use for a number of other services, such as maps and navigation. Some manner of digitalization, where the stand-alone transmitter is cut out in favor of the phone will be necessary to further minimize the encumbrance of the system, as this accounts for about half of the overall weight of the entire system [15].

A reduction in volume and weight presents a number of mechanical challenges, as a RFS has a few key features that are vital for it to function well in an operational context. Furthermore, it must adhere to certain standards of robustness and reliability (specifically, certain ingress protection degrees of protection). Furthermore, this ties in with a number of electrical challenges too, as smaller components must be selected without sacrificing any of the existing capabilities of current remote firing systems.

### 3. THE PROBLEM

---

Finally, as far as we know, there exists no software solution for such a system. The extent of remote firing systems in the 21st century is simply that of interaction between the receiver and transmitter, where the transmitter will be used to safely arm and detonate the receiver using coded signals [15]. Considering the greater autonomy and importance of decision-making on a more individual level in the modern military, it's critical that a proper software architecture is constructed that will facilitate the means by which a soldier can confidently make a well-informed decision in the heat of the moment.

In summary, the key problems of current remote firing systems are:

- First, the receiver size and weight. These impact the amount of equipment an operator can bring on a mission.
- Second, the current system in-use today consists of a two unit solution (a receiver and a stand-alone transmitter), which is in addition to a mobile device they carry.
- Third, there is no well-developed software solution for this sort of system, which is paramount of it is to be digitalized.

The objective of the system described herein is to develop a modern, solution to the remote firing systems in-use today, to either supplement or replace current solutions. To accomplish this, the following five objectives will be necessary:

1. Reduce the size and weight of existing remote firing systems without sacrificing any of the functionality or capabilities.
2. Replace the transmitter part of remote firing systems with a mobile device.
3. Develop a software solution that aids the soldier in their decision making, in an intuitive and user-friendly manner.
4. Develop training functionality inside the app, or with the app, to facilitate the training of its users.
5. Transform the classical RFS from a standalone system into a digitalized system.

3. THE PROBLEM

**Understanding the Problem**

We have constructed three context diagrams in an effort to visualize the external factors that must be considered for our system. At this point in time, we have considered three different angles: the system and its stakeholders; the receiver; and other systems that may interact with our system.



Figure 5: Stakeholder Context Diagram

3. THE PROBLEM

The stakeholder context diagram (figure 76) displays the various stakeholders that are likely to interact with our system in one way or another. Active stakeholders are displayed in a yellow color, while passive stakeholders are displayed in purple. Here, we are able to clarify where our requirements are likely to come from (particularly, EOD operators, the military, the police, and special forces operators), and we are thereby better equipped to solicit guidance from the most relevant stakeholders. Primarily, the customer will be the military, wherein there are EOD and special forces operators who will use our system, but seeing as the police does have EOD operators of their own, we consider this a possible customer as well.

Figure 6 illustrates a number of interacting systems we will need to address. Notably, Spectac expressed a strong desire to employ a Samsung phone with an Android operating system (as this is what is often used in the field in military context), and consequently these are of critical importance to realize a seamless system that will integrate into existing architecture. Furthermore, in order to facilitate testing of our system, it is very likely that we will need to set up a Wi-Fi network to simulate the mesh network created by the tactical radios of military personnel, as it is unlikely we will be able to obtain these radios ourselves.

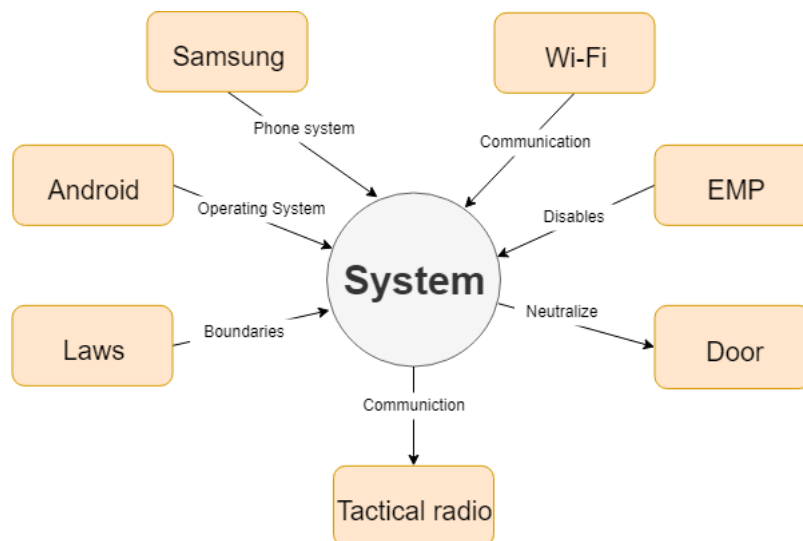


Figure 6: System Context Diagram



3. THE PROBLEM

Finally, we constructed a context diagram from the receiver’s point-of-view, shown below in figure 7. In it, A.Operator refers to the operator who planted the receiver, and P.Operator refers to the operator who didn’t plant the receiver, but serves in the same unit as the active operator. Of particular note are the challenges regarding the security of our system as a whole, arising from the necessity that the communication between the Android phone and the receiver must be wireless. Among other things, this exposes a plausible attack vector in which a hacker could potentially seize control of the receiver and detonate it, or the signal between the phone and the receiver could be jammed.

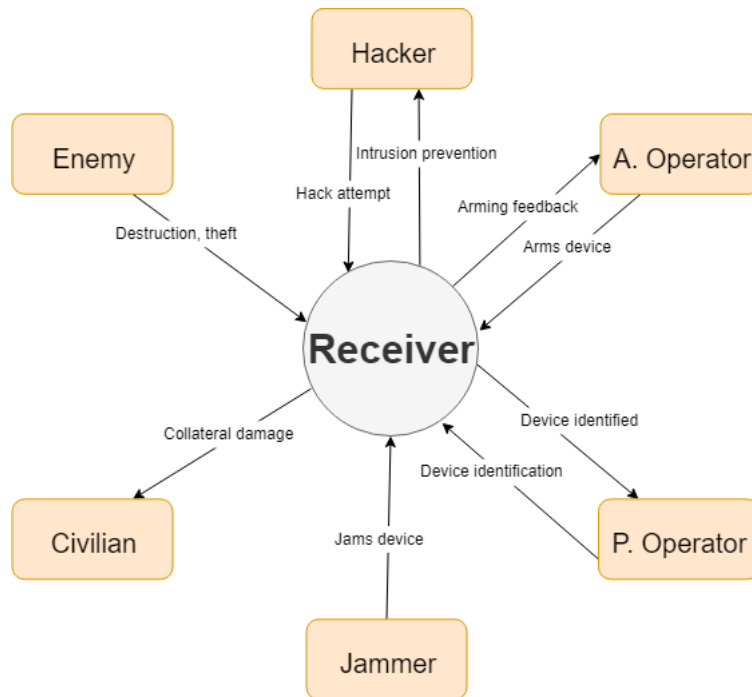


Figure 7: Receiver Context Diagram

Additional factors to consider include:

- The reliability of the receiver, as it can under no circumstance accidentally arm and detonate itself, necessitating a two-step activation process.
- When the receiver detonates, it does so in a controlled, predictable manner that minimizes collateral damage (particularly in urban areas where there may be civilians).
- If the receiver is stolen by the enemy, it must be rendered completely useless without the sender’s software.

### 3.2 Requirements Specification

In addition to experienced engineers, Spectac also possesses first-hand expertise when it comes to the military and how they operate. Our customer requirements are written in the form of user stories to fit our project model, and these are what our product backlog is composed of as well.

There is a system for how the requirements are arranged; at the top level there are customer requirements (pink color), written in the form of user stories. There are system requirements (green color) derived from the customer requirements, and each and every system requirement consists of multidisciplinary technical requirements (blue color).

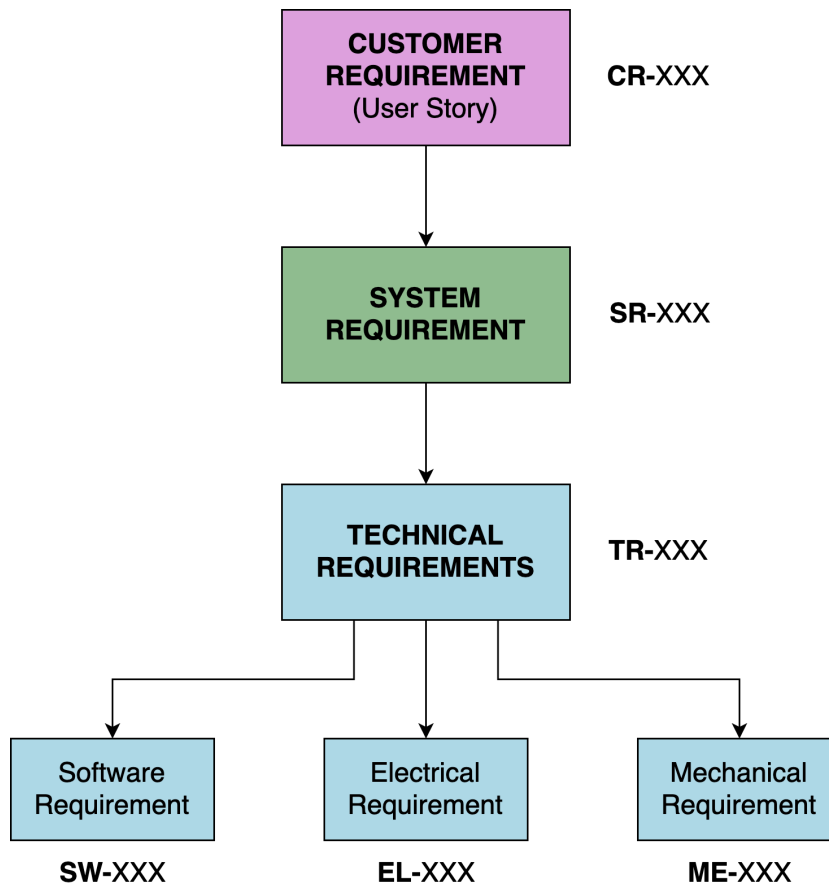


Figure 8: Structure of Requirements

System requirements define what the system shall do from a top-level perspective, whereas technical requirements provide technical specifications and constraints about the requirement to the development team. Our technical requirements are constructed to ensure we're engineering according to specific, measurable quantities. Each discipline have their set of technical requirements, and these allow us to ensure that the customer's wishes are translated directly into tangible quantities, and not left up in the air.

All of our requirements are located in the appendix, organized with a unique ID. There is traceability between the requirements, the testing and risk, which highlights the big picture view of how the different parts of our project are connected together. Whenever a requirement is mentioned in the thesis, it's noted that if the thesis is viewed digitally, these can be clicked to take you to the corresponding requirement in the appendix.

### 3.3 Validation Process

In our project we are trying to emphasize customer validation, meaning the verification of what our customer actually wants and needs in the end product [18]. There are different methods for validating a system and we are using some of them in our project, such as [129], which defines a *System Validation Plan* to drive acceptance testing frequently validating against the customer agreed upon expectations. Based on Scrum there is no predefined validation method, but via frequent meetings with the product owner and stakeholders, we believe this will provide an even better validation process for our system, which ensures a fitting product that aligns as close to the actual user experience as possible. A key milestone for our team will be to produce a prototype for a *Design Review* in the systems operative context at the end of March. Succeeding in this we will give us the chance to receive direct feedback on our product from those who will use it.

We have a good work-relationship with our employer, and we are able to interact on a weekly basis, ensuring feedback and input often with our backlog for reference, helping us validate our tasks in line with the customer and end user. As the company Spectac consists of people from both highly technical engineering backgrounds and operative special ops military backgrounds, we have the luxury of getting consistent input on our work, ensuring compliance to the original customer requirements.

### 3.4 Verification Process

To ensure that we are building a system that the customer needs, we need to verify all requirements. We have created a generic testing report that is intended to verify customer and system requirement, as well as technical requirements and ensure good traceability. We plan on verifying concurrently with the development of the system; our project model enables us to do this frequently. Despite using Scrum we are influenced by our semester learning about the System Engineering process, and we will employ some of the foundational practices such as identifying the initial requirements early on and deriving these to facilitate testable and measurable requirements [129]. Based further on Scrum's aim to make a "possibly shippable product" in each sprint, which entails an experimental and simulation based verification approach per the SE methodology [18].

# SPARK

## 4 The Proposal

### Contents

4.1	Software Design . . . . .	43
4.1.1	Evolution of Software Proposal . . . . .	43
4.1.2	Proposed Conceptual Software Model . . . . .	46
4.1.2.1	Functionality of the Proposed Software Solution . . . . .	47
4.1.2.2	Software Abstractions from Sequence Diagrams . . . . .	48
4.1.2.3	Software Architectural Model . . . . .	51
4.1.3	Elaborating Software Functionality in Preparation for Implementation	52
4.1.3.1	Expanding the Base Use Case Model . . . . .	53
4.1.3.2	Expanding Excerpts of the Proposed Software Architectural Model . . . . .	65
4.1.3.3	Example of the Section of the Final Software Architectural Model . . . . .	72
4.1.4	Paving the Way Towards Implementation . . . . .	73
4.1.4.1	Ad Hoc Networking . . . . .	74
4.1.4.2	Fault Tolerance . . . . .	83
4.1.5	Military Communication and Information Systems . . . . .	85
4.1.5.1	Cloud Computing in Military Applications . . . . .	85
4.1.5.2	Implementing Cloud Computing in Military Operations . . . . .	87
4.1.5.3	NATO Generic Vehicle Architecture (NGVA) . . . . .	88
4.2	Physical Design . . . . .	93
4.2.1	Design Challenges . . . . .	93
4.2.2	Casing Design . . . . .	94
4.2.2.1	Material and Production Methods . . . . .	95

4.2.2.2	IP, Atex and Test Standards . . . . .	96
4.2.2.3	Testing of Material and Casing . . . . .	99
4.3	Electrical Design . . . . .	101
4.3.1	The Electrical System . . . . .	101
4.3.2	Plasma Ignition System . . . . .	104
4.3.3	Step Up . . . . .	104
4.3.4	Electrical safety barriers . . . . .	108
4.3.5	Ethernet . . . . .	113
4.3.6	Hardware Paring and Authentication . . . . .	115
4.3.7	Battery status system . . . . .	117

## Chapter Introduction

This chapter of the bachelor thesis for the Spark remote firing system concerns each discipline's proposal for a solution to the problems and objectives presented in section 3. We herein first propose a conceptual software model consisting of a set of UML diagrams (including use case and sequence diagrams, as well as two software architectures on different levels of abstraction). The software team elaborates on their modeling by focusing on one piece of functionality in particular: Make Decision. Then, the mechanical team details important considerations when approaching the design of the receiver casing, addressing topics such as material alternatives, IP standards, and more. Finally, the electrical team presents a custom PCB and the various components necessary to fulfill our customer's wishes.

## 4.1 Software Design

### 4.1.1 Evolution of Software Proposal

In this subsection, the software team decided to keep some of our earliest drafts and thinking; despite the fact that it's outdated, we believe it conveys a blunt (if valuable) initial brainstorming when we were attempting to comprehend our objective.

#### Initial Drafts

A huge prospect of our remote firing system is to improve the current radio communication system within the military, which is based on radio communication system over a mesh network. This is something that needs to be carried by a soldier, along with a significant amount of other equipment that the personnel in the military must carry.

When a message gets transmitted through the communication system, it goes from the sender's device through a tactical radio, which is connected to the military's mesh network, and gets transmitted to the endpoint, which is also a tactical radio connected to the same network. The entire IP communication happens between these two tactical radios, which is considered secure and reliable by the military.

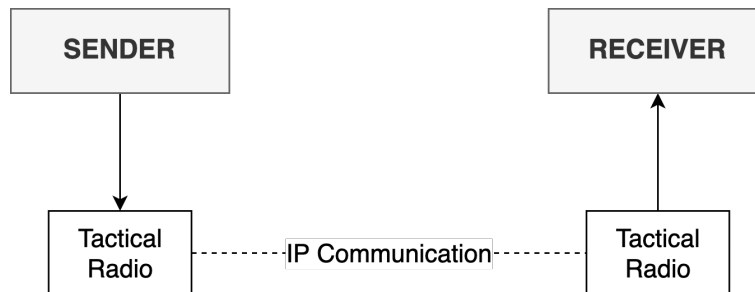


Figure 9: Current radio communication system

#### First draft

The communication channel of the remote firing system that we are developing is based on WLAN connectivity between the sender and receiver devices, and there is a Command and Control (C2) server deployed between the sender and the receiver in order to route and forward the requests, and to ensure an end-to-end encryption between the two parties.

The sender device is equipped with Android and the receiver is equipped with a WLAN chip in order to interact with our network. They are both connected to the same local intranet, where the C2 server is deployed. The sender device submits a message to the server, which then gets processed by our application programming interface (API) and

## 4. THE PROPOSAL

thereupon forwarded to the recipient, where the message gets decrypted.

In order to establish a mesh network within our proposed communication system, using WLAN routers, we need to deploy network repeaters within a range to extend the scope of our local intranet. This ensures that more devices can connect to the same network, and that these devices can communicate with each other over the same communication channel.

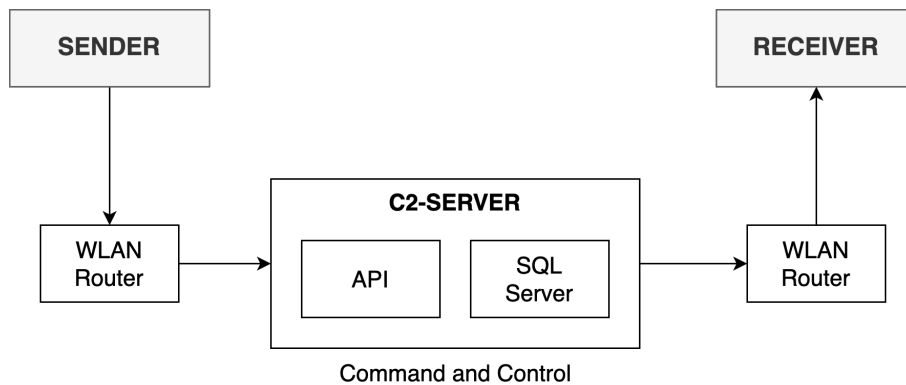


Figure 10: Proposed communication system

### Second draft

During our second sprint we've realized after consulting with our external advisor that a pure client to server based solution would not meet the initial requirements of the system when deployed to the field. Being a local network in a demanding environment out in the field, a static server mounted on one operator for the whole mission would expose the data and easily lead to problems if something were to happen with the operator or equipment. The team refined our thought and started thinking about a more peer to peer based solution as seen from the figure. Where each node represents an operator on a joint mission, with a "Data Token" making hops between nodes checking for new data to extract, and update the next nodes with updated information.

Inspired from the 1980's network protocol *Token Ring* (IEEE 802.5), we tried to make an attempt at a distributed server where each operator should have updated data as often as possible, with a data token (token ring) hopping between the operators.



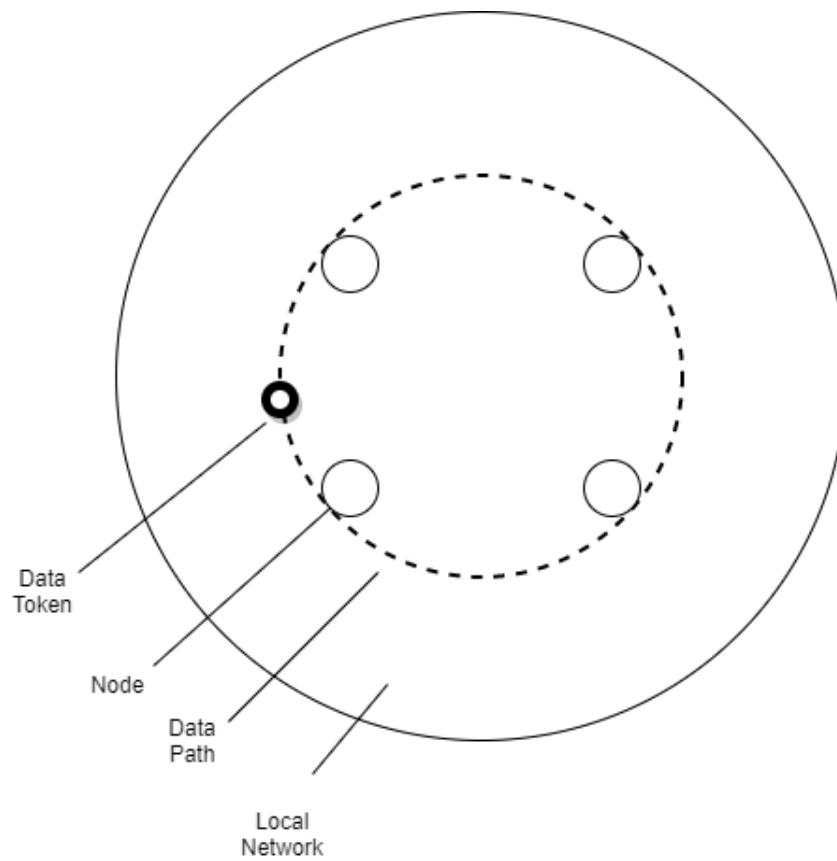


Figure 11: Block Diagram - Peer to Peer Network

### 4.1.2 Proposed Conceptual Software Model

The Proposed conceptual software model consists of a set of UML diagrams, which help to define functionalities of the software and discover abstractions within these functionalities which will become main parts of the proposed software. The proposed conceptual model is defined through software architectures of the proposal in which we show layers of software component which follow the Model-View-Controller (MVC) software pattern [28]. Its characteristics are known to separate software components into layers and distinguish between software components used for User interfaces and software components, which store computing programs and data. The success of the MVC pattern (or sometimes called MVC architectures) was guaranteed by component based software technologies such as J2EE [140] and .NET [91] in which software components form the MVC layers can be deployed in various Integrated Development Environments (IDE) such NetBeans [82], Eclipse [46], QtCreator [114] and similar.

The process of creating the proposed conceptual model consists of the development of UML diagrams which start with the definition of functionalities through a Use Case model, extraction of abstractions from this use cases model in the format of objects in the sequence diagrams, and finally a conversion of these abstractions into layered software components within the MVC pattern.

In section 4.1.2.1 we show an initial use case model, which illustrate the functionality of the software solution we develop and as such contains only base use cases and their descriptions. In section 4.1.2.2 we discover abstractions in the format of UI, control and data objects which will comprise sequence diagrams. The distinction between UI, control and data objects, in the top row of sequence diagrams, is essential when modelling with UML because it encourages the separation of concerns and clear definition on which type of UI and data we need in order to create a software solution. This section also emphasises that the most important part in the discovery of software abstractions are data objects of sequence diagrams and their potential sharing across various use cases within the initial use case model. In section 4.1.2.3 we show how a set of sequence diagrams developed in 4.1.2.2 can be converted into the proposed conceptual software model. In this model, software layering, created according to the type of objects discovered in sequence diagram, shows software component typical for the MVC pattern. In the same section we describe how the proposed model affects the software application we will develop and defines exactly which type of data we need to obtain in order to implement the functionality described in our initial use case model.

#### 4.1.2.1 Functionality of the Proposed Software Solution

Figure 12 shows the initial use case model which consists of base use cases only. Their short description is given below.

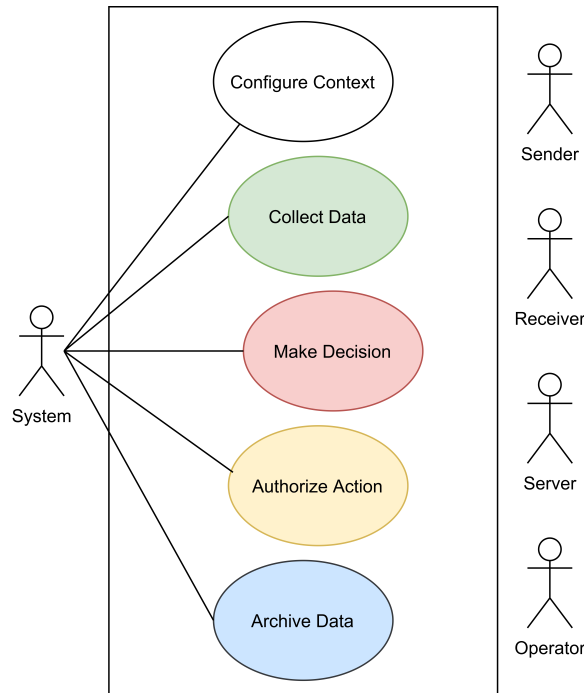


Figure 12: Initial Use Case Model

The Make Decision use case describes the functionality that will assist in tactical decision making; this includes all tactical decisions an operator may have to make in the field related to the RFS, such as controlling receiver units (e.g. arm it, disarm it, set a timer, deploy a waypoint on a map to indicate its location), initiating detonation sequences and operational context warnings for the operator. This requires sensor data, health monitoring data and connection status data.

Configure Context is a use case that uses all data relevant for planning scenarios beforehand, decisions on deployment of receivers, team configurations, and environment changes. These are to be prepared prior to the mission, and prepares such data to be used in decision making later on in active operations. Examples of relevant data for configuring the environment include: data on people involved in action data on the terrain and configuration of the field/urban area where detonation is required, location data, current local threats, evaluation safety and preventive measures to take in hostile environments.

The Authorise Action use case is needed for personnel access data, setting user roles, access privileges, pairing and connection details and allows for an overview of who has access to

what, what they are able to do, which devices will link and communicate to each other, and so on.

Collect Data and Archive data are use cases that are used for managing data on information which are either needed or generated by this software application. Example of archiving are user logs, training data, behavior analytics and component health measurements (e.g. CPU usage, memory usage). Collection of data may happen spontaneously (when a situation requires), with careful planning (such as logistics and known strategies) and sporadic data generated by sensors and data which results in changes in the environment, where decision making is needed.

It is obvious that the first two use cases will carry most of the functionalities of the software application we develop. Their successful implementation will depend on the way we discover abstractions from use cases which focus on the collection and archiving of data.

The use case model from Figure 12 is not the only model we could use for the development of our proposal. However, it is a good starting point to distinguish between two major functionality of our software: Configuration of the environment in which detonation is feasible and Decision Making based on the relevant information collected from or generated in such environments.

#### 4.1.2.2 Software Abstractions from Sequence Diagrams

In Figures 13 (these are 6 sequence diagrams created from Figure 12) we show the discovery of abstractions (objects) from each use case which are placed in individual sequence diagrams. This discovery is done according to the UML principles. Here, each use case generates one sequence diagram, and in each sequence diagram, we adhere to the following UML rules:

1. Finding a UI for each particular functionality described in the use case.
2. Defining Control objects, which will become a computing program in our software solution.
3. Discovering all data objects which are needed in order to perform the functionality defined in the use case.

Therefore 1. and 2. above are compulsory, which means that no sequence diagram derived from the use case model can start without these two types of objects. This means that our role was to discover exactly which data objects must be in each sequence diagram and

how the control object is going to use them.

At this stage, we do not have to show exactly the computations stored in the control objects, but we need to show through messages defined in the sequence diagrams, if the data is going to be inserted, manipulated/maintained, updated or just read. The sources of data (data objects) can be anything: from databases, data repositories on clouds or servers/web to “arrays” typical in programming languages. Consequently we do not need to name messages passed between objects yet. In our case, we have elected to not show any arrows (which is to say, sequence of events) for these sequence diagrams, as they are on a very high level of abstraction; for the moment (before we dive deeper in section 4.1.3) it is sufficient that we define which data objects are used by which controller.

The discovery of data objects has an interesting outcome:

1. It shows the scale of various data sources and potential data types which are needed for implementing the functionalities in use case.
2. It shows potentials for data sharing between various use cases which would be essential when creating a software architectural model.

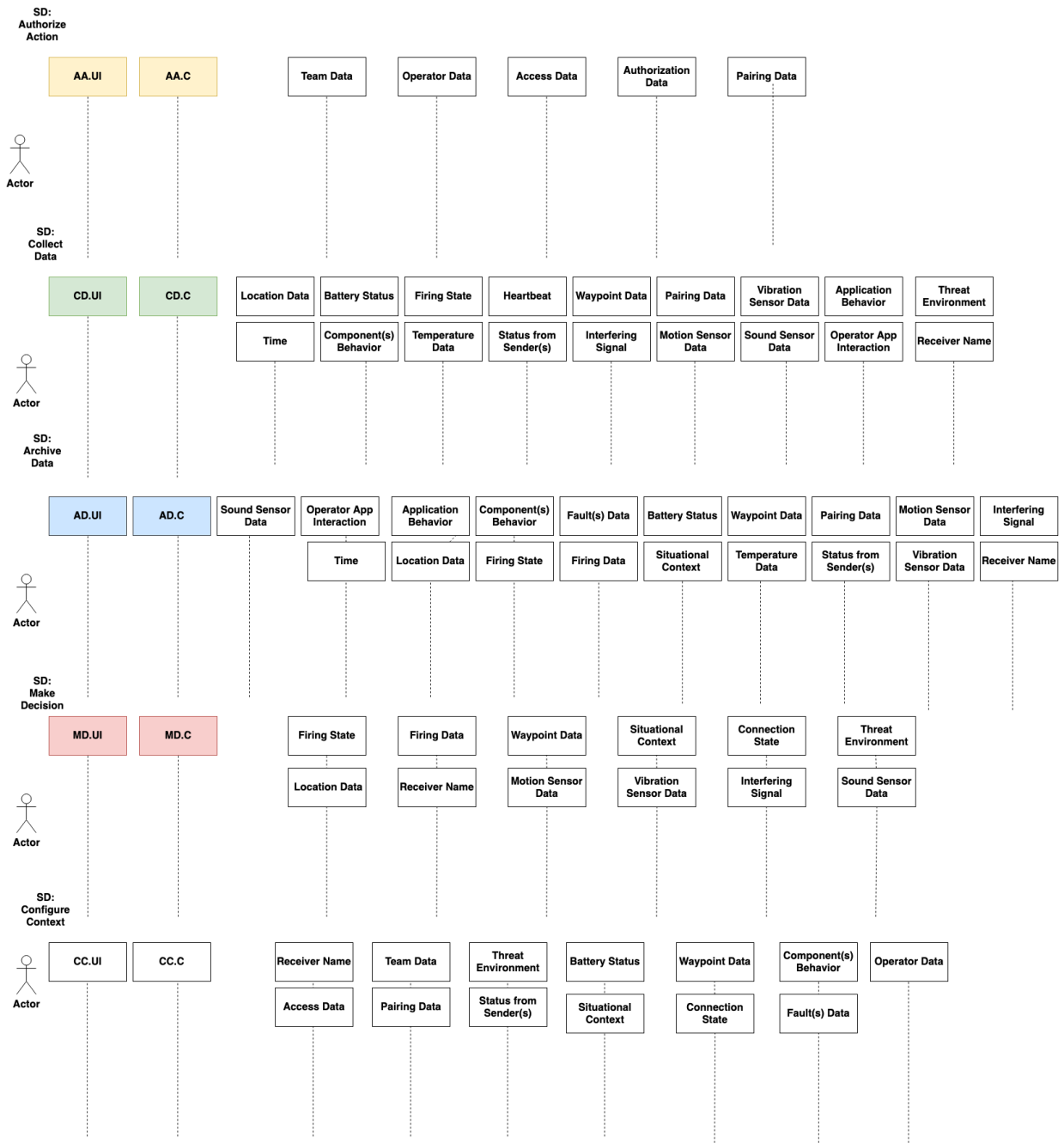


Figure 13: Sequence Diagrams for Figure 12

### 4.1.2.3 Software Architectural Model

Figure 8.7, placed in the appendix is a conceptual model of the proposed software solution. It is a layered and component-based software architectural model that adheres to the MVC pattern. Each type of objects discovered in sequence diagrams (UI, Control and Data objects) are converted into software components which are placed in adequate layers of the architectural model. This means that we group software components, according to their purpose into layers; all UI objects create the top row of the proposed architectures and all data objects are converted into software components stored at the bottom layer of the model.

The layers of the proposed model, which follow the MVC pattern are very suitable for deployment in any operating environments, which range from cloud and server computing, to mobile and wireless environment such as edge and fog computing [52] [25].

The model also shows a high degree of data sharing amongst all functionalities of the proposed software. From our architecture in 8.7, we see a large amount of data related to receiver units and their operational status linked to most controllers, especially data which will be distributed further across devices, such as sensor data, location data, component health monitoring and critical firing data.

Anomaly data which originates from the computing edge itself and initialization data are less frequently shared to the different components, which makes sense, seeing as these are all likely to be computed on the edge, and only emitted further upon faults or critical errors. It is clear that **Make Decision** depict heavy dependencies towards our data findings, especially sensory data and firing data, as these are all able to provide meaning of a situational context and define possible threats. Furthermore, In the software architecture (subsection 8.7) we see the **Authorize Action** with the least data dependencies, as this is intended to utilize data for initial setups and configurations, which may remain persistent for some time while the system is operational, such as key generation, connectivity management, access privileges and team configurations.

Due to the nature of the MVC pattern and our proposal, it is obvious that these layers can be implemented using different technologies and software application derived from the proposed model, and must use an IDE in order to deploy the model and create computations suggested by the proposal. At this stage, the type of data we discovered in Figure 8.7 and data accessibility is not relevant. The choice of software technologies available for the deployment of the proposed model will address issues relevant for the implementation.

### 4.1.3 Elaborating Software Functionality in Preparation for Implementation

Figure 8.7 from the appendix is a starting point for deploying the proposal and preparing it for the implementation. Before we start using software technologies for this task, we have to look again at the proposed conceptual model and analyse the relation between the number of software components, and determine which of them hosts computer programs and which hosts data. A high number of data components compared to only 6 components which store computing programs presents a clear message; we have to go back to the functionality of the software application and try to discover the level of complexity hidden in each use case defined in Figure 12. This does not mean that we defined wrong functionality in Figure 12; rather, this means that we have to look at possibilities of finding more use cases which are naturally hidden within our original base use cases from Figure 12.

Thereof rein the next few sections we actually prepare the proposal for the implementation by:

- Specifying a detailed functionality of each base use case from Figure 12, and creating a new use case model from each base use case in Figure 12 (section 4.1.3.1)
- Creating new sequence diagrams form each newly discovered use case from section 4.1.3.1. Due to the complexity of the proposed solution, we choose to illustrate this process by incorporating detailed analysis of ONLY ONE initial base use case from Figure 12: Decison Making (section 4.1.3.2).
- Creating an excerpt from the architectural model from Figure 8.7 where discoveries of new use cases and their sequence diagrams from sections 4.1.3.1 and 4.1.3.2 create a “mini” MVC pattern for the Make Decision use case model (section 4.1.3.3. The outcome of this subsection is an excerpt from the proposed model in Figure 8.7 which will be ready for the implementation in the next Chapter.

However, the deployment of this proposal needs more attention before we start the implementation. Therefore, in section 4.1.4 we address the practical issues related to the implementation and address promising technologies in WiFi Direct and Bluetooth as well as relevant security concerns.



#### 4.1.3.1 Expanding the Base Use Case Model

In Figures 14,16, 17, and 15 we illustrate new functionalities which were hidden in the base use cases of Figure 12. We address the Make Decision use case later, in section 4.1.3.2

Figure 14 shows the functionality for the **configure context** base use case from Figure 12. This is still a fairly abstract diagram, and is unlikely to be implemented in this project in its entirety. The intention of this functionality is to allow an operator (specifically a team leader, most likely) to set the scope and context of the mission they're about to embark on. The primary requirement voiced by Spectac related to this functionality is that they would like a group of operators to be able to view all receivers within their group (so if two groups are in the same area, they do not end up mixing receivers), but in an effort to provide a more covering solution, we have modeled slightly beyond this scope as well.

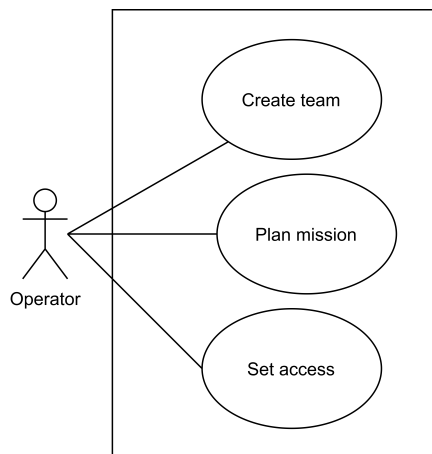


Figure 14: Configure Context Use Case Diagram

- **Create team** - Create a team in the application, wherein the operators within the same unit, as well as their receivers are grouped.
- **Plan mission** - Set the context and necessary information for the mission, so the application is better equipped to aid the operator in their decision making.
- **Set access** - Set access to receivers.

Figure 15 shows the functionality for the **collect data** base use case from Figure 12. This diagram relates to the collection of various data during the mission, as well as the reception of the heartbeat from the receiver to the transmitter.

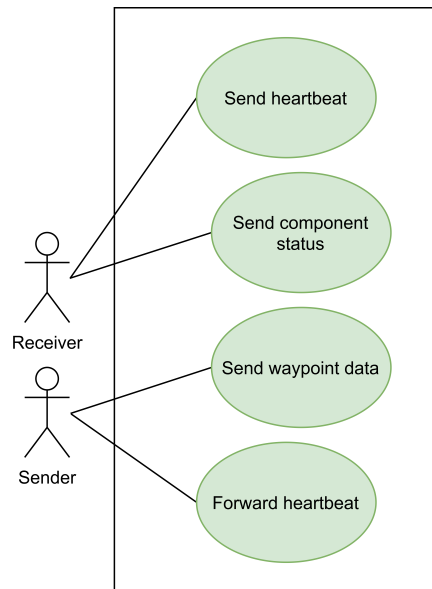


Figure 15: Collect Data Use Case Diagram

- **Send heartbeat** - Periodically send a heartbeat from the receiver to the transmitter to ensure the receiver is healthy. The heartbeat package includes various sensory data that can aid the operator in their decision-making. The specifics of the heartbeat protocol can be viewed in the network protocol section.
- **Send components status** - Send status of receiver components to the transmitter. If there is a component failure, the operator should be warned.
- **Send waypoint data** - Asynchronously update all transmitters in the same unit with waypoints created when deploying a receiver.
- **Forward heartbeat** - Forward the heartbeat from a connected receiver to all other transmitters in the same unit.

Figure 16 shows the functionality for the **authorize action** use case from Figure 12. This diagram relates to the authorization of various actions related to the system. Specifically, in its current iteration, it shows the system's capability of pairing with receivers and transmitters, as well as discarding said pairing. Within these, it's implicit that these pairings must be authorized based on some sort of security to be able to pair.

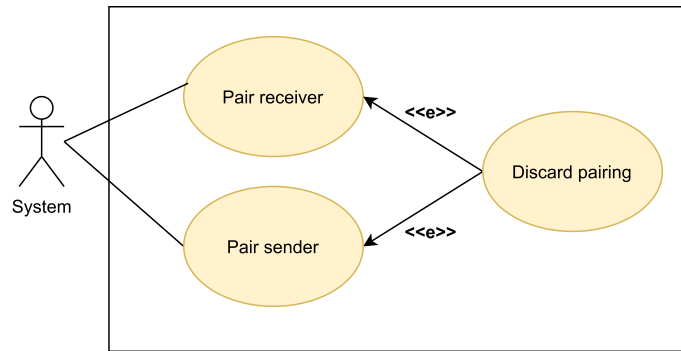


Figure 16: Authorize Action Use Case Diagram

- **Pair receiver** - Initiate the pairing between a receiver and transmitter. If authorized, the pairing succeeds.
  - extends to **Discard pairing** - Discard a pairing, if the authorization should fail.
- **Pair sender** - Initiate the pairing between two or more transmitters to form a network. If authorized, the pairing succeeds.
  - extends to **Discard pairing** - Discard a pairing, if the authorization should fail.

Figure 17 shows the functionality for the **archive data** use case from Figure 12. Similarly to the configure context diagram, this is also a fairly abstract diagram aimed at a more generic solution, and is unlikely to be implemented in our project. This diagram relates to the uploading and downloading of mission data to better equip an operator to make well-informed decisions in the field.

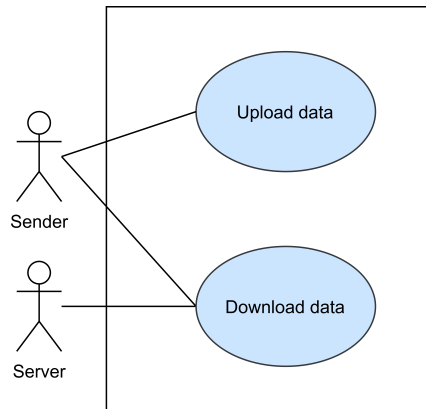


Figure 17: Archive Data Use Case Diagram

- **Upload data** - Upload data collected during a mission from the transmitter to a server. This data can include data from the "collect data" use case diagram.
- **Download data** - Download mission data onto the transmitter before a mission to aid operator in decision-making. For example, this could include information about the operational area, the threat environment, and so on.

Even though we focused on the Make Decision use cases, we have some other sequence diagrams related to the Configure Context and Collect Data use cases. These are included here as examples, since the diagrams are still valid and relevant.

### Send Heartbeat

The Heartbeat is a very important subsystem of the remote firing system, and it is necessary for both transmitters and receivers with the intention of ensuring reliable communication.

The heartbeat is implemented differently on the receiver and transmitter, and Figure 18 illustrates the process of the heartbeat from the receiver perspective. It consists of more sensor-generated data, compared to the heartbeat implemented at the transmitter side, which consists of more user-generated data and logs.

The heartbeat has one main operation, which is to send a signal to the transmitter every half minute, for the purpose of informing the operator that the receiver is online. The signal consists of several data, such as battery status, current temperature, the firing state of the receiver and the current location. This is relevant information that is shared with the operating team within the mobile application.

The heartbeat implemented at the receiver side has a one-to-one relation with the corresponding operator, and it only signals the transmitter that is paired with the receiver. If, however, the receiver access is transferred to another operator (through Request Control or Forward Control), then the heartbeat would automatically be transmitted to the new operator.

**SD: Send Heartbeat**

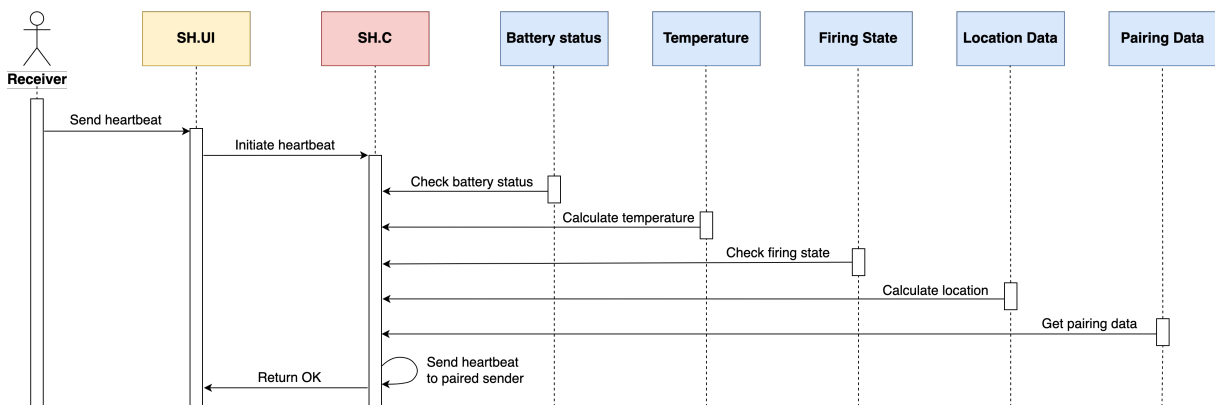


Figure 18: SD: Send Heartbeat (receiver-side)

The heartbeat functionality is absolutely significant in terms of continuously transferring data between the receiver and transmitter, and to indicate the team operators that the receiver is alive and ready for a tactical commands. Without the heartbeat functionality, there would be no simple way for the operator to know that the receiver is online and it would be complicated to transfer data amongst the peers.

### Send Component Status

It is important to have a self-diagnosis system to detect and report faults observed at the hardware components. The remote firing system consists of several sensors and safety-critical components, and it is crucial to continuously check that all hardware components are functioning as intended.

The component diagnosis system checks the GPS sensor, the temperature sensor and the component logs to observe any unusual activity, such as power loss or other hardware-related faults. The status signal is then transmitted to the paired operator, in order to inform the operator whether the receiver is fault-free or if there is something wrong.

SD: Send Component Status

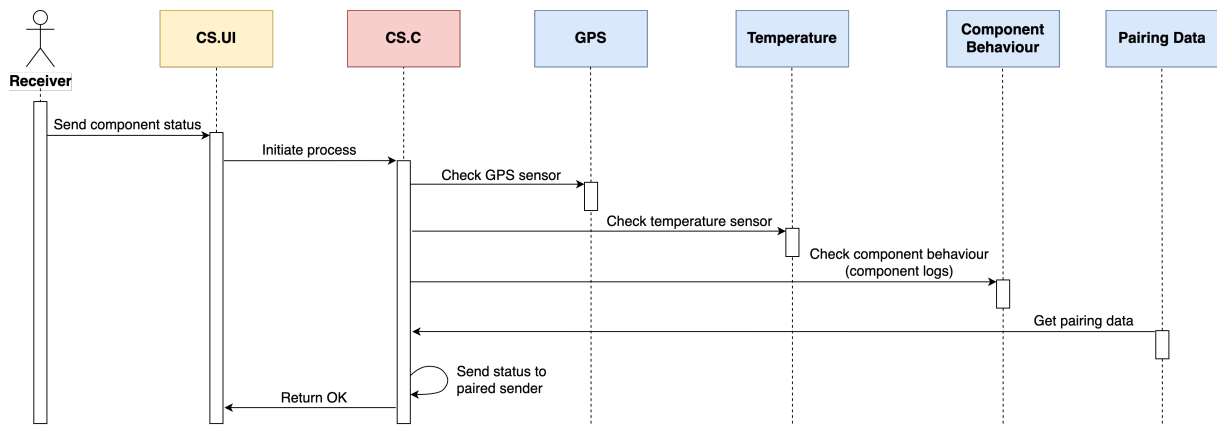


Figure 19: SD: Send Component Status

4. THE PROPOSAL

**Create team**

An important feature of the software application is the ability to create tactical teams, for the purpose of separating Special Force operators within the application and to assign operators to different missions.

There might be scenarios where there are multiple operators stationed within the same geographical area, but they might be on different missions out in the field. In such scenarios it is very important to separate these operators and group them into different tactical teams, so that there is no confusion when the team leader or mission planner is assigning tasks.

So even though the operators are located within the same geographical area, there should be no difficulty to identify which operators are assigned to a specific mission. It is usually the team leader that creates the tactical teams and it takes place before the mission starts, so that the operators are already assigned to teams when the mission starts.

The process of creating a team starts when the team leader enters a team name. The next step is to select which operators that should be assigned to the team, as well as setting the operator permissions to distinguish between high-ranked operators and other operators that should have less access in terms of the capability of arming and detonating receivers.

**SD: Create Team**

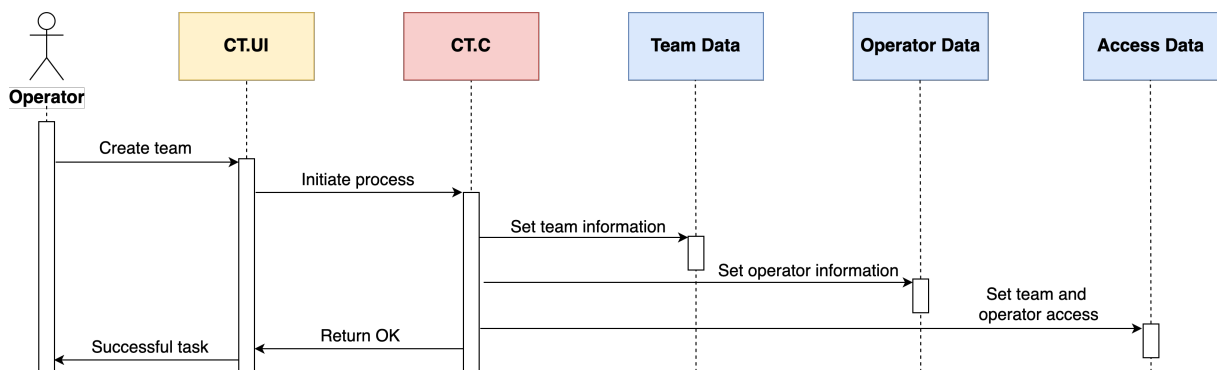


Figure 20: SD: Create Team

Operators from the same team can interact with each other, such as requesting or forwarding receiver control amongst each other, while operators from other teams are usually on other mission, and therefore it is not possible to engage with the other teams directly from the mobile application.

**Plan Mission**

Tactical military missions must be planned beforehand in order to save time out in the field, and to perform tactical operations as quick as possible. It is therefore possible to plan a mission in the software application, where the team leader can open the map and place waypoints at the locations where the receivers should be deployed.

These waypoints are small colored markers placed on the map, and these waypoints are supposed to help operators maneuver out in the field. The mission is assigned to a specific team, so that all team members can examine the mission details and prepare the tactical operations.

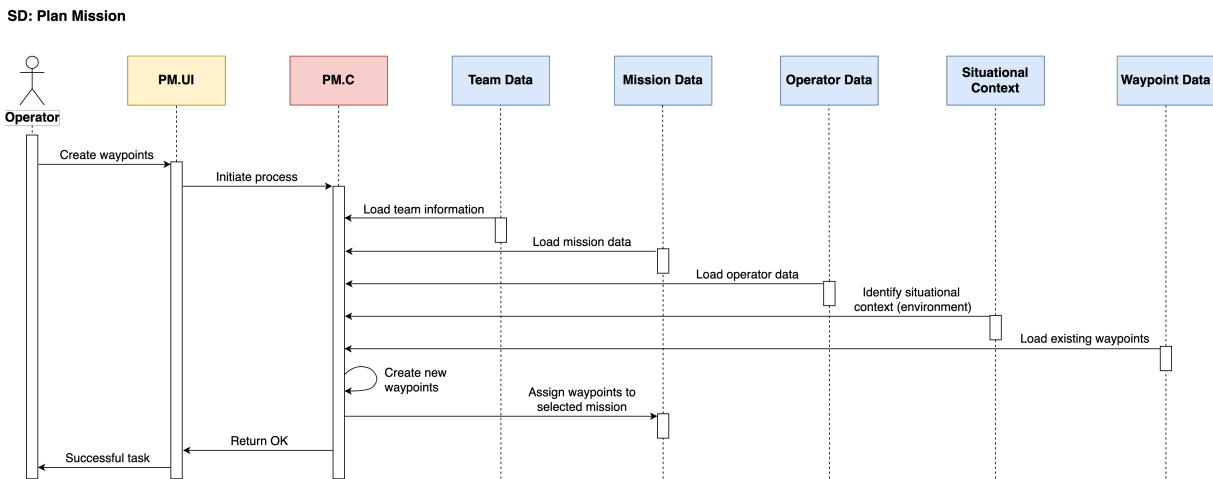


Figure 21: SD: Plan Mission

Figure 21 shows the process of planning a mission from the operator perspective. The entire process starts with deploying waypoints on the map, and ends with assigning these waypoints to a mission, which is available for the entire team.



**Set Access**

In order to have separated roles within the software application with different access levels in terms of permission to arm and detonate the explosives, it is important to set operator access based on how much access one operator should have during a mission.

There is an access table which connects each operator with a role within the application, such as Operator, Team Leader and Administrator. An operator will only have access to operative tasks, such as arming the receiver, detonating and assigning waypoints. A team leader will have more administrative access, such as creating teams, planning missions and managing operators within the application. Administrators have top-level access and will mostly perform administrative tasks, such as checking application logs and statistics.

**SD: Set Access**

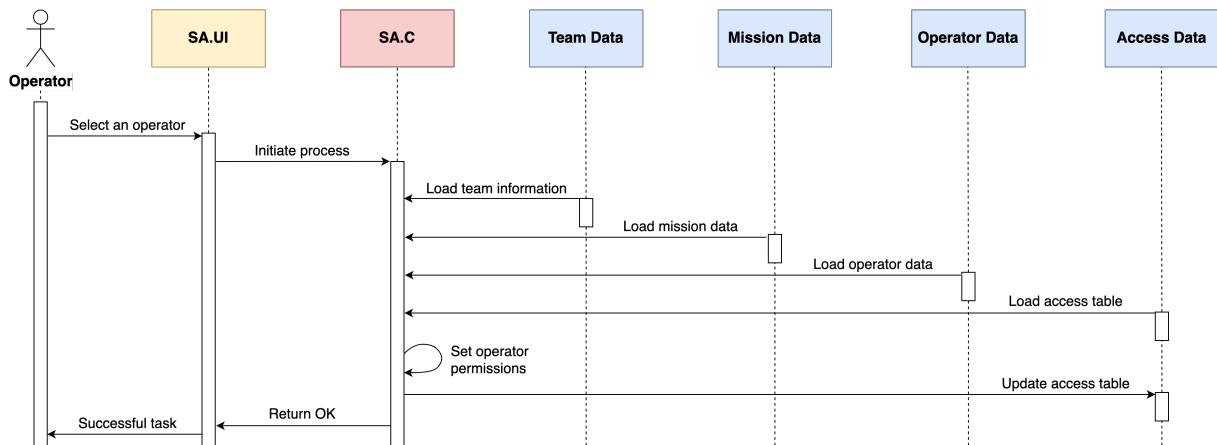


Figure 22: SD: Set Access

**User Stories****Authorize Action**

Operator Jones meets up on base with his squad. The squad is headed out on an EOD mission 5 km off base. The mission is planned to be executed daytime in sunny weather conditions. Operator Jones is a part of a 4-man team, where he will act as the EOD operator. All squad operators picks up a Samsung cell phone which they will use as a command & control device. Operator Jones also picks up a pack of 2 initiation receivers which he will be able to detonate from his phone. Operator Smith is the team leader of this squad, and powers up his phone as well. Operator Smith will pair his Samsung device with the main server, where he will extract information about the mission, the team he will work with and what skills the members of the team posses. Operator Smith also gives the go-ahead for all team members phones to extract mission and team information from the server. The server sets up the data, and defines roles and access privileges to each device, tagging operator Smith as the team leader, operator Jones as the dedicated EOD operator, and the rest of the team for other dedicated roles.

Operator Jones has set up his smartphone device for the mission, and turns to his 2 initiation receivers, powering them all on. Jones starts a pairing sequence with each receiver based on a unique serial code tied to every receiver, he picks them up one by one, and while keeping them close to his chest pocket where he keeps his phone, he is able to pair each receiver up within a few seconds. The pairing is successful because operator Jones' identity has been verified as an EOD personnel, and so is authorized to handle and configure the initiation receivers.

The team has reached the destination of the mission, and from his phone team leader Smith verifies the coordinates based on previous mission intel extracted from the server. The team leader changes the mode of operation for the squad to critical as the team enters a dangerous area tied to many IEDs placed in the ground. Based on the mission details, there has been located two IEDs which the EOD operator has been assigned to safely detonate, as to avoid harm to civilians.

Operator Jones has placed out his receivers close to the IED, ready for safe detonation when the phone suddenly stops working and shuts down, probably due to a bad battery and rough use over time. Team leader Smith is immediately notified on his device that Jones is no longer connected, and the system perform a "failover" claiming control over Jones' detonation receivers with the preset configuring and setup. The team completes the mission well, with Jones supervising Team Leader Smith and his Samsung device, ensuring a safe detonation despite the problems with Jones' device.

**Collect Data**

Operator Smith and his team are on a new mission 10 km off base. The mission will take place during midnight at an area marked as extremely hostile. They are set out to do a tactical rescue mission where they will have to enter a building with two entry points. From previous reconnaissance around the area, they know the two doors to be locked. Operator Jones is assigned to handle and breach open the two doors using C4 explosives and 1 detonation receiver per door so the rest of the team can enter. The doors will be breached simultaneously so that the tactical team will gain access from each endpoint concurrently. Each member of the squad is set up with a Samsung device where they can see a map of the location. Operator Jones has already paired up the receivers on base, and is seeing a good connection to each device, having battery levels ranging from 80-90 percent, this will easily suffice the time period until they are detonated. Jones lays out the receivers, and then marks a location waypoint on each door where the receivers are placed. He gets confirmation that the location has been extracted from a GPS satellite, and that his Samsung device has forwarded the waypoint to all team members connected on the local network. Operator Jones withdraws back to his fellow teammates and monitors the devices on the map on his phone, he assigns one waypoint the tag "Door North East", and the other "Door South" for his members to see. The rest of the team awaits Jones to arm the receivers ready for detonation. As Jones arms the receivers, the internal components starts generating energy, ready to ignite the fuses. Both waypoints changes color to red, as to clearly see that the detonation command will be sent out within a short period. The receivers are all done with the arming sequence, Jones is now ready to detonate.

Team Leader Smith has now full overview of the placed receivers and their current state. Jones suddenly receives new critical intel about the location from central command, and must pause the breaching. Smith shares the message with his team. Operator Jones complies, but all the receivers change their "Armed" state back to "Standby" mode regardless, as the time limit has passed for how long a receiver is allowed to be armed, and the energy charge is released. The waypoint on all Samsung devices has been turned back to color green, and the state on all receivers clearly says "Standby" except for one receiver which clearly indicates yellow for detected "Fault". The internal diagnostics system of the receiver has detected an error and notified Jones' Device of the problem.

**Archive Data**

The mission gets canceled from central command who observers the mission from base, and Jones packs down all the receivers. Despite being dedicated for "One time use only" the receivers were not detonated this time. The faulty receiver may also give new technical insight when troubleshooting the root cause of failure. On base the team connects their devices to the server, which does an automated uploading of the mission results, locations,

## 4. THE PROPOSAL

---

time spent on object with relevant data tied to external events and operator decisions. The Samsung device will contain logs from itself and the deployed receivers for usage statistics, network timing latency and receiver diagnostics. Including the faulty receiver which turned yellow, going into error mode when trying to switch back from an "armed" state.

### **Make Decision**

At the big city police station, Special Weapons and Tactics (SWAT) leader Filippo and his team has made good progress on the counter terrorist training they do on a frequent basis. The training involves breaching doors and disposal of IED's for search and rescue mission and as a first response tactical team on terrorist scenarios tied to urban areas. Officer Jenkins is a SWAT officer assigned as a demolition engineer for the teams EOD and breach missions. Reports come in to the station about a hostage situation in the city in a closed down factory building. Filippo's SWAT team rushes out along with officer Jenkins. On site, Jenkins gets an overview of the situation using the GPS map on his Samsung phone, and starts planning the breaching of the building with his team leader. He places out 4 receivers on the main entry door along with C4 explosives. He falls back to position and views the status of all the deployed units. The connection is stable and well. He places out a waypoint on the location, and his Samsung phone forwards in to all unit leaders on premises so that they can monitor the event. The SWAT team is waiting for the go-ahead. Jenkins receives order to fire when ready. He Arms the devices which instantly responds with a red color waypoint on screen. The waypoint starts blinking yellow, indicating an error. Jenkins notifies his team leader and halts the mission. Officer Jenkins disarms the devices and rushes to the door. He sees that the receiver were not mounted properly and 2 of them fell to the ground. He quickly mounts the up again correctly and falls back with the rest of the response team. Filippo gives the signal and Jenkins arms his devices once again. The detonation goes off, and the door breaches open, the SWAT team enters the building. The mission was a success, despite the problem with the receivers, offices Jenkins was able to act fast and agile, fixing the problem within seconds thanks to sensors integrated on the receiver which notified officer Jenkins, enabling a quick and responsive rescue mission for the rest of the team.

### 4.1.3.2 Expanding Excerpts of the Proposed Software Architectural Model

In this section we show the path of discovering more use cases for the initial base use case named “Decision Making”.

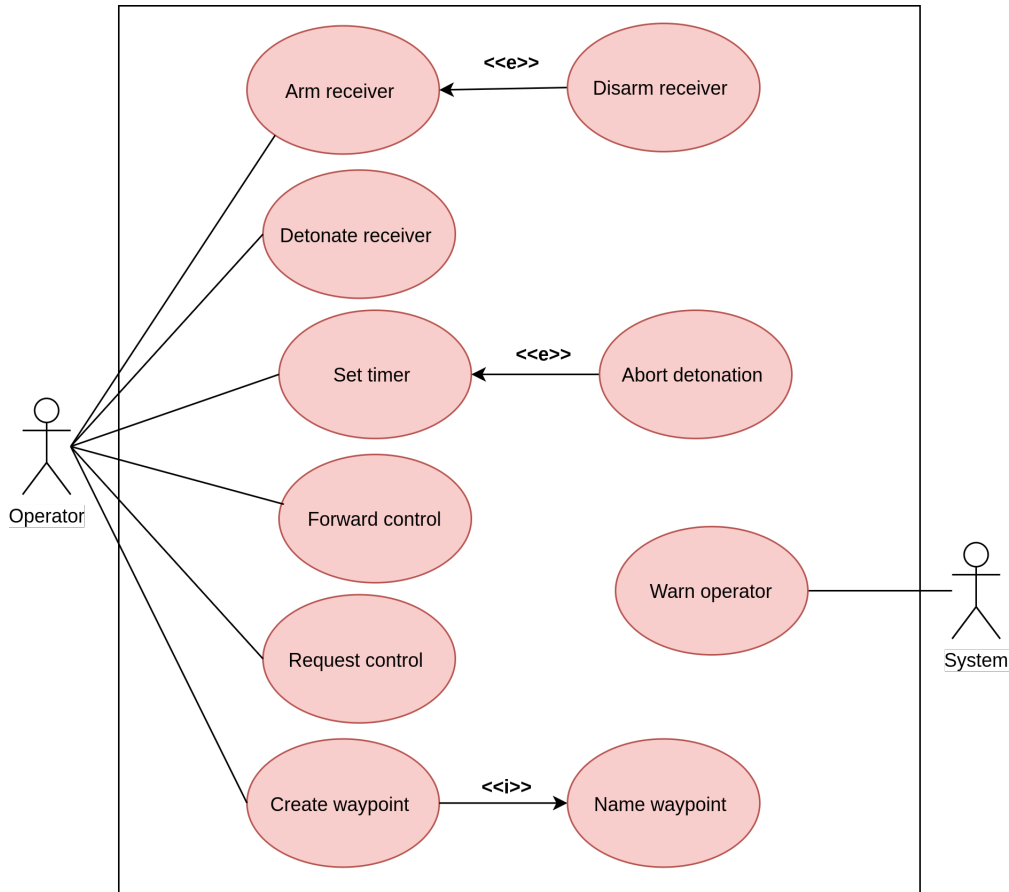


Figure 23: Expansion of Make Decision Use Case

Our vision of **Decision Making** is to capture and express tactical decision making of an end user of the system. This involves critical decision making during operations in the field or a mission, reacting to context of the environment and external factors which can drastically alter or inhibit the task at hand for the operator.

Its uses cases can be described as follows:

- **Arm receiver** - Arm the receiver, which puts it in a state wherein it can be detonated subsequently.
  - extends to **Disarm receiver** - Disarm the receiver, which puts it back into a standby state, wherein it cannot be detonated.
- **Detonate receiver** - Initiate the energy transference from the receiver that will set off the explosive compound it's attached to. Must be armed beforehand.

#### 4. THE PROPOSAL

---

- **Set timer** - Schedule the detonation to occur at a later time (e.g. in 60 minutes).
  - extends to **Abort detonation** - Abort the scheduled detonation.
- **Forward control** - Forward the control of a receiver to another sender in the same unit.
- **Request control** - Request control of a receiver from another operator in the same unit.
- **Create waypoint** - Create a waypoint on the map displaying the location and state of a deployed receiver.
  - includes **Name waypoint** - Once a waypoint has been created, it must be named to make it distinguishable from other receivers deployed in same area.
- **Warn operator** - Based on sensor data from the receiver, display a warning to the operator to aid in decision-making. For example, can be vibrations, sound, etc.

The following 10 sequence diagrams complements the use case model, revealing the data dependencies for each scenario. Fig 24, 25, 26 and 27, 28 consists of the actions of operating the initiation sequence of a receiver unit (which is to say, tell it to arm, disarm, detonate, or detonate on a timer) and thus are similar in message passing.

Figure 30 and 29 shows the handover transaction exchanging controls of receiver units. This is in the event that an operator would like to either request the control of receivers from another operator in their team, or forward their own control to someone else in the team. Once control has been gained by another team member, they can at that point arm it, disarm it, detonate it, and so on.

Figure 31 shows the computation of sensor data, and in the event of a critical context, change, or disruption in environment, emits a warning to the operator. This aids them in their decision making, as they have a better understanding of their surroundings at all times.

Figure 32 and 33 conveys the process of placing visual markers on a shared map when deploying a receiver unit in the field or before a mission. This way, all members in the team are able to see the location of all deployed receivers.

**SD: Arm Receiver**

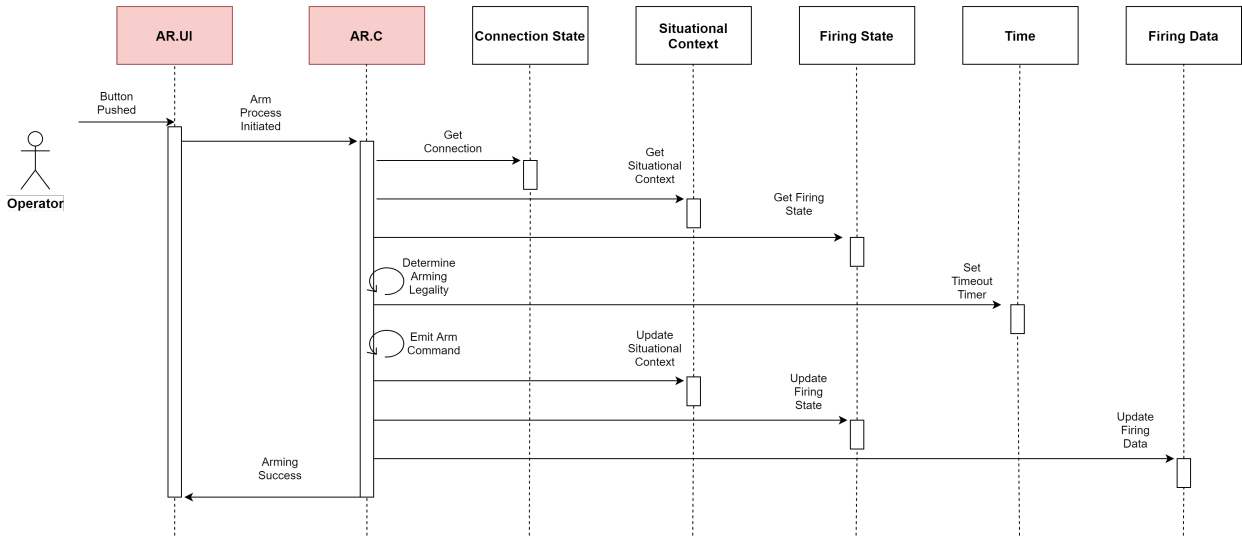


Figure 24: Arm Receiver Sequence Diagram

**SD: Disarm Receiver**

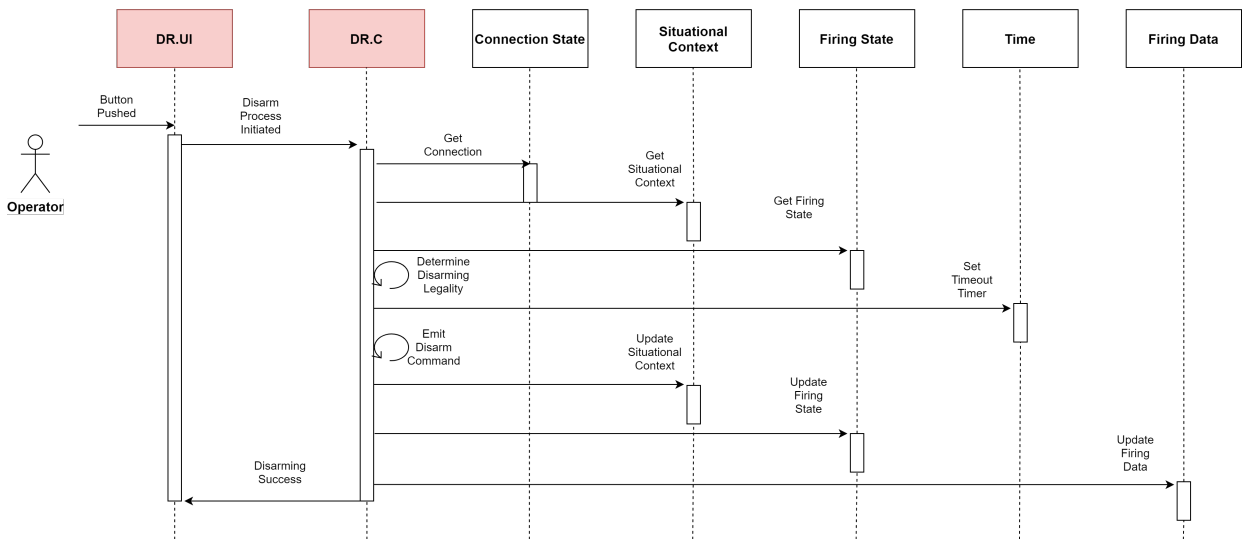


Figure 25: Disarm Receiver Sequence Diagram

SD: Detonate Receiver

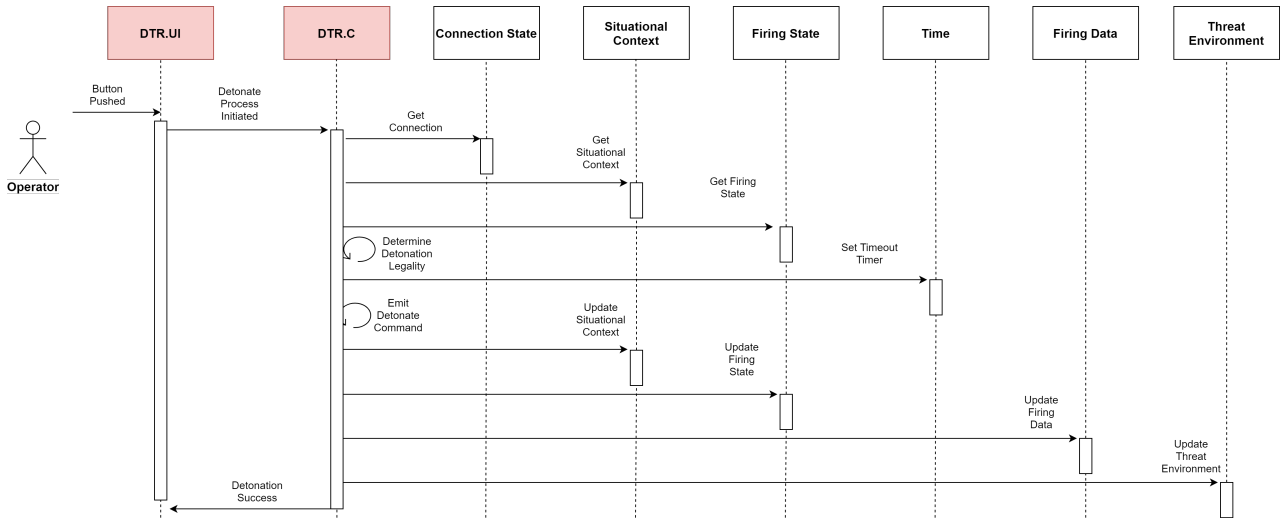


Figure 26: Detonate Receiver Sequence Diagram

SD: Set Timer

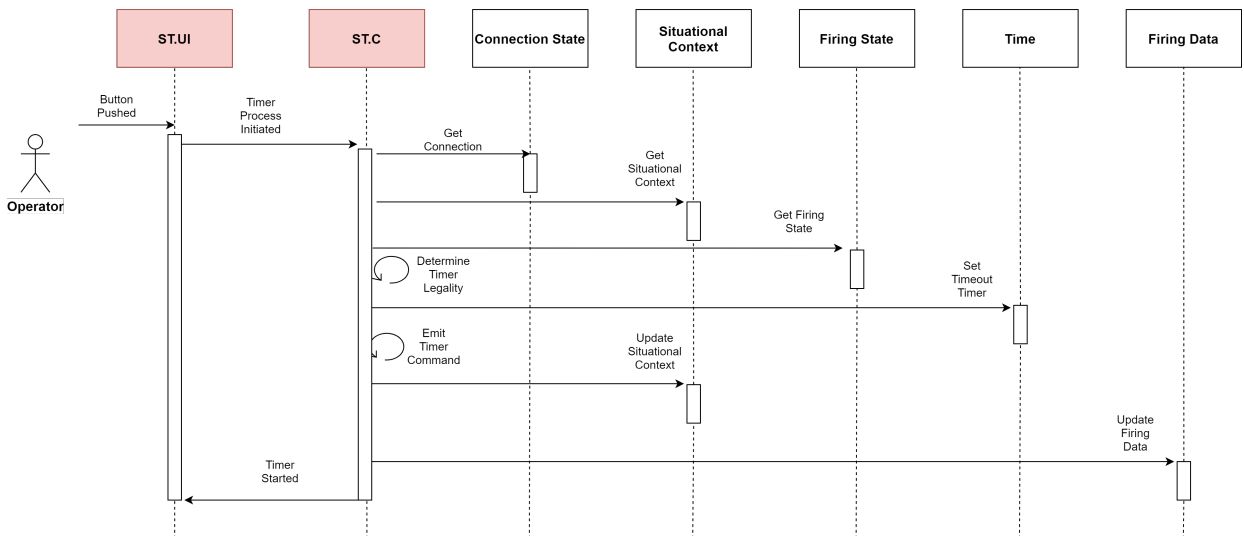


Figure 27: Set Timer Sequence Diagram



SD: Abort Detonation

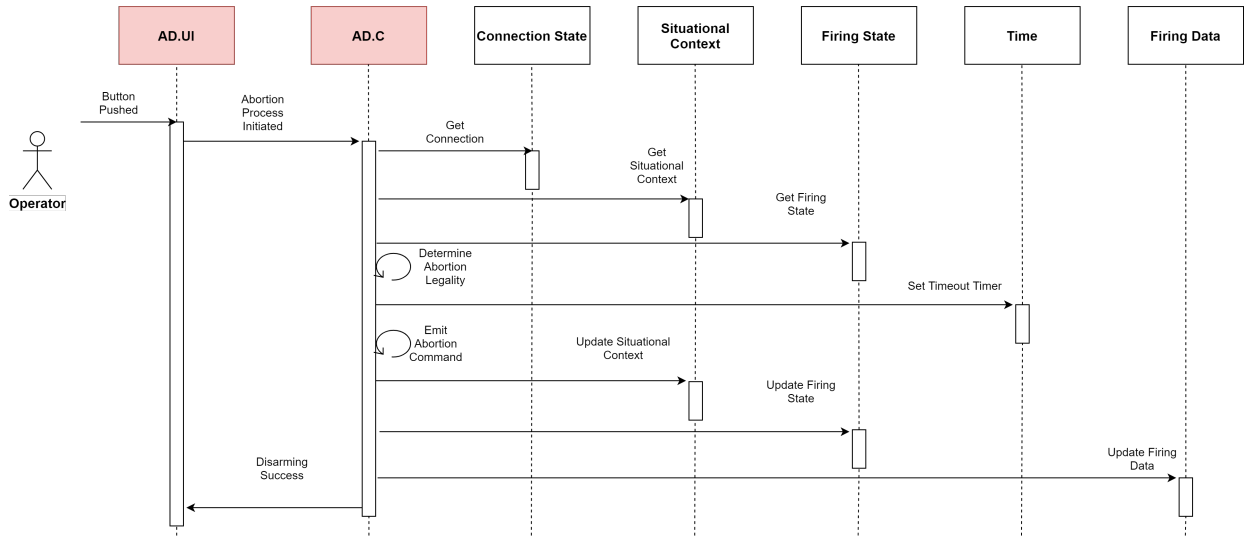


Figure 28: Abort Detonation Sequence Diagram

SD: Forward Control

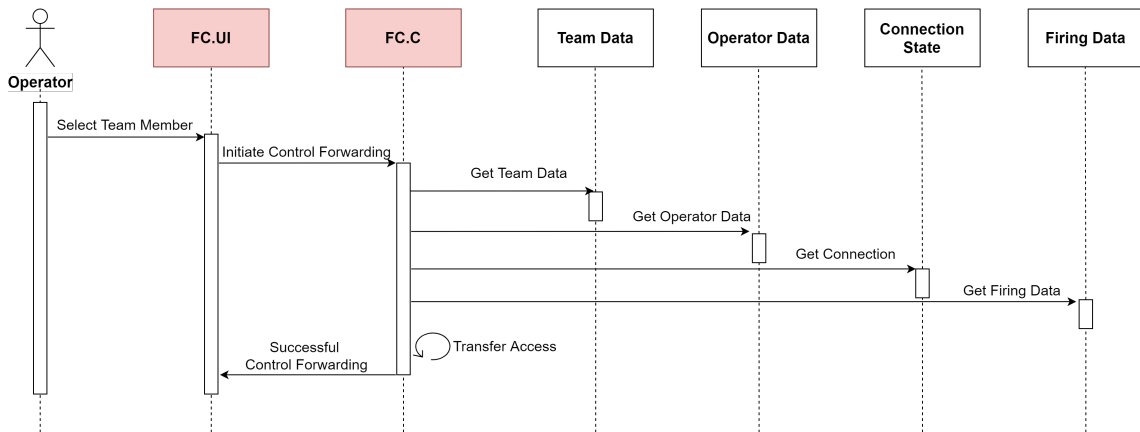


Figure 29: Forward Control Sequence Diagram

SD: Request Control

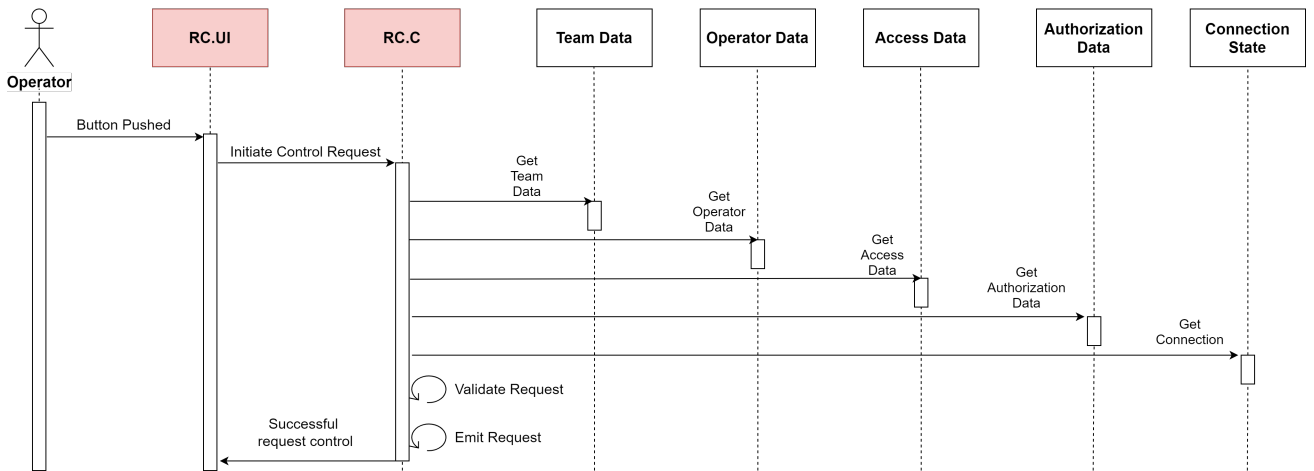


Figure 30: Request Control Sequence Diagram

SD: Warn Operator

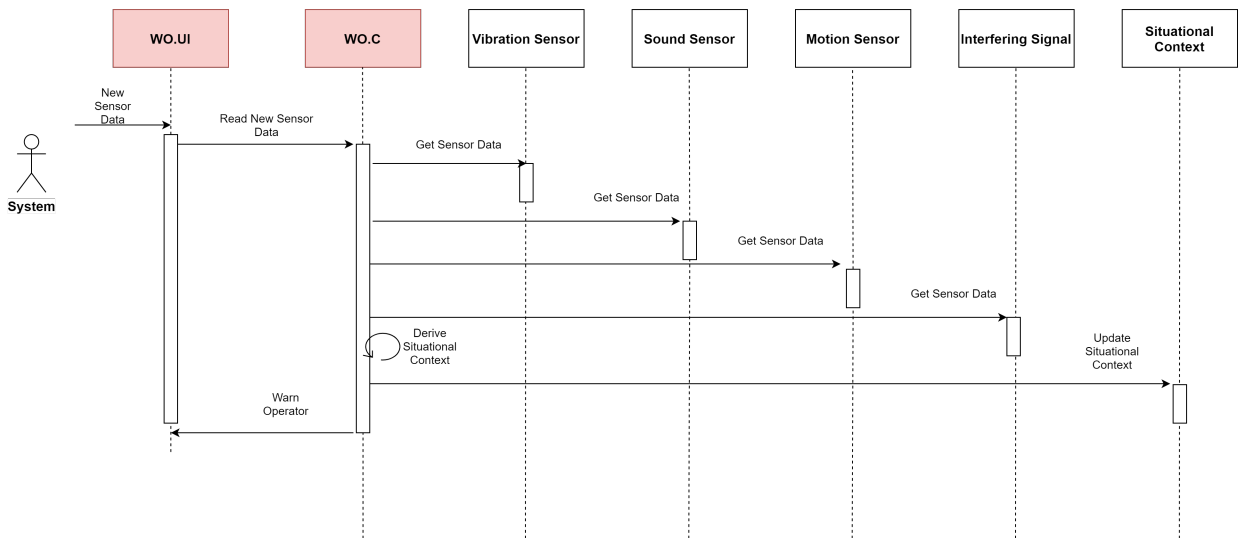


Figure 31: Warn Operator Sequence Diagram

SD: Create Waypoint

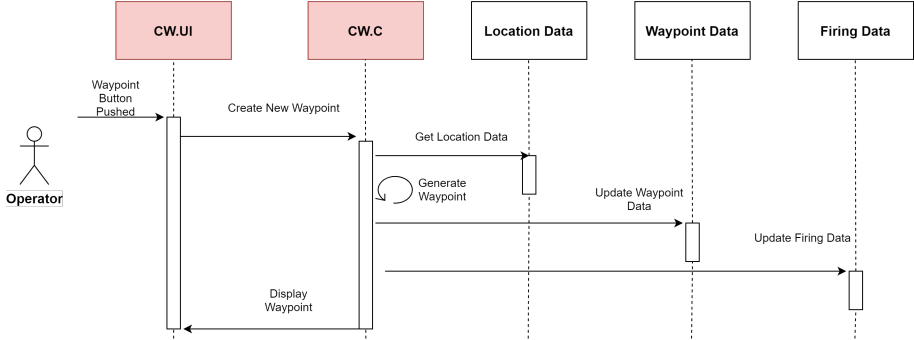


Figure 32: Create Waypoint Sequence Diagram

SD: Name Waypoint

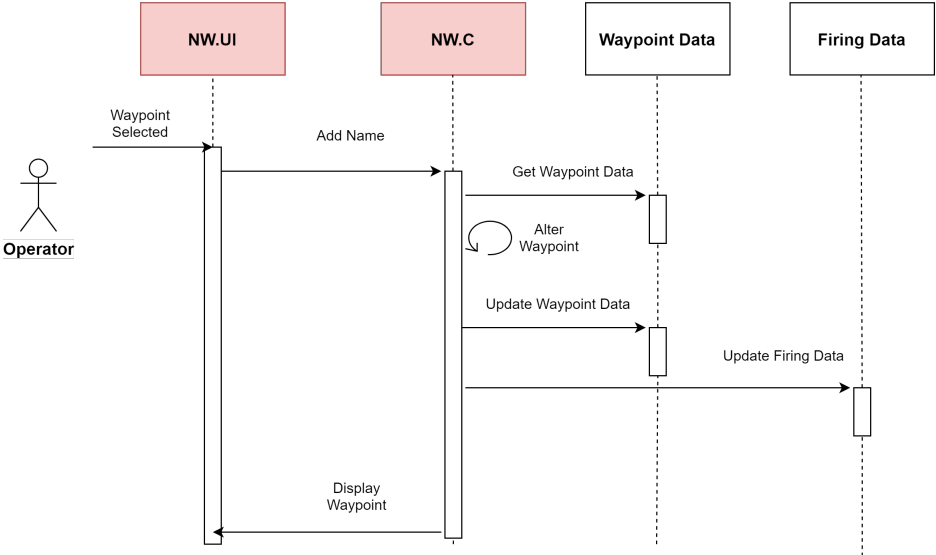


Figure 33: Name Waypoint Sequence Diagram

4. THE PROPOSAL

4.1.3.3 Example of the Section of the Final Software Architectural Model

A set of sequence diagrams from the previous section create an architectural model for the implementation of functionality defined in the Decision Making use case, which is shown in Figure 23. This “mini” MVC architectural model for Decision Making has also been identified in the overall architectural proposal 8.7, which is given in Figure 34. Therefore Figure 34 is a starting point towards the implementation of the excerpts from the conceptual proposal. It is interesting to note that software components which store data in Figure 34 show significant overlapping with the same components in Figure 8.7 from appendix. The only difference between Figure 34 and 8.7 is the level of precision when showing the functionalities to be implemented through computing programs (the middle layer of the software architecture).

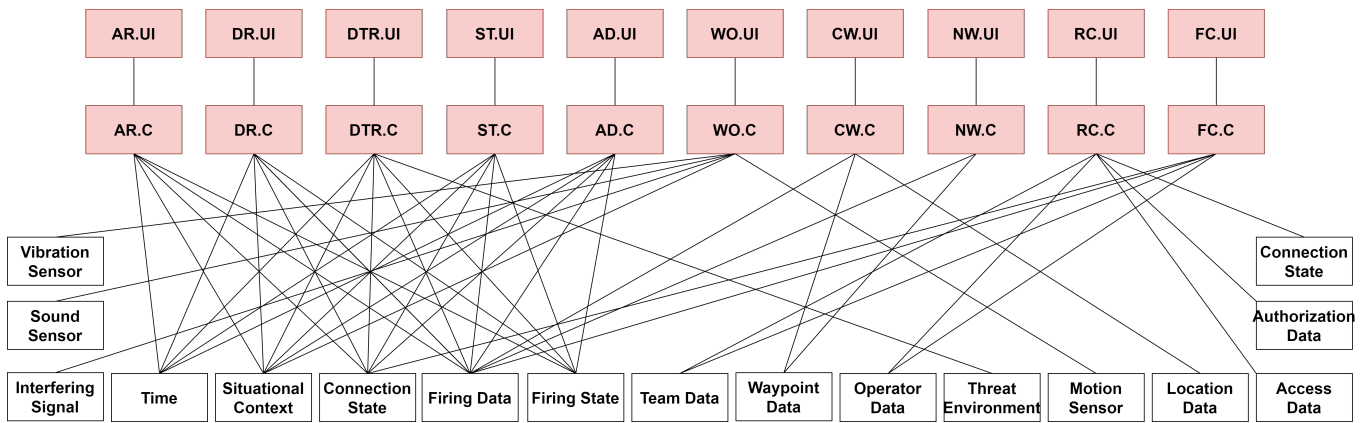


Figure 34: Component Based Architecture of Make Decision

#### 4.1.4 Paving the Way Towards Implementation

To realize our software proposal, we code using the QT framework which provides MVC software solutions [115] utilizing the QML programming language [119] for UI development and C++ language for controller software [118]. The QT framework provides their own integrated, IDE Qt Creator [114] which enables us to keep all components of the MVC pattern in the very same code base. The QT project itself offers a vast and highly detailed documentation base to aid in software development [116].

Additionally, the QT platform offers support for multiple platforms using the very same code base, allowing us to deploy our solutions to various systems and devices most suitable for our solution without altering the code, this includes Linux, Windows and Android devices [117]. These are all desired qualities which enables us to take advantage of the speed of C++ combined with native graphical user interface development in the very same environment, as well as allowing us to work and test our solution on different end systems. As the project grows in complexity, performance requirements change and product vision evolves, we believe QT will offer us a highly scalable environment for our solution to thrive in.

#### 4.1.4.1 Ad Hoc Networking

It is noted that the following subsections (addressing WiFi Direct, Bluetooth, and DDS/ROS) are purely theoretical with no practical application for the proof of concept. These subsections are meant to present a few ad hoc network solutions that could be implemented with the Spark RFS in the future, provided with enough time and resources to do so.

The use case for our system entails a fully functioning system in an area with no to minimal network infrastructure, off the grid and on the move. Operating in such an environment, passing messages between devices must happen without relying on centralized servers nor external routing. The end user of this system will need on demand network access whenever and wherever they go. These constraints introduce the following challenges:

- Each network node must be able to reach another node by themselves
- Each network node must maintain a robust, stable connection to each node, despite being actively on the move
- Each network node must be highly adaptive to changes occurring in the network topology
- A break or interrupt in connection must result in quick pairing and reconnect
- The Ad Hoc network must have low latency to provide real time update on critical data
- The Ad Hoc network must facilitate enough bandwidth for the packet transmission between nodes
- The Ad Hoc network must employ a method of avoiding possible packet collision

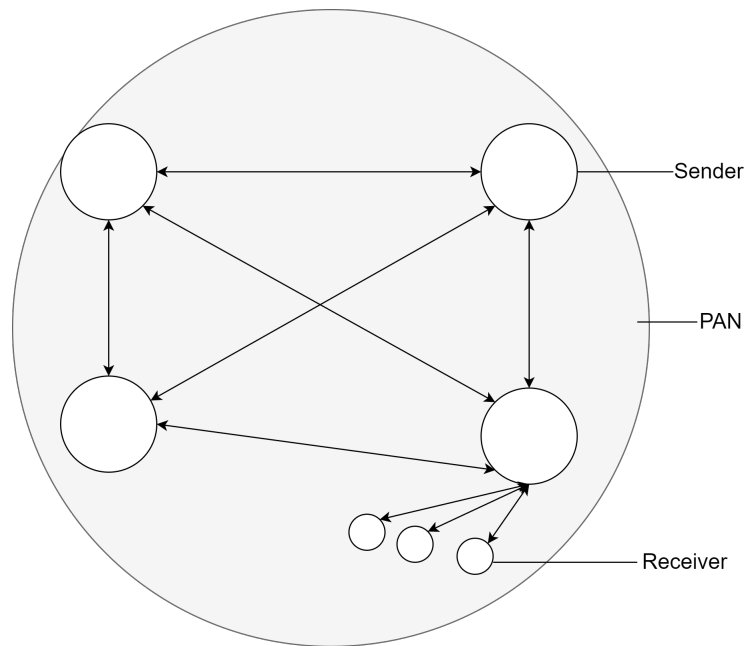


Figure 35: Many-to-Many Communication

In a pursuit of viable options to implement such a system, one would need an Ad Hoc solution to facilitate the transmission of data between nodes without dedicated entities to store data or route packets. A symmetric environment where each node keeps responsibility of receiving and transmitting data. From figure 35 we see an ideal scenario with a many-to-many relationship between each peer. All senders are able to reach their peer within a single hop. A dedicated sender maintains a connection with all deployed receivers.

Possible constraints in range and connectivity may prove to be a problem. A single hop solution could prove difficult in practice when nodes increase their distance or lose line of sight. An alternative design would be a peer-to-peer multi-hop scheme where each node maintains a personal routing table, and upon request shares this, allowing forwarding of packets on behalf of another node in the group. Acting as an intermediary routing point.

Possible technologies within Mobile Ad Hoc Networks (MANET):

- Bluetooth
- WIFI Direct

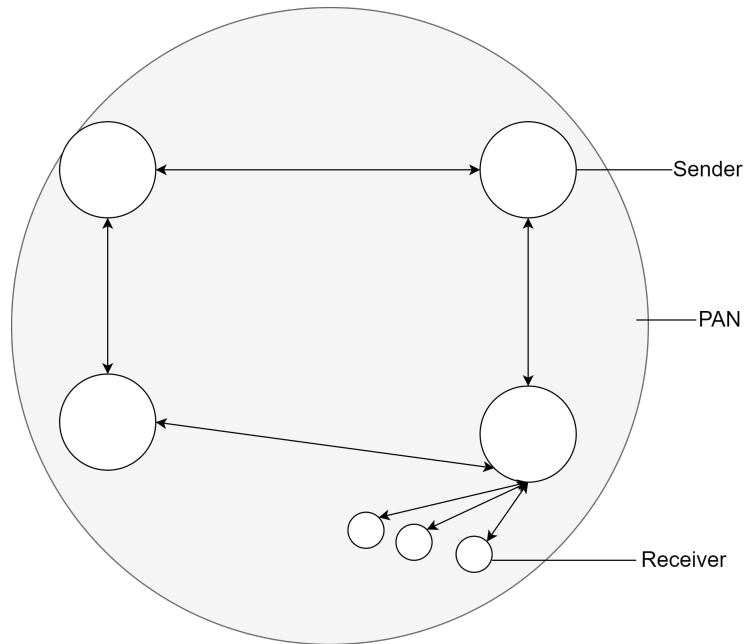


Figure 36: Multi-hop Routing Scheme

### WiFi Direct

WiFi Direct is a relatively new standard for peer-to-peer network setups enabling exchange of data without a centralized router or wireless access point[7]. WiFi Direct enables peer-to-peer network communication, by setting a mobile device to act as a software access point for all nearby devices. This access point forms a peer-to-peer group, allowing the exchange packets to all devices asking to connect to the access point. The initial device becomes the *Group Owner* and each connecting node becomes a *Group Client* [21]. The nodes address each other using their MAC-addresses as their GUID [21]<sup>1</sup>. WiFi Direct offers a bandwidth of up to 250 MBS [6], and a range of up to 200 meters [5], which looks promising for a remote data intensive system. WiFi Direct API is available for Android Development for software releases from 4.0 and later [8], which aligns with our requirements for our application to run in an Android environment.

---

<sup>1</sup>Globally Unique ID



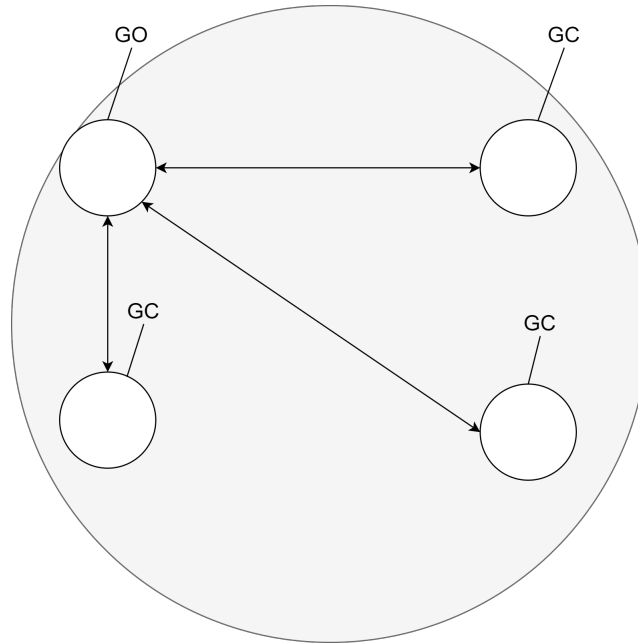


Figure 37: P2P Group in Wifi Direct

From figure 37 we see a standard peer-to-peer group, one sender becomes the GO, the remaining becomes a GC. Comparing this with our desired solution, we would lack the intercommunication between Group Clients, as each GC is only allowed direct communication with the Group Owner. This could be solved with the dedicated node carrying EOD privileges becoming a GO. This would put restrictions on for instance the required handover scheme, reducing the safety of the system. A senders safety requirements necessitates the option to do a handover of all EOD controls to a third party node if technical problems occur. Consequently, a solution based on WiFi direct straight out of the box would likely not be applicable here.

From [67] we see an attempt to utilize the functionality of this technology using the Android API, as well as bypassing the restriction of only allowing single-hop communication with the G0 of the group. The method in practice is to let each node in a network go back to being a G0 of their own temporary network when idle and no data is transmitted. This allows for full visibility of all network nodes, and allows for a multi-hop routing scheme based on the device's mac addresses, and pairing protocol in WiFi Direct. The results identifies some challenges, such as an unstable peer discovery due to frequent switching of Group Owners, possible packet collisions when flooding<sup>2</sup> the network for nearby node discovery when forming a personal routing table. To remedy this, Kecheng Liu *et al.* suggests a more advanced routing scheme: repetitive discovery of peers to reduce pairing

<sup>2</sup>Forwarding a packet from one router to all nearby except for the one who sent the initial packet. Allows for nearby node discovery in a network

overhead and a scheduling setup in data transmission to reduce collisions.

This approach seems to work but entails some overhead and can prove unstable in forming connections and re-connections with peers on the network. It may work for small group setups but could introduce scaling problems when adding more devices. Per [79] from S. L. Meshram and P. D. Dorge, we see a comparison of possible routing algorithms AODV<sup>3</sup> and AOMDV<sup>4</sup> suggesting the former for networks requiring lower latency. Concluding that a lot of the viability in a MANET scheme relies on having a robust routing protocol. In regards to connectivity, the WiFi Direct pairing processing initially requires user interaction when connecting to a peer. Findings from [29] introduces three different attempts in automated and self monitoring systems. Based on metrics such as Group Owner Selection Time and Total Group Reformation the researchers tried to find the optimal method of connecting nodes in a network seamlessly on Android.

From [64] we see similar attempts at solving the restrictions of WiFi Direct's single hop communication. J. H. Lee *et al.* Proposes a system comprising of a dedicated routing layer on all mobile devices within a group, enabling multi hop communication between nodes. The system facilitates routing by periodically broadcasting each node's routing table, so that nearby nodes can discover neighbours of neighbour nodes. This seems to further support the possibility of a multi-hop scheme in WiFi Direct, but does not explicitly mention possibilities of pairing challenges nor possible routing difficulties. Further research from [56] has an interesting approach with multi groups in concurrent mode acting as both a GO and GC, but [64] points out that this is not compliant to the standard. Although this is not explicitly described per [21], and that concurrent mode is meant for a possible peer-to-peer interface along a conventional WLAN interface. Yet, section 2.3 and 2.4 of the standard does say it does not preclude from a possible multi group peer-to-peer setup.

These findings, we believe, builds a promising case for WiFi Direct as our network solution for SPARK. Both bandwidth and range are within our desired criteria given that a multi hop scheme is feasible, yet routing and connectivity challenges must be handled in an intelligent manner and ensure an adaptive and dynamic system which can handle nodes on the go jumping in and out of range while still providing low latency, high uptime and an uncomplicated pairing process.

---

<sup>3</sup>Ad Hoc on-Demand Distance Vector Routing Protocol

<sup>4</sup>Ad Hoc on-Demand Multipath Distance Vector Routing Protocol

## Bluetooth

Bluetooth was the very first technology we considered to solve our ad hoc network communication, as our customer initially voiced this as a possibility when presenting the project to us. Bluetooth facilitates short-range data communication using radio links between two or more devices, and has since its inception become one of the most popular means of short-range wireless communication [73, 26]. This technology exhibits many of the traits we deem desirable for our solution, such as the ability to connect in an ad hoc manner, asynchronous data links, and low chip cost (around 100 nkr) [126].

The way Bluetooth works is that two to eight devices (more in the case of Bluetooth LE [26]) will connect and form a piconet in an ad-hoc manner, of which will consist of one master and one to eight active slaves. Additionally, there can be up to 255 parked slaves that are a part of the piconet, but does not actively participate. A master in one piconet may be a slave in another one, which would make this particular device function as a bridge between these two piconets, and can then relay data between the members of both piconet networks, forming a scatternet [126]. In Figure 38, the grey circles represents one piconet each, the red "S" circles represents slaves, the yellow "M" circle represents a master, the blue "P" circles represents parked slaves, and the "M/S" circle represents a device that is a slave in one piconet, and a master in another, serving as a bridge, and forming a scatternet.

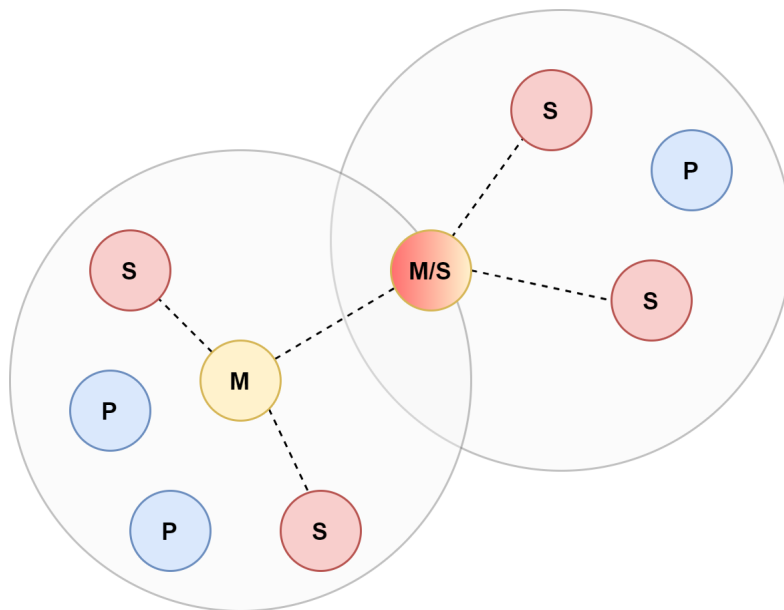


Figure 38: Bluetooth Scatternet (inspired by [126])

#### 4. THE PROPOSAL

---

To initiate a connection, a device will leave the standby state (the default, low-power state) to either initiate or receive an inquiry or page command. The former would be the case if the address of the targeted device is unknown, and must subsequently be followed by a page command, and the latter would be the case if the address is known already; it's the page command that connects the devices [73, 126]. Once a connection has been formed, the slave will synchronize to the master's clock and frequency-hopping pattern. The connection will utilize the 2.4GHz signal range.

Frequency hopping is a necessity for Bluetooth networks as they operate in the unlicensed ISM frequency band, which is occupied by a number of signals from other types of devices such as garage door openers and microwave ovens [126]. If a Bluetooth piconet did not have frequency hopping, these interfering signals would disturb the communication in the network, which would make it severely unreliable and undesirable for a vast majority of applications. Each piconet will have its own frequency-hopping pattern, consisting of 1600 different frequencies it will hop through each second. Upon connecting with a master, all slaves will agree on the pattern set by the master, and be assigned a unique time slot for transmitting [26, 126].

Concisely, Bluetooth provides three means of security: authentication, confidentiality, and authorization. For authentication, the devices will calculate a response based on randomly generated numbers (including a secret link key) and the slave's unique address. If the responses do not match, authentication fails. For confidentiality, a stream cipher called "Eo" is used. For authorization, a local database is checked on the device to determine if a device has previously been authorized and labeled as trusted. If not, trust must be established first [26].

Bluetooth presents a number of problems even in its latest version that renders it undesirable for military applications, particularly in the areas of security [26, 9], range [73, 126], and battery life [71], although the latter two have seen significant improvements ever since version 5.0.

According to [71], version 5.0 is capable of a range of 200 meters outdoors, provided it's unobstructed, and about 40 meters indoors, which would indeed suffice for our application. Furthermore, despite the extended range and speed of Bluetooth 5.0, the energy consumption has not increased – on the contrary, it has decreased thanks to some ingenious designing, and can in the best case consume about two times less power than the previous version (4.0).

#### 4. THE PROPOSAL

---

Another problem, listed in [73], is that of the amount of time and effort that goes into a successful, consistent pairing. However, since we employ NFC for our pairing, initiating a Bluetooth connection on a successful NFC read would likely minimize this problem considerably. Finally, according to [26], the fact that Bluetooth shares the 2.4 GHz radio frequency spectrum with a number of other appliances could result in interference with these devices. Due to the frequency-hopping scheme however, this interference would only last for 1/1600th of a second, as the piconet would remain in that particular frequency for only that amount of time before hopping again.

It appears, then, that a number of the most prominent problems attributed to Bluetooth would not have a significant impact on our solution (in practice, the range problem would likely present challenges related to interior walls). There is, however, a critical flaw of Bluetooth that severely impacts its desirability for our use: its security.

A very brief overview of Bluetooth's security was introduced earlier, wherein one of the measures was that of authentication. Bluetooth allows for an unlimited number of authentication challenges, which means an attacker can collect a significant number of challenge responses, which they could in-turn use to obtain information about the secret link key. Furthermore, the storage of link keys must be carefully considered to prevent an attacker from viewing or modifying them. Finally, this form of authentication is related to the device only, rather than its user; if someone were to obtain the device, they would have full access to the network [26]. Consequently, if Bluetooth is to be considered, some manner of external authentication would be a necessity.

The confidentiality security measure of Bluetooth also presents a number of vulnerabilities, namely related to its cipher and encryption key. In general, the stream cipher E0 is considered to be weak as a result of its simplicity. Additionally, the encryption key used in the cipher can be as small as 1 byte [26]; in [9] they present an attack on Bluetooth's encryption key as recently as 2019. This attack, named "Key Negotiation of Bluetooth" (KNOB) attack, allows a third party to make two victims agree on an easy, 1 byte encryption key, from which the third party can gain access to the communication. In [26], they were also able to perform spectrum analysis, packet sniffing and packet decoding using a few easily accessible tools.

Finally, [26] also lists a number of other vulnerabilities, such as jamming, packet dropping, wormholes, and localization. In conclusion, it's evident that if we were to choose Bluetooth as our communication technology, we would have to dive even deeper into the literature in order to discover the means of mitigating all of these vulnerabilities.

### **Data Distribution Service (DDS)**

The Data Distribution Service (DDS) is a protocol and API standard for integrating system components together, providing low-latency data connectivity and communication between the nodes within a publish and subscribe-based messaging architecture [45].

This architecture provides a system for exchanging information between modules in a publish-subscribe (PS) system. This PS model is based on information producers (also called publishers) that are connected to information consumers (also called subscribers) [97]. DDS is often preferred in tactical military environments, such as air-traffic control and autonomous vehicles. There are three main entities in the DDS architecture [4]:

- Topics (T) are data buses or pipelines where nodes can exchange messages.
- Publishers (P) defines the data sources, and it publishes messages to appropriate subscribers who have subscribed to a given topic.
- Subscribers (S) reads a given topic and thus receives the messages from the publisher based on which topic it's subscribed to.

### **Robot Operating System (ROS)**

Robot Operating System (ROS) 2 is a set of software libraries built in order to implement DDS in an actual application. The communication within ROS 2 is based on the DDS architecture, and it uses the same publish-subscribe pattern to transfer data in-between different nodes. This makes ROS 2 perfect for industrial applications, along with Internet of Things, since there is no centralized master node or server that the other nodes are depending on.

This provides a great peer-to-peer infrastructure that can be developed in C++, and thus it can be implemented into the mobile application of our remote firing system to establish a communication link between our senders and receivers.

## 4. THE PROPOSAL

### 4.1.4.2 Fault Tolerance

For a mission-critical system that cannot afford to be rendered unresponsive, some manner of fault tolerance or high availability functionality is imperative. Fault tolerance, in its simplest form, is a way of ensuring that despite the occurrence of some failure event within a system, it's still able to operate properly [2, p. 491]. According to [2, p. 491], there exists a few different techniques to facilitate fault tolerance, such as: replication, where several identical instances of the failed system receives the tasks meant for the failed system; redundancy, where an identical instance of the failed system is switched to (failover); and diversity, where several different implementations of the same failed system is used.

Consider Figure 39. Here, the receivers R1 and R2 have been deployed by an operator with transmitter T1. These receivers have been paired and maintains a wireless, constant connection with its transmitter, and assures the operator of its health by periodically sending a heartbeat to their transmitter. The transmitter, in turn, will reply with an acknowledgement upon receiving the heartbeat.

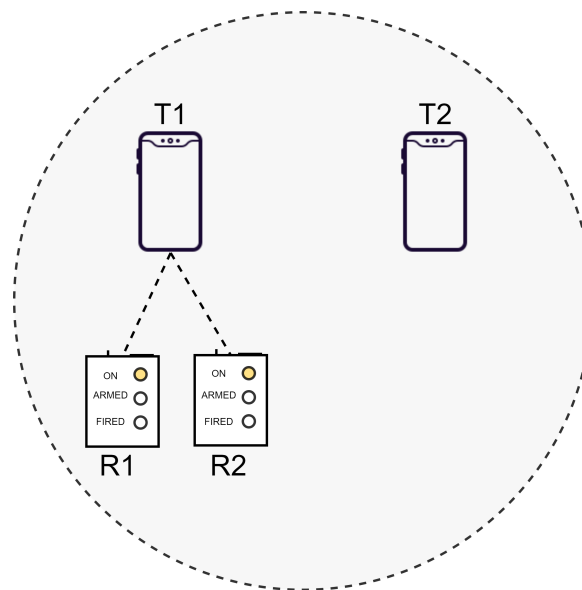


Figure 39: Transmitter / Receiver Network

Let's then assume that transmitter T1 becomes indisposed, perhaps because of physical damage, and is no longer capable of maintaining a connection with its receivers. Without some manner of fault tolerance, the receivers would send their heartbeats, receive no acknowledgement, and simply remain where they are, deployed in the field, with no easy means of being controlled by other members in the team. If receivers R1 and R2 were critical to the mission's success, it's possible the mission would fail as a result.

For our purpose, we believe some type of redundancy is the most appropriate solution. The reasoning for this is that we have several identical transmitters within any given group, all of which could handle the receiver communication. Specifically, we believe a failover technique wherein the receivers automatically determines a new host (transmitter) within its group to connect to would be desirable.

Considering the importance of constant receiver uptime and interactability, failover is of the utmost importance. Therefore, consider next Figure 40.

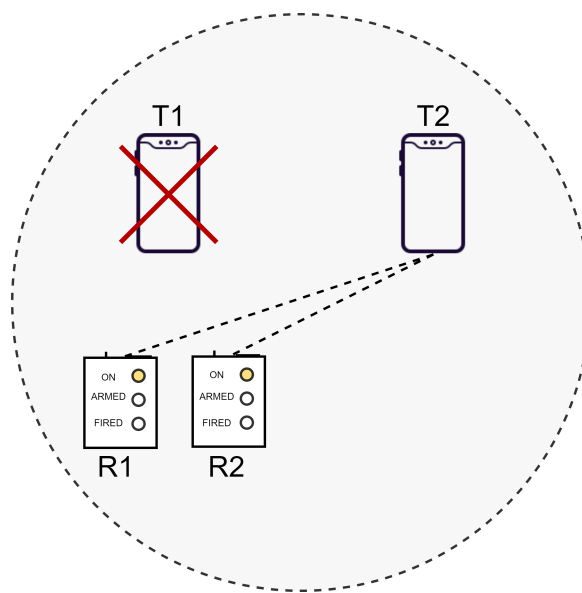


Figure 40: Receiver Failover

In the same scenario, T1 is indisposed, but with failover, the receivers are able to determine a new host (T2) and establish a connection, allowing the operator of T2 to control R1 and R2. Consequently, this operator can now arm and detonate the receivers as planned.



#### 4.1.5 Military Communication and Information Systems

The trend towards building open system architectures for military land vehicles and equipment has been increasing in the last decade, and is becoming more popular, as the demand for various military systems increases [84]. The more military systems are built on open-source technical standards, the more flexible and interoperable it is to integrate these systems, in order for one system to seamlessly communicate with another, especially when these military systems are deployed and operable in different nations around the world [92].

In cases of using manned or unmanned military land vehicles, flexible and open source system architectures is in high demand and may allow full data-exchange among military systems and any other source of information relevant to military operations. They may range from economical and financial to political data sources, which may impact any type of decision making in military operations. If data sharing and development of interoperable systems in finance, management, governance and healthcare proved to be feasible and helpful in making them efficient, then we could transfer the same ideas when developing software which support military. Consequently, this may provide improved situational awareness and enhanced combat proficiency, as it also saves the economic costs of developing separate and single-purpose military systems [92].

##### 4.1.5.1 Cloud Computing in Military Applications

Cloud computing has become a backbone technology in today's modern society, and it started being a significant trend in military platforms and environments, as well as in the public sector. Cloud computing can be described as a model for enabling on-demand network access to a variety of computing resources, such as mobile applications, servers and services [37]. Cloud computing is known to provide flexible and scalable software solutions, with increased performance, it is often quite affordable with practically unlimited storage capacity, since everything is hosted in the cloud instead of local storage.

There are obviously numerous benefits of cloud computing, but there are known disadvantages, which have been published in the literature.

First, the main threat of moving and using to cloud computing is definitely security, as it provides a security risk in terms of sharing sensitive information with a third-party hosting provider [37]. It is almost impossible to ensure that the data stored at the third-party hosting provider is completely secure and kept away from hackers.

#### 4. THE PROPOSAL

---

Second, if we know that cloud computing relies on good internet connectivity, we could assume that the lack or low quality of the internet connectivity is a huge problem in military environments, as battlefield environments are often unreliable due to a large number of movements and perhaps obstacles that are blocking the internet connection.

Third, information delivery is urgent in battlefield situations, as it must get through the low bandwidth radio links in the fastest time possible. It is considered as high priority in military environments [81]. Military radio systems are often characterized by a high amount of error rates, as it is often affected by temporary outages, such as terrain blocking of signals. A reliable information delivery system over these military channels are therefore highly essential, as it usually involves combat and the consequences of failure might result in loss of life, cause severe property damage or cause damage to the environment.

Fourth, in military context it is an absolute demand to provide fault tolerance and some sort of disaster recovery [143]. In battlefield situations and military tactical operations, any downtime is unacceptable and it must be prevented at all costs. This is due to the fact that downtime within the network prevents the operators from interacting with the military platform, and it might even disable the communication channel amongst the operators, which might lead to severe consequences in terms of mission progress and potentially result in loss of life.

Fifth, security is undoubtedly the greatest concern when it comes to adopting a cloud model in military context. The worries often derive from data confidentiality, encryption needs and data management. However a private cloud infrastructure enlightens most of the common security-related concerns, as private clouds are managed by the organization itself and the complexity of the private network security is determined by the organization [143]. Furthermore, Cloud computing and centralized server solutions are often prone to distributed denial-of-service (DDoS) attacks as well, which is a major concern in military tactical environments and battlefield situations [37]. Such an attack will immobilize the cloud computing access, which prevents all connected operators, devices and servers from using the cloud service. The availability is extremely important when it comes to cloud computing services, and this is often challenging to guarantee as well.

These drawbacks above do not necessarily eliminate cloud solutions from military needs. We have both positive and negative sides of adopting a cloud-based infrastructure in military context. However, long term experience in businesses and governance, which are now almost completely dependent on cloud computing, might highlight approaches or give advices on how to overcome critical drawbacks and make cloud computing a viable part of solution for military software development. There is a possibility that, by managing

strict guidelines for security, and ensuring data confidentiality and integrity, cloud-based computing model might be possible to implement as a part of the military platform.

#### 4.1.5.2 Implementing Cloud Computing in Military Operations

We have to conclude that the decision of moving military systems and platforms to the cloud will open a great number of integration possibilities, as the systems are more portable when it is communicating with a centralized data storage unit on the cloud. The military systems might be able to share information through web services, or perhaps use APIs to share data amongst the military networks.

Therefore we need to look at different cloud computing deployment models, such as public cloud, private cloud and hybrid cloud [143]. Public clouds are often sold to the public and is usually run by normal hosting companies. Private clouds are however enterprise owned and is often configured within the organization's private network. In military context it is often favorable to go for private cloud solutions, since the physical data control is restricted from the public and the entire platform is controlled by the military organization [143]. Organizations that value data confidentiality, security and interoperability often choose private cloud solutions.

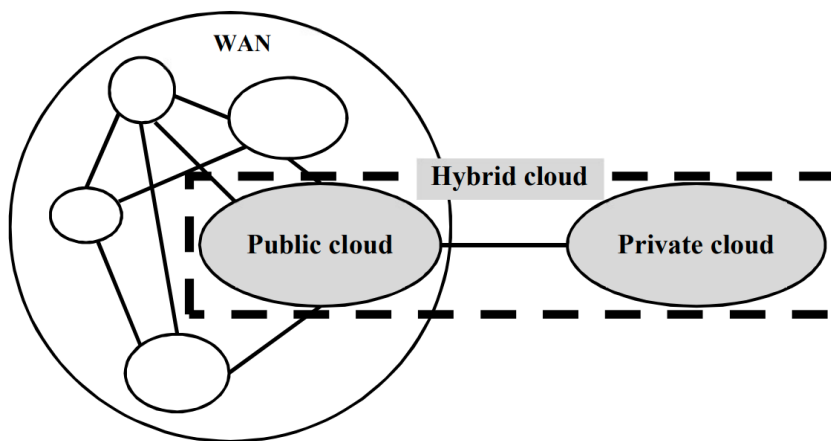


Figure 41: Cloud Computing deployment models [37]

There are hybrid clouds as well, which often is a combination of two or more clouds. An example from the military aspect would be a combination of public and private cloud solutions, where the physical data is stored within the private cloud and there is an additional public cloud within this hybrid model, which enables access to some public information within the organization, such as public data accessed through an API [143]. The public cloud within the hybrid model does not have access to any resources from the private cloud,

so this hybrid model is not less secure than the pure private cloud model.

There is also another option for military software development where cloud computing is balance against fog [138] and edge [65] computing. This means that certain parts of military operations can be supported by cloud computations, but fog and edge computing may take over in situations where instant decision making is needed and it is not advisable to use cloud computations. Edge computing brings computational solution very close to the source of data and military environment where computing can be safe and time efficient [65], and thus leave cloud computations for situations where its use is unavoidable.

#### **4.1.5.3 NATO Generic Vehicle Architecture (NGVA)**

NGVA is a NATO Standardisation Agreement (STANAG 4754), based on open standards to integrate military systems and to exchange data among vehicle subsystems [84]. The NGVA standard is built upon the DDS infrastructure (read more in section ) and it adopts the data distribution protocol to interoperably exchange information amongst the various subsystems.

The usage of NGVA makes military system integration less risky and more compatible, and it is cheaper as well to integrate subsystems on military platforms that are following the same standards. The NATO standard opens up an opportunity for new markets, such as smaller companies that are developing military systems and equipment. Since all subsystems are following the same open standard, there should be less risk when performing maintenance and updates in the future, as the subsystems are capable of cooperating more competently with less risk of breaking the system due to divergent infrastructures.

### Integration of NGVA

The process of integrating the remote firing system into a NGVA-based infrastructure consists of several steps of compatibility checks. These steps are necessary in order to verify that the system in development is compatible and operable with the military platform, so that these subsystems can interact and exchange data.



Figure 42: NGVA integration compatibility levels [43]

1. **Connectivity compatibility:** This is the top-level compatibility check, which ensures that the system in development can be physically integrated into the military platform without needing to modify the existing infrastructure. This means that both the hardware and software must be supported on the target platform in order to successfully perform an integration.
2. **Communication compatibility:** This next level ensures that the system data model is properly implemented based on the NGVA standard, such as various topic types, video streaming standards or messaging types among the subsystems of the military platform. This is necessary to ensure that all messages that are being sent or received by any subsystem can be fetched and understood by other subsystems.
3. **Functional compatibility:** The last compatibility level ensures that the subsystems are functioning properly in terms of functional and performance requirements. Even though it is possible to exchange data among the subsystems, each subsystem should be able to operate separately as intended, since the integration must not have a negative impact on the existing infrastructure in terms of functional capabilities.

If these three compatibility levels are met and there is no particular issue otherwise, then it is safe to consider that the remote firing system is capable of being integrated into a military platform following the NGVA standard.

There is a practical example shown in Figure 43 with the illustration of the remote firing system integrated into a military land vehicle. The subsystems of a NGVA-based platform are network nodes that are acting as peers to the NGVA infrastructure, allowing the nodes to establish a peer-to-peer relationship within the network.

4. THE PROPOSAL

There is, however, a gateway present in-between each node and the NGVA data network, which is required to map and route the data throughout the infrastructure. The RFS mobile application and receiver (marked in red) are exchanging messages through the data network, alongside other military subsystems deployed onto the military vehicle, as well as some off-vehicle grounded subsystems.

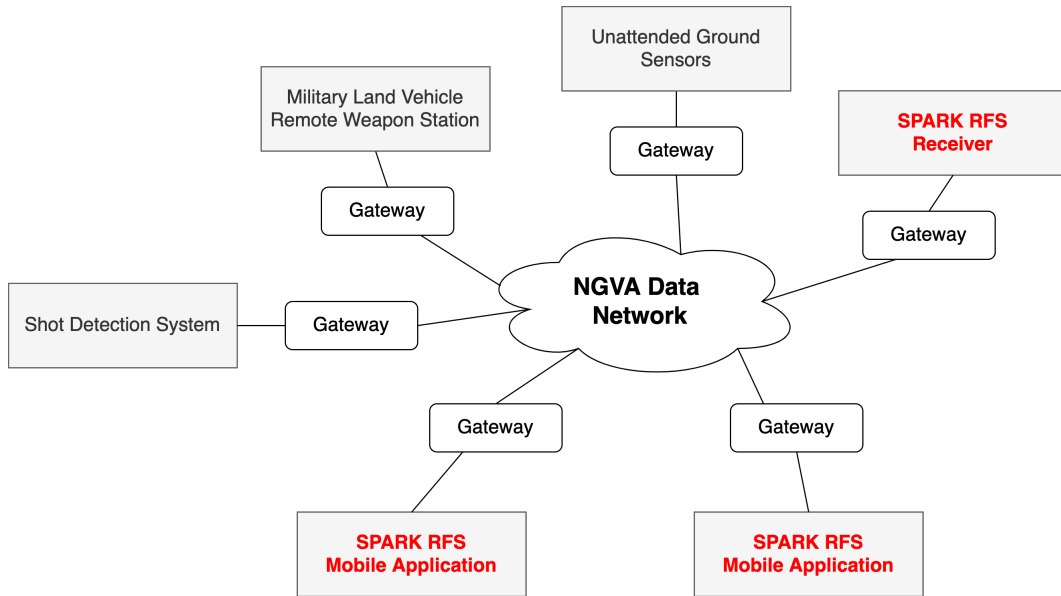


Figure 43: RFS integrated into a NGVA data network

The data that is exchanged between these nodes are packed as DDS messages, where some nodes are publishing information to a specific topic, and some nodes are subscribing to these given topics in order to fetch the data. This is the publish-subscribe pattern that the DDS infrastructure is built upon, which is the main approach for exchanging data amongst nodes of the NGVA data network.

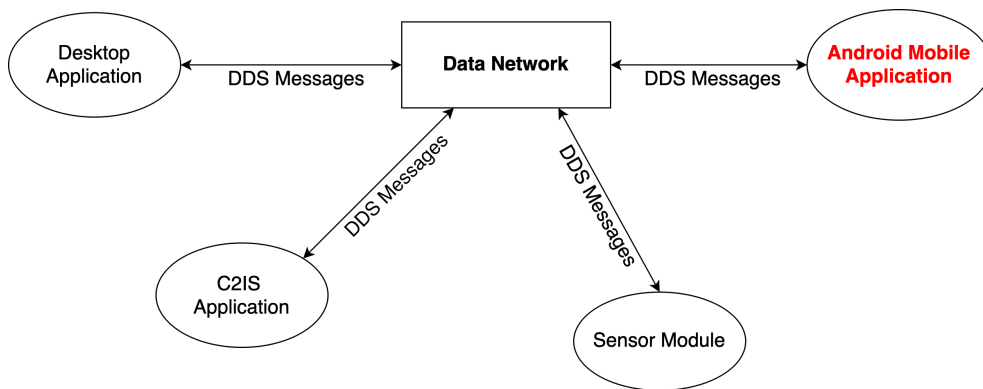


Figure 44: Abstract illustration of the NGVA data network

## 4. THE PROPOSAL

Digging deeper into the publish-subscribe pattern, each node within the network can both publish and subscribe to messaging topics, such as shown in Figure 45. In this example there are two nodes connected to the data network, "C2IS Application" and "Android Mobile Application". The latter would be the mobile application used to communicate with the receiver of the remote firing system.

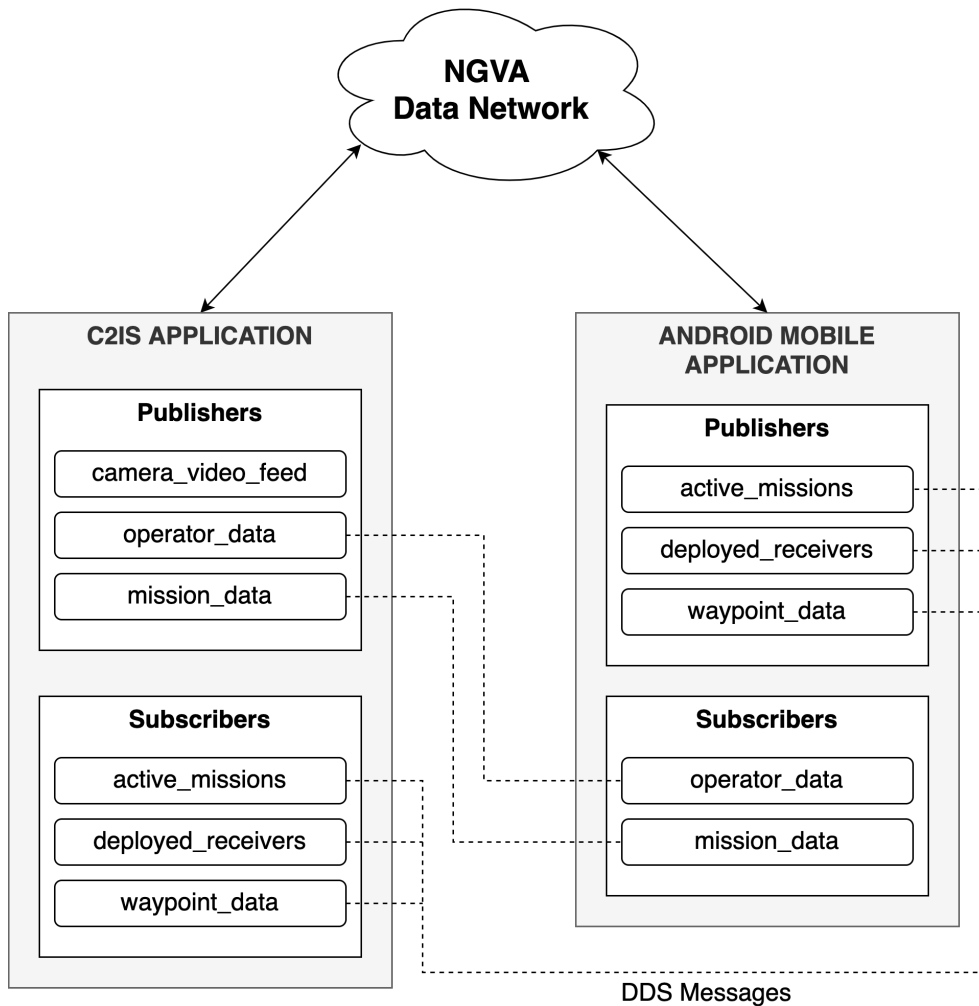


Figure 45: Publish-subscribe pattern amongst two nodes

There are numerous data topics in this example, such as *camera\_video\_feed*, *mission\_data*, *waypoint\_data*, and so on. These topics are used to exchange information amongst the two nodes above, such as for example special forces operator data that is being shared from the C2IS to the mobile application (C2IS is publishing data to the topic, and the mobile application is subscribing to this specific topic in order to fetch operator data).

Likewise, the Android mobile application is publishing waypoint data of deployed receivers to the C2IS application through a specific topic, in order to inform and update the other Special Forces operators about the new deployed receiver within the network.

#### 4. THE PROPOSAL

---

This is a pure peer-to-peer communication channel amongst the nodes within the network, and this infrastructure does not rely upon a centralized server or any cloud solution to transmit the data back and forth.

By following this data model and infrastructure, it is possible to integrate practically any subsystem into an existing military platform, which again opens up a large number of opportunities to improve the combat proficiency within military systems.



## 4.2 Physical Design

### 4.2.1 Design Challenges

When it comes to the physical design of the receiver there are a lot of concerns to take into consideration. To design a product for EOD personnel within agencies such as the military and law enforcement, some of the most important features to evaluate are weight, size and manageability. Operational personnel like this will always carry an extra weight in addition to their own; the packing can vary from 27-66 kg depending on the nature of the mission [101]. Consequently, the receiver needs to be as small and lightweight as possible, as dictated by requirement CR-005, 77.

During a mission, whether it is at the local train station or in Afghanistan, the operator will be under a lot of pressure and be subject to very stressful and demanding situations, and therefore it is of the utmost importance that the receiver is manageable and intuitive. For a development team this means a simple user interface that provides easy assembly, a crystal clear state of the receiver, solid grip and strong attachment to whatever it needs to be attached to.

In addition to the aforementioned hands-on conditions, another element to consider is the environment of operation. It is important that the receiver does not get adversely affected by any environment the operator is located in. To ensure the receiver box unit can withstand a number of environmental conditions, some standards needs to be addressed.

In this case the standards that need to be implemented are IP. In this context IP stands for "Ingress Protection" and is defined according to IEC 60529 [83] and CENELEC (European norm) EN 60 529 [134]. This standard describes a classification of how airtight and waterproof a container is. More about IP can be found here [95] and in the Standards paragraph 4.2.2.2.

When a casing has an IP classification another challenge arises: the components inside transfer heat, and the environment outside is changing so it will become moist inside the casing. This can, in time, corrode the electrical components and then the receiver may malfunction [55].

The product owner has provided a requirement that states that our product will need to adhere to at least IP65, and if possible IP67/68; this is requirement CR-004, 71. From the listing in figure 46, it can be read that it will have to be dust tight, have no entrance point for solid objects and have protection from a water jet with 6,3mm nozzle and up to

## 4. THE PROPOSAL

---

12, 5l/m( $\pm 5\%$ ) for at least 3 minutes.

Furthermore, it must be considered whether some ATEX standards needs to be considered, since the receiver is to be located in an explosive environment, this will be discussed in the ATEX/EX paragraph at the end of this section 4.2.2.2.

Finally, the material of the receiver unit must be considered. The casing needs to withstand sudden impact as an operator carrying the receiver could be hit or falls over It needs to withstand a heavy load, if, for instance, it is under several units of gear under a mission transport stage where there is no time to consider where equipment is packed. It is also crucial that the receiver is not adversely affected by the weather conditions in the operating environment. There will be a closer look and analysis of materials in the Material 4.2.2.1- and Material testing 4.2.2.3 section.

### 4.2.2 Casing Design

The product owner has stated the system is to be powered by a single CR123 battery. This is because CR123 is the most commonly used battery in military- and law enforcement contexts. More about this battery type can be read in paragraph 4.3.7. It is also a requirement (SR-023: 70), that the battery compartment must be designed in a manner that makes a battery change easy and doable without the use of any tools; this in addition to the fact that it needs to be as water resistant as the rest of the receiver, (SR-024: 72).

Some research was conducted to determine if there was a different solution available that does not require plasma igniters and nonel for ignition. This would be to make the receiver smaller and more manageable.

It was discovered that the next best thing after nonel, in this context, was a electric detonator. The drawback with this is that it is vulnerable to electromagnetic interference (EMI) or other radio frequencies (RF). EMI and RF can then cause the receiver to malfunction or in worst case, detonate when it not in operation [121] [23] [22].

To make this work the receiver would need to have a Faraday Cage or shield. One possibility to obtain this without making the receiver larger is to use nanotube induced resin to build the container unit, and in that way eliminate or to greatly reduce these risks. However, after a meeting with our customer where we proposed to use something else than nonel, we were served another reason why this will not be acceptable: the police force exclusively use nonel in EOD missions, and our customer will not use anything else at this point because nonel is the safest way to go.

#### 4.2.2.1 Material and Production Methods

For the time being the receiver will be 3D printed, this is because it is cost efficient in small quantities. As of today the product owner uses PA-12 filament when 3D printing other company products. PA-12 stands for Polyamide-12 and is made of nylon 12 resin. Although nylon is known to be a tough material in addition to being flexible, it presents a challenge when used in nordic countries and other countries with high humidity. According to the product owner, the challenge seems to be that it absorbs moisture from the air, so when the product is delivered in Norway some of the parts no longer fit like they are supposed to.

The technical specifications on the PA-12 data sheet [51], shows that the recommended environmental conditions for PA-12 are 50-70 % relative humidity (RH). Weather-and-climate.com shows that in 2019, in for instance Norway, there were an average annual percentage of humidity in Trondheim was 78.0 %, in Oslo 74 % and in Bergen 78.0 %, and beyond the Norwegian borders Copenhagen had in 2019 79 % RH, Stockholm had 75 % RH and Helsinki had 80 % relative humidity. This might explain the challenges the product owner has experienced with previous products.

With this in mind, one alternative is to use an advanced Polyethylene Terephthalate (PET) filament called Glycol-modified Polyethylene Terephthalate (PETG) or Amphora AM1800. This is a thermoplastic copolyester and the glycol-modification will keep the receiver from crystallizing when exposed to heat. The lack of crystallization will prevent the material from becoming easy breakable. Using PETG will also eliminate the relative humidity challenge. This is a FDA (Food and Drug Administration) approved filament, this aligns with the project goal to keep the engineering as green as possible; Life Cycle Engineering. Read more about LCE in paragraph 2.5.

From the information on the technical data-sheet for PETG [98], and 3D Insiders filament page [1], we believe that PETG will provide a casing that, when 3D printed, is strong, flexible, 100% recyclable, impact resistant in both high and low temperatures. Because of the glycol modification the 3D print will most likely not degrade in water or absorb moisture and it will have a low degree of warpage and shrinking under printing.

Another production alternative is to cast the casing. By casting the receiver one can control and design the desired properties in every batch and the design will be stronger. Unfortunately we are not able to cast at the time being, so the alternative is to build the receiver by using laminates made of fiberglass and resin.

Since the receiver will be employed in harsh environment it can be advantageous to enhance

## 4. THE PROPOSAL

---

the properties of both the fiberglass and resin as much as possible. One solution to this is to disperse single wall carbon nanotubes(SWCNT) in the resin.

Single walled carbon nanotubes has very high specific mechanical properties and a higher aspect ratio than other carbon nanotubes. These properties will improve the mechanical properties of both the resin and the fiberglass [48].

For this project quadraxial fiberglass fabric has been chosen, the fabric has non-woven layers in 0,45,-45,90 degree orientation. This alone gives an exceptional longitudinal, transverse, and sheer strength [76].

Furthermore we have chosen to use a clear epoxy, SvaPox110, for matrix material. The epoxy has a low emission of smoke/steam- and shrinking is at a minimum, when hardening. This provides low degree of residual stress. Epoxy is the matrix material which provides the highest strength to a composite [78].

Quadraxial fiberglass fabric and epoxy together with SWCNT will hopefully make a casing that is exceptionally strong and lightweight, extremely impact resistant at any temperature, will not absorb moist from the air or degrade in water, may function as a Faraday shield or cage, will be reusable and will not shrink during manufacturing.

The nanotube paste we have used is Tuball Matrix 301, produced by OCSiAl, a Russian nanotechnology company. OCSiAl operates worldwide, with offices in the US, China, Hong Kong, India, South Korea, Luxembourg and Russia [89]. The nanotube paste was pre-dispersed, according to OCSiAL guidelines, in SvaPox 110 (epoxy resin) with a 0.3% weight ratio and then combined with quadraxial glass fabric (850g/m<sup>2</sup>) in approximately 40/60% weight ratio.

We made 3 laminates so that we could produce 3 casing. Two casings were tested and one was for the final product.

### 4.2.2.2 IP, Atex and Test Standards

As mentioned in the subsection Design Challenges 4.2.1 the container unit, or the components inside the container unit, needs a certain amount of protection against foreign objects and liquids. This is called Ingress Protection, and describes how tight an enclosure is. The ICE-60259 standard describes IP the following way:

*This standard describes a system for classifying the degrees of protection provided by enclosures of electrical equipment for two conditions: 1) the protection of persons against access to hazardous parts and protection of equipment against*

*the ingress of solid foreign objects and 2) the ingress of water. The degree of protection against these two conditions is designated by an IP Code.[83]*

As can be observed in figure 46, the IP code consists of two digits. The first digit describes the degree of protection against foreign objects to enter the enclosure and the second digit is the degree of liquid protection. An important thing to remember is that there has to be separate tests for all the IP codes the system shall have. That means that even if the container unit manage IP68 classification, it will not automatically have IP65. This because IP65 is tested with a water-jet from all possible angles and IP68 is just tested 1m under the surface for 30 min or more.

To test the laminates: *ASTM D2344/D2344M-13 Standard Test Method for Short-Beam Strength of Polymer Matrix Composite Materials and Their Laminates (ILSS)* and *ASTM D3039/ D-3039M-00 Standard Test Method for Tensile Properties of Polymer Matrix Composite Materials*.

There were made test specimens from each of the tree laminates made.

To test the PETG: *ASTM D638-00a Standard Test Method for Tensile Properties of Plastic* and *ASTM D790-03 Standard Test Method for Flexural Properties of Unreinforced and Reinforced Plastics and Electrical Insulating Materials*<sup>1</sup>.

Some research on ATEX and EX classifications was conducted to figure out if these are standards that the development team had to consider in this project. The result of the research was that ATEX and EX is mainly for equipment permanently installed or functioning in explosive atmospheres [99], for instance an oil rig. This was then discussed with the product owner, who verified that there is no need for ATEX/EX standards on this project.

IP Ratings Guide

**IP (Ingress Protection) Ratings Guide**



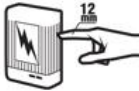



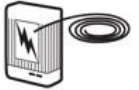







SOLIDS		WATER	
<b>1</b>	 <p>Protected against a solid object greater than 50 mm such as a hand.</p>	<b>1</b>	 <p>Protected against vertically falling drops of water. Limited ingress permitted.</p>
<b>2</b>	 <p>Protected against a solid object greater than 12.5 mm such as a finger.</p>	<b>2</b>	 <p>Protected against vertically falling drops of water with enclosure tilted up to 15 degrees from the vertical. Limited ingress permitted.</p>
<b>3</b>	 <p>Protected against a solid object greater than 2.5 mm such as a screwdriver.</p>	<b>3</b>	 <p>Protected against sprays of water up to 60 degrees from the vertical. Limited ingress permitted for three minutes.</p>
<b>4</b>	 <p>Protected against a solid object greater than 1 mm such as a wire.</p>	<b>4</b>	 <p>Protected against water splashed from all directions. Limited ingress permitted.</p>
<b>5</b>	 <p>Dust Protected. Limited ingress of dust permitted. Will not interfere with operation of the equipment. Two to eight hours.</p>	<b>5</b>	 <p>Protected against jets of water. Limited ingress permitted.</p>
<b>6</b>	 <p>Dust tight. No ingress of dust. Two to eight hours.</p>	<b>6</b>	 <p>Water from heavy seas or water projected in powerful jets shall not enter the enclosure in harmful quantities.</p>
<p>Rating Example:</p> <p><b>IP65</b></p> <p>INGRESS PROTECTION</p>		<b>7</b>	 <p>Protection against the effects of immersion in water between 15 cm and 1 m for 30 minutes.</p>
		<b>8</b>	 <p>Protection against the effects of immersion in water under pressure for long periods.</p>

Figure 46: IP standards. Picture from Nema Enclosures

#### 4.2.2.3 Testing of Material and Casing

The plan was to test the PETG and PA-12 ourselves. But, there has been some changes at campus due to the COVID-19 situation. For a while we had no access to the composite lab or the test lab. In light of this we had to put our faith in the technical data sheets we have found online. At least this will give us a rough estimate of the material properties of PETG [**PETG**] and PA-12 [51] so that we can compare them.

For IP testing the plan was to collaborate with a student group and Kåre Særen at Tinius Olsen Company at USN. Their assignment this year was to create a system and test environment for performing such tests. Due to the restrictions that occurred this fell through. Hopefully we can find another valid solution before the end of this project.

After the restricted reopening of USN we had enough time to do material testing on the laminates. Kåre Særen and Tinius Olsen Company lent us the ASTM test standards. There were made test specimens from each laminate. The results and execution of these tests can be read in full in the test report in appendix 8.12.

We also performed assembly tests; we exposed the three casings (without components inside and without battery lid) to a steady pressure in Tinius Olsen Company Super-L 300 test machine, (picture at 47). Since we are testing an assembly instead of pieces there will not be any official computer engineered report of the results, but the only result we are interested in are how many kgf (kilogram force) each casing can hold (Ultimate Force) before collapsing, and the Super-L 300 provides just that, the results from these tests are in appendix 8.13.

The PETG casing withheld 2810 kgf before it exploded. The first nano casing was assembled with super glue, cyanoacrylate. This casing withstood 2860 kgf before it collapsed in the interfaces of the assembly. All pieces but one was intact. The piece that had any damage was on the side with battery- and RJ45 entrance/hole, the piece is marked with a red circle in picture 48. The second nano casing was assembled with Araldite AW4858/HW4858 adhesive. This casing withstood an astonishing 7460 kgf before it collapsed, also in the interfaces of the assembly and the only piece with any damage was the same as the test before, so it is safe to say that this piece, without the battery lid in place is the weak link of the assembly.



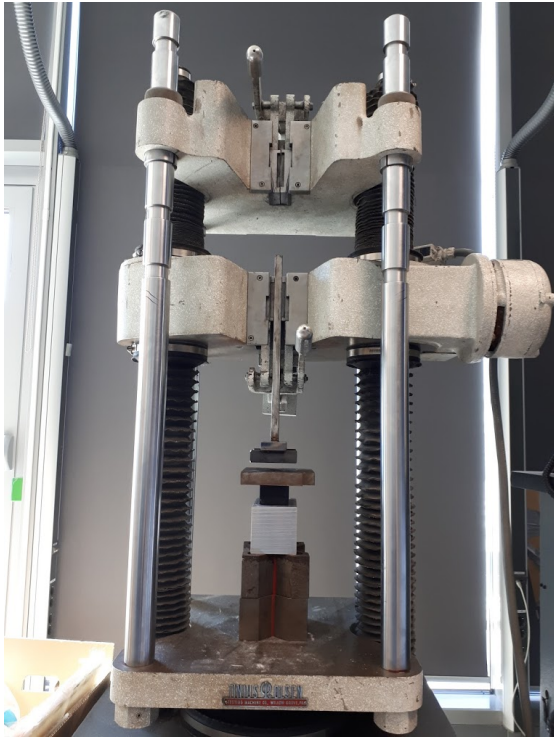


Figure 47: Tinius Olsen Company Super-L 300



Figure 48: PETG- and nano casing after test



## 4.3 Electrical Design

### 4.3.1 The Electrical System

This section is a proposal for how the electrical system in the receiver could be designed. In Figure 49 we can see a block diagram of the proposed system thus far. The block diagram is composed of subsystems and some key components; the subsystems will be derived further later in this section.

The primary objective of the electrical system is to ignite the plasma igniters to initiate the energy transference. To do this, the system must be able to accumulate the energy needed and deliver it to the plasma igniters. Consequently, the system needs two energy storage units: "Cap.1" and "Cap.2" in Figure 49 serves as the energy containers. To fill these containers with the necessary energy we have the "Step-Up 1" and "Step-Up 2" subsystems more about this in section 4.3.2. They drain energy from the battery and transform it to the potential we need. The battery specifications are derived from CR-003.

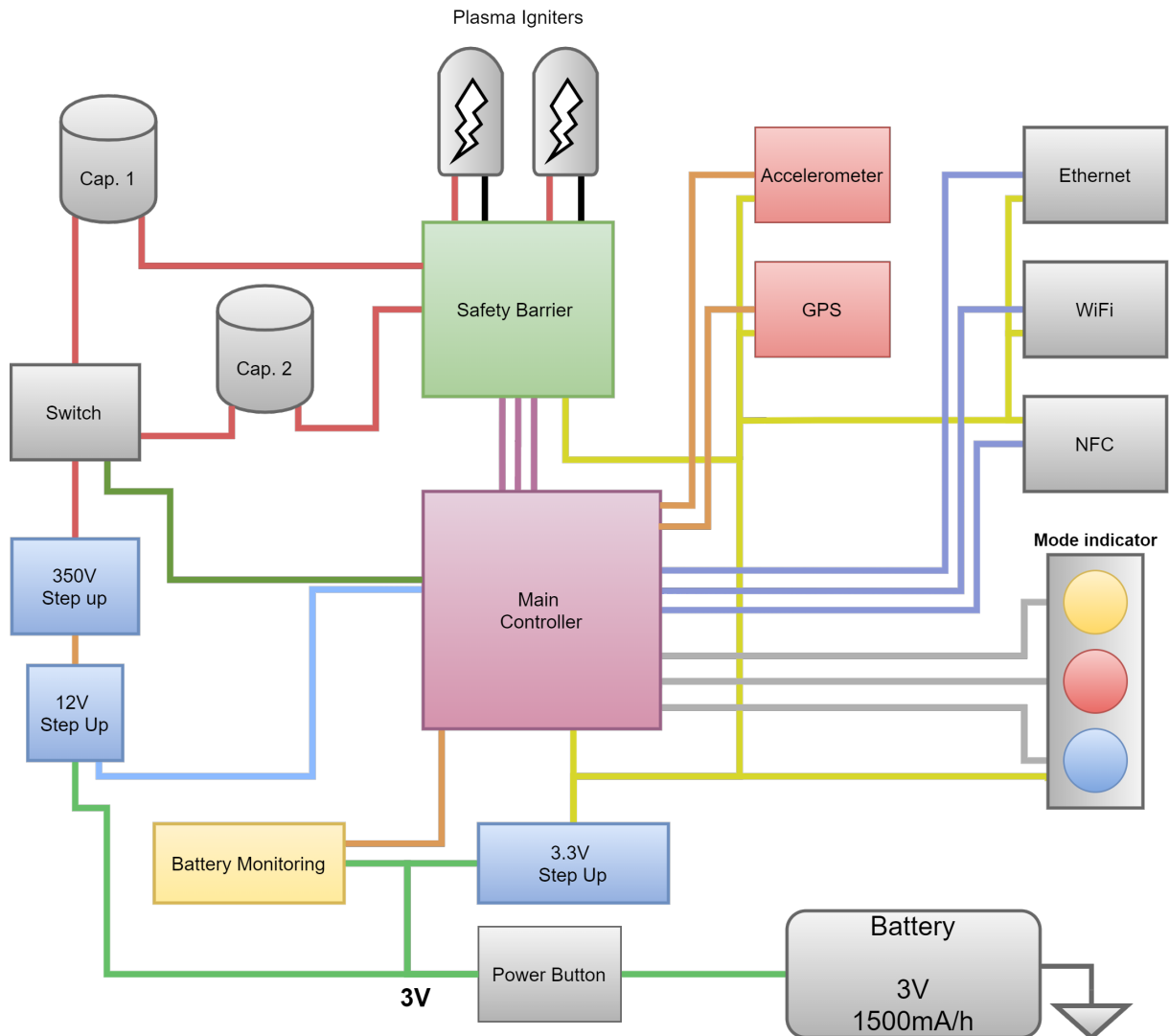


Figure 49: The Main Electrical System

To be able to do this in a safe manner and minimize the risk of a premature initiation we need some safety barriers to ensure that the energy doesn't reach the plasma igniters accidentally. Another thing that the safety barrier subsystem does is the discharging of the capacitors if an abort initiation instruction is received.

For the system to be able to communicate with the software application we need a communication channel and something to handle the data sent and received. Our customer desires two communication channels: one through RJ45 (IP) and the other through WiFi. Therefore, the electrical system has a RJ45 interface and a WiFi antenna connected to a micro-controller dedicated to communication and handling of data. The data received will be transmitted to the main controller which will reply if necessary.

#### 4. THE PROPOSAL

---

The data received through the NFC antenna will be the pairing sequence, and the communication through RJ45 and the WiFi will be the commands and configuration data. The main controller will act according to the instructions received and perform the actions needed to fulfill these instructions. It acts as the "brain" of the electrical system and sends out control signals to the subsystems to arm, disarm or initiate.

To show the mode of the receiver the Mode Indicator in Figure 49 will consist of 3 LEDs that will be colour coded to display the different modes: powered, armed, and fired. The Power Button will be a two-step switch that the operator can turn the receiver on and off with. This comes from CR-009 and CR-010.

The battery monitoring system enables the main controller to assess the battery status and relay it to the operator. It also alerts the operator if the battery no longer holds the necessary energy to arm the receiver, which corresponds to SR-022 and CR-018.

The 5V Step-Up subsystem will ensure that the rest of the electrical system gets a steady supply of 5V DC. It will also compensate for the voltage change that occurs in the battery when exposed to different temperatures or when the battery starts to run out. With this proposal we aim to fulfill the functionality needed to give the customer what they want and at the same time make a robust and safe product to use.

## 4. THE PROPOSAL

### 4.3.2 Plasma Ignition System

The requirement from our customer regarding the use of a plasma igniter as our interface is immutable. We are unable to obtain the datasheets of these igniters, as they state that "Further information on authorised user request only". Consequently, our customer has provided us with the requirements of 350V and 9 joules of energy, based on their knowledge of the field. This is the minimum amount of energy to ignite the plasma igniter they intend to use with our device.

Together with the requirement of using a CR123 battery as our only energy source, we need to find a way to accumulate the power needed for the plasma igniter. One of the challenges here is to step up the voltage of the CR123 battery from 3V to 350V, and the second challenge is to deliver 9 joules to the plasma igniter.

The third requirement that affects this subsystem is the overall size of the system. Our customer wants the receiver to be as small and lightweight as possible. So when we look for solutions on how to solve the two first challenges we need to evaluate function against size.

### 4.3.3 Step Up

The first challenge is to step up from 3V to 350V, this can be accomplished with boosters. After some research we found that most boosters can step up 2-20V more than their input voltage. If we connected them in series with a voltage divider between each of them as shown in figure 50 we can get up to 350V, but this is not sufficient when it comes to size or cost.

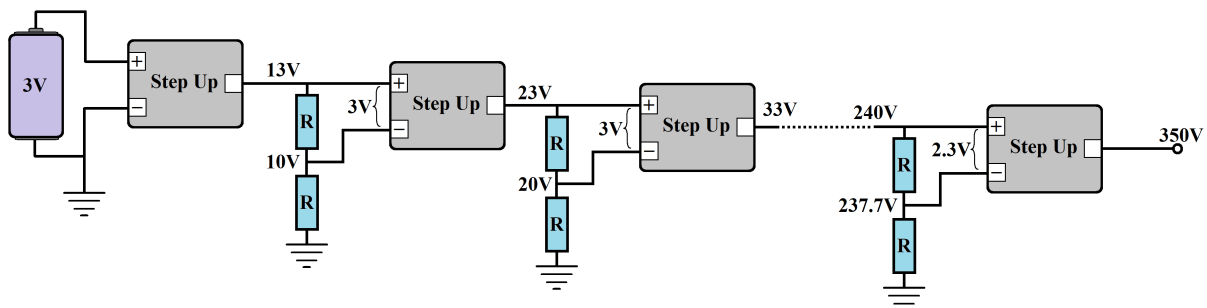


Figure 50: Booster in series

There are boosters that have a higher range; for example, we found some that could deliver 40V and more than the input voltage. But even if we reduced the number of booster we

## 4. THE PROPOSAL

will lose some efficiency to the voltage dividers. We managed to find a booster that would allow us to only use two stages to reach 350V, but they were large relative to the other options.

This will solve the first challenge of stepping up 3V to 350V, but it's costly in terms of size. Therefore, we started to look into the possibility of designing a simple step up circuit ourselves. After some research we understood the general idea and made the circuit shown in figure 51 for more details see page 873-875 in [44]. Subsequently, we calculated some rough estimations on component values and ran some simulations, the results of which can be seen in figure 52.

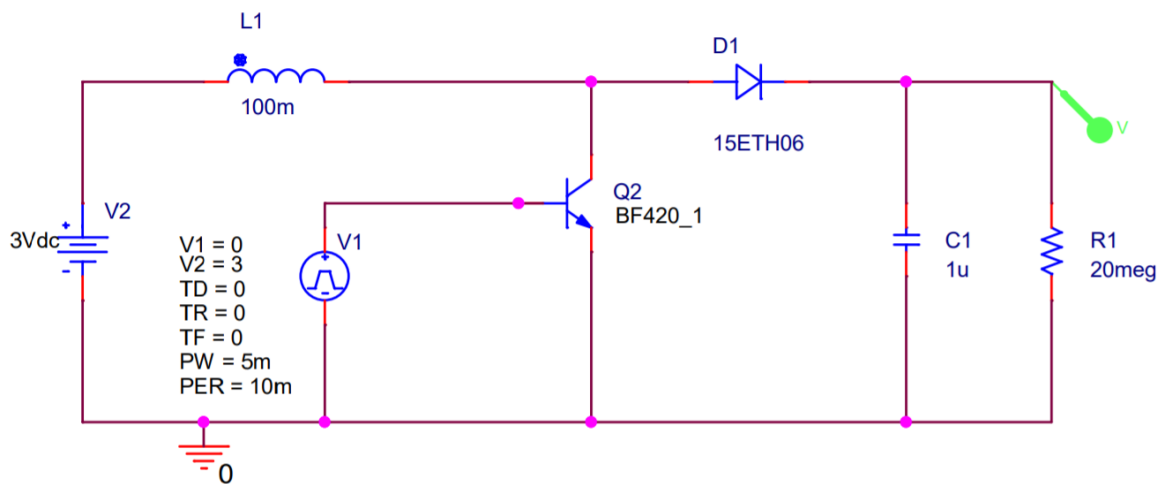


Figure 51: Step up circuit

This is the first design of our step up circuit, and it will require more refinement in the event that this is the solution we end up proceeding with. For now, we need to conduct more research and look at more specific solutions. The main goal was to determine if it was possible to step up 3V to 350V, given the small size of the receiver and it is.

The second challenge is to deliver 9 joules of electrical energy to the plasma igniter. If we had a energy source with 350v as output voltage we could design some circuitry to deliver the 9 joules. Since we need to step up the voltage from the CR123 battery we have a limited amount of current at our disposal. This necessitates the accumulation of the needed energy in a container with a fast discharge characteristic.

A properly sized capacitor will be able to hold this energy and discharge the energy to the plasma igniter. This is a very common component when it comes to energy storage.

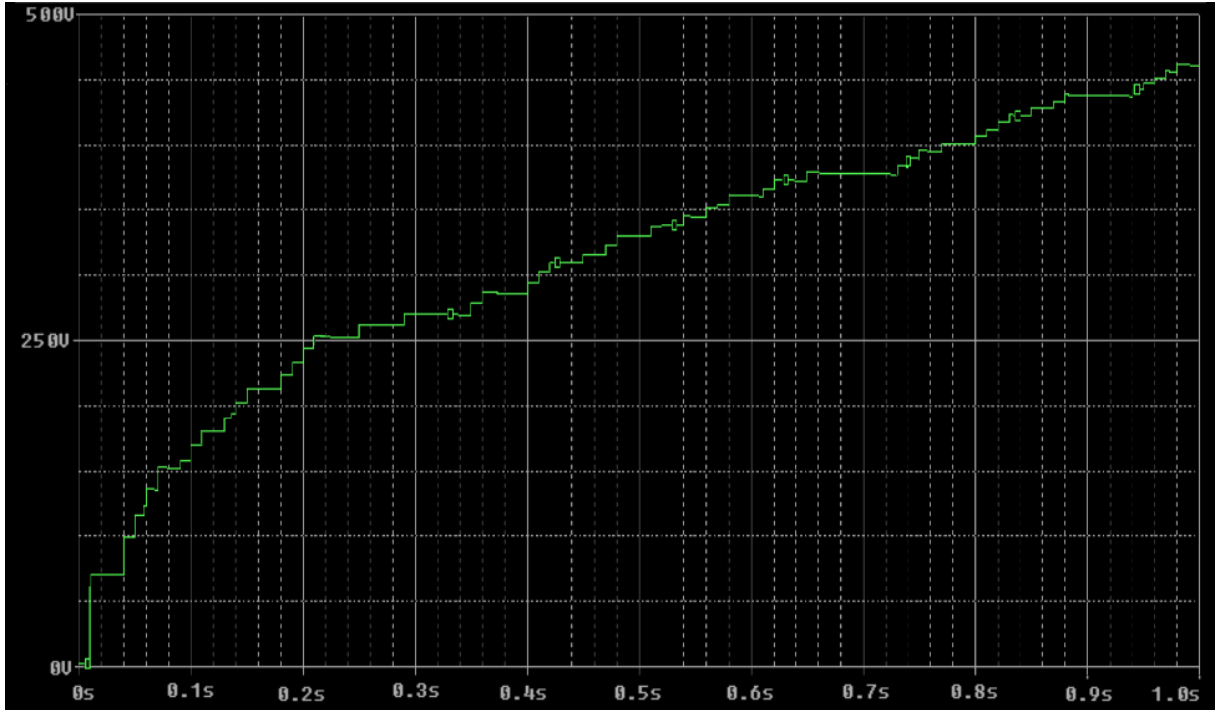


Figure 52: Step up circuit simulation

To calculate the proper size of the capacitor we can rewrite the standard capacitor energy storage equation nr. 1 to find the capacitance  $C$  as in equation 2.

$$J = \frac{1}{2}CV^2 \quad (1)$$

$$C = \frac{2J}{V^2} \quad (2)$$

We want the capacitor to store 9 joules of energy at a voltage of 350V, if we use equation 2 we get:

$$C = \frac{2 \cdot 9}{350V^2} = 146.938\mu F \quad (3)$$

This capacitance results in a relatively large capacitor, but it is not out of the ordinary. There is many electrolyte capacitors to choose from in 150  $\mu F$  range. Which is close enough  $150\mu F \approx 146.938 \mu F$  which will give us a little more than 9 joules. This is the minimum requirement for the plasma igniter to ignite, but there will be some lost energy that we need to consider too. All in all, we can use a capacitor to store the energy we need and still fulfil the other requirements.

In table 5 we compare some of the characteristics the component that fits our system best and the circuitry we plan to design if we go for that solution. The circuitry to be designed

4. THE PROPOSAL

	Design Step Up Component	Buy Step Up Circuit
Voltage	>350V	>350V
Ripple	Medium	< 0.05%
Ampere	150mA	50mA
Size	Unkown	1.4 x 1.1 x 0.5 Inch
Implementation time	>Short	Short

Table 5: Step up design vs step up component

is unknown in size, but the step up components deliver a voltage which is refined to a degree much higher then what we need. We can design a step up circuit with higher ripple voltage and get a way with less circuitry. Step Up Component in table 5 can be found at [63]. The parameters in 5 is derived from the data-sheet and the simulations.

After a lot of research the third step up option is to buy a integrated circuit and choose the external components values according to what we need. In Figure 53 we can see the integrated circuit LT3757 from [70]. Here most of the external component values are already chosen, so we can more or less just use the design shown in Figure 53 because its already configured to what we need. This is properly the solution we will go for.

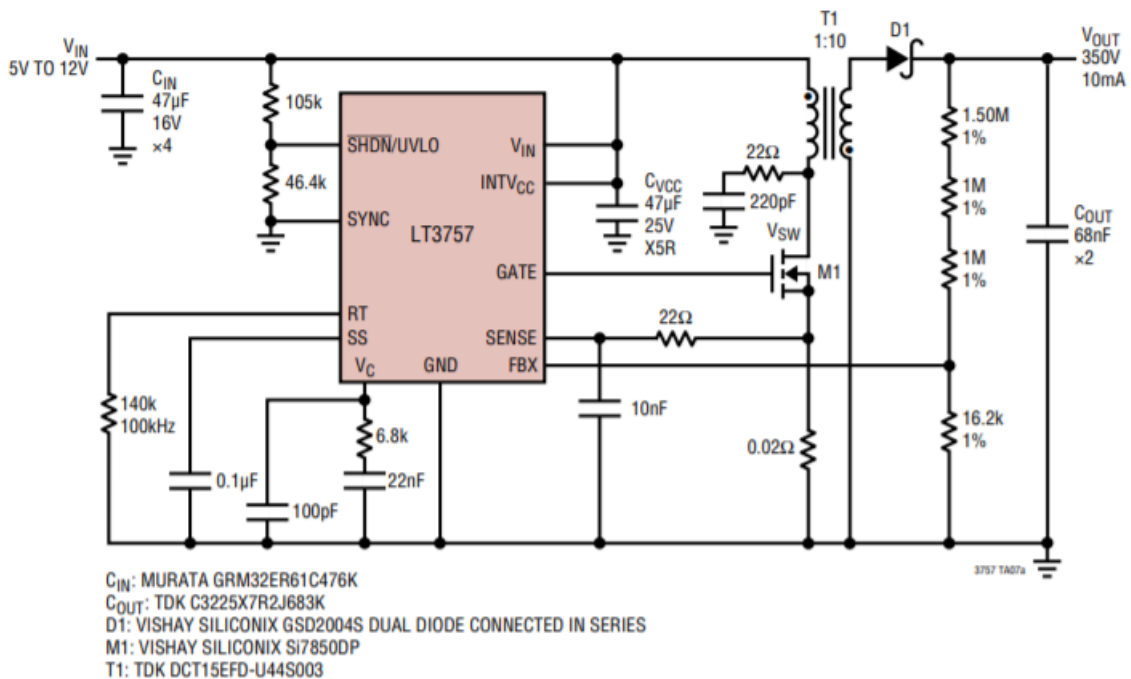


Figure 53: LT3757 from Analog Devices

#### 4.3.4 Electrical safety barriers

Since we are developing a system that deals with explosive ordnance, we need to minimize the risk of potential failures. Failures in a remote firing system can lead to tremendous consequences like the risk register shows. Based on requirements SR-014, SR-033 and SR-016 we need to develop an independent electrical system with safety barriers to make the system safe to use.

The main purpose of the electrical safety barriers is to prevent unintentional energy delivery from the capacitors to the plasma igniters (the load). We have investigated different possible solutions in different parts of the electrical system to avoid that from happening.

##### Switching component between plasma igniters and rest of the circuit

In this case we only illustrate with one plasma igniter even though there are two of them. A possible general design for this kind of circuit could be the one shown in Figure 54. The very basic discharge circuit requires two independent input signals to start delivering energy to load, if set up correctly. With this design we avoid charging the capacitor accidentally. Switch G and L is controlled by the same signal because either we want the energy to go through the discharge resistor or the load. Switch D is controlled by its own signal independently. The general discharge safety circuit can be seen in Figure 54 and the load is modelled as a variable resistor

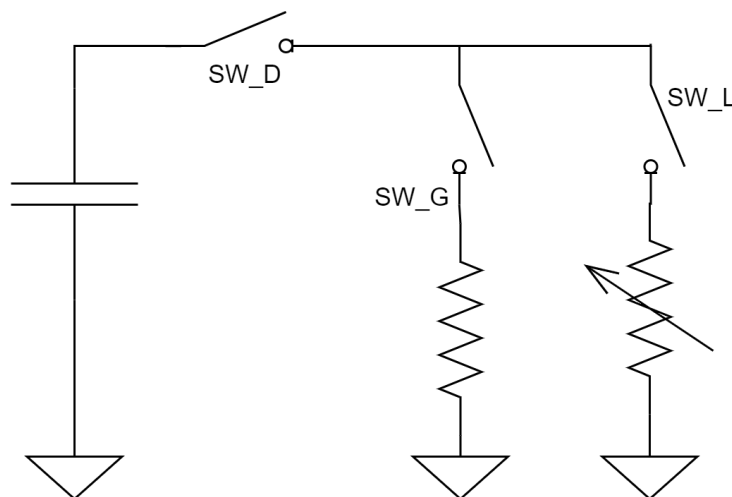


Figure 54: General discharge safety barriers



Switch	Operation
SW D	Normally closed
SW G	Normally closed
SW L	Normally open

Table 6: Switch normal operation

In Table 6 we can see the normal conditions for each switch. The normal switch configuration is chosen so that normally there will be no voltage buildup over the capacitor, and the plasma igniter is normally out of the circuit.

To control this switching mechanism we can use digital logic. See the table below for the combinations of the mechanism operation. We assume that a high signal toggles each switch and a low signal does nothing.

Signal to D	Signal to G & L	Result
0	0	Avoid voltage build up over capacitor and load separated from circuit
0	1	Deliver energy to load
1	0	Charge capacitor
1	1	Charge capacitor and load connected connected to ground

Table 7: Different states based on different inputs

By looking at Table 7 we see that we are interested in all combinations except 11, because it is unsafe and serves no purpose to us. In Figure 55 we can see a possible solution for the digital logic using one NAND gate and two AND gates. This will make the unwanted combination 11 be replaced by 00. More about the logic gates and their truth tables can be found in [61].

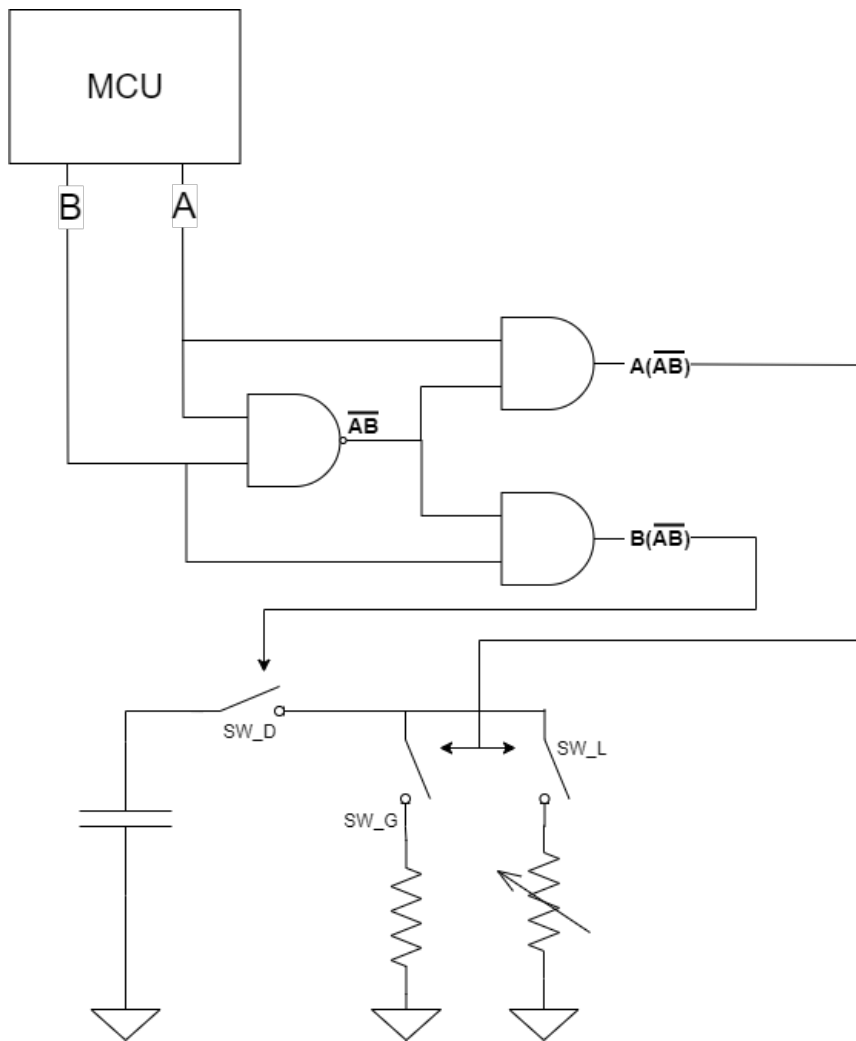


Figure 55: Digital safety logic from MCU to discharge circuit.

For the switches that is in the circuit in Figure 54, these must be able to withstand a high voltage and high current. A relay could be suitable for high currents, but not high voltages without being unnecessarily big for the receiver. A thyristor such as a silicon controlled rectifier (SCR) is a possible solution. The SCR characteristics can be seen in Figure 56

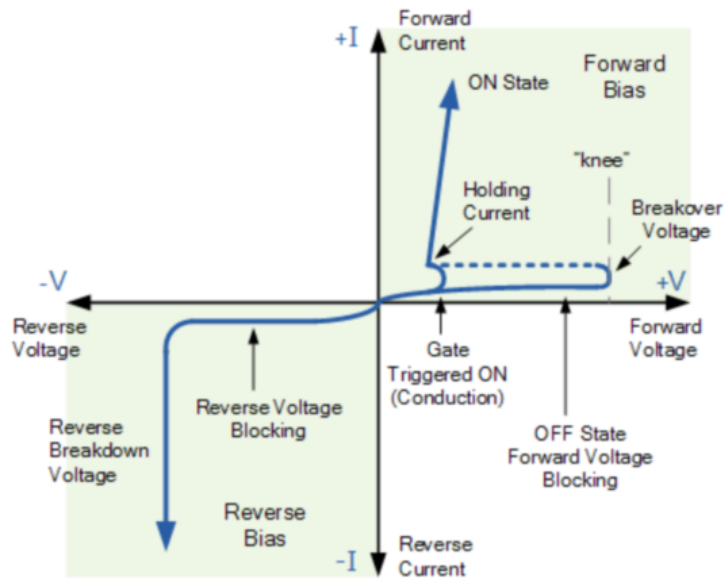


Figure 56: Silicon controlled rectifier (SCR) thyristor [62]

As you can see, it acts as an open switch until forward-breakdown voltage. We can control the gate trigger current to turn it on, and it will discharge the capacitor until the current drops below holding current [62]. Another possibility is to use a FET transistor. Nevertheless, once components are chosen, signal conditioning elements such as op-amps need to be used to make everything compatible.

### MCU Stability verification

To make the electrical system more robust against errors from the MCU we could employ circuitry that checks that the signals from the MCU are consistent. For instance if the voltage is fluctuating between low and high, i.e. is not consistent over time of let say half a second, the electrical subsystem would not be triggered. This way we disable the subsystem if the MCU is not exhibiting the expected behavior. A possible solution is to use the design illustrated in Figure 57.

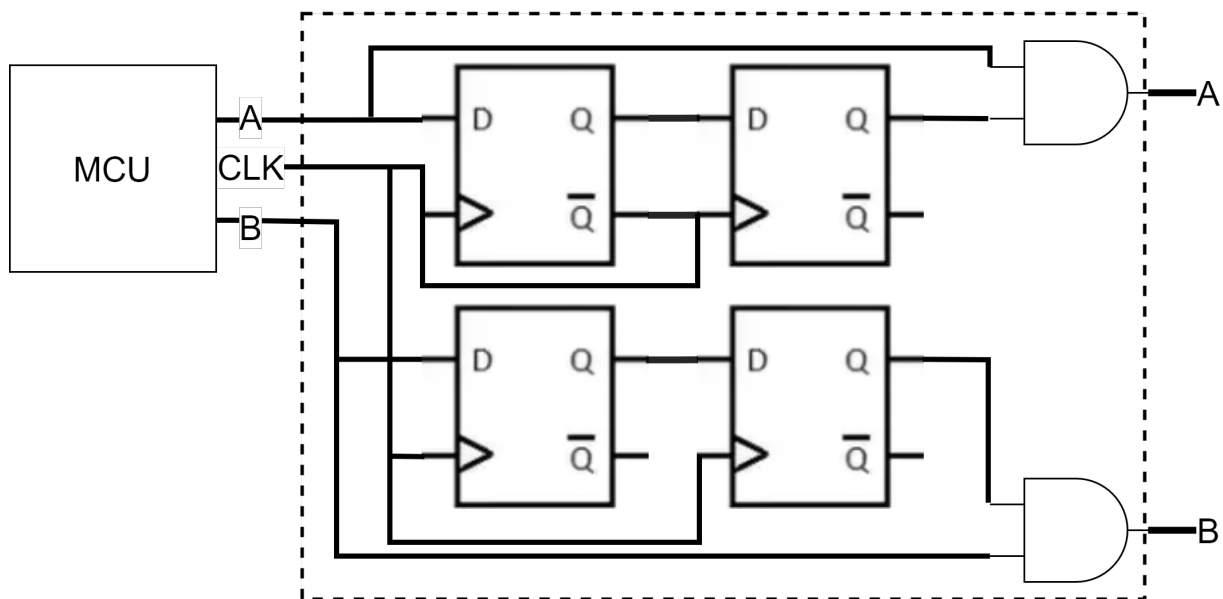


Figure 57: Stability verification using D-flip flops

The logic in Figure 57 will use two clock cycles to check if the signal is consistent. If the clock frequency is set to half a hertz we would use 1 second to perform this check. The problem that this solution solves is somewhat uncertain, and will have to be investigated more in depth and benchmarked against other alternatives for the final design.

### Other safety measures

We have also discussed filtering of noise and a self diagnostics check as potential electrical safety barriers. Filters placed in sensitive parts of the circuit, critical parts or where noise is generated could be a good idea. Another idea is a self diagnostics system where, for example, the MCU could check a GPIO port for the electrical system status. The signal on the GPIO port could be generated from a combinational logic setup that is connected to each electrical subsystem. More information about combinational logic can be found in [61]

### 4.3.5 Ethernet

#### **Ethernet controller**

The receiver needs a network interface that enables it to connect to a network (R-001). According to requirement EL-001 we need a RJ45 interface that supports Ethernet (IEEE 802.3) integrated into the receiver. Since we wish to create a PCB that integrates the different electrical subsystems we are looking for an integrated circuit (IC) that supports Ethernet.

The Arduino MKR Eth Shield is an open-source hardware released under the CC-SA-BY-NC licence [49] that we have researched. The Arduino even provides the Eagle CAD files together with a schematic for people to make their own circuit. After examining this licence agreement we saw that it is not possible to use this circuit directly in commercial use. It is possible to use it for commercial use if we integrate the shield as a whole into our product, but this seems rather impractical when we want to minimize the size of the receiver. Below we will take a look at two different alternatives for Ethernet controllers.

#### **WIZnet W5500**

The Arduino MKR Eth Shield uses the WizNET W5500 IC. Although we cannot use the exact schematic or Eagle CAD files from Arduino, we can still use the same IC that they used, W5500 for our product. The W5500 chip is a Hardwired TCP/IP embedded Ethernet controller that provides easier Internet connection to embedded systems [137, 142].

The IC has good documentation about different applications and configurations. For example, they have Internet Offload Library (ioLibrary) which includes drivers and protocol implementations. With good examples and documentation we would expect it to be easier and faster to integrate. Good documentation is very important, and should not be neglected because of the amount of hours we use to understand the IC will be associated with a cost (in this project that cost is time).

To integrate this IC we only need to set up a SPI with MCU, clock, RJ45 and transformer for the communication. The block diagram of the IC with peripherals can be found in [137].

## 4. THE PROPOSAL

**Microchip ENC28J60**

According to the datasheet [39] the ENC28J60 is a stand-alone Ethernet controller with an industry standard Serial Peripheral Interface (SPI) and it is designed to serve as an Ethernet network interface for any controller equipped with SPI. This means it can be integrated with a microcontroller that has SPI support. More about how to implement the IC in embedded systems can be found in [139].

In Figure 58 we can see how the IC could be integrated with MCU and the surrounding interfaces.

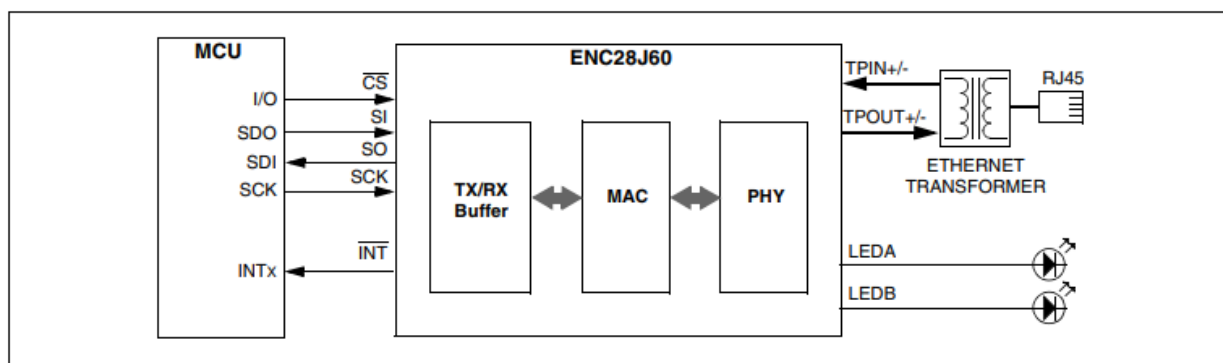


Figure 58: Typical ENC28J60-based interface [39]

**Considerations**

There are many factors to consider when choosing the correct IC for our application and needs. For example, this could be bit rate, security, implementation complexity, support, power consumption, robustness, durability and price. After considering the two Ethernet controllers above we see that they are two useful alternatives, but they also differ. The main difference between them is that the W5500 includes a hardwired TCP/IP stack on the IC. The IC can either be setup in client or server mode, depending on the need. The MCU can communicate with the IC by having an established TCP/IP socket connection[137]. If you use ENC28J60 you will need to implement the TCP/IP stack inside the MCU or an equivalent. This could mean more integration and possibly higher development complexity.

4.3.6 Hardware Paring and Authentication

When it comes to pairing the receiver with the mobile application the requirement SR-043 states that this must happen in an easy and fast manner. The operator must manage to do this under demanding circumstances even if they are stressed and short on time. This demands that the pairing procedure must be simple so that it can be performed under a lot of stress. Their hands might be shaking, their vision might be blurry and their way of thinking might be illogical. Therefore, the paring procedure cannot rely on steady hands, clear vision, many steps or complex routines.

Another problem with the paring procedure is security, since the receiver holds the potential it does, only authorized personnel should be able to pair with it. The authentication will then need some kind of key or password handling from the soldier. This is not ideal when the soldier is in a stressed situation.

One way to solve this problem is to use NFC to let the receiver and mobile application handle the paring and authentication. Then there is very little left for the soldier to do; they only need to bring the receiver up to the chest where they have the mobile device and let the pairing happen.

NFC is a wireless communication standard for exchanging data over a very short distance, typically 0-10 cm which is standardized in ISO/IEC 18092 [54]. We investigated the use of a NFC module to integrate with the processor of the main system.

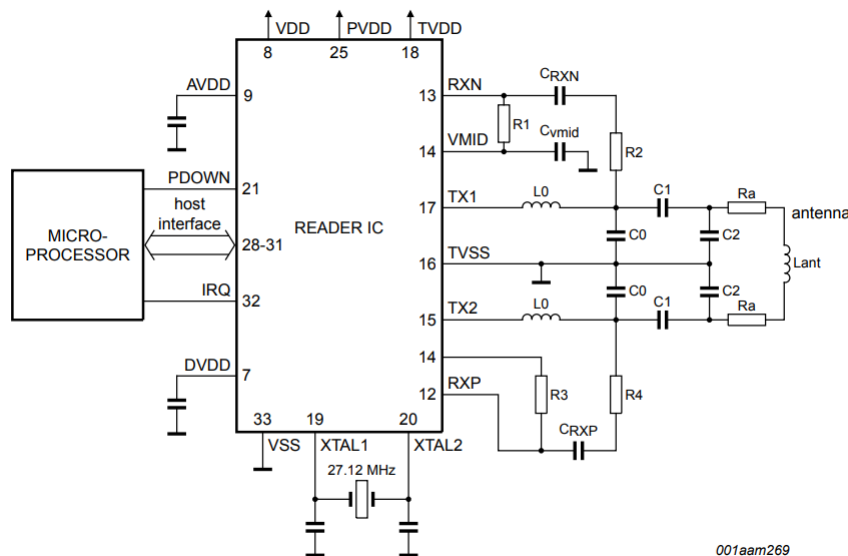


Figure 59: Wiring diagram for NFC system[80]

#### 4. THE PROPOSAL

---

In figure 59 we can see such a component, this is the MFRC63102HN from NXP [80]. This is a integrated circuit that handles the NFC connection and communicates the information it receives to the main micro controller in the system. We have to choose the external component values for the circuitry around this component. This gives us the opportunity to customize the NFC implementation to what we need.

Most of the external components build up filters and a way to match the impedance of the chosen antenna. This will allow us to filter out noise and to pick the antenna with the range we need. One thing that we can do different is the oscillator shown on input pin 19 and 20 in Figure 59, here we can either buy a oscillator with 27.12MHz or we can use the main microcontroller to generate a clock signal. We will look more into this when we get to the power consumption of the electrical system.



4. THE PROPOSAL

**4.3.7 Battery status system**

CR123 is the battery type that will be used for the system, according to requirement CR-003. In the analysis of the CR123 battery type we want to examine a few different alternatives to ensure our design is within compliance of normal commercial batteries. This is important so the end user is not dependent on the battery producer to use our system. The battery specification comparison can be found in the table below and is constructed from [38], [40] and [66]

<b>Specification</b>	<b>Energizer</b>	<b>GP Batteries</b>	<b>Duracell</b>
Classification	Lithium	Lithium	Lithium/Manganese Dioxide
Chemical system	Lithium/Manganese Dioxide	Lithium/Manganese Dioxide	Lithium/Manganese Dioxide
Designation	ANSI-5018LC,	3.0 Volts	3.0 Volts
Nominal voltage	3.0 Volts	3.0 Volts	3.0 Volts
Storage temp	40C to 60C	N/A	N/A
Operating temp	-40C to 60C	-40C to 60C	-40C to 60C
Typical capacity	1500 mAh (to 2.0 volts) (Rated at 100 ohms at 21C)	1500mAh (Discharge at 10mA to 2.0V at 23C)	N/A
Typical weigh	16.5 grams	App. 16g	App. 16g
Typical volume	7.0 cubic centimeters	N/A	7.0 cubic centimeters
Max discharge	1500 mA continuous (3500 mA pulse)	N/A	N/A
Max rev current	2 $\mu$ A	N/A	N/A
Typical Li Content	0.55grams	N/A	N/A

Table 8: CR123 battery specifications from different manufacturers

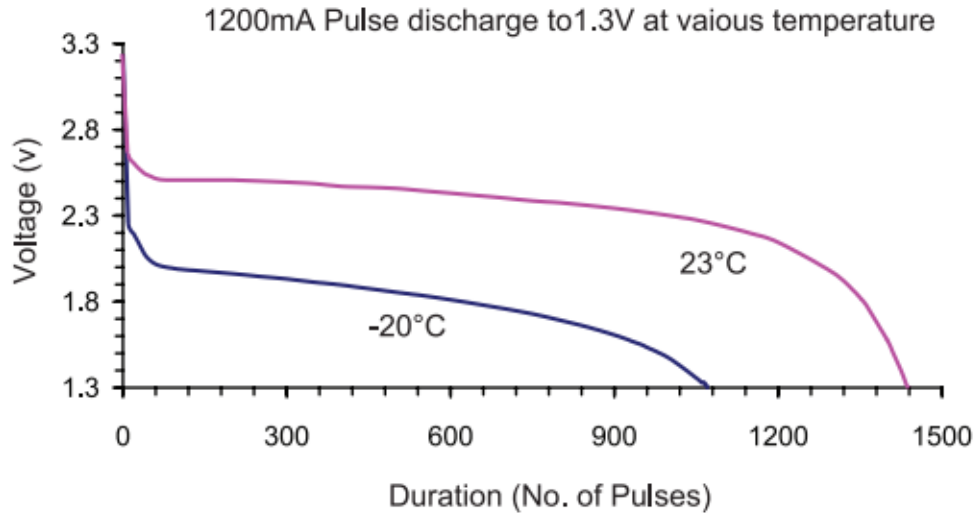


Figure 60: Discharge curve of the CR123 from GP Batteries [66]

$$Energy(J) = Amperhour \cdot voltage \cdot 3600 \quad (4)$$

The total amount of energy in a Energizer CR123 battery can be calculated using (4) and is roughly equal to 12960J. This is assuming the voltage is constant around 2.4V (widest part of the battery characteristic), which we know is not always the case. This means that since the battery needs to deliver 18J of energy every time the plasma igniter needs to activate we can have a total of  $\frac{12960J}{18J} = 720$  cycles roughly speaking. We also need to consider that we will lose some energy from the battery to the plasma igniter in form of heat to operate the different circuits. Another condition that will impact the amount of energy available for the plasma igniter is the temperature of the battery. In 60 we can observe that at  $-20^{\circ}\text{C}$  the capacity is substantially lower than at  $23^{\circ}\text{C}$ . This implies that we need to take into consideration the temperature of the battery when trying to estimate its status. Furthermore we need to choose a boost converter that accepts a voltage range that is suitable for the CR123 battery characteristics.

When we have figured out what kind of voltage level that is reasonable as input to the boost converter and the rest of the circuits we must monitor the battery voltage. From Requirement SR-018 we know that the electrical system needs to deliver 9J at 350V to the plasma igniter. To be able to fulfill this requirement we must be able to notify the operator if the battery is not capable of operating the circuit to meet requirements SR-021 and SR-022.

## 4. THE PROPOSAL

**LM3914**

After some initial research we found the LM3914, which is a monolithic integrated circuit that is capable of sensing analog voltage levels to drive up to 10 LEDs. The initial thought is that since the circuit can trigger up to 10 LEDs we could instead provide a microcontroller with a binary word. That way the embedded software could notify the user about the battery status. The temperature range of the IC is also within range of our compliance. The block diagram and specifications of the LM3914 can be found in [68]. After looking through the block diagram, we simulated parts of the circuit in LTSpice to verify its characteristics as you can see in Figure 61. We believe it is a possible solution, but we need to investigate the power consumption and integration with the rest of the system further before making a decision.

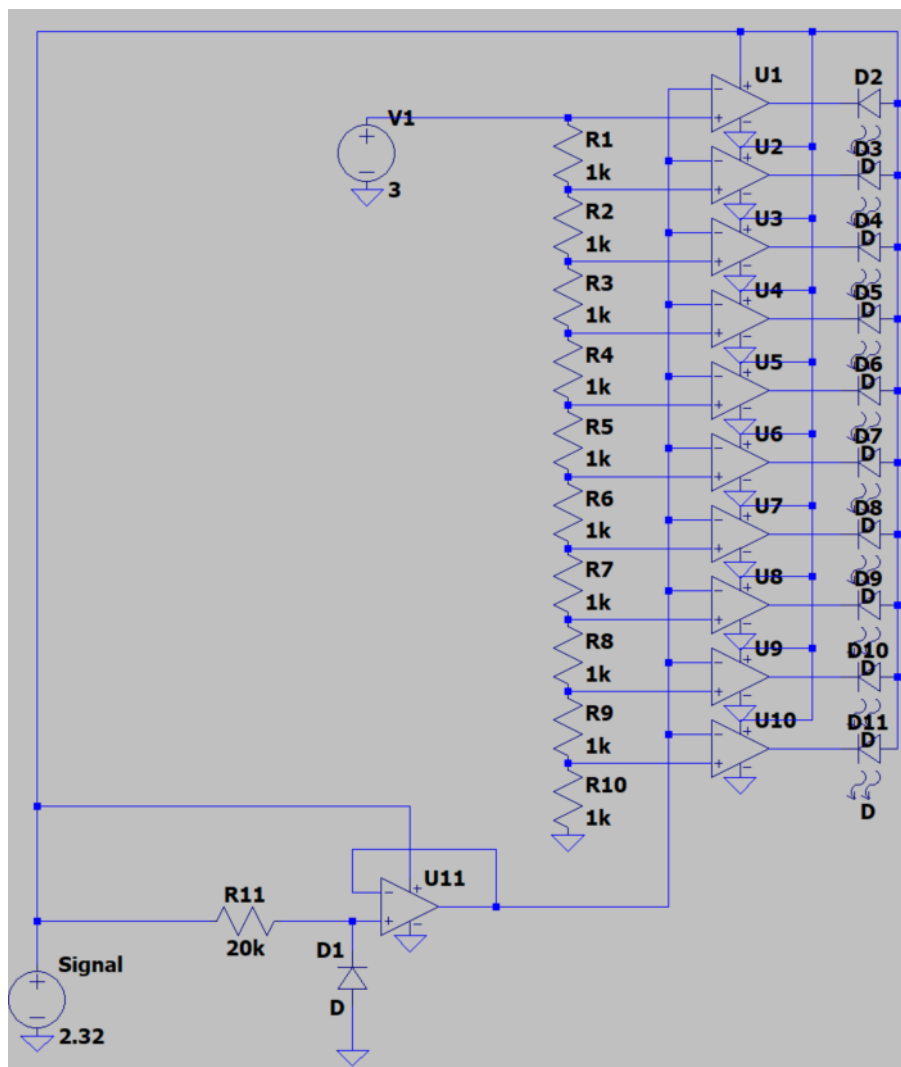


Figure 61: LTSpice simulation of LM3914 main circuitry

**Analog to digital converter (ADC)**

Another alternative is to use an ADC IC to read the battery voltage and use a microprocessor or another digital device to interpret and act upon the data.

The Least Significant Bit (LSB) voltage can be calculated using the equation below from [20]

$$LSB = \frac{V_{SPAN}}{2^N} \quad (5)$$

Now, let's consider the GPBatteries CR123 discharge curve (60). Our voltage span ( $V_{SPAN}$ ) can be selected to be 3.3V and by using Equation 5 with a eight bit resolution we achieve a LSB of 0.0129V. This is voltage difference between each discrete output level (number of counts). For an eight bit resolution we get  $2^8$  levels (counts).

The accuracy of the ADC is important because the CR123 discharge characteristics are quite sensitive. Quantization error is inevitable due to the output being discrete, but we wish to minimize the effect of it. Quantization error is the difference between the ideal/theoretical value and the actual value. The maximum error is usually specified as 1/2 LSB in datasheets, so we want to minimize LSB, and we see from Equation 5 that this can be done by increasing the resolution.

Looking at a 8 bit ADC, MAX153CPP+, it has a power rating of 2mW when operating, but also offers a powerdown mode with only  $5\mu W$  consumption. ADC is also a fairly common part integrated in development boards and microcontrollers. Currently, alternative 2 is most desired because it does not need modifications and it is a widely available component.

**Temperature sensor**

Since the system is to be used in all kinds of environments(TR-003), we know that this affects battery capabilities. At this time we are considering using a temperature sensor that can be used together with the voltage reading. This way we will be able to estimate the battery capacity better.

# SPARK

## 5 Implementation

### Contents

5.1	Software Components . . . . .	124
5.2	Software Application Development . . . . .	125
5.2.1	Qt Framework . . . . .	125
5.2.2	Initial Development Phase . . . . .	126
5.2.3	Networking Protocol . . . . .	129
5.2.4	Verification of Network Communication . . . . .	137
5.2.5	Near Field Communication (NFC) . . . . .	139
5.2.6	Software Safety Barriers . . . . .	144
5.2.7	Software Security Concerns . . . . .	154
5.2.8	IEC Functional Safety and IEC 61508 . . . . .	155
5.2.9	Android Application, Implementation Overview . . . . .	157
5.2.10	Communication Controls . . . . .	159
5.2.10.1	TacticalOperationsHandler . . . . .	159
5.2.10.2	ConnectionManager . . . . .	160
5.2.10.3	PacketManager . . . . .	160
5.2.11	Device Data . . . . .	161
5.2.11.1	ReceiverHandler & SenderHandler . . . . .	161
5.2.12	IP Interface . . . . .	162
5.2.13	Functional Behavior . . . . .	165
5.2.14	Map & Waypoint Functionality . . . . .	169
5.3	Electrical Design . . . . .	173
5.3.1	MCU . . . . .	173
5.3.2	Communication . . . . .	175

---

5.3.2.1	NFC Module . . . . .	175
5.3.2.2	Ethernet . . . . .	177
5.3.2.3	Wi-Fi . . . . .	179
5.3.3	Plasma Igniter System . . . . .	182
5.3.3.1	350V Step Up . . . . .	182
5.3.3.2	355V Capacitive Discharge Circuit . . . . .	185
5.3.4	Sensors . . . . .	193
5.3.4.1	Accelerometer . . . . .	193
5.3.4.2	GPS . . . . .	194
5.3.4.3	Battery capacity estimation . . . . .	196
5.3.4.4	Watchdog . . . . .	198
5.3.5	Power Supply . . . . .	199
5.3.6	PCB Design . . . . .	202
5.3.7	System integration . . . . .	206
5.4	Final Mechanical Implementation . . . . .	207
5.4.0.1	Nanotube enhanced laminates . . . . .	207
5.4.0.2	Casing V3.1, 3D Printed . . . . .	208
5.4.0.3	Casing V3.2, Nano Laminate Assembly . . . . .	208
5.4.0.4	Moisture Inside the Casing . . . . .	211
5.5	Early Stages of Prototyping . . . . .	212
5.5.1	Prototype V1 . . . . .	212
5.5.2	Prototype V2 . . . . .	214
5.5.2.1	Casing V2 . . . . .	220

## Chapter Introduction

This chapter of the bachelor thesis for the Spark remote firing system concerns the implementation of the proposed solution in the proposal section. In it, our prototyping and each discipline's part of the implementation is addressed. For software, this means a user-friendly application running on an Android operating system programmed in C++ (back-end) and QML (front-end). This application has NFC and IP capabilities, and facilitates the means by which an operator can pair receivers and control them over a network. As described in section 4, the Make Decision functionality has been the primary focus of implementation. For electrical, the circuitry of the receiver unit is considered; due to time constraints, the custom PCB has not been used for prototyping. For mechanical, the casing of the receiver is addressed. At the end, we detail two prototypes that were constructed during the project; a third is not detailed in the same manner, as the rest of the implementation chapter constitutes this prototype as the final implementation of our proof of concept.

### 5.1 Software Components

The software implementation consists of the network and communication elements of the remote firing system, along with an Android mobile application (sender) to communicate with the receiver. The remote firing system contains numerous software components, which are positioned on both the receiver and sender. The receiver would be the remote firing device itself, and the sender would be an Android device running our mobile application to communicate with the receiver.

There are some similar software components present in both sender and receiver, such as the network interface, connection handler and the packet handler. These components provide a communication channel between both sender and receiver, along with the communication between multiple receivers within a local network.

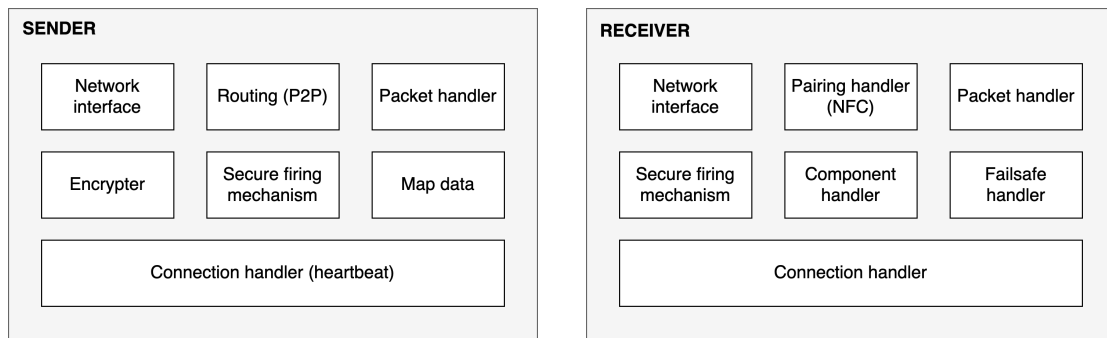


Figure 62: Software components (sender) Figure 63: Software components (receiver)

The network interface establishes the connection between sender and receiver, and the connection handler (also called heartbeat) is a pinging mechanism that checks if the connection is alive or not. The packet handler is the process of packing, unpacking and transferring the data between senders through our peer-to-peer network. This process is explained in more detail in section 4.1.4.



## 5.2 Software Application Development

One of the main requirements from the customer is that an Android mobile application must be developed in order to communicate with the receiver of the remote firing system. The mobile application consists of multiple software components, which are illustrated in Figure 62. Some of these components are programmed as separate blocks within the application, where there is an user interface at the top-level that combines these blocks together. The mobile application is developed in Qt Framework with C++ and QML.

### 5.2.1 Qt Framework

Qt is a framework that is extensively used for developing cross-platform applications in C++ and Python, often with a graphical user interface (GUI). The framework is especially popular among desktop and mobile application developers, since Qt applications are supported on multiple platforms, such as Linux, macOS, Windows, Android, iOS and a few others [113].

It is essential to note that Qt is not a programming language itself, but rather it is a framework provided with extended functionality to create better and more feature-rich applications in C++. Even though Qt is a framework, there are some addon frameworks that can be used alongside Qt to extend its functionality even further, such as user interface kits and APIs provided with tons of extra features.

One popular addon framework for Qt is Felgo, which is a framework based on Qt, provided with many features to develop native mobile apps and games [42]. The Qt framework does not provide any native support for Android and iOS development, and therefore it must be combined with Felgo in order to get an application with native look and feel.

### Choice of programming language

There are numerous programming languages and frameworks available to develop Android mobile applications. It is quite common to develop Android applications with Java or Kotlin using for example Android Studio, but there are however other options, such as Qt framework, which can be used to develop applications in C++ with the same native design as any ordinary Android application developed in Java.

The reason why we decided to develop our application in Qt framework with C++ is because the framework integrates well into hardware, as C++ is often used in embedded development. Qt framework is very well-documented and provides great support as well.

### 5.2.2 Initial Development Phase

There was a initial application design phase during the early stage of the mobile application development. This phase was based on brainstorming sessions and mockup designs to plan and illustrate the user interface (UI) of the application, without actually developing anything yet. This is a crucial step to avoid unnecessary development and massive adjustments in the future, as it scopes the backbone and structure of the application based on the requirements from our customer.

The UI mockup design sketches are constructed in Adobe Photoshop, and they provide an abstract perspective of how the mobile application would look like in a real-life scenario. One essential requirement from the customer is that the application must be easy to use and handle, especially during stressful situations. This mobile application will be used by special forces, and the user interface must be easy to understand and not too complex in terms of the amount of available information and options on the current screen.



Figure 64: Main view (sketch)

The main view is the first page that the user observes, and this page will be as simple as possible. It consists of the product logo and perhaps some general information about the application connectivity, such as "connected to the network", "missing network connection", or anything in that direction.

The overview page shows a list of all deployed receivers, along with the ability to pair a new receiver. This pairing process is performed by using NFC.

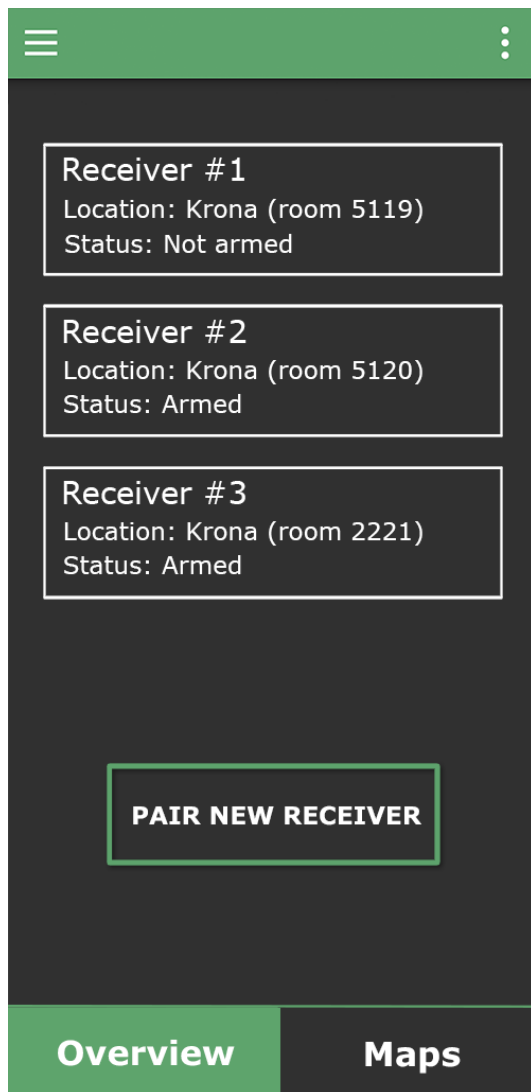


Figure 65: Receiver overview (sketch)

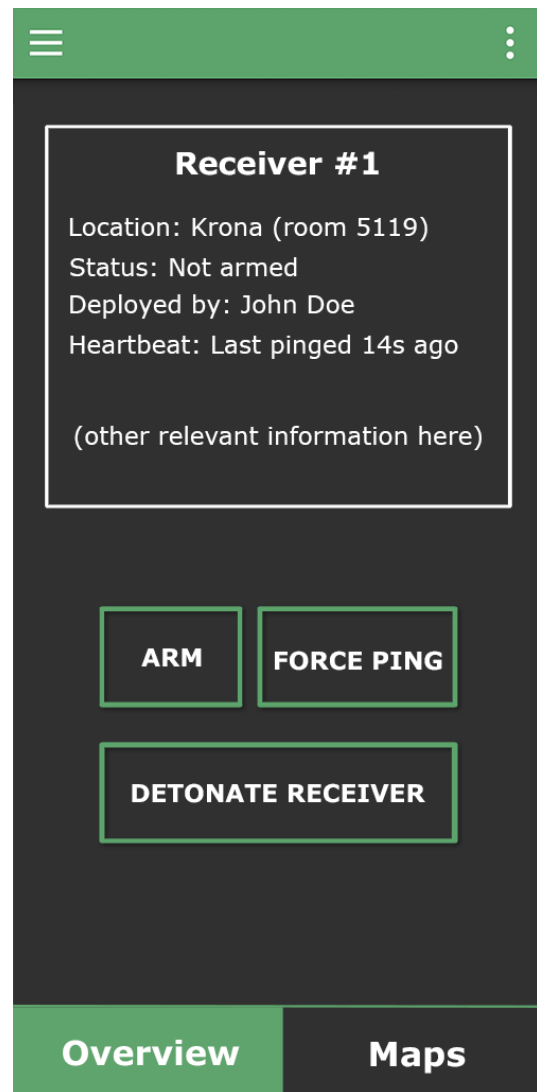


Figure 66: Detailed overview (sketch)

The ability to observe deployed receivers on an actual map is important in order to acknowledge where the receivers are located. This is a customer requirement as well, as the operator must be able to effortlessly track down and deploy receivers based on a real-time map. This map is accessible by all special forces operators, so that any operator from the same team can locate receivers on the map, which includes receivers deployed by themselves and other operators as well.



Figure 67: Map view (sketch)

The UI mockup design sketches will most likely change as time goes, and the actual implementation of the mobile application will probably not be identical.

### 5.2.3 Networking Protocol

The protocol sets out to facilitate functionality for the use cases in Figure 12, in addition to satisfy compliance for SR-047 and parts of SR-001. As well as to provide a method of verification against technical software requirements;

- SW-022 - SW-018 - SW-017
- SW-016 - SW-015 - SW-014
- SW-013 - SW-012 - SW-010

The SPARK system comprises of  $N$  transmitters where a dedicated transmitter controls any given  $M$  receivers. The communication between these devices will consist of a stateful binary based protocol with an  $X$  byte frame, layered on top of the TCP protocol.

#### Protocol Header

HEADER						PAYLOAD	SIGNATURE
0	1	2	3	4	5	6	7
0	Start of Frame					2 Bytes	
1	Length of Payload					2 Bytes	
2	Sender ID					1 Byte	
3	Recipient ID					1 Byte	
4	Data Category					1 Byte	
5	Data ID					1 Byte	
6	Payload					$N$ Bytes	
7	Authentication Signature					$M$ Bytes	
						8 + $N$ + $M$ Bytes	

Figure 68: Protocol Frame

The protocol frame has 6 fields with a 7th optional signature field. There is no form for error handling or control of packet loss, as TCP provides both in regards to correct sequence ordering of packets, as well as a modest cyclic redundancy check [136]. The RFS networking protocol should be used with an encryption scheme for data confidentiality, as well as the optional signature field for authenticity of packets, providing non-repudiation of operational decisions.

From Figure 68 field 0 indicates the start of a new frame initiated by value  $0xFF$ , followed by the length of the packet payload. Field 2 through 3 consists of transmitter and receiver

5. IMPLEMENTATION

unique ID to indicate the initial transmitter of the packet as well as the intended recipient, ensuring scalability for a peer-to-peer communication setup. Field 4 through 5 consists of data category of the given payload, as well as a subset data ID providing context for interpretation of the payload. Field 6 contains the payload itself.

Being a stateful protocol, it is critical for all packets to be received in the sequence they arrive while also avoiding all packet loss. Recipients will change local context dependant on received messages and will be interpreted differently depending on order of processing.

**Protocol Commands**

From Figure 69, each field will in overview contain data related to one of four categories;

FRAME FIELD	FRAME COMMAND	
4	<b>0x01</b>	<b>Pairing</b>
5	0x01	Acknowledge
	0x02	Pairing Request
	0x03	Pairing Response

FRAME FIELD	FRAME COMMAND	
4	<b>0x02</b>	<b>Tactical Operations</b>
5	0x01	Acknowledge
	0x02	Standby Receiver
	0x03	Arm Receiver
	0x04	Detonate Receiver
	0x05	Request Control
	0x06	Forward Control
	0x07	Situational Context

FRAME FIELD	FRAME COMMAND	
4	<b>0x01</b>	<b>Pairing</b>
	<b>0x02</b>	<b>Tactical Operations</b>
	<b>0x03</b>	<b>Collection</b>
	<b>0x04</b>	<b>Archive</b>

FRAME FIELD	FRAME COMMAND	
4	<b>0x01</b>	<b>Collection</b>
5	0x01	Acknowledge
	0x02	Heartbeat
	0x03	Forward Heartbeat
	0x04	Component Status
	0x05	Forward Waypoint

FRAME FIELD	FRAME COMMAND	
4	<b>0x01</b>	<b>Archive</b>
5	0x01	Acknowledge
	0x02	Upload Data
	0x03	Download Data

Figure 69: Header Fields

The four data categories comprises of commands used to control and monitor the transmitters and receivers, **Pairing** handles the process of accessing sensitive data and monitors the initial pairing sequence. **Tactical Operations** (TacOps) contains all operative execution of the Receiver-Transmitter activity to facilitate EOD controls. **Collection** contains all data gathering both for intermittent data use in the field, as well as persistent data meant for storage and further analytics of mission. **Archive** will handle the process and uploading of these persistent data for a long term storage entity. Each category will have a **Acknowledge** command to provide additional context and control when ensuring that the right data has been received.

In Figure 69, we can observe that the leftmost table is a close representation of the original use case model described in paragraph 4.1.2.1. Furthermore, it is noted that **Tac-**

**tactical Operations** represents the **Make Decision** functionality detailed in section paragraph 4.1.3.2. Here, all the functionality laid out by the aforementioned model have been translated into frame commands in the top right table. Additional functionality described by the original use case model (Figure 12) have also been implemented to varying degrees, though the Make Decision use case has been the primary focus, as established in section 4.

### Pairing

- **Pairing Request:** Initial request to establish a connection. Sent to a Transmitter device. Should contain necessary encryption key data. System ID and optional User ID of the requesting device.
- **Pairing Response:** Response to a pairing request. Sent from a Transmitter Device. Should respond with necessary encryption data, System ID and User ID.

### Tactical Operations

- **Standby Receiver:** Signal a receiver to go in standby mode. Sent from a transmitter device.
- **Arm Receiver:** Signal a receiver to go in armed mode. Sent from a transmitter device.
- **Detonate Receiver:** Signal a receiver to initiate a detonation process. Sent from a transmitter device.
- **Request Control:** Signal a transmitter to handover controls for a given set of receivers. Sent from a transmitter device.
- **Forward Control:** Forwards control of a given set of receivers. Sent from a transmitter device to a receiver device.
- **Situational Context:** Emits situational context from a receiver device based on sensory data. Sent to a transmitter device.

### Collection

- **Heartbeat:** Periodic connection pulse containing relevant status data from a deployed Receiver device. Sent to a transmitter device. Should include the receiver state and battery status.
- **Forward Heartbeat:** Forwards status data from a deployed receiver device. Sent from a transmitter device to other transmitter device(s).

- **Component Status:** Component status data sent from a receiver to a transmitter device.
- **Forward Waypoint:** Deployed receiver waypoint data, emitted when a waypoint representing a deployed receiver has been created on a transmitter device. Sent to other transmitter device(s).

### Archive

- **Upload Data:** Upload necessary data from a transmitter device to a persistent storage entity.
- **Download Data:** Download necessary data from persistent storage entity to a transmitter device.



## Protocol Behavior

### Pairing Process

The initial pairing handshake between a transmitter and a receiver will be initiated by the receiver upon startup. The receiver seeks for transmitter device on a given IP range, and upon detection performs a pairing request. The transmitter device responds with a pairing acknowledge, followed by a periodic heartbeat pulse.

A transmitter to transmitter pairing process will be the same sequence, but with the possibility of being bidirectional, as both devices can initiate the request, as well as excluding the periodic Heartbeat in the end of the handshake.

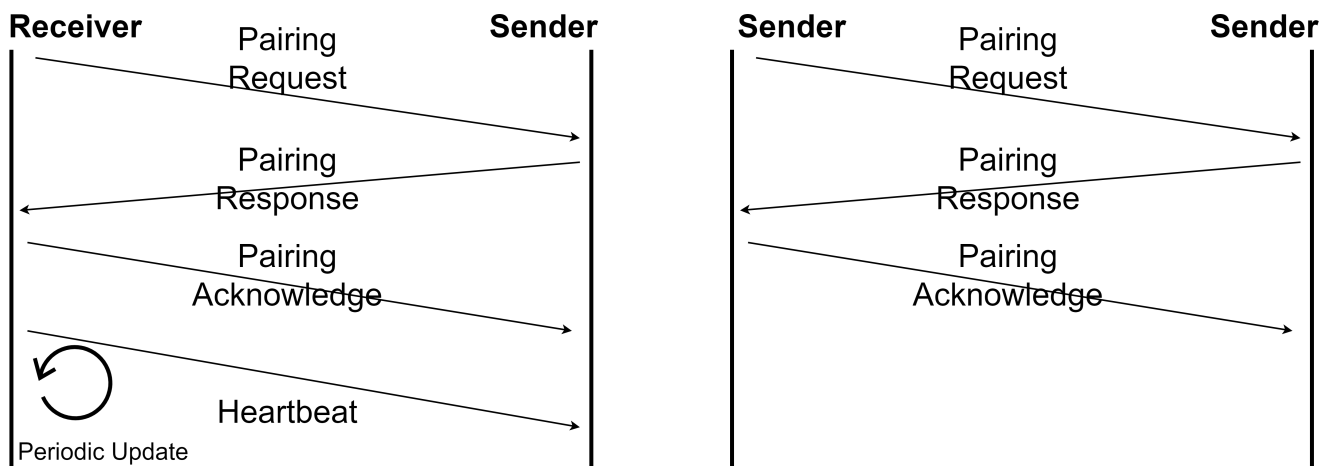


Figure 70: Protocol Pairing Handshake

The heartbeat will contain necessary periodic data which will be displayed on screen on the transmitter device, to enable health monitoring and overview of the deployed receivers. Each heartbeat will contain in their payload the current state, and a set of health indicators such as battery life, storage usage, sensory data and processing usage. Some will be displayed for the operator to enable a tactical overview of the situation, while some are meant for logging purposes and further training analytics.

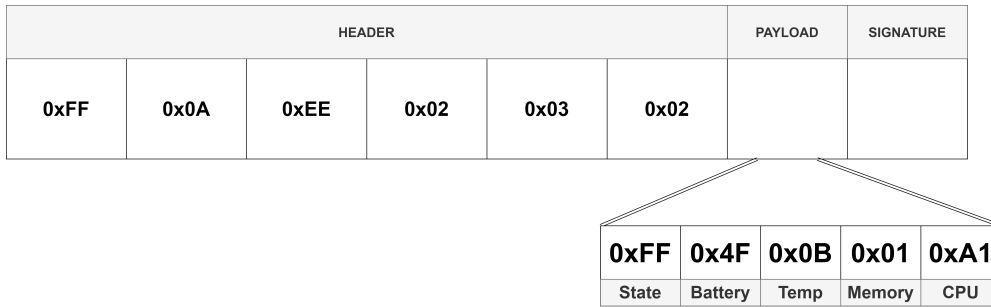


Figure 71: Heartbeat Command

As an example case, 71 indicates a new frame from 0xEE sent to 0x02, carrying a payload of 10 bytes with a heartbeat packet as per category 3 message ID 2, containing a status update on the receiver’s current state, battery levels, temperature, memory and processing usage. Each payload field has a 2 bytes offset from payload start, indicating a new value.

**Operative Execution**

When deployed, a receiver will by default be in standby mode. When initiating the detonation signal, the sequence must be in the specific order of **STANDBY - ARMED - DETONATE**. When the receiver is in armed mode, it should be able to disarm again by receiving a standby mode command from the transmitter device. The frequency of the periodic heartbeat pulse should also increase upon entering the armed state, reducing the time window of when a transmitter device expects to receive new Heartbeat updates.

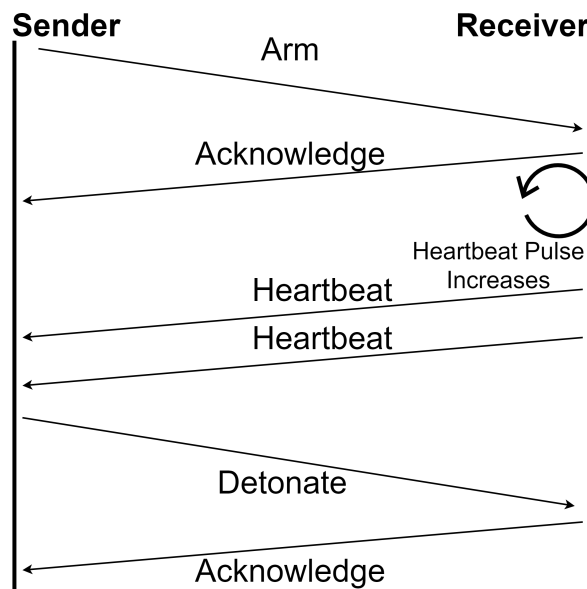


Figure 72: Detonation Sequence

Context Update  
Periodic

The main functionality in the **Data Collection** category, is to provide updates on the context and external environment surrounding the deployed devices. In a field setting, a dedicated transmitter should forward relevant data to nearby transmitter devices, to ensure access to updated context and threats. This is aggregated data forming the heartbeat and waypoint. A forward heartbeat shall contain the relevant data for a deployed receiver to transmitter devices partaking in the mission. A forward waypoint shall contain location data.

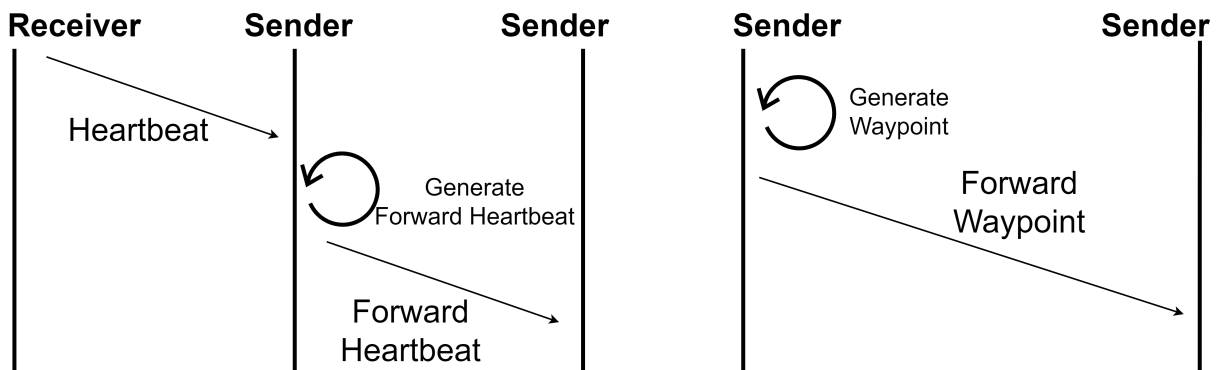


Figure 73: Forward Heartbeat and Forward Waypoint

**Event Based**

**Component Status** and **Situational Context** are event based packets which will trigger when emitting a critical status of a receiver component or physical warnings based on electronic sensors in the receiver device.

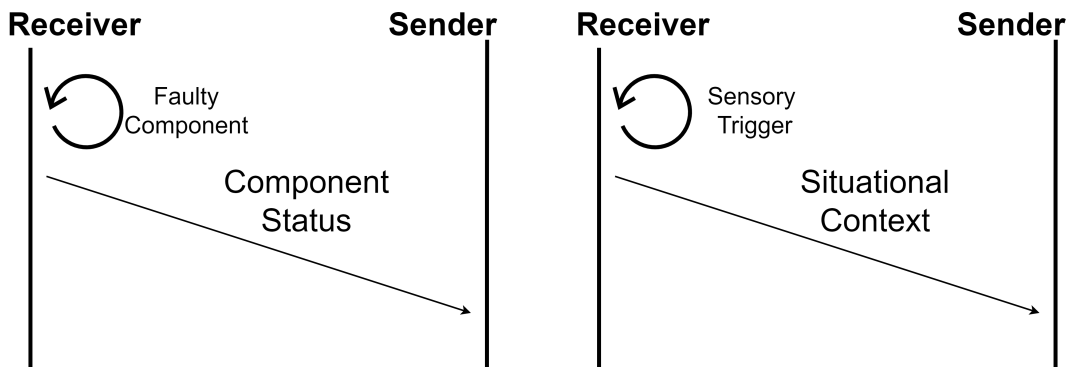


Figure 74: Component Status and Situational Context

### Handover Scheme

To exchange control of receiver devices, a handover transaction must occur to signal deployed receives to connect to a new transmitter device. The transmitter that is to gain control must initiate a "request control" packet, sent to the transmitter that it wishes to extract control of. The in-control transmitter should acknowledge the request, process and prepare a new packet with the necessary data, and emit a "forward control" packet to the deployed receivers. Deployed devices should then close the existing connection and seek for a new device with the given IP address, received from the "forward control" packet.

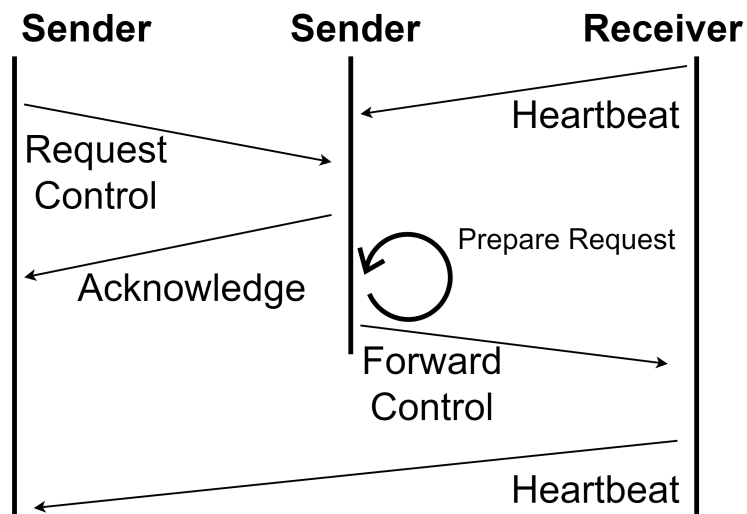


Figure 75: Handover Transaction

### Protocol Security

To avoid third party eavesdroppers, tampering of packets and exploitation of system functionality, a secure communication channel should be implemented to ensure that all interaction between users of the Spark system remains confidential. In terms of the protocol, initial pairing over IP facilitates usage of public key exchange to distribute a secret key for further encryption of packet data, enabling end-to-end encryption. A receiver, upon pairing with a sender, should receive the secret key to use for encryption and decryption of data.

Furthermore, the packets may implement a last header field for digital signatures, enabling packet authenticity. This ensures verification of actions, which traces it back to the original device and operator, providing non-repudiation.<sup>5</sup>

<sup>5</sup>Non-repudiation provides assurance that a person or party cannot deny an executed action. Such as the ability to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on an action or order based on the sending of a message that they originated

### 5.2.4 Verification of Network Communication

To ensure communication between devices is correct and behaves as expected, we've used **Wireshark**[141] for testing and verification of network packets, hooking into the network interfaces and capturing traffic comparing it to our network protocol.

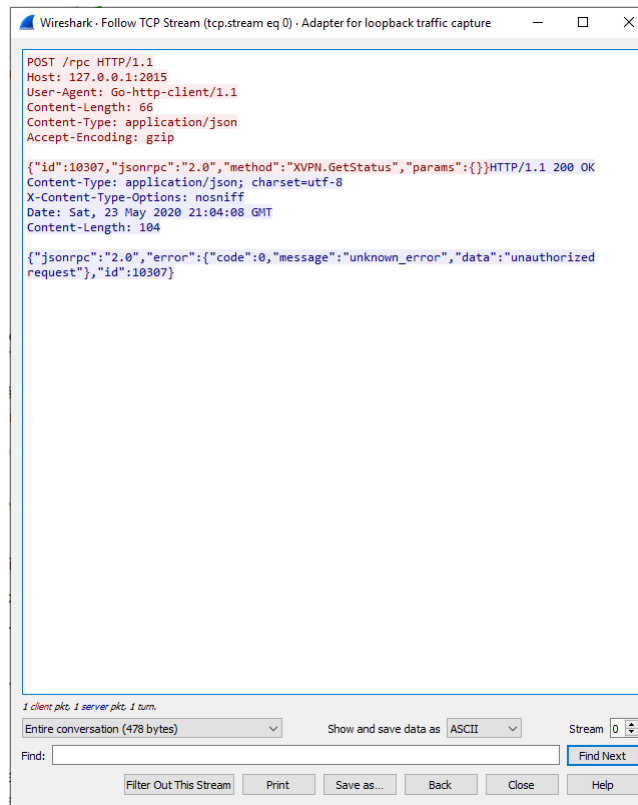


Figure 76: Packet Analysis in Wireshark

To verify the IP communication sequence is correct, as well as packets are intact over time we've simulated the Receiver as an opposite endpoint in software using Python scripts to acknowledge commands and emit heartbeat pulses, testing IP functionality in the Application with network packets designed in subsection 5.2.3 and verifying requirements derived from SR-047 and SR-001:

- SW-022 - SW-021 - SW-021
- SW-019 - SW-018 - SW-017
- SW-016 - SW-015 - SW-014
- SW-013 - SW-012 - SW-010

The testing packets is made up of byte arrays looping upon commands sent from the Android application, and allows us to verify the functionality in practice. The low cost

and overhead to Python allows for a simple yet efficient testing environment. Code is in appendix 8.8.

### 5.2.5 Near Field Communication (NFC)

To facilitate the primary means by which the receiver and transmitter establishes an initial connection, our customer voiced a strong preference towards using NFC, as denoted by SR-010 as part of CR-001. The primary goal initially was for the transmitter to successfully read an NFC tag, and upon reading said tag, visualize in the application that this reading had succeeded. This proved challenging, however, as we were unable to locate any step-by-step tutorial detailing such a NFC implementation using C++ and QT. Consequently, we had to make use of the NFC documentation and an example provided by QT, as well as some help from stack overflow [111, 103, 94], and attempt to boil the functionality down to the bare essentials, and work our way up from there. In hindsight, this proved to be a good thing, as it forced us to think and test for ourselves, as opposed to being "fed with a teaspoon".

Notably, NFC is critical to the Make Decision functionality described in paragraph 4.1.3.2, as it serves as the primary means by which an operator will pair a receiver to the transmitter; without NFC, there can be no decision to make in the first place, as there would be no receiver to control.

QT makes use of a .pro file, which is a project file that defines all files and functionality to be used by the project [104]. In this file, we must declare our intent to use the NFC functionality, which is accomplished with the following line: "QT += nfc" [110]. With this, we are able to make use of the QtNfc library in our header and cpp files (full code in subsection 8.6:

The above snippets of code of the header and .cpp form the heart of the NFC functionality of the application. The two first lines in the header file includes the two most vital parts of the QtNfc library: the *qNearFieldManager* and *qNearFieldTarget*. The former is the class that's responsible for the detection of NFC tags that enter the range of our device [111]. In the third line in the .cpp file, a new object of the *qNearFieldManager* class is created, which is later leveraged to begin the detection of NFC tags. The latter (*qNearFieldTarget*) is a class that provides the interface by which the application can communicate with the target NFC device [107]. In practice, this will allow the creation of an object of the detected NFC device later on.

In order to begin the detection of NFC tags, the `qNearFieldManager` object created in the constructor is used, as shown in the snippet above. When the application is started, the target detection starts, and the transmitter will be ready to detect nearby tags. Once a tag has been detected, another function, called "targetDetected" is called automatically because of QT's signals and slots mechanism (specifically because of the fifth and sixth lines of code in the `.cpp` file showing the constructor). Signals and slots is used to notify one or more objects of a change in another object [112]. In this case, the manager (object 1) has a signal (`&QNearFieldManager::targetDetected`), and if a change happens to this signal (for example, a tag is detected), the object of the NFC class (object 2) has a function that is called as a result (`&NFC::targetDetected`).

### Exposing C++ Objects to QML

One of the greatest challenges the software team faced when developing the NFC functionality emerged from the integration between C++ and QML. Initially, the application made use of a button to manually update QML when a new receiver was to be paired. This was highly undesirable for our customer ([CR-008](#) explicitly necessitates the application to be as easy to use as possible), and therefore, a better, more streamlined solution would have to be devised.

The problem came from our inability to automatically update QML when a change occurred in C++ with the NFC object. Therefore, the best we could do at the time was to fetch the pairing status of the C++ NFC object whenever this button was pressed, and update QML accordingly, as shown in the snippet below.



Eventually, we discovered the means by which we could do this update automatically. To do this, three things were necessary:

- Register the C++ NFC class as a type for QML to create objects from (which later turned out to be a mistake).
- Leverage QT's Q\_PROPERTY, by defining a NOTIFY signal in the C++ NFC class, which is invoked whenever the pairing status of the NFC object changes.
- Use QML's "onStatusChanged" to read the NOTIFY signal defined in C++, and enact an update accordingly.

With the QML now being automatically updated when an NFC tag was read, another problem was discovered: when entering the overview tab (in which the paired receivers are displayed), then exited and re-entered the tab, the receiver had disappeared from the UI! It was discovered that since the C++ class had been exposed as a type from which QML could create instances (which, incidentally, violates the MVC software design pattern), it was recreating the object every time the overview tab was accessed. Consequently, we had to remove control of the object from QML, and instead ensure C++ remained in control, independent of what was occurring front-end.

To this end, it was discovered that rather than exposing the type itself to QML, it would be preferable to expose an instance of the object, controlled by C++, to QML, shown in the snippet below.

Here, instead of using the "qmlRegisterType", which exposes the type itself to QML, an instance of the class is created in C++ (nfcObject), and is subsequently exposed to QML by using "setContextProperty". A connection to this specific instance is created on the QML side, at which point it can be used as normal, as shown in the below snippet.

### The Android Manifest & NFC

Every Android application makes use of an `AndroidManifest.xml` file to provide essential information about the application (such as what functionality and permissions are used) to, among others, the Android build tools and the operating system [30].

Android employs a tag dispatch system wherein the OS will analyze a scanned NFC tag, parse it, and subsequently attempt to hand the tag over to the most relevant application. Since NFC tags can consist of a number of different technologies, Android provides three filters in which an application can define what kind of tag it's looking for. In order for Android to be aware what sort of tags an application is prepared to handle, an *intent filter* can be set in the manifest [31].

The three intent filters Android provides are as follows [31]:

- *ACTION\_NDEF\_DISCOVERED*, wherein an application will be started based on the NDEF (NFC Data Exchange Format) payload of the tag. This is the most specific of the three, as it allows the application to filter for specific kinds of payloads, and therefore there will be no doubt which application should receive the tag, if the filter is set correctly.
- *ACTION\_TECH\_DISCOVERED*, wherein an application will be started based on the technology of the tag itself. For example, Mifare classic is a type of tag that could be filtered for.
- *ACTION\_TAG\_DISCOVERED*, wherein an application will be started if any tag at all is detected.

It's crucial to use one of these filters, as the operating system must be made aware of the application's intention of using the inbuilt NFC chip; failure to do so can (and have) resulted in the OS stating its inability to read NFC tags, as it's not aware of the application's intent of receiving these tags. Particularly if the application is minimized and maximized again, or if the phone is locked and unlocked, the application will lose its ability to receive tags if an intent filter has not been set.

For our purpose, we have opted to make use of the least specific of the three, ACTION\_TAG\_DISCOVERED, as we were interested in rapid prototyping, and this filter let us get started testing quickly and without worrying about what sort of tag we were using. If our solution was to be commercialized, however, it would be more desirable to make use of the ACTION\_NDEF\_DISCOVERED, so that only specific tags could interact with the application.

### **NFC Battery Bug**

As of the writing of this thesis, it must be noted that a number of Android devices (including the S9) can suffer from a bug wherein the NFC will become unreliable when the battery percentage drops below 70%. Attempts to fix this bug have been futile, and according to [16], this is a problem that can only be fixed by a future software update. Since our objective is to develop a proof of concept we have deemed this bug acceptable (we wouldn't be able to acquire a new phone at this point anyway), but if the solution was to be developed further, it's possible a new phone (of the same type, or at least one that can read the same sort of NFC tags as the S9 can) must be acquired. Alternatively, perhaps an OS fault could be discovered with enough time and resources.

### 5.2.6 Software Safety Barriers

A remote firing system is a safety-critical system in terms of dealing with explosives, and therefore a number of safety barriers must be in place in order to prevent consequences of failure. A system is considered safety-critical when a failure could result in loss of life, cause severe property damage, or cause damage to the environment [59].

Along with our electrical safety barriers (section 4.3.4), there are some software safety barriers in place to provide a safe initiation of the remote firing system and to prevent accidental or unpredicted initiation, as well as minimizing the risk of potential failures. This implies that the software components that has been developed must be evaluated in the same manner and just as thoroughly as the hardware itself.

#### Deadlock detection and avoidance

The safety barriers are enabled in both the mobile application and the receiver. There are some safety implementations related to the Android operating system, such as process synchronization and scheduling with focus on deadlock detection and avoidance.

Deadlock is a situation where a set of processes are blocked because they are sharing the same resources. A resource cannot be used by more than one process at the same time, which leads to processes that are forced to wait for each other to complete before using the same resource. Since the processes are blocking each other from using the resource, it eventually leads to both processes stopping.

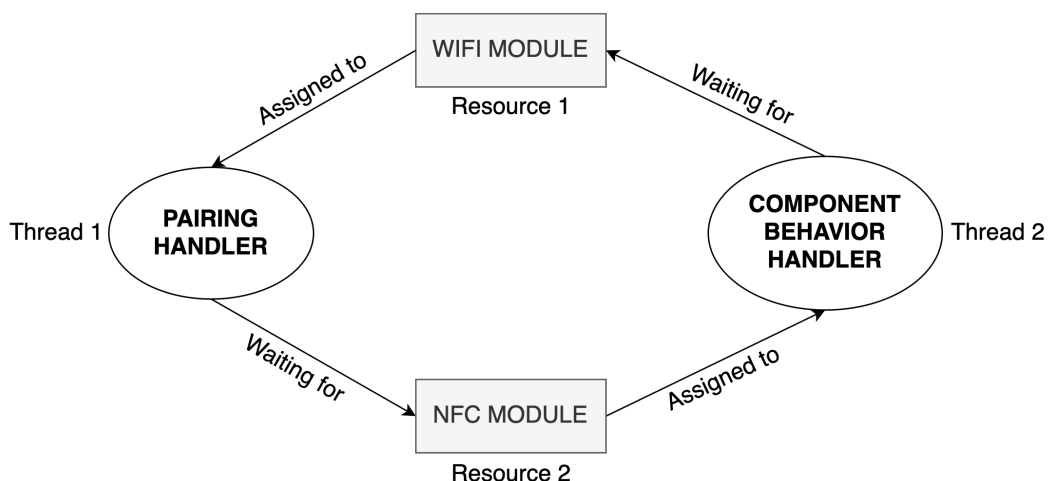


Figure 77: Example of deadlock between two processes

Figure 77 shows an example of two processes waiting for each other, because Thread 1 is waiting for a resource that is assigned to Thread 2, and simultaneously Thread 2 is waiting

for a resource assigned to Thread 1. Since both of the processes are waiting for each other to release the lock, a deadlock condition has occurred.

Deadlocks are however often challenging to debug, especially when there is a large amount of concurrent code in the application.

### **Prevent accidental or unpredicted initiation**

One of the main operations of the remote firing system is to arm and detonate explosives connected to the receiver in a safe manner. There are numerous customer requirements fronting the importance of implementing a safe and failure-free system to prevent consequences of failure, which in worst-case scenario can be loss of life.

There are multiple software safety barriers in place to communicate with the electrical components upon arming and detonating the receiver. The software components are flashed onto the microcontroller unit (MCU), which interacts with the electrical elements every time the receiver goes into arm or detonation mode.

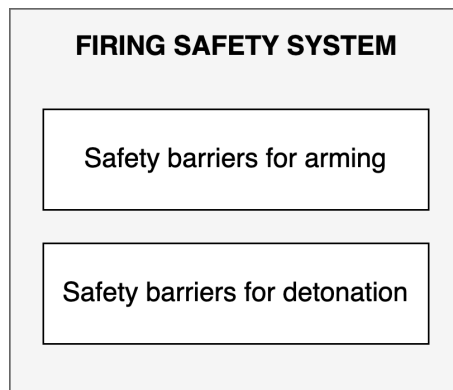


Figure 78: Firing Safety System

The Firing Safety System (FSS) is a subsystem of the receiver, and consists of software safety barriers for arming and detonation. The safety barriers for arming consists of several conditional checks, such as checking the battery level and the capacitor charge before pursuing with the arming process.

As shown in Figure 79, the process of arming the receiver starts with estimating the battery capacity by checking if the capacity is above a threshold of for example 2.5V. This is to ensure that there is enough power to drive the operation through the circuit.

Both of the capacitors are checked upon arming, in order to examine whether they are charged with electricity or not. The electrical switches are tested as well, to ensure that

the detonation mechanisms are working properly to initiate the plasma igniters. It is crucial to perform this test before charging the capacitors, because otherwise the electrical switches would lead current past the switches and cause an premature initiation.

The last part of the arming process is to update the mode indicator (LED), in order to inform the operator that the receiver is armed. This will update the receiver status on the mobile application as well.

All of these safety checks are controlled by the software components implemented onto the microcontroller unit, and they must be passed before the receiver is considered as armed.

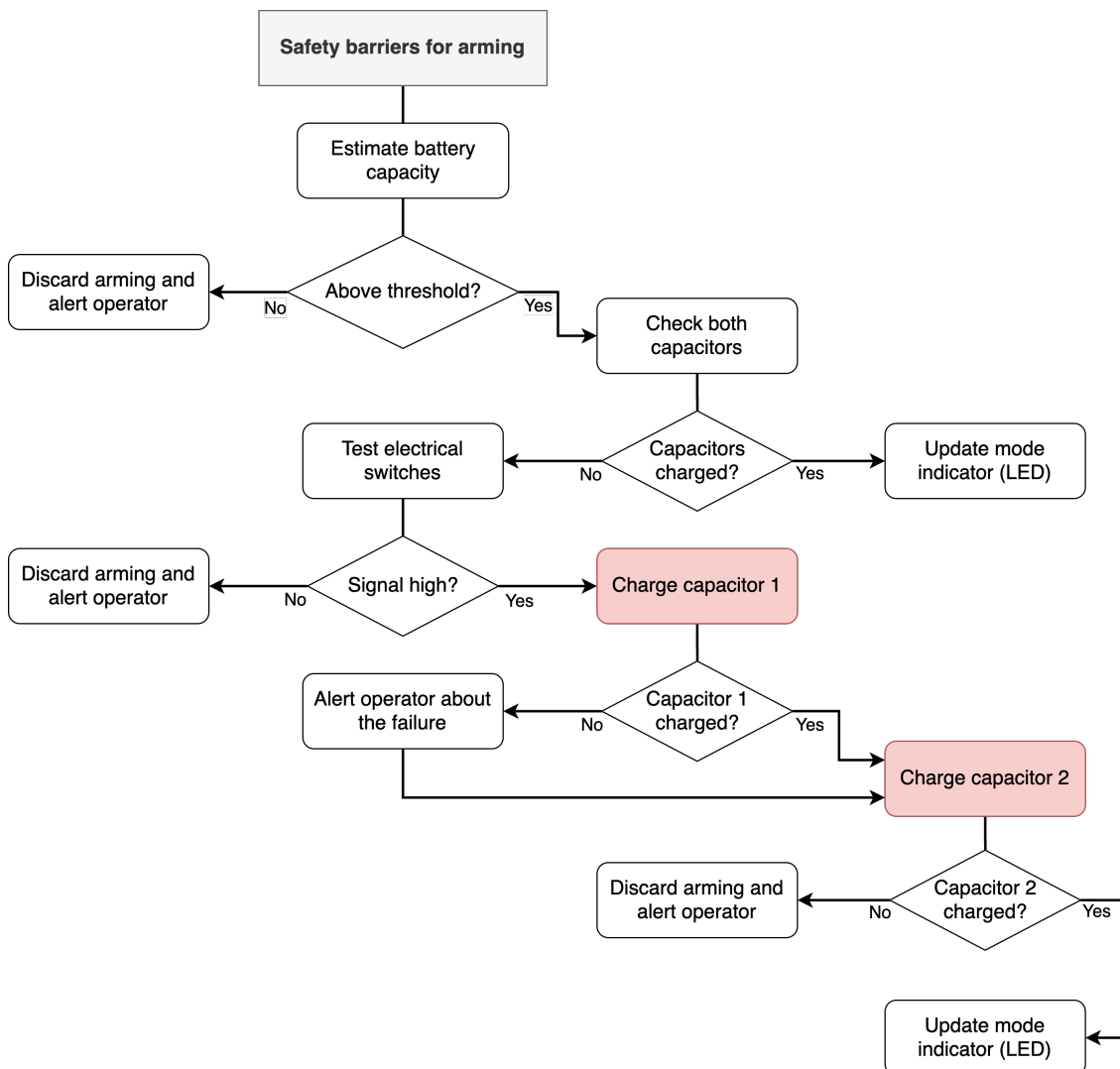


Figure 79: Safety barriers for arming

The red-colored boxes in Figure 79 are subcomponents of the Firing Safety System, and they are also controlled by software running on the microcontroller unit upon charging the capacitors. This process is shown in Figure 80.

The arming operation will be paused while the capacitors are charging. If the process is complete and the capacitor is fully charged, the charge status variable will be set to high in software, which allows the arming operation to continue.

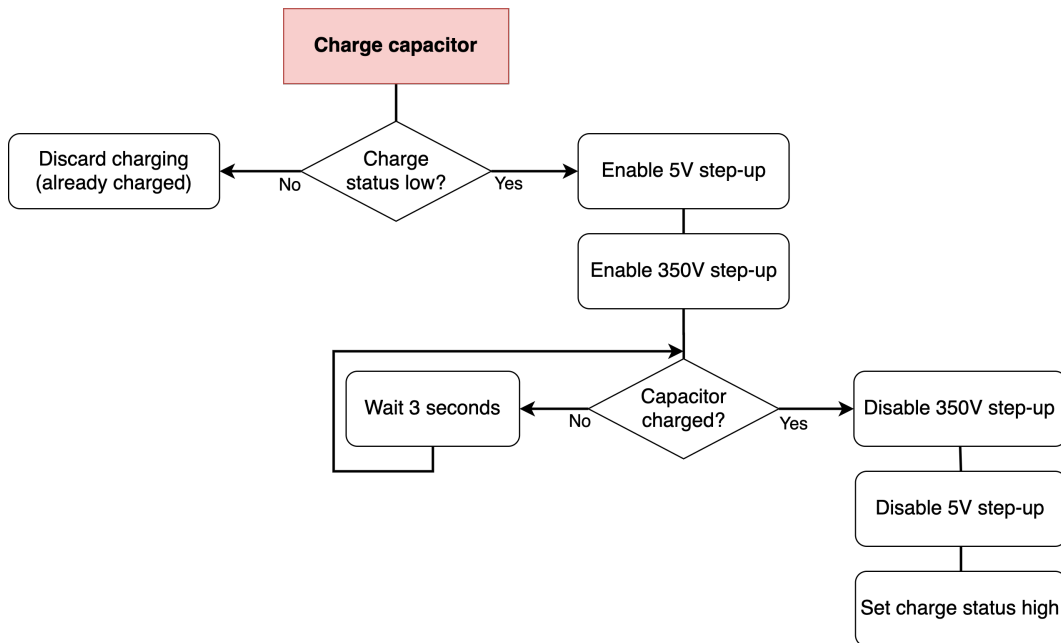


Figure 80: Capacitor charging subcomponent

The process of detonating the receiver is somewhat smaller than the arming process, and there are less software safety barriers in detonation as well. This is because most of the important safety checks have already been performed and the electrical components have already been tested during the previous arming process.

Similar to the arming safety barriers, the first safety check for detonation is to ensure that there is enough battery capacity to drive the entire detonation process, which requires the battery capacity to be above a specific threshold, such as for example 2.5V, which is more than sufficient for this operation.

The next software safety check is to ensure that the capacitors are charged before proceeding with the detonation process. Simply by reading the charge status, which was set during the arming process, it will know whether the capacitors are charged or not.

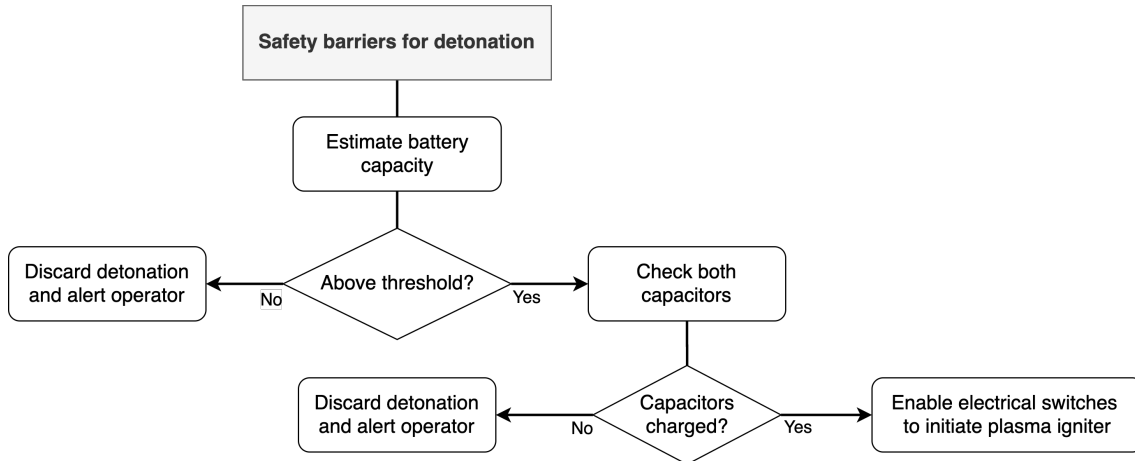


Figure 81: Safety barriers for detonation

The last step is to enable the electrical switches by changing its state, which will initiate the plasma igniters to subsequently initiate the explosives connected to the receiver. The operation is complete at this point, and the mode indicator (LED) and the mobile application will be updated to inform the operator that the receiver has been detonated.

The software-electrical interactions are necessary to provide a safe initiation, since the electrical safety barriers are providing safety in terms of hardware failure and the software safety barriers are providing safety in terms of ensuring that the digital circuits are functioning properly before pursuing with an arming or detonation operation.

The Firing Safety System reads the digital states of the electrical components, and thereafter performs an operation based on these inputs and outputs. The system would not be so reliable in terms of safety if we only had electrical safety barriers in place, and it would neither be reliable if we only had software safety barriers. It is the combination of these two subsystems that provides a safe initiation of the remote firing system, especially in such a safety-critical system where the consequences of failure are severe.



### Watchdog implementation in software

Similar to the watchdog implementation in hardware (section 5.3.4.4), there is a watchdog timer implemented in software as well. This is implemented a little differently, even though the concept is still same. The main operation of the software watchdog is to continuously perform specific operations in the background, based on a regular interval.

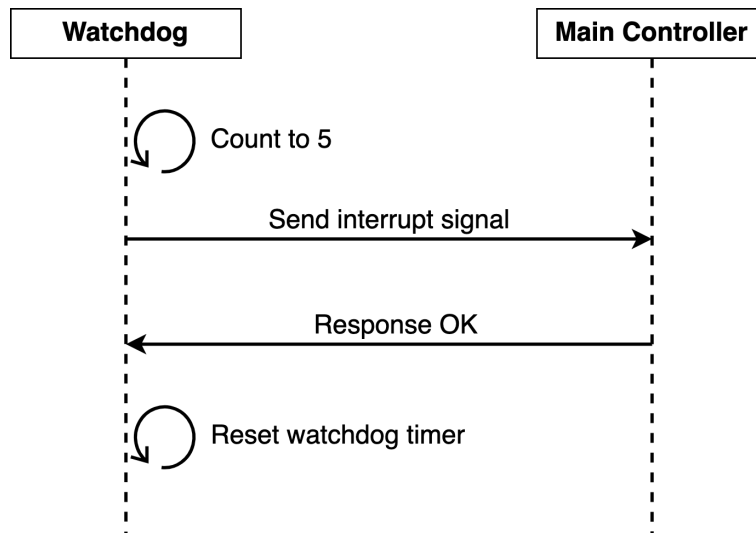


Figure 82: Watchdog timer operation

A watchdog timer is used to prevent functional flaws in both software and hardware, such as when the program hangs or the electronic device malfunctions. The watchdog is implemented using a timer which increases in parallel alongside the main controller.

As the timer increases and reaches a specific given value, the watchdog drives an interrupt signal to the main controller and waits for response. If there is no response from the main controller within a given time frame (a timeout), then it is safe to consider that the program flow has hung somewhere, or the main controller is stuck in a bad or invalid state.

The watchdog will then emit a reset signal to the main controller, in order to release the invalid state and get back in routine. This operation is shown in Figure 83.

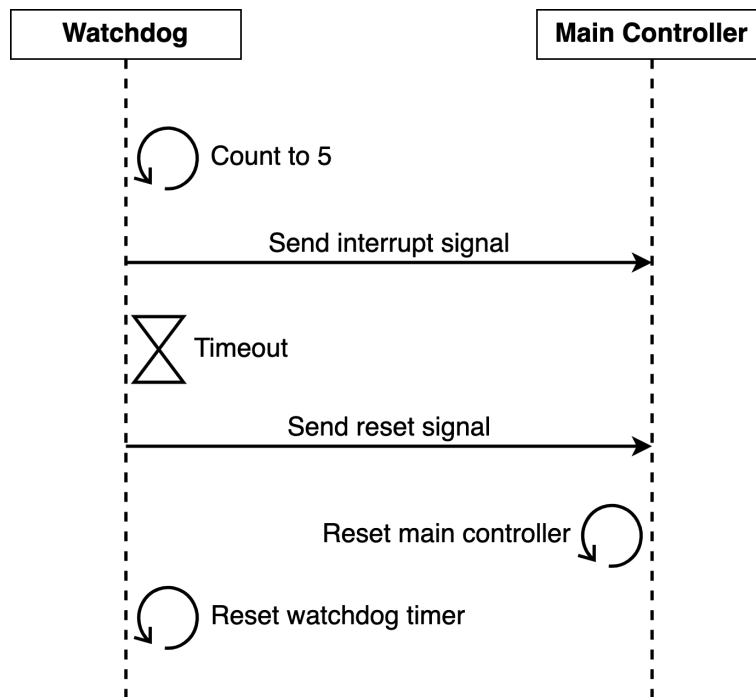


Figure 83: Watchdog timeout operation

The watchdog timer can be used to detect a deadlock in the software as well. For instance, if there is an infinite loop where the watchdog timer counts up to five seconds and then resets itself by receiving a response signal from the main controller, then it would be quite easy to react if there suddenly is no incoming signal.

There should be no running process that locks any resource for more than five seconds, which indicates that a resource has been blocked by a running process in one of the software subsystems, which again prevents the top-level main controller from triggering a reset signal as response to the watchdog timer.

Since the software code is stuck in a bad state somewhere and it is not able to recover itself from the deadlock, an easy solution to this problem is to reset the entire main controller, which forces the software to reload itself and start over again.

The watchdog in software is not only used for preventing malfunctions, as it can also be used to perform scheduled operations alongside its main purpose, such as checking component status or reading sensor data.

The most common operation is to check the battery status, in order to be aware of the battery level on a regular basis. The watchdog would trigger an interrupt signal at for example every half minute and interact with the battery component to check its capacity, and then forward this data to the paired sender.

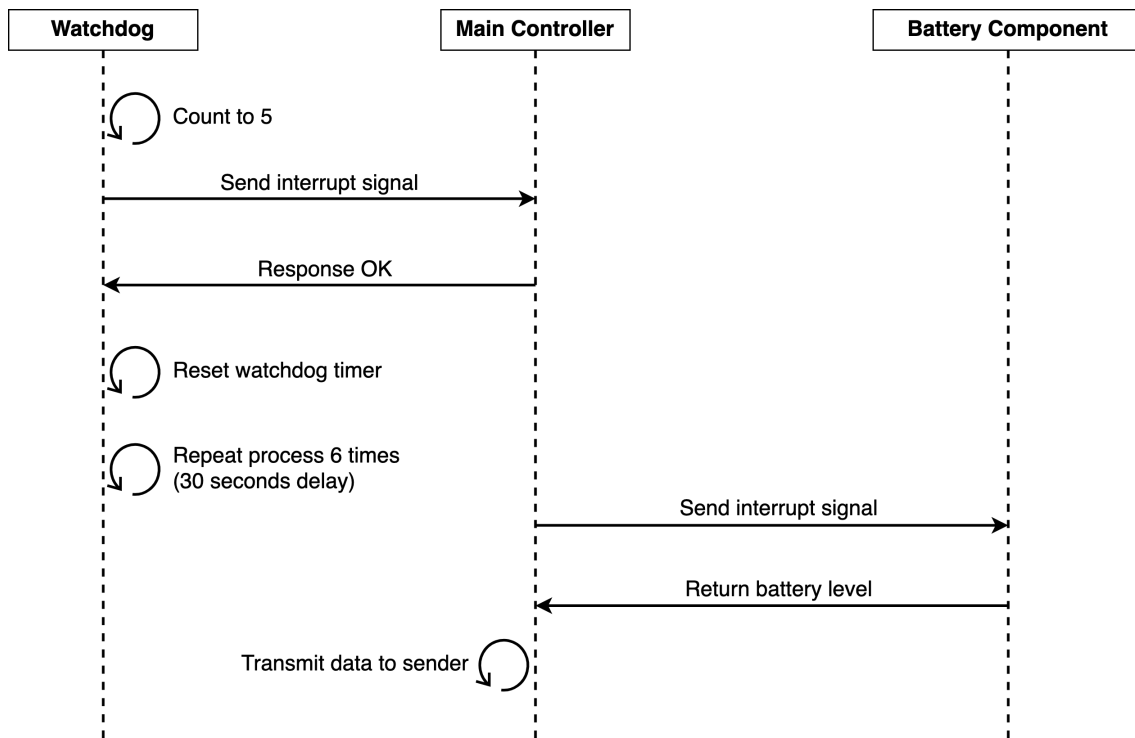


Figure 84: Using watchdog timer to check the battery level

Other data that can be transmitted through the software watchdog is for example component behaviour or diagnostics, which should be reported to the paired mobile application on a regular basis. The watchdog is suitable for this kind of operation, since this data is not critical and it causes to harm to delay it as a timed schedule.

But if any component fails or a critical error occurs, then it would transmit an instant signal to the paired mobile application and it would not wait for the watchdog timer to count first.

The watchdog timer can also be used to wake up the electrical circuit board from sleep, for instance if the circuit is forced into a power-down mode to save power and the watchdog timer wakes the circuit up every minute to read some sensor data, and then forces the circuit back to power-down or sleep mode to conserve battery life.

There are specific instructions custom-designed to control the watchdog timer in software, and each instruction consists of 1 byte. Along with the ability to enable, reset and interrupt the watchdog timer, there are four additional data bits used for transmitting extra data as a payload, such as the battery capacity or the state of a component.

#### Watchdog Control Register

Bit offset	7	6	5	4	3	2	1	0
	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>WDS</b>	<b>WDR</b>	<b>WDIE</b>	<b>WDE</b>
Initial value	X	X	X	X	0	0	0	0

WDE = Watchdog Enable    WDIE = Watchdog Interrupt Enable    WDR = Watchdog Reset  
 WDS = Watchdog Sleep    D1-D4 = Data bits

0 = LOW    1 = HIGH    X = Don't care

Figure 85: Watchdog Control Register (custom-designed)

The watchdog timer starts by setting all bit offsets to zero, in order to completely reset the register. The counter starts increasing when the watchdog is enabled, and it keeps counting until it is set to interrupt mode. This mode will transmit an interrupt signal to the main controller, and it will wait for a response as well.

#### Watchdog Control Configuration

WDE	WDIE	WDR	WDS	Mode
0	0	0	0	Stopped
1	0	0	0	Enabled (count)
1	1	0	0	Interrupt mode
1	X	1	X	Reset mode
1	0	0	1	Sleep mode

Figure 86: Watchdog Control Configuration (custom-designed)

If the response is valid then the watchdog timer will reset itself, and restart the timer process. The control register is shown in Figure 85 and the configuration of this register is shown in Figure 86.

The watchdog operation is shown in Figure 87, where it goes from stopped to enabled state, and then stays in this state for five seconds. The instruction bitstream would be 1000XXXX, until it is finally ready to transmit an interrupt signal to the main controller, which sets the bitstream to 1100XXXX (WDE to high, WDIE to high, WDR and WDS to low, and the data bits don't matter in this instruction).

### Watchdog Timer Process

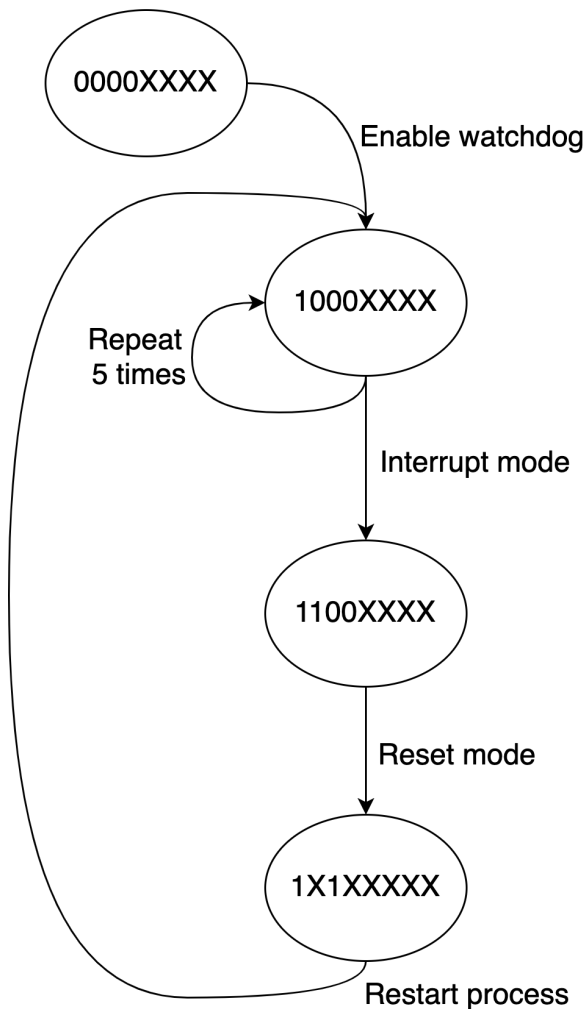


Figure 87: Watchdog timer process

### 5.2.7 Software Security Concerns

From a software perspective it is essential to develop a safety-critical system involving advances in areas such as specification and architecture, along with problems and threats related to information systems security [59].

It has become clear that the attacks towards information systems are growing and it is a major challenge in today's modern world of technology. Cyber security attacks can have devastating effects upon a system, and therefore must be prevented in terms of properly implemented safety barriers in both hardware and software.

Some of the major security challenges from software perspective are data confidentiality and integrity, to prevent hacking and to ensure that the system can be trusted as safe and secure. There are different security concerns within the different aspects of our systems, such as the mobile application, the receiver, or the communication channel between them.

It is significantly more challenging to ensure that the mobile application is secure, due to the fact that the application is running and relying on the Android operating system. There might be small or large vulnerabilities within the operating system that might affect our application in one way or another.

The receiver of our remote firing system is however somewhat less complicated to secure in terms of data confidentiality and integrity, because the embedded software is running on our own designed printed circuit board and there is no operating system running in the background with lots of other applications and bloatware.

One solution to provide a level of data confidentiality would be to encrypt the messages within our systems, and to encrypt the messages that are transmitted amongst our senders and receivers. Encryption transforms the data content into ciphertext, which means that it cannot be read as plaintext. This means that the message needs to be encrypted before leaving one system, and it must be decrypted when entering another system.

Even though encryption and data confidentiality are necessary to provide secure communication, there are other aspects of software security and system safety that might be useful to investigate as well. There are many standards available with guidelines and requirements for functional safety and security, such as the known IEC 61508 standard.

### 5.2.8 IEC Functional Safety and IEC 61508

IEC 61508 is an international standard for the functional safety of electrical programmable systems. The primary purpose of the standard is to create requirements intended to achieve reliable systems that work properly or fail in a predictable manner [77, p. 3].

The overall scope of the IEC standard is to help system designers create reliable systems that functions properly, or fail in a foreseeable manner. Most of the requirements derived from the IEC 61508 are considered to be classical and common sense practices, extended from prior quality standards and general software engineering practices [77, p. 4].

The standard provides requirements for many different aspects of a product development process, such as documentation management, functional safety management, hardware and software design, verification and much more.

Some of the most relevant requirements from IEC 61508 for software design are for example the need to create an architecture that meets the software safety requirements, along with the importance of designing software that is verifiable in terms of functional testing, as well as the ability to modify the software safely [77, p. 131].

Another important requirement from the standard is the importance of designing a system that meets the requirements for system behavior upon detecting a fault. The system shall detect the flaws and then act accordingly [77, p. 131]. This is especially a very high priority in safety-critical systems and real-time systems, where there might be serious consequences upon failure, such as in the case of our remote firing system.

For software implementation, there are some general requirements such as the concern of implementing a system according its detailed design, and to make sure that the software and hardware safety requirements are fulfilled as well. The system must have a detailed plan in terms of validation test procedures, with clear acceptance criteria in order to pass or fail the tests [77, p. 153].

Even though the standard covers large aspects of functional safety for software, it does not describe much about the need for cyber and information security. That is because cyber security was considered as too new to be included in the 2010 release of the standard [77, p. 5].

It has however become a more popular topic during the last couple of years, and cyber security for software and embedded products is expected to become a part of the IEC

61508 standard during future releases [77, p. 5].



### 5.2.9 Android Application, Implementation Overview

The application as of now is made to facilitate functionality derived from **Make Decision** Use Case, as seen from Figure 23. The static overview is colored and divided into *three* areas of either an **Interface**, **Controller** or **Data** object. The Application supports the following functionality and use cases;

- IP Communication
- NFC Communication
- Administrative Team Overview
- Tactical Warning System
- Tactical Map System
- Detonation and Arming Controls

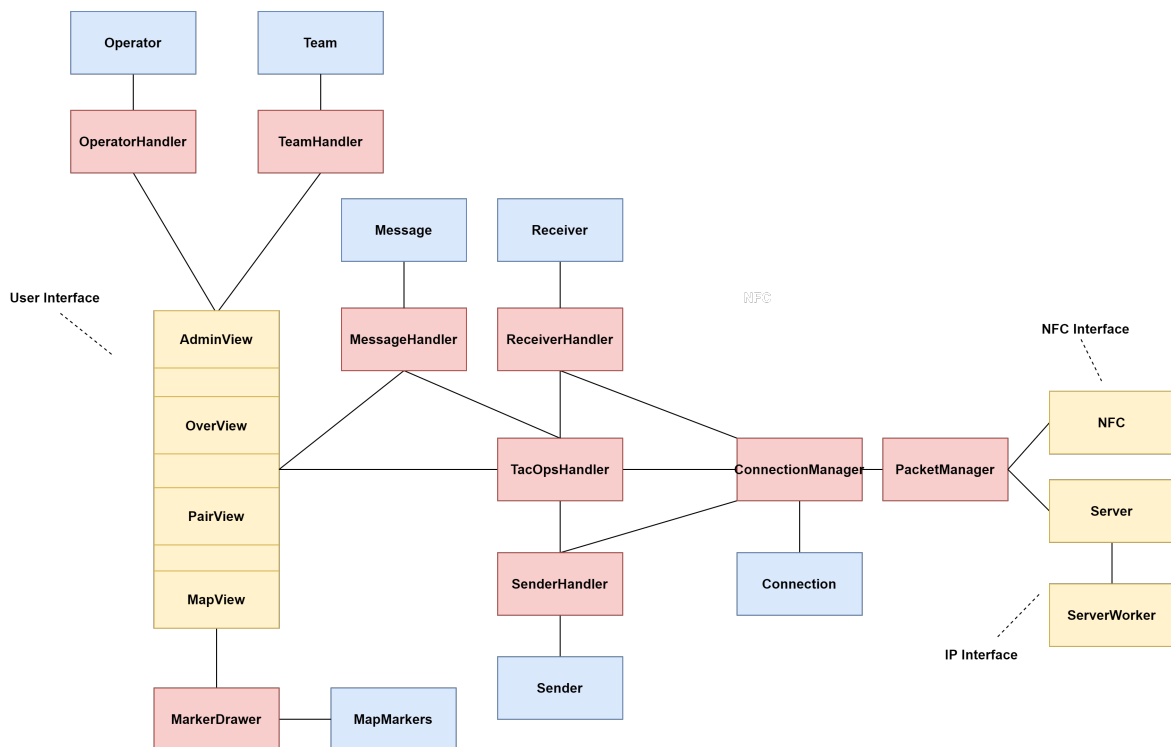


Figure 88: Overview Android Application

All implementation is made with intent to be a *Proof of Concept* System aiming to cover the most important functionality with an emphasis on Requirement CR-001 as the initial waypoint for software application development. Current work comprises of a front-end implemented in QML and a back-end implemented in C++. The overall application is

split into 4 QML components and 19 C++ Classes. Lateral Object communication is realized using the Signals and Slots event handling setup [36], utilizing QT's integrated event handling system for callback communication and message passing between front-end and back-end, as well as via NFC and IP interfaces.

### 5.2.10 Communication Controls

Backend Communication Controls of the Spark App is divided into 5 areas;

- Packet Management
- Connection Management
- Sender Device Handling
- Receiver Device Handling
- Tactical Operations

For each of these areas there is a dedicated object which will serve as a controller for its domain and oversee normal functional behavior.

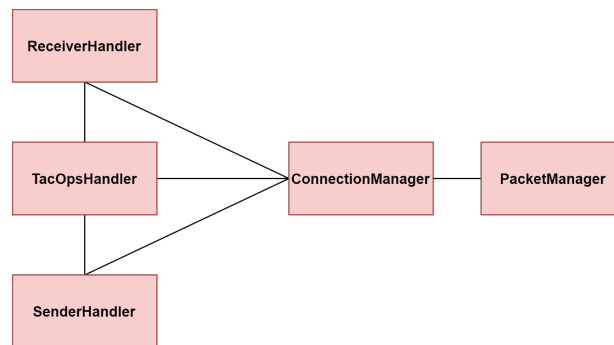


Figure 89: Controller Static Overview

#### 5.2.10.1 TacticalOperationsHandler

For tactical operations, all operative decision making from the app graphical user interface, will signal this object, then prepare and forward operational commands to be sent out to deployed receivers. This includes the arming-detonation process as well as the handover controls scheme. All app user interaction will be signaled to this object. The TacOps handler will be interfaced directly with the ConnectionManager, messageHandler, ReceiverHandler and SenderHandler.

In the TacticalOperations header, we've focused on the ability to Arm, Detonate and Disarm the device based on button pushes from QML and initiating a process further backend. Detected operational anomalies from connected devices is relayed to the messageHandler to notify the operator. Whenever a command or action is sent out to a device, the TacOpHandler will notify the device handler of a pending transaction. A timer will be set off in

the ConnectionManager, and notify if the command is not acknowledged by the receiving device.

### 5.2.10.2 ConnectionManager

All incoming connections will be overseen in the ConnectionManager object, which will maintain a table for all pending and active connections via a **Connection** class, as well as process and respond to packets received from the two communication interfaces IP and NFC. It is the heart of the controller objects, and will forward data on behalf of the receiver, transmitter and TacOps handlers, acting as a message relay. The ConnectionManager communicates directly with the PacketManager object.

A lot remains to be implemented in the ConnectionManager class, especially notification of terminated connection to dataHandler objects such as ReceiverHandler to do memory cleanup and deletion. The class, like PacketManager, is one of the larger classes, and so far has implemented functionality to facilitate the tactical operation derived from the **Make Decision** Use case in Figure 23.

When it comes to Connection.h, objects are instantiated upon setting a new active connection by ConnectionManager. This class is further used and handled as connection data, status changes and provides timer functionality for incoming and outgoing communication between devices, such as timeouts for heartbeat and TacOps commands when emitted to receivers for acknowledgment.

### 5.2.10.3 PacketManager

PacketManager is the endpoint for all incoming data sourced from either the NFC or IP interface, before forwarding it to the ConnectionManager. The PacketManager cleans and interprets the meaning of all packet header content, before either discarding the packet if non-valid or forwarding it further down the control chain for processing.

Being on of the larger classes in the application, there remains quite a bit of functionality in the PacketManager class. Future work would be an encryption scheme for distributing keys and encrypting/decrypting packets as they are transmitted. Functionality for the *handover transaction* as well as intercommunication between Sender devices has not been prioritized, and is not implemented.

## 5.2.11 Device Data

### 5.2.11.1 ReceiverHandler & SenderHandler

The ReceiverHandler object will receive updates on the status of deployed receivers and is the only object able to tamper with the data of receiver object. The object communicates directly with the ConnectionManager object.

The SenderHandler, similar to the receiver setup, will fetch and handle updates about connected devices and its users, and is the only object allowed to alter or change status about transmitter devices. It interfaces with the ConnectionManager object.

The system will not implement a persistent database, and will therefore only monitor status updates from the connected devices as a proof of concept for the SPARK system. As the design is implemented now, it should be trivial to interface new functionality with existing ones to provide for a static storage of analytics, training stats and app-user interaction logs. Our solution will contain data in dedicated device structures instantiated either by the ReceiverHandler or SenderHandler as new connections are established, per Figure 90.

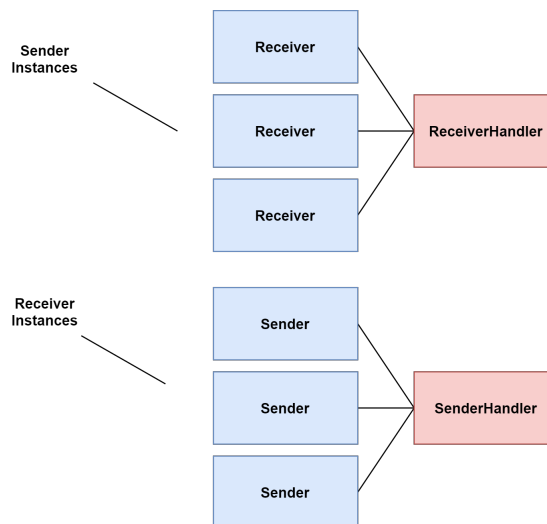


Figure 90: Handling of receiver and transmitter Data

Only the ReceiverHandler and Receiver classes has been implemented so far, as communication between Sender devices had lower priority. The ReceiverHandler is so far the only classes implemented as a singleton class, but most controller classes could fit this as they are only instantiated once. It contains management functions to instantiate and update receiver data as they are connected, if anomalies are detected it will emit signals to the TacOpsHandler. Proper object cleanup and deletion of receiver objectst as connections are

terminated still remains, and there is not implemented a trigger in ConnectionManager to notify about old connections. This would be a next step to focus on in this class.

### 5.2.12 IP Interface

The IP interface of the application will comprise of a TCP socket server which will handle communications between deployed receivers and other team members transmitter devices. The server shall provide an interface for new incoming connections, and upon a request, should handle the pairing sequence and relevant data sent by the requester. The server should maintain a upper limit of allowed pending connections as to not hog too much processing power or risk malicious exploitation of app functionality.

The server is implemented in a worker pattern originally inspired by [35] which upon a connection, will instantiate a new ServerWorker object dedicated to handle that client alone. This aids control of each connection and isolates them from one another, as well as allowing for a multi-threading scheme to increase performance and enable more connections simultaneously.

This class builds its functionality upon the QTcpServer and QTcpSocket libraries in QT to allow for socket management and data transmission over TCP protocol. The ServerWorker communicates with the Server on a 1 to 1 basis with the Signals and Slots functionality, notifying new packets and upon creation by the Server, directly receives the handle to a socket file descriptor which it further operates on.

The current implementation is realized in a single thread process, but is built to be scalable with a multi-thread implementation via the signals and slot callback method [36] which supply thread safe event calling connections in the QT framework.

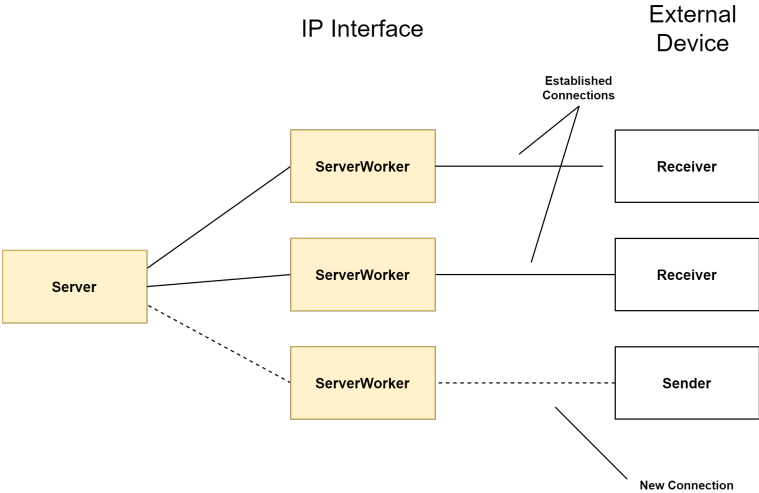


Figure 91: Server Static Overview

Overall, the architecture is designed and implemented in a *Divide & Conquer* style to enforce a single purpose principal<sup>6</sup> of each object. This also enables ease of testing and verification of each module, as well as reducing the dependency between objects to be able to perform their tasks. All object interaction is implemented using the *signals and slots* functionality of the QT framework [36]. Do note that the way of programming has been focused on reaching as much functionality as possible, in the shortest amount of time. As code grows complex and dependencies increase, the code should be refactored to provide a good programming overview for further work, as well as encapsulating classes and member data between objects providing a safer class interface and enforcing good coding style.

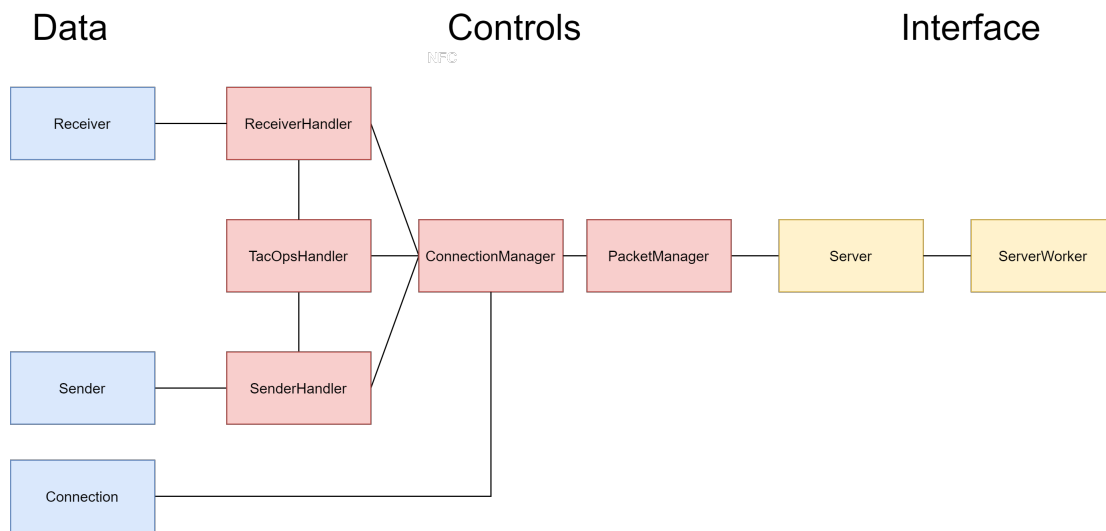


Figure 92: IP Communication Object Interaction

<sup>6</sup>Each object should have a clearly set specialized function and purpose.



### 5.2.13 Functional Behavior

From the diagrams in appendix, subsection 8.9 shows the object interaction upon a new pairing handshake. The handshake will allow for the client to initiate the process, the server to exchange and share its public encryption key, and the client to verify its success by the final pairing acknowledge. Based on Figure 91, each existing ServerWorker will only interact with the top level server, and upon a connection close or timeout, will terminate the connection completely within a given timeframe. Yet a sudden unplanned interruption or loss of connection should not terminate the connection immediately. The worker should notify its server parent which further notifies the ConnectionManager about an anomaly. A device which then reconnects to the given device, should allow further communication without having to go through the pairing process again.

From subsection 8.9, the pairing will set a connection as pending upon initial request, and after the pairing acknowledge, set the connection as active. Although not yet implemented, if a request or pairing packet deviates from the protocol, it should discard the packet. In a top view to the right; the data will then be able to flow through the IP interface, either from a new device, and then be dealt with accordingly, or through an existing connection already paired via the NFC interface. The packets will be parsed, evaluated and forwarded to the correct instance of the device to update relevant data.

When incoming heartbeats are received at the IP interface of a transmitter device, there will be initiated a callback chain to forward and process data from the payload. From subsection 8.11 in appendix, we can see data incoming from a receiver device, met by the ServerWorker which forwards it to its parent. The data gets evaluated and filtered, as well as forwarded to the ConnectionManager to assure that the data is expected. If so, the data ends up at its rightful receiver instance, updating relevant data. Figure 94 depicts the sequence of events for a Heartbeat process.

A heartbeat serves two main purposes; update relevant data and if an anomaly occurs, notify the operator. Furthermore, it must act as a connectivity verification for the communication channel between the transmitter device and receiver device, ensuring that the connection is stable and up to date. The heartbeat timer begins upon the initial creation of a receiver device, and will reset whenever a new heartbeat packet is received. If a timer reaches its limit, it should notify the operator via the app that a heartbeat has not yet been received for a deployed receiver. Note that the timer countdown will vary depending on the given state of a receiver. An armed receiver will emit heartbeats with a higher pulse than a deployed receiver in a standby state.

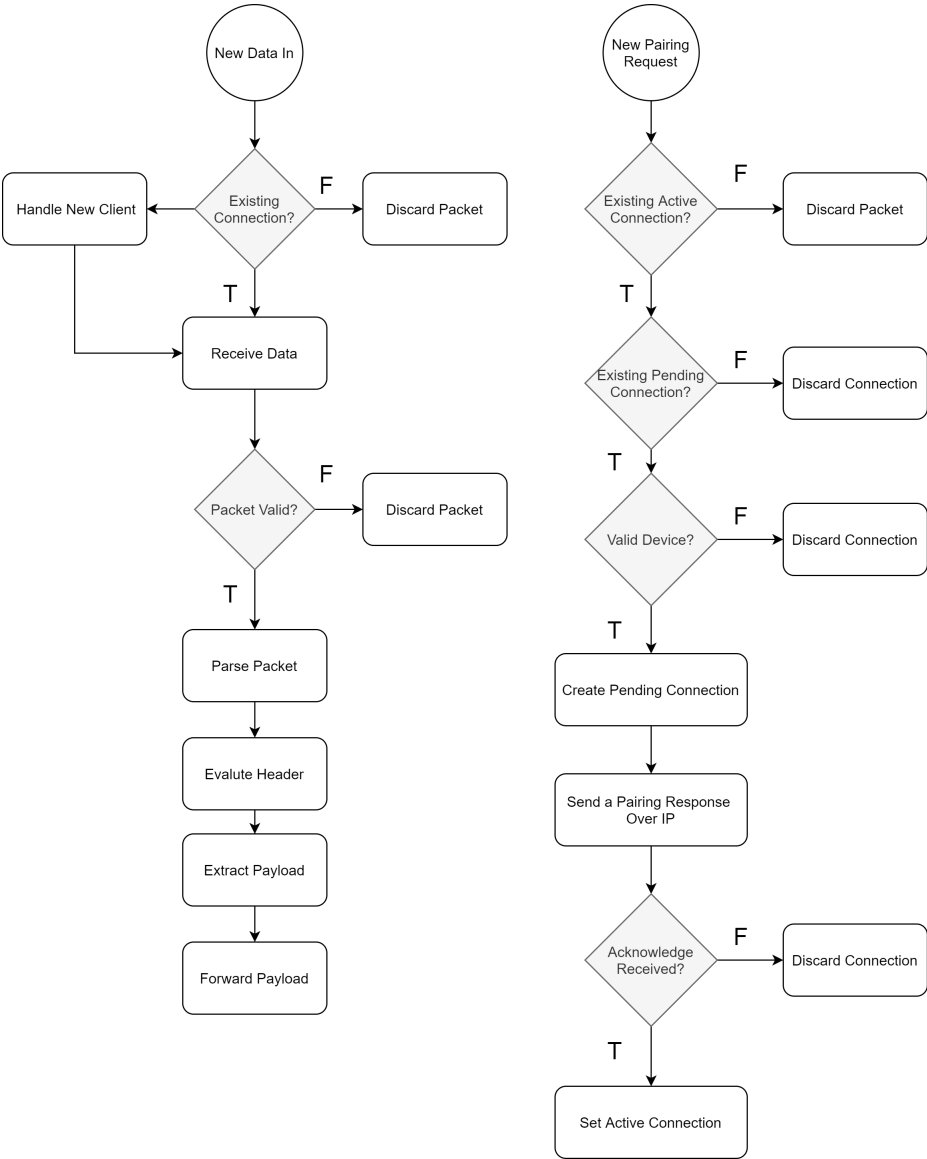


Figure 93: Flowcharts, Pairing Process and Incoming Data

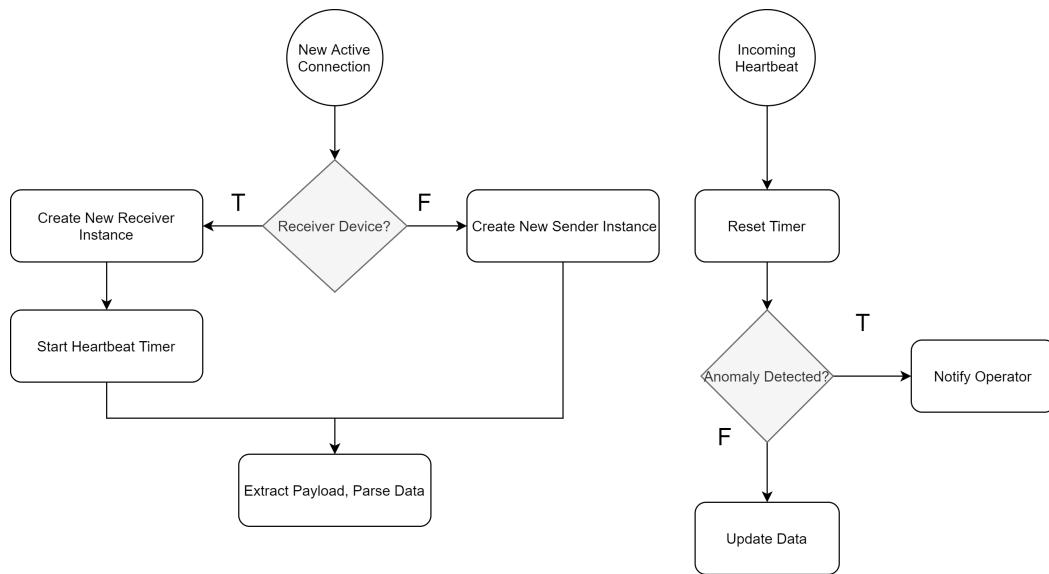


Figure 94: Flowchart Heartbeat Functionality

When it comes to tactical operations, there are three main commands which take precedence; **ARM, DISARM, DETONATE**. These are all critical commands that directly initiate a receiver to either detonate its charge, prepare for a detonation or abort. In addition, one should be able to transmit a *timed* detonation for specific scenarios where the receiver will initiate a countdown. For the back-end handling of these operations, the front-end will signal the back-end when a TacOps button is pushed, all of these actions will notify the TacOpsHandler to initiate the process of sending out the command. First it must verify and validate that the operation is feasible and that the given command complies with whatever states the receiver already is in.

For instance, if a receiver is in Standby mode, it should not be able to transmit a detonation command. The TacOps handler will ensure this by fetching data from the receiver(s) via the ReceiverHandler before proceeding with the command process, signalling the connectionHandler which will further engage the packetHandler in preparing a TacOps packet, sent out by an already established communication channel, represented as a serverWorker.

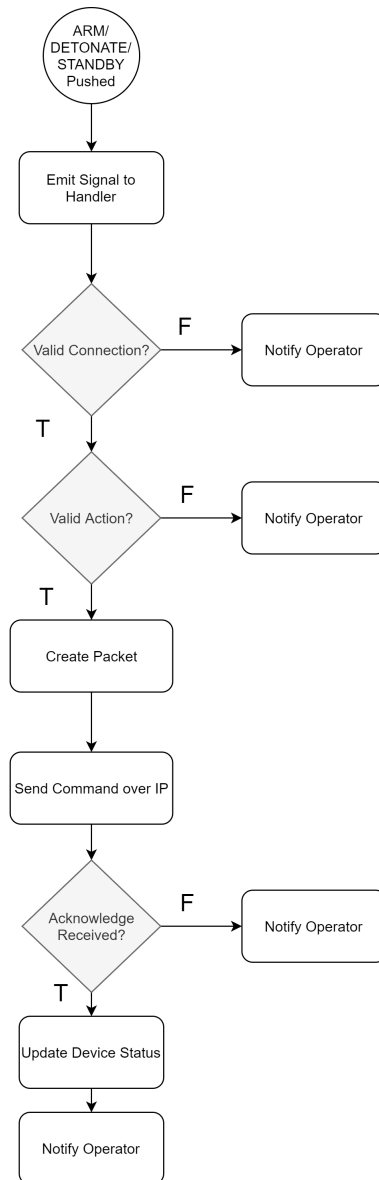


Figure 95: Flowchart Tactical Operations

Actions initiated by the operator should go through a verification check before allowing the process to continue further. If allowed, the packet will be created and transmitted through the IP interface, and a TacOps acknowledgement packet is expected back from the receiver. If not, the operator should be notified that a TacOps action was not successful per SW-019, SW-020 and SW-021. Diagrams of message passing in TacOps commands can be examined in appendix 8.10.

### 5.2.14 Map & Waypoint Functionality

One of the functionalities described in paragraph 4.1.3.2 is that of creating a waypoint on a map; the following section describes the implementation of this use case.

To gain access to Map functions and visualizations in the QT framework, we've used a QML plugin<sup>7</sup> via the location API[108] in QT.

```

1  import QtLocation 5.12
2  import QtPositioning 5.12
3
4      Plugin
5      {
6          id: mapPlugin
7          name: "mapbox"
8      }

```

QML is the front-end side of the QT framework and offers various functionality to build upon in the QML language, so that tools such as map data has no need for back-end handling in C++ and is fully supported and ingrained in the project beforehand, ready to be used by our application. This aids our software development when working in teams, as the dependencies and modules are clear and preset, as well as makes it easier for the team to integrate their work. These specific Geo services reside in the location and positioning libraries, native to the QT framework. For our prototype we've landed on three third-party Geo services to use for map functionality.

#### Open Street Map

Out of the box, Open Street Map (OSM) would be the default choice as it is a purely open source service, free to use. It provides data, images and map information from various contributors under the *Open Database License*. Building on this, the OSM plugin requires strict SSL support to enable communication over the HTTPS protocol in the *Application Layer*, and has no alternative for unsecure HTTP alone. Based on the documentation of QT; QT provides support of secure communication services for HTTP using the OpenSSL library, but so far due to legal restrictions[102] this is not available in precompiled form for android applications in QT. Meaning it must be built from source and then integrated to the .APK<sup>8</sup> file manually. In addition, since QT 5.6.2 available map types are suspect to sudden changes or removal without further notice[109], and so would be an unstable option not suited for critical tasks where the intuitive usability from operators in the field is a critical requirement from the customer.

<sup>7</sup>A QML plugin is an extension module built in C++ which offer additional functionality.

<sup>8</sup>Android Application File Extension

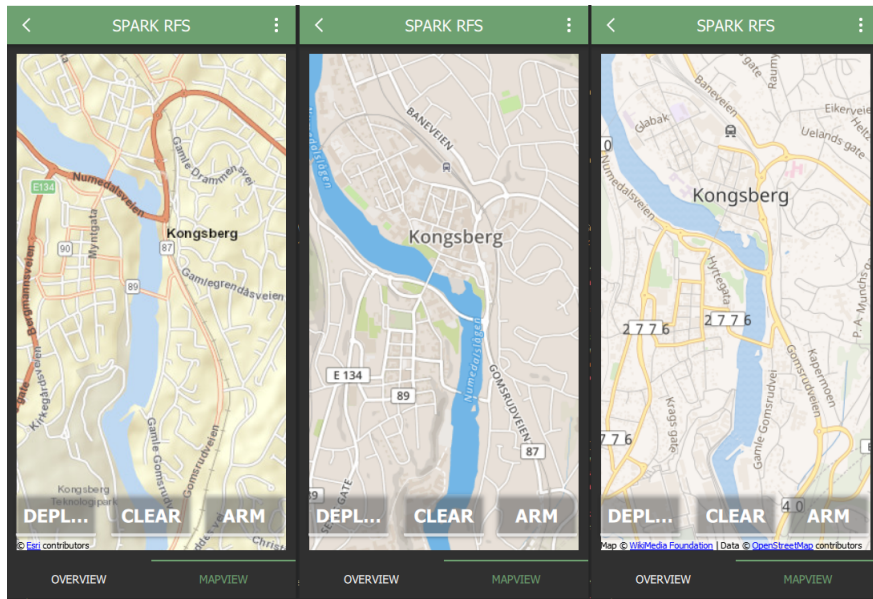


Figure 96: Esri, OSM and Mapbox alternative map views

### Mapbox(GL)

Mapbox and Mapbox GL seems to be a well fitting choice for our product. Mapbox is available with many customizable map options and graphics choices to give our customer the best user experience as possible. Mapbox offers a free tier option for up to 25,000 monthly users, which for our team and product vision is considered an acceptable scalability as of now[75]. Using the Mapbox on our test platform, the maps have a very different look and focus on details which may fit our customers needs quite well. As for use in the field, the zooming of the map will be a very narrow scope, and in need of details such as houses, roads and other urban areas, and Mapbox meets these constraints in a well done manner with their design.

### ESRI

QT also supports map data usage based on ESRI, world market leader in GIS services as well as a free option. The maps seems to be a valid choice for our our usage but lacks customizability in QT and one is forced to accept the stock settings and visual effects provided by QT and ESRI. In addition, all use of the ESRI geo services is subject to the ESRI Terms of Use but has a developer subscription with the following options;

- An app can request up to one million maps per month
- The app must not directly generate revenue and must be free of use for the end user.

As of now these conditions are well inline with our current prototype, but for end use and a push to production, this will not satisfy our customers product vision and plans further

down the production cycle.

Our current setup will be using either the ESRI or Mapbox geo services as map data provider for this iteration of a *Possibly Shippable Product* per Scrum.

### Waypoint Functionality

Using our map solution, we want to be able to add our own waypoints with necessary tactical details for deployed receivers out in the field. We want to be able to do the following;

- Drag and drop a waypoint at current location of an EOD operator
- Assign a relevant name for the waypoint in regards to current operational context
- Arm and disarm modes should be clearly visible on the waypoint map, indicating RED or GREEN colors

To realize this in software, we're making our own visual waypoint markers with QML and subclassing main QT classes such as *QAbstractListModel*[105] and *QGeoCoordinate*[106] which are available in QT to build further upon. Subsequently, we then use this to provide necessary data forming an overlay on top of the map view, at the corresponding coordinates of a deployed receiver (decided by the operator).

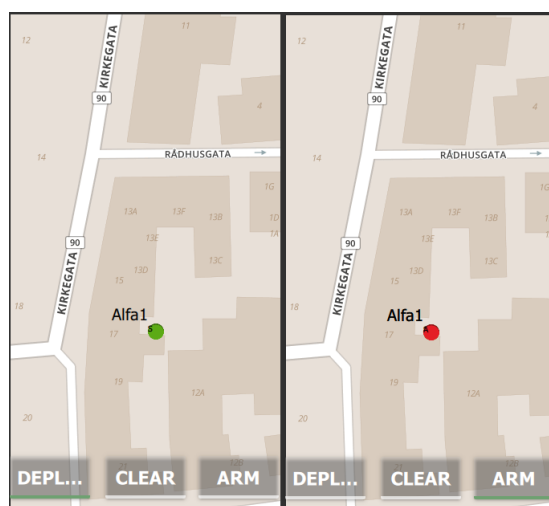


Figure 97: Waypoint States Overlay

The models are implemented using the two QT classes, and is then built back-end in C++ forming "Setters" and "Getters" for initiating callback functions which shall change the states and names of the waypoint markers. StackOverflow threads such as this[93] has

been used to build further upon for solving this problem. The implementation is based on a main *MarkerModel* class which inherits from the QT *QAbstractListModel*, and then uses a *Waypoint* class inheriting QT *QGeoCoordinate* for handling the relevant data in each deployed waypoint on screen. The Main Markermodel class will handle a list of waypoints, which it will push to front end QML on demand as the user interacts with the tactical map view, deploying receivers.



## 5.3 Electrical Design

### 5.3.1 MCU

Microcontroller unit is the device that will integrate the different subsystems to form one system. There are a lot of criteria when selecting the MCU to use for a project. In our case we selected one based on size, power consumption, durability, interfaces and functionality. For the Spark PCB we selected the ATmega2560 microcontroller from Microchip. This is mainly because of its rich functionalities such as many GPIO ports, ADC, UART, USART, SPI, I2C and it's broadly used in many systems today. It provides a solid base of official documentation, and since it is broadly used, a lot of user's experience is shared online. The ATmega2560 has a variable clock frequency that is dependent on the input voltage to the controller. In our case we run the ATmega2560 on 5V input voltage, so that we get the fastest clock speed to comply with EL-011. Since SPI is supported by ATmega2560 and it is used for both Ethernet and Wi-Fi we shall discuss it in the next section.

SPI is a common technology used nowadays for communication with peripheral devices where we want to transfer data quickly and with real-time constraints [100]. SPI is a synchronous, full duplex master-slave-based interface, which means that it uses a shared clock signal to synchronize and that the devices can send and transmit at the same time [53]. The SPI master is the device that generates the clock signal used for communicating. In Figure 98 you can see two devices communication via SPI.

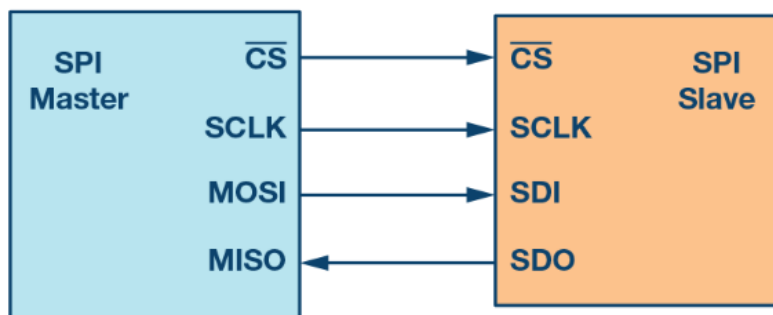


Figure 98: SPI communication [53]

The device that is the master uses one chip select signal to select which device it wants to communicate with. This enables all devices that communicate with the master to share the same signal lines. The only independent signal that each device needs is the chip select signal. In comparison to other communication interfaces such as UART or I2C, SPI is faster because UART and I2C is only half duplex.

Since UART is asynchronous sender and listener needs be set to the same baudrate before-

hand. This is not necessary with SPI since the shared clock is driving the communication. The main downside with SPI is that it uses many signal lines in comparison with UART and I2C [53].

Since we use the ADC on multiple occasions in the Spark electrical system we will study the ADC of our chosen MCU. The ATmega2560 features a 10-bit successive approximation ADC with an absolute accuracy of  $\pm 2$  LSB [12]. With a 10-bit resolution we can calculate the minimum LSB for applications using Equation 6 [20]

$$LSB = \frac{V_{SPAN}}{2^N} = \frac{5V}{2^{10}} = 4.88mV \quad (6)$$

Therefore we are limited to an accuracy of  $LSB = 4.88mV$  for applications in the Spark electrical system. In our design we only use the ADC single channel inputs, because we are only interested in a signal conversion with reference to ground. If an analog input connected to one of the ATmega2560 ADC ports and activated, it experiences a leakage current through a  $100k\Omega$  resistor and a sample and hold(SH) capacitor [12]. Since this SH capacitor needs to be discharged and charged as the input signal varies, it is optimized for analog sources with an output impedance less than  $10k\Omega$  [12]. Since we are only measuring DC voltage in our designs, we don't need to take this into account. The ADC reference can either be set externally via AREF or use internal reference voltage.

### 5.3.2 Communication

#### 5.3.2.1 NFC Module

The Near Filed Communication or NFC module for our system will be the communication channel between the Android device and the receiver when it comes to pairing the devices. When the paring sequence is complete the NFC module will be disabled and the communication between the receiver and the android device will switch to one of the two main communication channels, Ethernet or WiFi.

The NFC module must support at least one of the same standards as the Samsung S9 Android device for the paring to be possible. But for further development and testing of NFC functionality, the NFC module is better off supporting more standards than just the ones of the Samsung S9.

The NFC module chosen for the receiver is the PN7150 from NXP [86] which support the same standards the Samsung S9. But it also supports more standards than what is shown in Table 9.

Supported Standards	Samsung S9	NFC module
MIFARE Classic® 1k (M1K S50)	Yes	Yes
MIFARE Ultralight®	Yes	Yes
NTAG®	Yes	Yes
MIFARE DESFire	No	Yes
Sony FeliCa	No	Yes

Table 9: NFC module supported standards

This module is well documented from the vendor and will make further development on the software and hardware side much easier. One thing in particular is the antenna matching circuitry needed to obtain a good transmission of the signal. Here the antenna impedance has to be matched with the IC input impedance as mentioned in [88].

The basic approach is to have an equally balanced impedance through the transmission line to minimize reflections of the signal. In equation 2.95 on page 110 in [24] we can see that if the impedance is matched we get no reflection between the coupling of the parts of the transmission line and the full signal strength is absorbed by the receiving part .

It is also recommended to use some filtering to avoid unwanted frequencies and noise. Since the carrier frequency of these standards is at 13.56MHz, we will use a filter to match this. After some searching we found an antenna to match the criteria of the PN7150 found in

[88] and at the same time match the circuitry shown in Figure 14 in [87].

We started on the path of calculating the circuitry needed to match the impedances of the antenna with the PN7150. Unfortunately, we underestimated the impact of the ambient factors of the PCB board and components, so we ended up with the suggested component values from the antenna and IC vendor and made space on the PCB to configure the circuitry after it's manufactured and tested properly.

This involves the  $R_Q$  value for the Q-factor of the antenna [19], as the PN7150 guide recommend a Q-factor  $\leq 35$ . For now there is only a 0ohm resistor in its place, but this can be changed if needed. We expect that the  $R_Q$  must be greater then zero as the antenna Q-factor is 49. The same thing is done for the  $R_X$  and  $C_X$  on the receiver pins where they were not recommended to use for antennas  $\leq 800^2mm$  as we are planning to use an antenna of only  $600^2mm$ . But the antenna can be changed and therefore the opportunity to implement  $R_X$  and  $C_X$  at a later time is possible.

With this approach the NFC module will cover all functionalities needed for our system. It will also give our customer the possibility to experiment and test out new ways to use NFC in the future. This is our work towards fulfilling SR-043.

### 5.3.2.2 Ethernet

The W5500 was selected to be used with the Spark PCB. This IC offers up to eight socket connections simultaneously. W5500 can either run in client mode or server mode [137]. This just means that it can either act as a server that is listening for incoming connections on a specific port and IP, or a client that connects to such a server. In Figure 99 you can see the sequence diagram of establishing a socket connection, both in client and server mode.

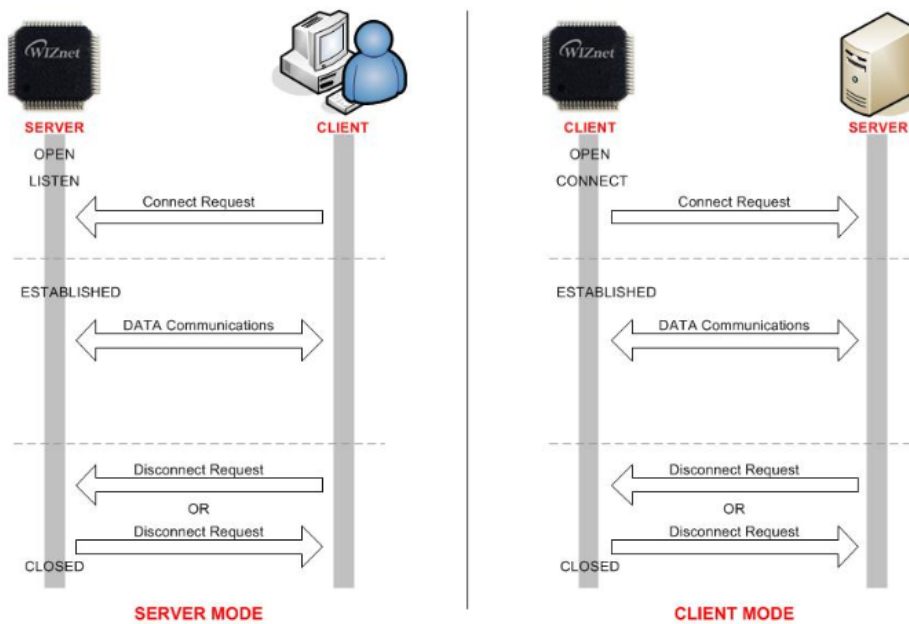


Figure 99: Sequence diagram of a TCP socket initiation with W5500 [137]

A great feature about the W5500 is that the manufacturer, WIZnet, provides the customer with drivers to use with the hardware. These drivers can be viewed as a hardware abstraction layer for development, where the drivers make it easier and more intuitive to interact with the hardware registers.

In addition to supporting TCP, which is a great protocol if you need a reliable transmission through packet acknowledgment, it also supports a variety of other protocols [41]. It also supports ICMP and ARP. ICMP can be very helpful when trying to troubleshoot network devices, and to check connectivity between devices on the network layer such as the ping command, or tracert. ARP on the other hand is very beneficial for troubleshooting on the data link layer in Ethernet (802.3) networks. On the datalink layer devices operate with physical addresses (MAC addresses) as source and destination. Both protocols makes for good tools in troubleshooting throughout the system life cycle.

The W5500 also offers flexibility by providing an interface for changing the communication modes through software after the IC has been installed. Changing the communication modes during operation is done by setting a 3 bit array in a predefined pattern according to the datasheet. Some of the modes are for instance to enable full duplex communication, half duplex or power down mode. More information on the modes can be found in [137].

The components needed for the W5500 to function includes passive and electromechanical components. The W5500 signal transformer is used to isolate the communication signals from the rest of the circuit. The transformer ratio is therefore a 1:1 transformer since we don't want to attenuate or amplify the signals. The signal transformer has an builtin impedance matching network [57]. The W5500 does not need a logic level shifter because its I/O pins are 5V tolerant [137]. The electromechanical oscillator was selected to full fill EL-010.

### 5.3.2.3 Wi-Fi

The system might be used without a wired network connection in the cases of the absence of a network provided by a tactical radio. In this case a wireless connection that supports the TCP/IP stack is desirable. According to EL-004 we need support for 802.11. 802.11 (commonly referred to as Wi-Fi if certified by WiFi Alliance) is a member of the IEEE 802 family, which is a series of specifications for local area network (LAN) technologies [47]. The IEEE has different 802.11 standards that offers different characteristics and are built and used differently. We wish to use a Integrated Circuit (IC) or module that can be integrated into our receiver PCB. There are many characteristics to consider when selecting a WiFi subsystem. Power consumption, range, bit rate, compatibility, communication interfaces and more.

#### **ATWINC1500 module**

The ATWINC1500 module was chosen because it's proven to be integrated with the ATmega2560 microcontroller. The module integrates Power Amplifier (PA), Low-Noise Amplifier (LNA), Switch, Power Management, and a printed antenna or a micro co-ax (u.FL) connector for an external antenna resulting in a small form factor (21.7 x 14.7 x 2.1 mm) design [13]. Easy integration is a contributing factor in developing a functioning system in the least amount of time as possible. The ATWINC1500 has a proven track record and is used in commercial products such as the Arduino MKR series. This also enables the developers of the system to use rebuilt Arduino libraries in a way that will speed up the development. A rich library of embedded drivers with good documentation is crucial in minimizing the time to start working with the hardware. Documentation needed to get started using the module can be found on here [14].

Since we are managing the interfaces of each component carefully, the less interfaces the better. The reason why we selected the module version, instead of the IC version was because it included the necessary peripherals that otherwise would add a greater complexity to our design. The material density is somewhat higher when using a module, but it is still the best alternative in this project. The module uses a SPI communication protocol together with some general purpose input output (GPIO) pins.

A logic level shifter (LLS) is an electrical component that converts one or more electrical signals to another voltage logic level. It is used to integrate systems that use different voltage logic levels, for instance CMOS or TTL. Since the ATMEGA 2560 microcontroller operates with 5V I/O logic level and the ATWINC1500 module operates at 3.3V a LLS is needed to comply with EL-007 and EL-008. In our case the signals that needs to be shifted is communication signals and some general purpose signals. Since communication signals tend to have high frequency components because of the nature of the signal (transmit a bit stream as fast as possible) the LLS needs to be fast enough to avoid aliasing (by considering the Nyquist sampling theorem) of the signals. This frequency constraint adds complexity to selecting the correct LLS, because one needs to make sure the output capacitance of both sides of the LLS is within the compliance of the LLS.

In Figure 100 you can see the illustration of an abstract LLS used in the the Spark electrical system. The LLS is a auto sensing bidirectional LLS. This means that the signals can be transmitted both ways in one channel without an additional signal to control the direction of the signal. This is very beneficial because it allows for easy integration since all signals can use one LLS.

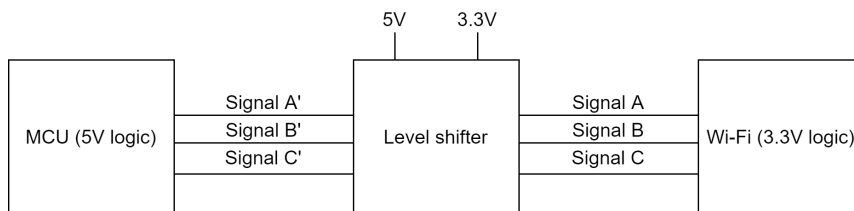


Figure 100: LLS between MCU and Wi-Fi subsystem

To get a better understanding of how a simple LLS can be constructed let's analyze the LLS circuit in Figure 101

We identify the transistor as an N-channel Enhancement MOSFET transistor. Let's say this circuit in particular represents the Master-In-Slave-Out(MISO) transmission line for the SPI inteface between the ATmega2560 and ATWINC1500. The 3.3V transmit side, namely marked 3.3V-TX is the transmitting side and the 5V-TX receiving side is the ATmega2560. Now if the 3.3V-TX side is driven low that would result in the gate-source voltage of the transistor to be greater than 0V, thus closing the transistor. Since the transistor closes we get a voltage divider and it is represented mathematically in Equation 7

$$V_{5V-TX} = 5V \cdot \frac{R_{DS(on)}}{R_{DS(on)} + R2} \approx 5V \tag{7}$$



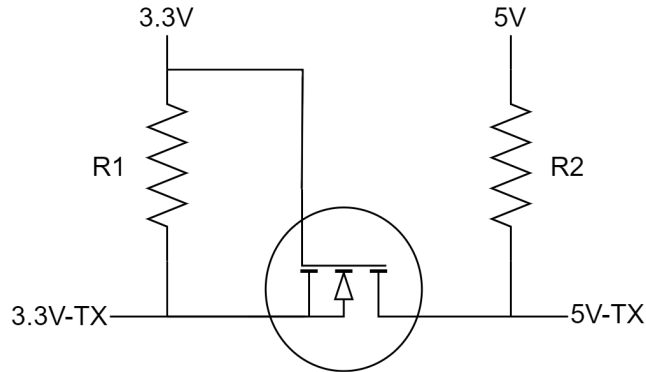


Figure 101: Transistor based bidirectional LLS

Since  $R_{DS(on)}$  which is the internal resistance of the transistor when it is conducting is much smaller than  $R2$  it can be approximated to  $5V$ . The result is that both sides are driven to ground potential. In the case of the MCU drives  $5V-TX$  low the drain of the transistor will be at a lower potential than the source, and thus it will start conducting. Therefore the  $3.3V-TX$  side will be at  $0.6V \approx 0V$  because of the  $0.6$  voltage drop over internal structure from source to drain and the left side will be low.

TXB0108 was selected because it can easily be integrated as an standalone IC with two independent voltage sources between the ATmega2560 and ATWINC1500 system. It was also a possibility to use the standalone transistor setup that was explained in the previous section, but since the requirement for a LLS emerged late in the development process we need to make a smart decision. If we were to use E-MOSFET's like explained above, we would need many passive components as well because there are 8 channels between ATmega2560 and ATWINC1500. It would also take up more space on the PCB that we were trying to minimize. The late change imposed a substantial technical risk and time problem to integrate the LLS into the rest of the design. By choosing the TXB0108 we eliminated a lot of uncertainty because we can trust its specifications, and one chip can handle up two eight individual channels. We researched the logic threshold levels to make sure they would comply with EL-007 and EL-008. The thresholds specifications can be found in [13, 12]. The idea is that for a device to read a high or low, it needs to define what range of input voltages actually is considered as high or low. In the datasheet this information is usually expressed in DC characteristics.

### 5.3.3 Plasma Igniter System

#### 5.3.3.1 350V Step Up

The functionality of the plasma igniter system is primarily to accumulate and deliver the energy needed to initiate two plasma igniters simultaneously. To do so, the system will store the energy in capacitors which will be charged by a step up circuit. Now, from Equation 3, we know that the two capacitors will be  $150\mu F$  in size and that they will hold a little over 9J each when there is 350V applied to them.

The battery voltage will vary according to temperature, it can be as low as 1.3V at minus 20 degrees or a new battery can be up to 3.8V at room temperature. To be able to counter this we have chosen to split the 350V step up circuit in two, where the first part will handle the voltage change of the battery and give a little boost while the second part handles the big boost.

The first step up circuit must be able to have a varying input voltage and at the same time deliver a steady output voltage. The range of the input voltage must be less than 1.3V and higher than 3.8V to handle the voltage change in the battery according to temperature. The battery voltages to temperature change is derived from [38] and [66] where they draw up to 1.8A in pulses to test the battery's durability. The step up circuit will be design to only draw around 1A to prolong the operation time of the receiver.

The output voltage of the first step up circuit should be close to the optimal input voltage of the second step up circuit when it comes to efficiency. The efficiency of a typical step up circuit is around 80 to 90% when given the optimal input voltage and its allowed to deliver the optimal current. So we start by finding the 350V IC and then we find the first step up IC with the sufficient voltage input range.

The criteria the 350V step up must fulfill is:

Component Criteria	
output voltage	350V
charge time	$\leq 10$ seconds
high efficiency	$\geq 80\%$
draw current	$\leq 1A$
size	small
external components	few

Table 10: Component criteria for 350V step up

The LT3420-1 IC from Analog Devices fit this criteria and was chosen, many other ICs were considered, but the LT3420-1 was the smallest and had the least amount of external components. For the LT3420-1 to be the most efficient, the input voltage need to be 4.5V or more.

The first step up IC has the criteria:

Component Criteria	
output voltage	5V
minimum voltage input	$\leq 1.3V$
max voltage input	$\geq 3.8V$
high efficiency	$\geq 80\%$
draw current	$\leq 1.2A$
size	small
external components	few

Table 11: Component criteria for input step up

The LT1308B, also from Analog Devices, fits to be the first step up circuit. Since both step up circuits is from Analog Devices we can run LTspice simulations and analyse their behavior. In Figure 102 we can see a simulation of the two devices charging up a  $5\mu F$  capacitor, this takes about 180ms. The reason we don't use a  $150\mu F$  capacitor in the simulation is that we don't have a computer that can handle a big simulation like this over the time needed to reach 350V on a  $150\mu F$  capacitor. So we calculate that  $\frac{150\mu F}{5\mu F} = 30$  and multiplying with the time used to charge the  $5\mu F$  capacitor. This gives us  $30 \cdot 180ms = 5.4s$  to charge one  $150\mu F$  capacitor.

The theoretical total charge time for both capacitor will then be  $2 \cdot 5.4s = 10.8s$ . The two 350V step ups have each a pin that will be set to HIGH when the ICs are done charging its respective capacitor. This gives software the feedback needed to know when to start charging the next capacitor. With this configuration, the system will be able to charge the capacitors. The charge time is within the expected range. The ICs also have the feature of indicating when they are finished charging the capacitors, so there is no need for extra components to do so, which is a bonus. This is our work towards fulfilling SR-016.

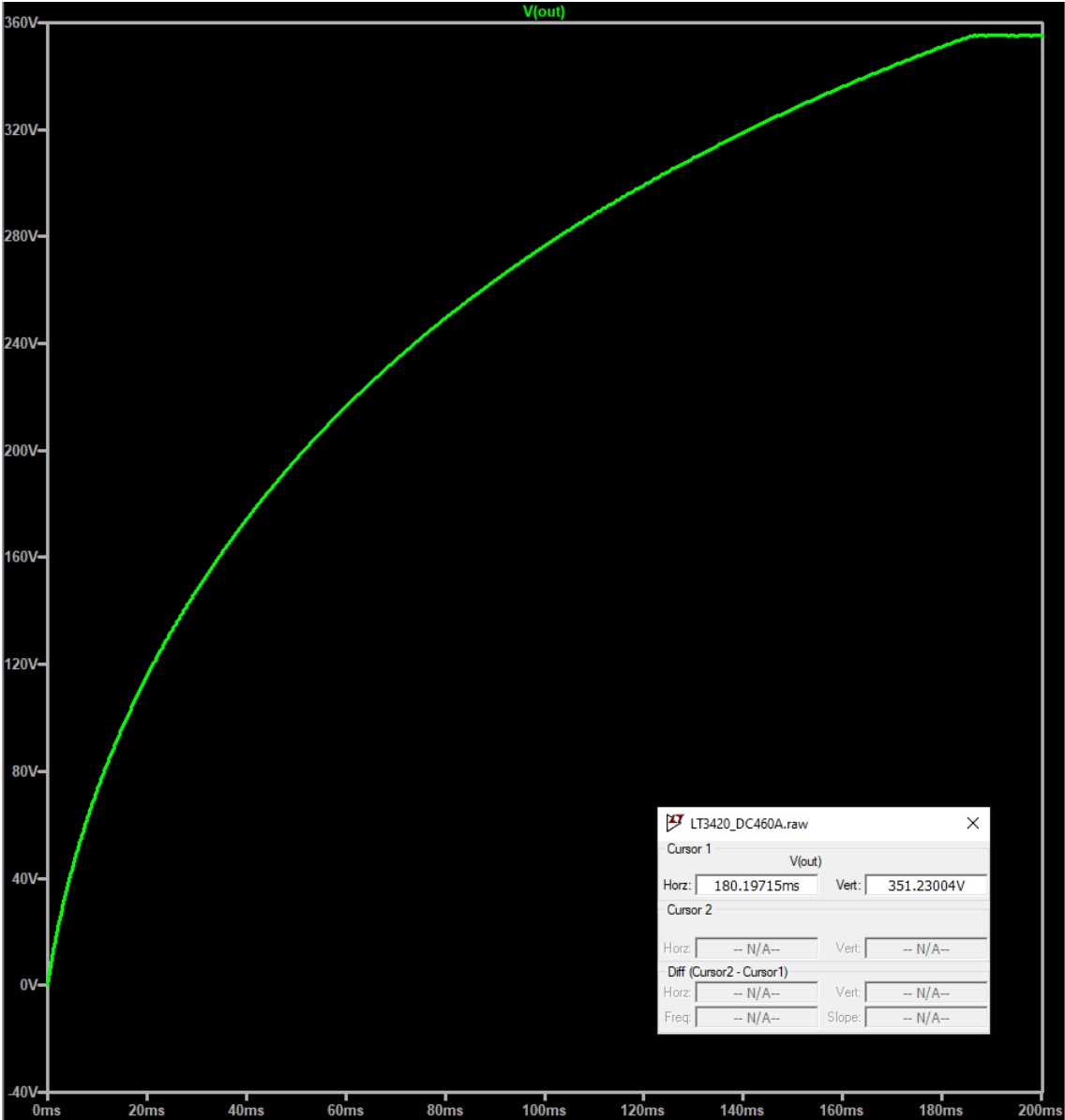


Figure 102: LTspice simulations with  $5\mu F$

### 5.3.3.2 355V Capacitive Discharge Circuit

The discharge circuit has a capacitive discharge characteristic because it is effectively a RC circuit. In Figure 103 you can see the discharge circuit schematic. There are three main focuses that need attention for the circuit to comply with the requirements. Safety thinking, performance and physical size considerations. As opposed to the electrical design in section 4, where we created a generic solution using abstraction, in this design phase we will go down to component level. On the generic level we only used switches in a general manner, assuming such switches with the desired characteristics existed.

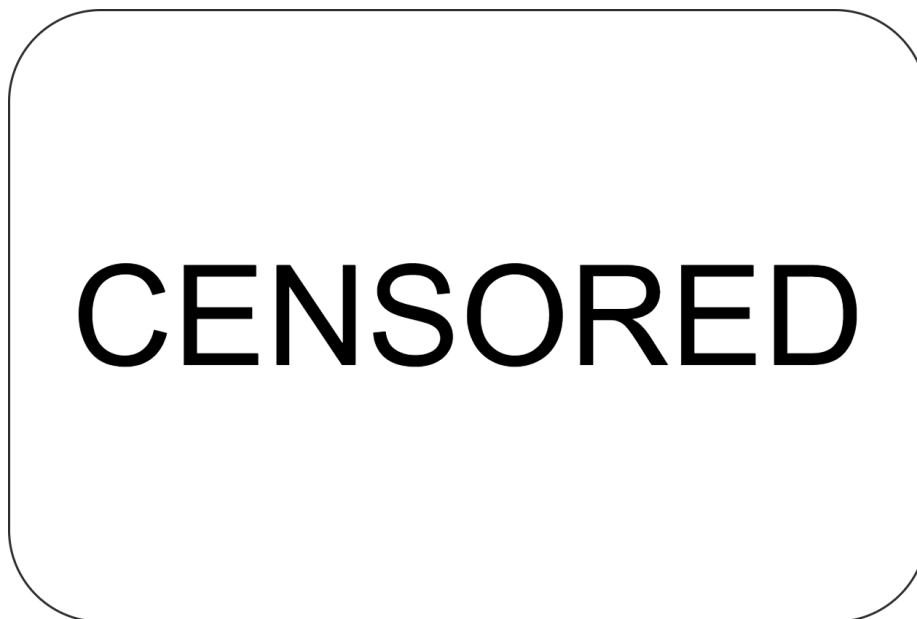


Figure 103: Discharge circuit that is implemented in the SPARK PCB

The circuit is composed of passive and active components. Resistors are used to limit the current and power dissipation in the active components, namely D-MOSFET and silicon controlled rectifier (SCR). The D-MOSFET is used to operate as a normal, closed switch. The reason for choosing a D-MOSFET is because of its high voltage capability between drain and source [62]. In contrary to the relays we researched that can handle high current, but not high voltages without being too physically big in comparison. It also offers an easy way of controlling it using a relatively small gate-source voltage. The limiting factor for the D-MOSFET in this design is the fact that it is not many commercially available for high currents. Furthermore, the D-MOSFET does conduct from source to drain (with a small voltage drop) which is undesired because of the switch self check mechanism, but that will be discussed in a later section. The D-MOSFET is therefore only used together with the 20k $\Omega$  discharge resistor, which gives a current that is well below maximum drain current, but also offers an acceptable discharge time. To be able to make a self check functionality we needed something that would act as a high voltage switch with high reverse voltage

capabilities.

5. IMPLEMENTATION

The SCR behaves like a high voltage switch that is normally open, but it can also handle high reverse voltage. It has a different structure than the D-MOSFET and can be viewed as a PNP junction composition [62]. Therefore, we can apply the same logic as we would to a PNP and NPN bipolar transistor coupled together. Actually, since the SCR is a current controlled device just like the bipolar transistor, it is much easier to control with MCU. The ATmega2560 can source up to 20mA per I/O port [12] which is more than enough to trigger the SCR. In comparison with a D-MOSFET, as the drain to source voltage demand increases, so does the gate-source voltage that needs to be applied. For a 400V drain-source D-MOSFET a typical gate-source voltage is around -10 volts. Furthermore, to drive a D-MOSFET into the open state you would need a negative gate-source voltage, thus increasing the design cost with another step up circuit. More information about the D-MOSFET and SCR can be found in [62].

Safety is a very high priority for the Spark system and consequently the discharge circuit must facilitate safety measures as well as other parts of the system. When we designed the safety measures, we have based it on multiple redundant safety checks and safety vs availability trade offs. It takes two or more independent signals/functions to initiate a critical action. If the correct combination of signals/functions is not present the system will not deliver energy to the plasma igniters. The circuit also facilitates a design such that the MCU can do a safety check of the switches before charging the capacitor. This will make sure that the energy will not be transferred into the discharge capacitors before the system has verified that the switches that control the energy flow is working properly. A system must also be reliable to be safe. If the system is developed with too much complexity, the safety measures of the system may not function in some cases. The consequences of a non-functional system can be enormous for the operator.

In Table 12 you can see the different system states in regards to the discharge circuit and what kind of input signals that needs to be present for each state.

System states	D-MOSFET	SCR_NO_EN	SCR_TEST_NO1	SCR_TEST_NC1	Watchdog	MCU	355V subsystem
Power on	Closed	Open	Open	Open	Enabled	Enabled	Disabled
Power off	Closed	Open	Open	Open	Disabled	Disabled	Disabled
Armed	Open	Open	Open	Open	Enabled	Enabled	Enabled
MCU in unexpected state	Closed	Open	Open	Open	Disabled	Enabled	Disabled
Fired	Open	Closed	Open	Open	Enabled	Enabled	Enabled
Self check NC	Open & closed	Open	Open	Closed	Enabled	Enabled	Disabled
Self check NO	Open	Open & closed	Closed	Open	Enabled	Enabled	Disabled

Table 12: System states and its corresponding signals

Self check for the normally closed and normally open discharge switches is important and is designed to comply with requirement EL-014. The idea behind the requirement is that the capacitor will not be charged from the battery before the system has verified during operation that the switches work. To examine this function let's analyze Figure 103. The self check works in steps described below.

1. Enable a high voltage level from MCU on V\_TEST\_IN1 signal
2. Close the respective pair of SCR's you want to test
3. Measure the voltage of V\_TEST\_OUT1 using the built in ADC of the MCU
4. Read the digital value and compare it to a predetermined threshold in software using relational operators
5. Compare the results with what is expected from the components depending on their input signals
6. Act upon the result of the previous steps based on a success or failure

This method is very simple, but is also cost-effective since the ATmega2560 already has an ADC on board. The voltage V\_TEST\_OUT1 will always be  $\frac{V\_TEST\_OUT1}{2}$  or  $\approx 0V$  and this is the way we can determine if the switches works.

Another important measure that was implemented to avoid discharging the capacitor unintentionally is to make sure that the normal operation of the components is a safe state. If the power is abruptly removed all the components will go to their usual operating state. That is why the D-MOSFET and the SCRs are placed in the configuration that they are. Also, it is important to realize the function of the  $1M\Omega$  resistor. Since the leakage current of the SCR is about  $10\mu A$  [125] the  $1M\Omega$  makes sure the voltage over the plasma igniter is as small as possible even when the SCR is open.

In this section there will be the calculations and thoughts behind how the different components and values were chosen. The DN2540N8-G is a surface mount 4 pin D-MOSFET with a (2.6 x 4.25)mm footprint[34]. The D-MOSFET are rated for 400V, 10% above the maximum voltage of 355V. This is because we do not want to push the component outside the recommended operating area. The limiting factor when choosing a D-MOSFET for the SPARK PCB is it's voltage rating, size and current handling capabilities. From practical experience we know that the maximum current rating of a D-MOSFET increases with it's package size because it needs to dissipate more heat. Therefore we want to minimize the current and as much as possible to achieve a small PCB footprint. Ideally the D-MOSFET acts as a open switch when the proper gate-source voltage is applied [62]. The DN2540N8-G has a maximum current rating of 500mA and a maximum of  $BV_{DSX} = 400V$



[34]. Because of requirement CR-019 we need to make sure the characteristics holds over the entire temperature range. In Figure 104 you can see the breakdown voltage of the DN2540N8-G over the required temperature range.

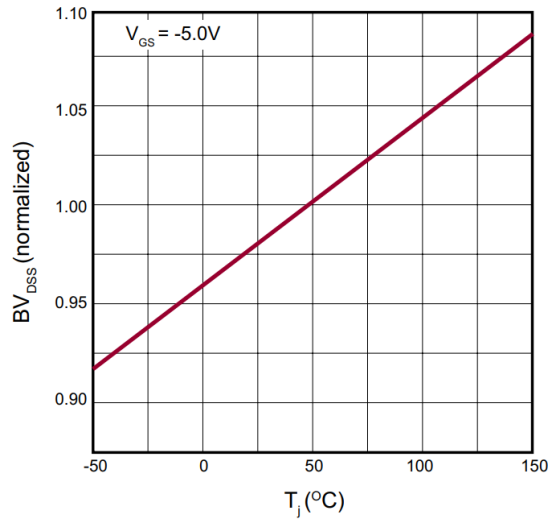


Figure 104: Breakdown voltage variation with temperature

At  $-40^{\circ}\text{C}$  junction temperature we get

$$BV_{DSS} = 0.91 \cdot 400V = 364V \quad (8)$$

and at  $+85^{\circ}\text{C}$  junction temperature we get

$$BV_{DSS} = 1.03 \cdot 400V = 412V \quad (9)$$

At both these temperature extremes we will comply with EL-002 and CR-019.

5. IMPLEMENTATION

The S4N1RP is a surface mount 3 pin SCR thyristor with a (3.95 x 5.60)mm footprint [125]. The SCR are rated for 400V, 10% above the maximum voltage of 355V for the same reason as the D-MOSFET. The limiting factor of S4N1RP is not the current rating, but rather the size and gate current trigger. The gate trigger current of the S4N1RP is 10mA [125]. A relatively high gate trigger current is selected because we don't want the smallest of interference to close the device. Also since the ATmega2560 is capable of sourcing up to 20mA [12] it is a good choice, since the gate trigger current varies with temperature. In Figure 105 you can see the gate trigger current variation with temperature

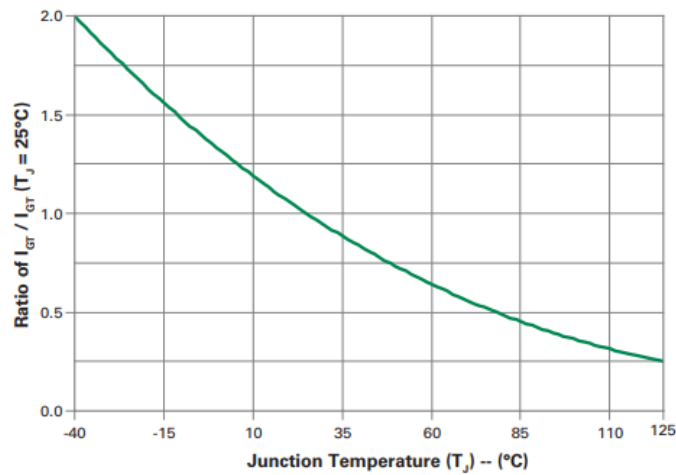


Figure 105: Gate trigger current variation with temperature[125]

At -40°C junction temperature we get

$$I_{GT} = 2.0 \cdot 10mA = 20mA \tag{10}$$

At +85°C junction temperature we get

$$I_{GT} = 0.45 \cdot 10mA = 4.5mA \tag{11}$$

At both temperature extremes the ATmega2560 current sourcing capability is within the operating range and it complies with EL-002 and CR-019

When selecting resistors for such a high voltage design there is at least four main considerations to take into account; resistance value, voltage rating, power and size. Since the 355V source is a capacitive source we will get very high peak values, which can be misleading when selecting the appropriate power rating. The maximum voltage rating is 400V and is very important [123]. The  $20k\Omega$  resistor value is a product of the fact that we need to limit the current though it to less than  $I_{D(Pulsed)}$  of the D-MOSFET. In Figure 106 you can see the simulation of the discharge through the  $20k\Omega$  with the D-MOSFET characteristic on-resistance.

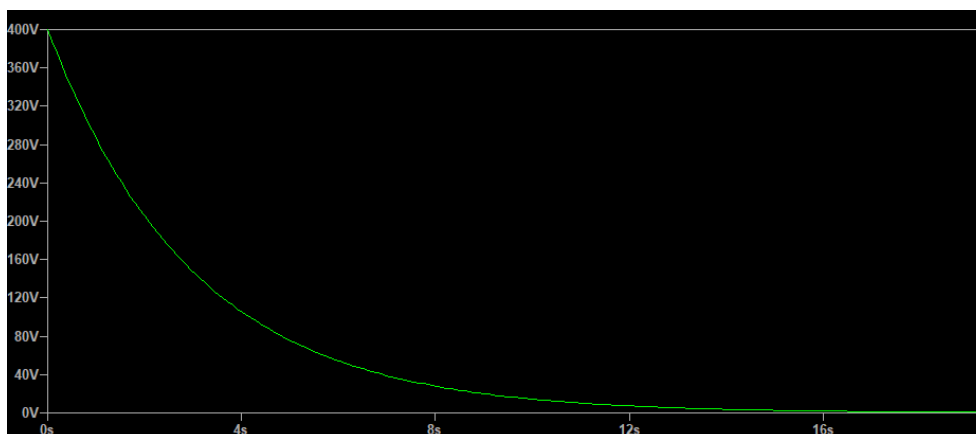


Figure 106: Discharge curve with  $20k\Omega$  in series with  $R_{DS(ON)} = 17\Omega$

As you can see the capacitor is discharged from 400V, which is above the absolute requirement, down to approximately 0 V in 16s. This is well within the EL-014. The peak power for the resistor can be calculated as follows:

$$P_{20k\Omega} = \frac{(400V)^2}{20k\Omega} = 8W \quad (12)$$

Comparing this to the Single Pulse Power rating in [123] reveals that it is compliant. In most, if not all situations the voltage would not be any more than 355V, therefore the power would be 6.3W maximum, but precaution must be taken.

5. IMPLEMENTATION

The three 133Ω resistors are used to trigger the SCRs. We know that the input of the SCR is a gate-source PN junction with a voltage drop equivalent of a NPN bipolar transistor [62]. The gate voltage of the PN junction drop also varies with temperature and can be seen in Figure 107. By knowing the trigger current and gate voltage over the range of temperatures in Figure 105 and Figure 107 we can calculate the resistor value. Since the ATmega2560 is driving the trigger current, we know from [12] the minimum output voltage. In Equation 13 you can see the calculation of the resistor value using the worst case scenario values to comply with EL-002 and CR-019.

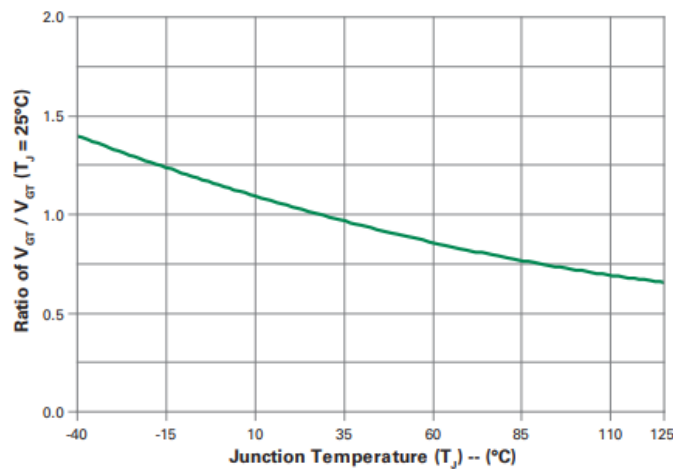


Figure 107: Gate trigger voltage variation with temperature[125]

$$R_{133} = \frac{V_{O(ATmega2560MIN)} - 1.5V}{20mA} = \frac{4.2V - 1.5}{20mA} = 135\Omega \tag{13}$$

The closest real resistor value to be found was 133Ω, and for this reason it was chosen. The power rating of these resistors are of no concern because of the low current from the ATmega2560.

The remaining resistors are calculated more or less on the same principles as discussed previously. The 200Ω has a higher power rating than the others, the 1MΩ has a lower power rating because of the low leakage current through the SCR. 100Ω is just to form a voltage divider and it also has a lower power rating because of the low current from the ATmega2560.

### 5.3.4 Sensors

#### 5.3.4.1 Accelerometer

The purpose of the accelerometer for our system is to detect movement after the receiver is deployed. For example, let's say the receiver is attached to a vertical surface and the operator has relocated to a safe location without line of sight to the receiver. If the receiver dislodges and falls down the operator has no way of knowing whether the receiver is still functional. With the accelerometer implemented, the operator will get a warning that the receiver has moved since it was deployed.

To cover this functionality we have implemented a 3-axis accelerometer that has a measurement range of  $\pm 3.6g$  [32]. This is a general purpose accelerometer, not something you would use in high performing applications, but it will be more than good enough to determine if the receiver has moved or not. It is also shockproof for up to 10000g [32] which should make it robust enough to handle the demanding circumstances our system has to endure. This is a low priority subsystem for our customer so there is not much time spent in its development or design.

### 5.3.4.2 GPS

The GPS module in our system will provide the location of the deployed receiver. This will allow the operator to monitor the receiver's whereabouts and see if someone has moved it from its original location. It also makes it easier to retrieve the receivers after the operation is concluded, assuming the receiver is still intact.

There are two fully functional GPS satellite systems, the russian GLONASS system and the NAVSTAR owned by the U.S. but there is more coming, as both china and the EU is working on finalizing their own equivalent systems. China is working on the BEIDOU system and the EU is developing the Galileo GPS system.

All the mentioned systems will have, or has, a global coverage; how much they differ in performance is still unclear. We plan to implement a GNSS module in our system, this is a module that is capable of receiving and handling data from all the previous mentioned systems and will provide our customer with more flexibility when it comes to further development.

For the GPS system to calculate the location of the GPS receiver it needs to receive a signal from 4 different satellites, 3 of them are used to calculate the location, and the fourth is used as a clock signal as every GPS satellite has a high precision atomic clock. The GPS receiver can then use the time delay to calculate its position.

How long it takes for a GPS to get a connection and a location depends on several factors, most of them from the surrounding environment. The ones we want to focus on is the data needed for the GPS receiver to determine its location. When starting up the GPS has to check if it has valid Ephemeris and Almanac packages [72] for the area it is and if not, it has to download the packages from the satellite network.

If the GPS has a cold start it must download the Almanac package from the satellites. Each satellite sends a signal every 30 seconds and part of that signal holds a fragment of the Almanac package. The download normally takes about 15 minutes with normal reception [50], but can take longer depending on the number of satellites in range.

In a warm start-up the GPS has a valid Almanac package but must update the Ephemeris package. This normally takes a few minutes but is also dependent on the number of satellites and can take longer.

## 5. IMPLEMENTATION

---

The last is the hot start and has a typical time of a few seconds before the first fix, after which the system holds valid Ephemeris and Almanac packages, and only needs to calculate its position.

When the GPS system is turned on for the first time it will be in a cold start situation where it must update all of its data. This will also be necessary when the GPS system has been off for a long time or when the system has been moved far from the last active location, like traveling to another part of the world.

The warm start is common when the GPS system has been off for a few days or you traveled to another part of the country. A hot start is when the GPS system has been off for a few hours and you are close to the same location where the GPS was last active.

The system we are implementing is not bound to be used in just one country or one location, so some sort of routine or protocol around when to start up and let the GPS system update will be necessary. Especially for operations where the location of the receiver is mandatory. Maybe towards SR-035.

The active antenna for the GPS has been chosen as a result of the bandwidth and the built-in filtering capabilities, as this saves us some time in development because we don't need to design any filter circuitry, which in turn means we get less components that take up space on the PCB. This is a low priority subsystem for our customer and therefore little time is used on the design and component selection.

The GNSS module will be able to cover all the functionalities needed for our system. It will also be a building platform for further development and testing as it is capable of utilizing the two existing systems and the once that are on their way. The size of the component and the few external components needed to implement this IC was two major contributors for why this IC was chosen.

### 5.3.4.3 Battery capacity estimation

To fulfill SR-049 we must be able to estimate battery condition during runtime. We use the built in Analog to digital converter (ADC) of the ATmega2560. By using the ADC together with an transistor controlled voltage divider we can measure the voltage of the battery. By knowing the voltage of the battery it is possible to estimate the battery condition, and estimate when it will need to be replaced. The battery discharge curve can be found in Figure 60. We observe that the flat regions is where the battery will have a fairly high capacity. The battery voltage is strongly correlated with the temperature of the battery as can be seen in Figure 60. Therefore the design integrates a temperature sensor as described in EL-006. The battery capacity can therefore be calculated in software using voltage and temperature data. A simple switch case function could be used to correlate different predefined temperature ranges with the output voltage, thus estimating the capacity.

The STLM20 temperature sensor is an ultra-low current 2.4 V precision analog temperature sensor for low current where maximizing battery life is important. It can be ordered as an IC and only uses 4 pins. The output voltage is ideally proportional with the ambient temperature of the device, and the parabolic transfer function can be expressed as:

$$V_O = (-3.88 \cdot 10^{-6} \cdot T^2) + (-1.15 \cdot 10^{-2} \cdot T) + 1.8639 \quad (14)$$

Solving for T in Equation 14 yields

$$T = -1481.96 + \sqrt{2.1962 \cdot 10^6 + \frac{1.8639 - V_O}{3.88 \cdot 10^{-6}}} \quad (15)$$

By using Equation 15 we can estimate the temperature of the device with an accuracy of  $\pm 2.5$  °C[132]. The STLM20 can handle captive loads up to 300pF. Since the ATMega2560 ADC has an input capacitance of 10pF the two devices are compatible.



In Figure 108 you can see the schematic of the battery measurement circuit with off-page connectors to the ATmega2560 and power supply. Here, we are using an enhancement MOSFET, which is normally open. This is to ensure power consumption is minimized as much as possible. The ATmega2560 needs to activate the E-MOSFET to take a measurement, and only then will there be a small current. The leakage current through the transistor in series with 499k  $\Omega$  resistor is negligible. A future improvement of this design is to add additional components to protect against reverse polarity, since this design does not take that into account.



Figure 108: Schematic for battery measurement for SPARK PCB

## 5. IMPLEMENTATION

### 5.3.4.4 Watchdog

The demand of the implementation of a watchdog in the system was given to us 16.05.2020 , where the watchdog shall serve as an extra safety barrier that ensures that nothing goes wrong in the event that the main controller fails. The main concern was aimed at situations where the system was connected to explosives, in which case a failure would be catastrophic.

The watchdog will be implemented between the main controller and the circuitry that controls and executes the ARM, DISARM and DETONATE functionality of the electrical system shown in Figure 109.

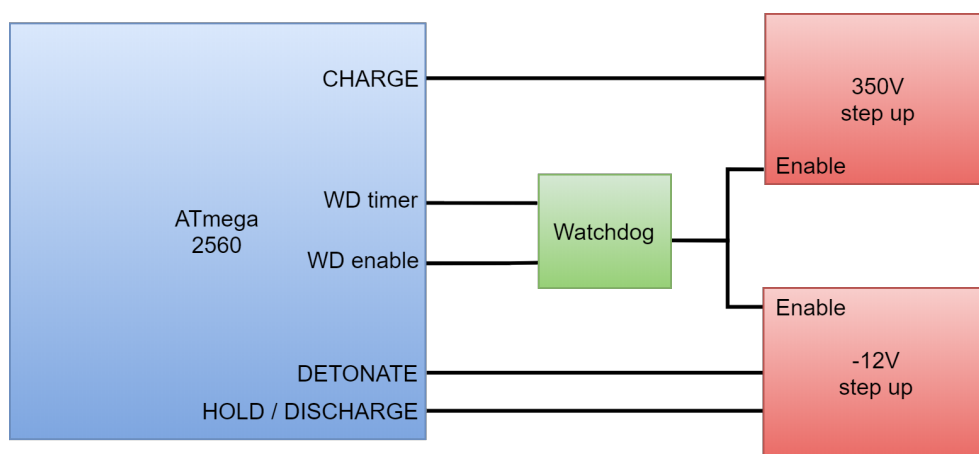


Figure 109: Watchdog Implementation

The watchdog is powered by the ATmega 2560 through the WD enable pin and the timer is reset with a rising edge on the WD timer pin. The duration of the timer is 102ms [133] and when the timer runs out the output of the watchdog goes low which in turn disables the 350V and -12V step up circuits. With the 350V step up disabled, the system cannot charge up the capacitors and hit the -12V, and with the step up system disabled the system cannot send the energy to the plasma igniters. The circuitry that uses the -12V step up as supply is by default in discharge mode 5.3.3.2.

With this configuration we minimize the chance of an accidental detonation even further. If the MCU fails there is no telling which state the I/O pins take, but with the watchdog there to disable the most critical subsystems, the MCU would have a hard time energizing the plasma igniters.

### 5.3.5 Power Supply

Our system has many components with varying needs, one of them is the supply voltage. We started with setting a standard of 3.3V early on and tried to find components that met this criteria. We change of 5V to 3.3V because it would serve as a more conservative power consumption than what was proposed in 4.3.1. 3.3V is a typical voltage supply and most components are designed to be driven by this voltage. However, our system is powered with a battery, so the voltage will depend on the state of the battery.

To get a steady supply to our components we will use step up circuits to handle the different states of the battery. The battery voltage depends on the energy still stored in the battery, the temperature of the battery and how much power the system draws from the battery. The step up circuits must have a range in the input voltage of less than 1.3V and over 3.8V as mentioned in paragraph 5.3.3.1

To pick the right 3.3V step up circuit we must know how much current it must be able to deliver and at the same time are we interested in a high efficiency. The efficiency is measured in % and is calculated from the power of the input over the power of the output. The efficiency for most step up circuits have a peak at a certain current at the output, but since the 3.3V step up must be able to deliver current over a range. This range depends on how many sub systems that are active at the same time. So we must choose a step up circuit that has its best efficiency somewhere in the middle of the current range.

Current Range		
Subsystem	Active (max)	Stanby
WiFi	290mA	4 $\mu$ A
Ethernet	132mA	13mA
NFC	180mA	14 $\mu$ A
GPS	30mA	0.1 $\mu$ AA
Accelerometer	300 $\mu$ A	0.1 $\mu$ A
Temp. sensor	8 $\mu$ A	0.02 $\mu$ A
Total	632mA	13.1mA

Table 13: Current Range derived from components data-sheets

As we can see from Table 13, the voltage range can vary from 13.1mA to 632mA. In practise all of these subsystems will not just be on or off, but a combination of the two. The three most demanding subsystems are the communication channels; the NFC will only be active before the receiver is paired with the phone. After that the communication will go

through WiFi or Ethernet and then only one of them will be active.

The GPS and accelerometer will only be active when the receiver is deployed and the temperature sensor will only be active in short periods when the the battery measurements are taken. So the current range for 3.3v step up need to be efficient between 334mA and 133mA, as this is the highest and lowest configuration of subsystems currents in practice. So, when we looked for different step up circuits we used the criteria in Table 14 to pick one.

Component Criteria	
input voltage min	$1.3V \geq V_{IN}$
input voltage max	$V_{IN} \geq 3.8V$
output voltage	3.3V
output current	$\geq 350mA$
current range efficiency	$\geq 80\%$
size	small
external components	few

Table 14: Component criteria for 3.3V supply

The LTC3539 from Analog Devices fulfilled all of the criteria shown in Table 14 and from the data-sheet we pick out the graph in Figure 110 to show that the efficiency is over 80% for most of the input voltages, but that the IC struggles to keep up the efficiency when the input voltage goes below 1.8V. That being said, we consider this acceptable. The same process was used to pick out the 5V and -12V step up circuits. This is our work towards fulfilling SR-021.

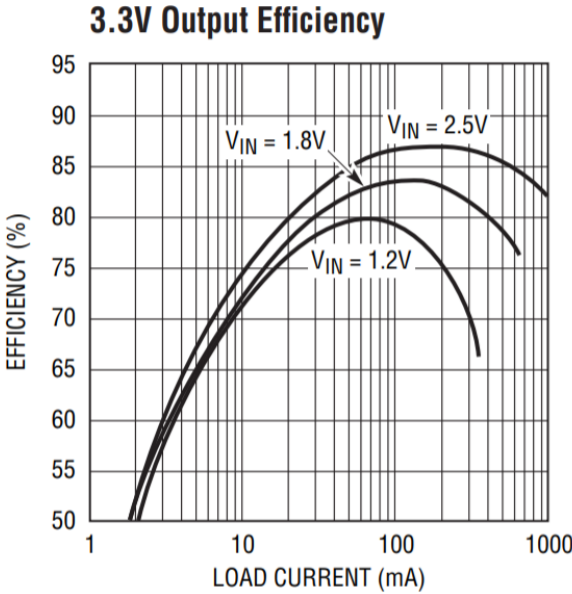


Figure 110: Component Placement

### 5.3.6 PCB Design

The PCB design is one of two big factors that impacts the overall size of the receiver unit. The other factor is the component size and quantity. This is the main interface between the mechanical and electrical design.

Our customer wants the receiver to be as small as possible, but no larger than ME-006. The system's three largest components in volume is the CR-123 battery and the  $150\mu F$  capacitors. If we choose the underside of the PCB to hold the larger components and use these three as a max height measurement for other components, then this will be the best solution when it comes to size. To reduce the size as much as possible, we then place the smaller components on the top side of the PCB.

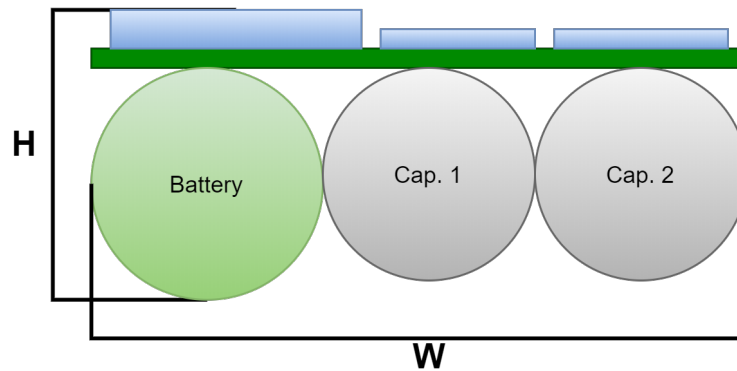


Figure 111: Component Placement

In Figure 111 we can see that the smallest width possible of the PCB is determined by the battery and the capacitors. The minimum width of the PCB then amounts to

$$D_B + 2D_C$$

From [40] and [124] we can derive  $D_B = 17mm$  and  $D_C = 16mm$ , which gives us the minimum PCB width of

$$17mm + 2 \cdot 16mm = 49mm$$

The minimum height is harder to find, but we know that  $D_B = 17mm$  and we can see in Figure 111 that the battery is the biggest contributor to the height. Therefore, if we use an estimation of 5mm height for the upper components, and that the PCB board builds around 3mm we get a estimated lowest height of 25mm.

5. IMPLEMENTATION

Now for the length of the board, we have to look at 3 other components. The antennas for the GPS, NFC and WiFi. The typical dimensions for these type of antennas that we need is 25x25mm or 35x35mm. After some research we found the 25x25mm antennas to be sufficient for the WiFi and the NFC, but the GPS needed the bigger 35x35mm antenna.

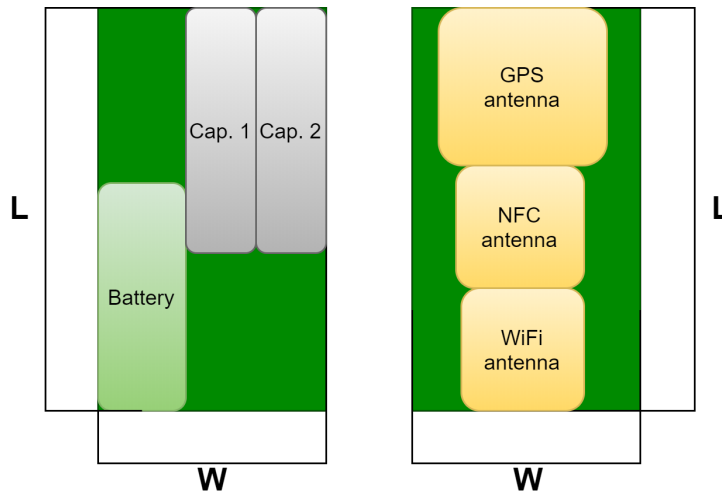


Figure 112: Component Placement

In Figure 112 we can see that the size of the antennas will determine the minimum length of the PCB board  $35mm + 2 \cdot 25mm = 85mm$ . After some discussion with mechanical we agreed to aim for a PCB + component size of 90 x 50 x 30mm as L x W x H. According to how many subsystems we are going to have on this PCB the space available is sufficient but it will be a compact design.

To reduce the noise or EMI that the sensitive parts (like the communication systems) are exposed to, we will place the 350V step up circuit and the discharge circuit on the bottom side of the PCB. Then the PCB will work as a shield and isolate most of the sensitive (or less noisy) systems on top. The only exception is RJ45 port because of its height.

In Figure 113 we can see an overview of the subsystem placement on the bottom side of the PCB, where we try to keep the 350V step up circuit and the discharge circuit at the other side of the board. The orange systems in the figures 113 and 114 are the ones we want to shield from noise, and the red ones generate noise. The white systems are somewhere in the middle of the red and orange.

On the top side of the PCB shown in Figure 113 we see that there is more space between the systems, but there are a lot more signal tracks to handle since all the systems will have signals going to and from the ATmega chip. The communication systems is placed in each

corner of the PCB and the RJ45 plug is through a hole in the receiver casing.

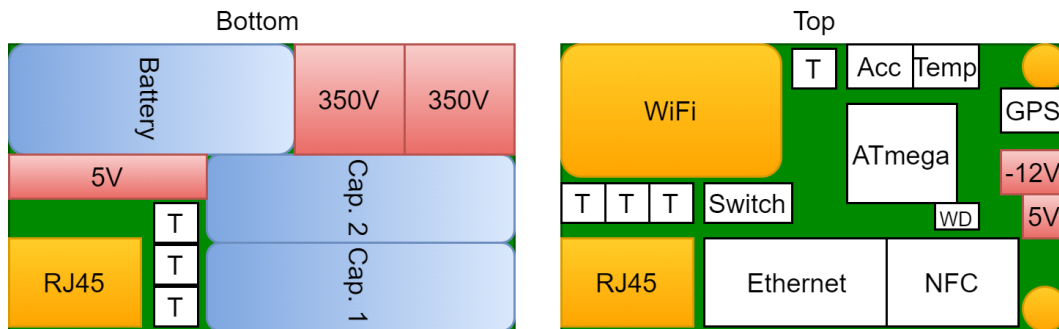


Figure 113: Component Placement

The two main communication channels are placed furthest away from the step up circuit's both from above and from the bottom side. This way we hope to ensure that the connection between the receiver and the Android device is not affected by the noise generated on the board.

There is also some test-pads included on the board that are connected to various control signals and voltage outputs. They can be used to validate the different signals in the testing and analysing phase. It will also make it much easier to troubleshoot if something is not behaving or working as planned. We are not yet sure of how many we will be able to include on the PCB but they are the first thing to go if there is no more space left.

When we selected pins on the ATmega to the different subsystems we used Miro as mentioned in subsection 5.3 to keep track of all the different signals. One thing that greatly influenced how we chose pins aside from the functionality needed, was where to minimize crossings of the signal path. When signals cross over each other on a PCB one of them has to be moved to a different layer and then brought back to the same layer again. This increases the production cost and the complexity of the PCB and is something we want to avoid.

When most of the components and traces on the PCB was done, some unexpected changes unfolded: the first thing was that we had forgotten to add a through hole pin connector and we realised that two of our resistors where too big. The second thing was that the battery compartment had to be bigger then first estimated.



## 5. IMPLEMENTATION

The first thing we did was to find some new resistors. Subsequently, we added the connector and altered the PCB layout so it would fit. Then we figured out how much the new size of the battery compartment would effect the bottom side of the PCB layout. The new battery compartment was taller than before, this allowed for capacitors that were shorter and with a bigger diameter. After some searching we found new capacitors that made it possible to place them as shown in Figure 114. Then, we had to stretch out the tall components of the 350V step up circuits alongside the board outline, and place the rest underneath the capacitors.

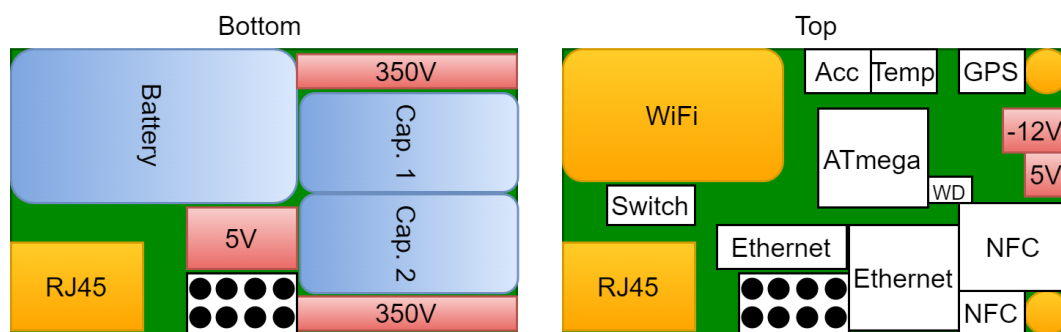


Figure 114: Component Placement

For the top side we had to move the Ethernet and NFC components. To the right in Figure 114 you can see that we also had to spread them out a bit as well. This removed a lot of the space for the test-pad, so we picked out some of the key points we needed and removed the rest. This set the progress of the PCB design back approximately two days.

The PCB design was completed and sent to the customer for validation, it was then discovered that some of the via's that were used on the PCB made it impossible to manufacture. By the time we got this feedback we were already buried deep in the documentation and did not have the time to change the PCB. The plan is to do the changes, but not before the documentation is delivered. So for the end presentation we hope to have a working PCB.

5. IMPLEMENTATION

5.3.7 System integration

When designing a system on component level it's crucial that the specifications of all the components are within range of its operating conditions. The process of selecting the correct components for a design can be time consuming because they often depend on each other.

Managing changes can be very difficult and could result in the change propagating through the entire design without being noticed. In electronics it is very common that one variable or component can influence or radically change the rest of the design, rendering it useless for its purpose. Therefore we decided that we wanted to use a common online whiteboard where we kept the overview of the interfaces between the different electrical subsystems. Managing and monitoring the interfaces while developing systems is crucial [17]. This helped us to easier manage changes, and see how the change impacted different subsystems or components. In Figure 115 you can see our online whiteboard in relation to the component integration with the MCU.

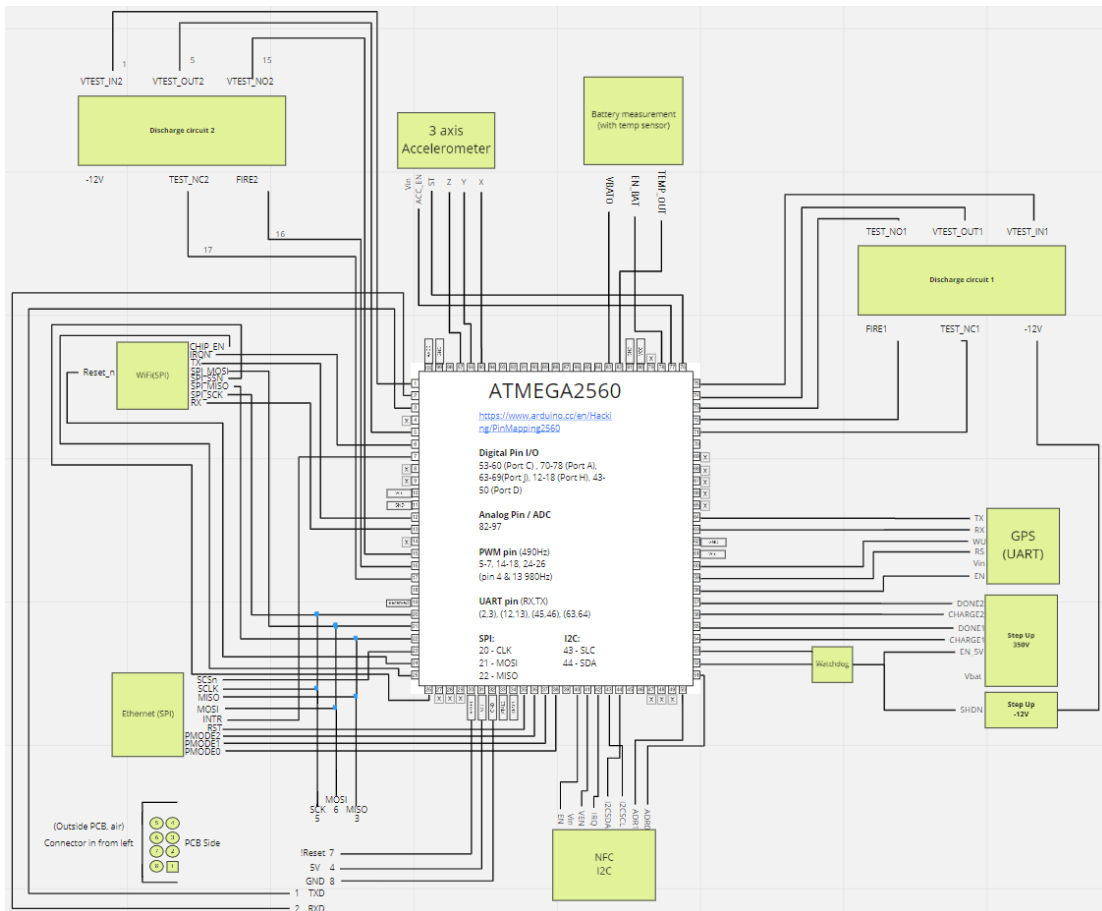


Figure 115: Online whiteboard for system integration

## 5.4 Final Mechanical Implementation

The third official version of the casing has the same volume as the technical requirement 86, but the measurements are different. We have tried to implement as many technical customer- and system requirements as possible to get a, as mentioned earlier, proof-of-concept prototype. A Bill of Materials for both casings can be found in appendix 8.14.

### 5.4.0.1 Nanotube enhanced laminates

This version is made in two casings. Version 3.1 is 3D printed and Version 3.2 is an assembly made from hand-made laminates. Due to the lockdown at the university, these laminates were made in the garage at home, but thanks to a build in heat pump there the temperature was at a constant 20°C. All laminates were post-cured in an oven at 60°C for 4 hours.

Laminates:

635 g quadraxial glass fabric (850g/m<sup>2</sup>)

422 g SvaPox110

101 g TL-1 (curing agent)

Ratio: 24% curing agent to SvaPox110

More specifics on how the laminates were made can be found in appendix 8.12.



Figure 116: Laminate in the making

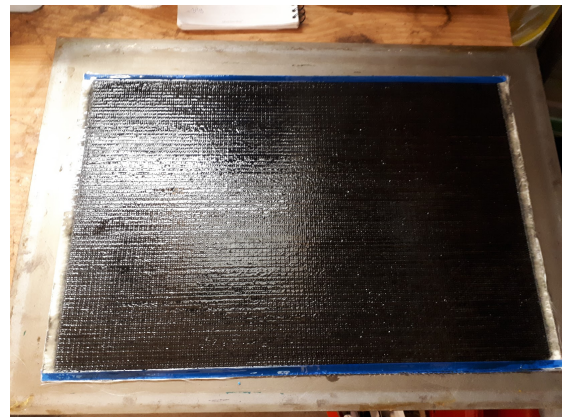


Figure 117: Laminate ready for hardening.

## 5. IMPLEMENTATION

### 5.4.0.2 Casing V3.1, 3D Printed

The 3D designed receiver unit is, as previously mentioned in the Material section [4.2.2.1](#), made of PETG. The design is made to fit 3D printing so to use other production methods from these drawings will be difficult and there will have to be some changes. This casing weighs 125gr without the battery compartment, gaskets and components inside.

To make sure that this version will be able to classify to at least IP65, all interfaces has a custom gasket, Figure 125, so that there will be no leakages between the lids and the casing body and the edges around RJ45, plasma igniters and on/off switch will also be sealed with Araldite AW4858/ HW4858. This will be so for both casings, version 3.1 and 3.2. Another element that will be common to both versions is that the lid is fastened with M3 rivets and countersunk machine screws.

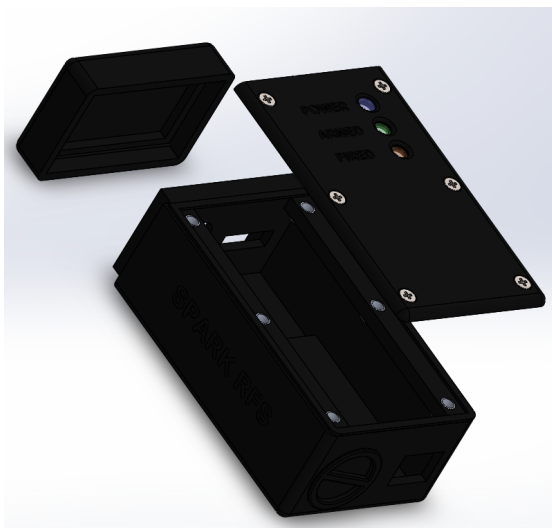


Figure 118: PETG Assembly Open



Figure 119: PETG Assembly Closed

### 5.4.0.3 Casing V3.2, Nano Laminate Assembly

Since the time and knowledge for casting the nanotube casing was too short, we decided to make a piece-by-piece assembly, which is why the laminates were made. While it is too expensive to make an assembly error with these laminates, test assemblies with laser cut plywood (3mm) would work as a try-and-fail/succeed arena. The plywood has the same thickness as the laminates. It took a few attempts and some design changes before we could move on to the laminates.

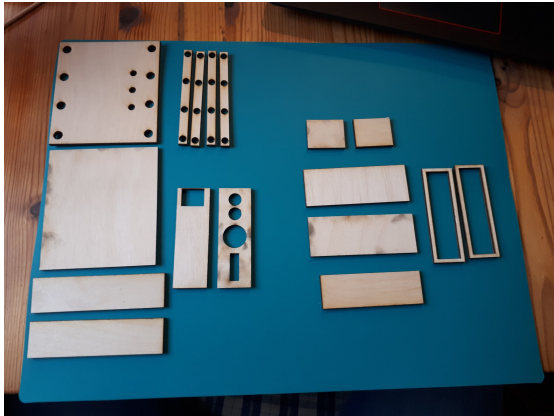


Figure 120: Test assembly with plywood

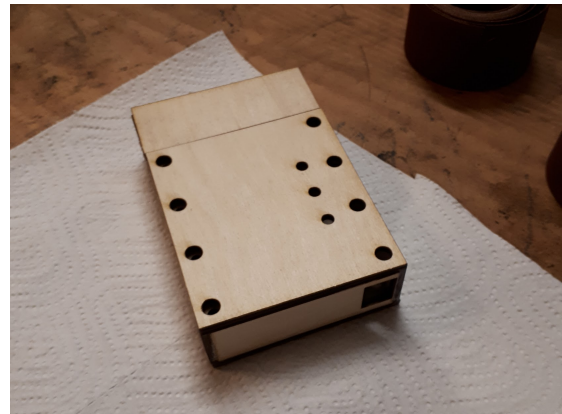


Figure 121: Plywood Assembly

The parts and test specimens were cut on a machine, Figure 122, lent to us by Kåre Særen. To use this we had to trace the cut-line manually based on Solidworks drawings and feed the data into a software, Figure 123. All parts were then cut with a tungsten carbide drill and after cutting, they were sanded before assembling and after to ensure smooth edges and a good grip, 87.



Figure 122: Laminate cutting



Figure 123: Machine program



To ensure that the nano casing can be classified as minimum IP65 the parts were assembled with Araldite AW4858/ HW4858 and all the exposed cut and sanded surfaces was sealed with Araldite. The assembly was post-cured in the oven at 60°C for 5 hours, Figure 126. This to get the most out of the Araldite.

This casing weighs 175gr without the battery compartment, gaskets and components inside.

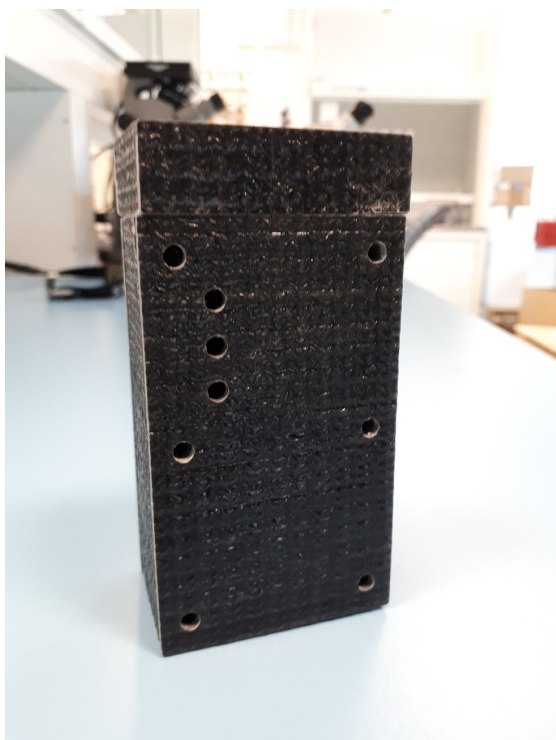


Figure 124: Nano casing for testing in Super-L 300



Figure 125: Custom gaskets

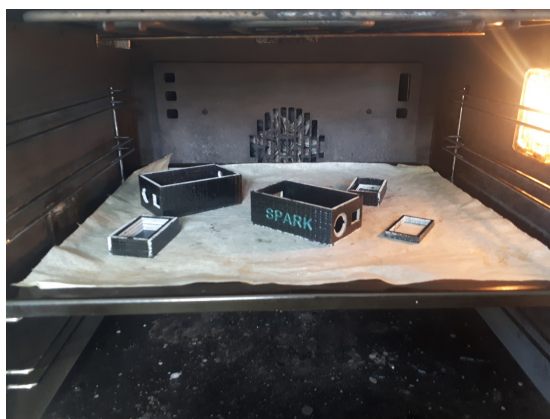


Figure 126: Post-curing assembly

#### 5.4.0.4 Moisture Inside the Casing

The way we have chosen to address the challenge with moisture inside the casing, as mentioned in section 4.2.1, is that we found that a gel with high dielectric strength that would protect the components and ingress protection ability of the casing. Because of the worldwide situation we did not order a gel like this online, but found one in a local power supply store. The RayTech Wondergel Invisible is a single component insulating gel. The gel has a wide operating- and installation temperature range, dielectric strength  $\geq 23\text{kV/mm}$  and has an IP68 classification [122]. This gel can be used on both casings.

All designs shown at the presentations were inspired by other training products from Spectac [131].

## 5.5 Early Stages of Prototyping

### 5.5.1 Prototype V1

This prototype was created for our first presentation. We did this because we wanted to give our customer something they could evaluate and also to easier showcase our project for the audience. Furthermore it gave us an opportunity to dive into some technical work to better understand the system to possibly identify challenges within the different disciplines.

The prototype is designed and 3D printed by the team. The electronics is composed of an Arduino Uno R3 development board and two nRF2401 radio transceivers.

The physical architecture of the prototype can be seen in Figure 127

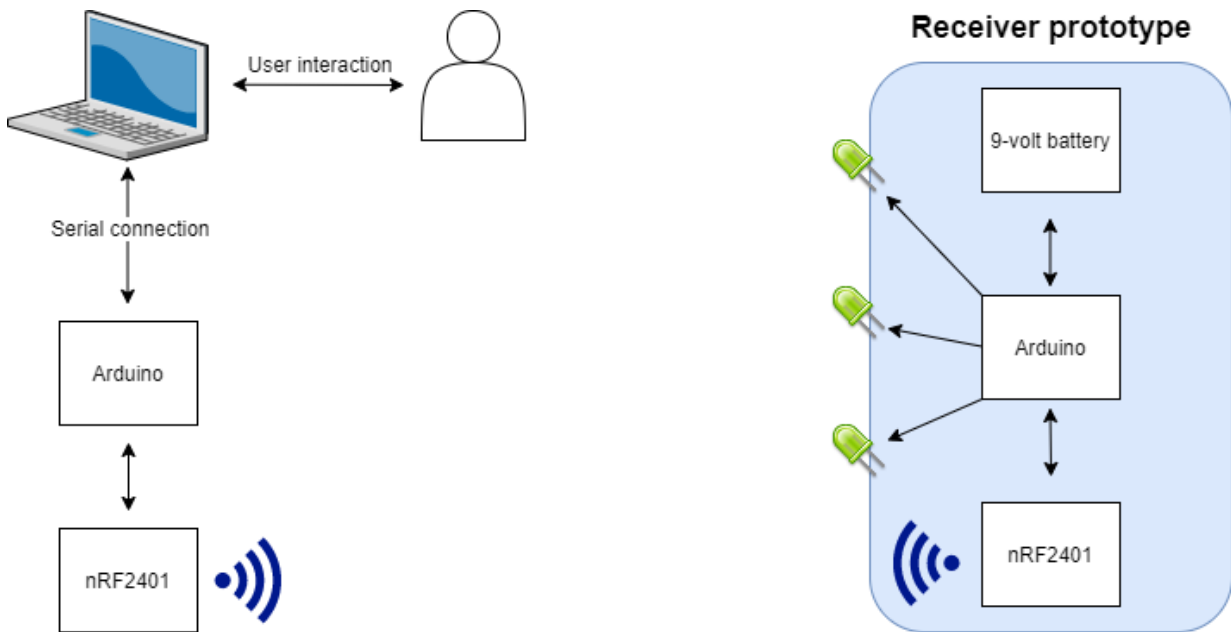


Figure 127: Prototype 1.0 physical architecture



### Arduino Uno R3

Arduino Uno R3 is a microcontroller board based on the ATmega328P microcontroller and provides different I/O and easy integration with various sensors. Arduino Uno R3 can be used for rapid prototyping and is well documented. More about the ATmega328P can be found in the data sheet [11]

### nRF2401

The nRF2401 is a low powered 2.4 GHz transceiver that was selected because of its easy interface. More information about the nRF2401 can be found in the data sheet [85]

### Code

To program the Arduino Uno R3 we used the open source RF24 library [135]. This made it easy to integrate nRF2401 with the Arduino Uno R3

### Physical encapsulation

The box to encapsulate the electronics was designed in Fusion 360 and printed using PLA filament. The inspiration for the box design was based on discussions with the customer, other designs they have [131] and imagination.

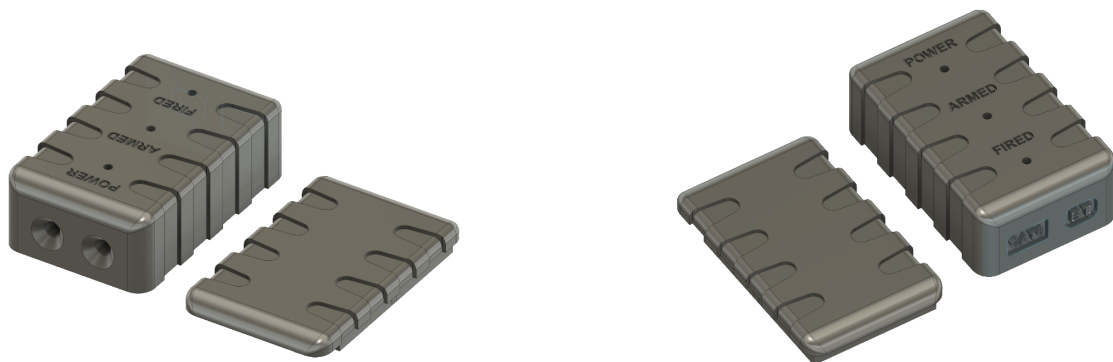


Figure 128: Prototype V1

**5.5.2 Prototype V2**

Spectac want to have something that they can show potential buyers of the system and other collaboration partners. That is why we want to focus early on getting them a prototype that can do just that. This prototype’s main task is to only include the functionality that brings value to the operator. In practise that means to keep things as simple as possible, only focusing on the core functionality. After this proof of concept prototype, we want to spend our time developing a realizable product, taking all the requirements and technical solutions into consideration.

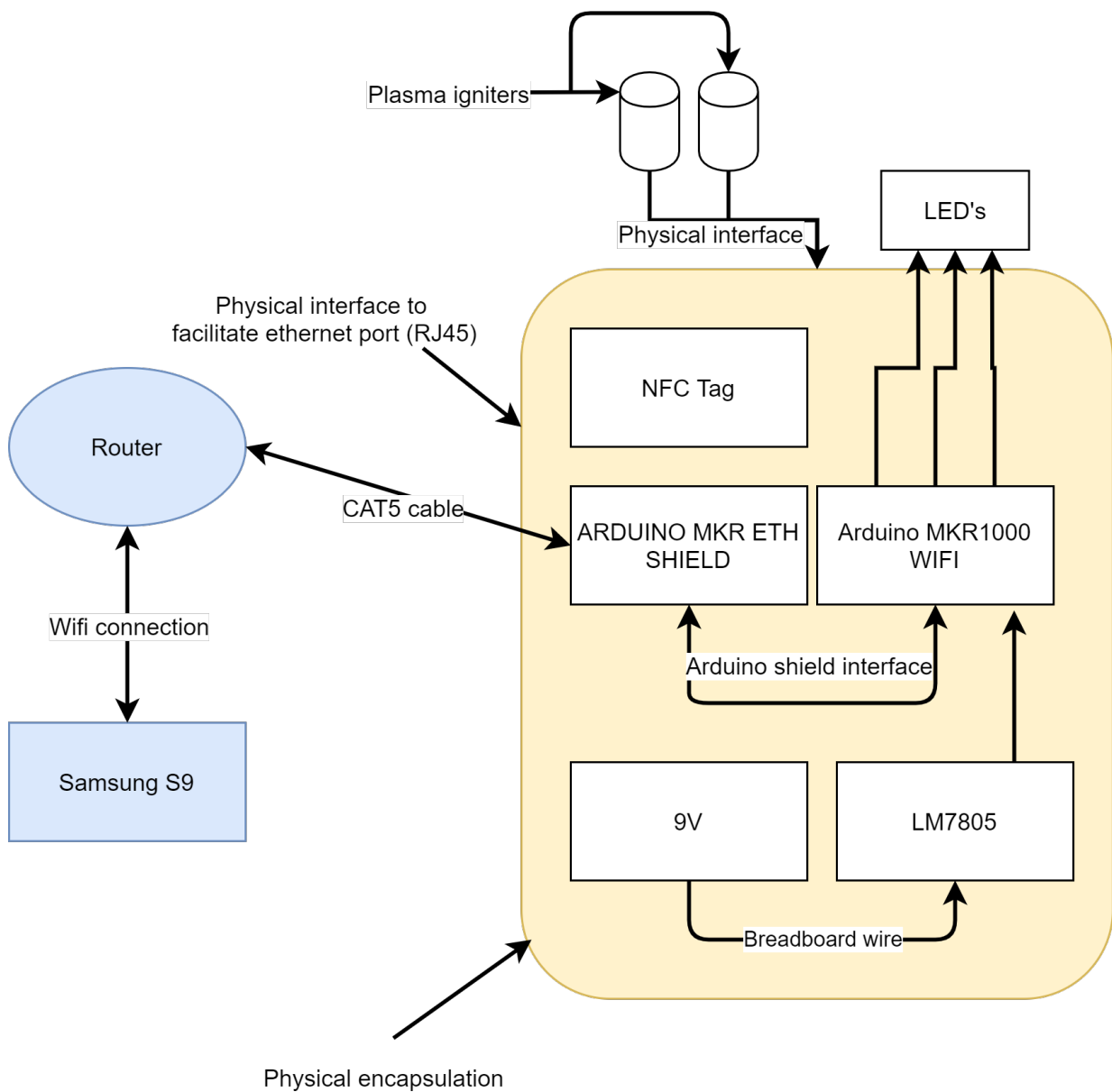


Figure 129: Prototype V2 physical architecture

### Mobile application prototype

This second iteration of the prototype development consists of a mobile application, which is developed to communicate with the receiver of the remote firing system. This is a huge improvement from the first prototype, since the communication link was established between a laptop and the receiver using a radio transmitter, and now there is a mobile application that communicates with the receiver over the standard Internet Protocol (IP).

As mentioned in section 5.2, the mobile application is developed in Qt Framework with C++ and is operative on the Android operating system. Since there are numerous customer requirements directed towards the mobile application, it was necessary to set boundaries for how much functionality that should be implemented for this second iteration of the prototype. Therefore it was determined to develop the NFC pairing functionality, a real-time map with waypoints functionality, along with some network infrastructure to communicate with the receiver of the remote firing system.

Figure 130 displays how the main view of the application appears on an Android emulator. The application is developed in landscape mode, because the mobile phone that the special force operators use at the field is oriented in landscape mode. The main view is the first page that the special force operator observes, and therefore it is designed to be as simple as possible to avoid any complications that might occur during a stressful situation.



Figure 130: Main view (landscape)

However the user interface of the mobile application is fully responsive, which means that it looks the same and behaves similar in both portrait and landscape mode. This is a benefit, because the application interface adjusts based on the screen resolution, which means that the application can be used on devices with different screen sizes, and even larger tablets running the Android operating system.

The mobile application comes with a navigation sidebar on the left side, which consists of links to different pages of the application. There is only one page available in the sidebar at this moment, but more pages will be added as the application grows.



Figure 131: Navigation sidebar (landscape)

The overview page shows a list of all deployed receivers, and this page allows the special forces operator to pair a new receiver as well. The overview page contains two panels, as shown in Figure 132, where there is a list of all deployed receivers in one panel (called Overview), and there is a real-time map where the deployed receivers are visualized and pinpointed in the other panel (called Mapview).

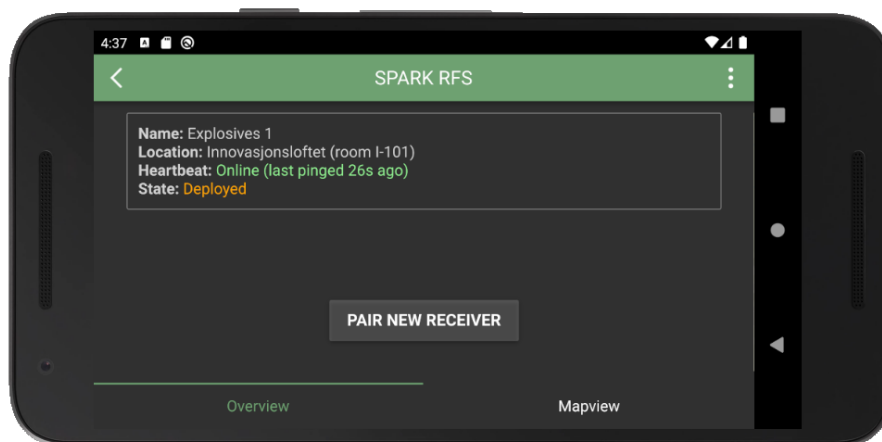


Figure 132: Receiver overview (landscape)

The mapview page shows a real-time map of all deployed receivers, along with the functionality to add new receivers to the map as well. This is an effortless way to interact with deployed receivers, as it gives the special forces operator the location of different receivers.

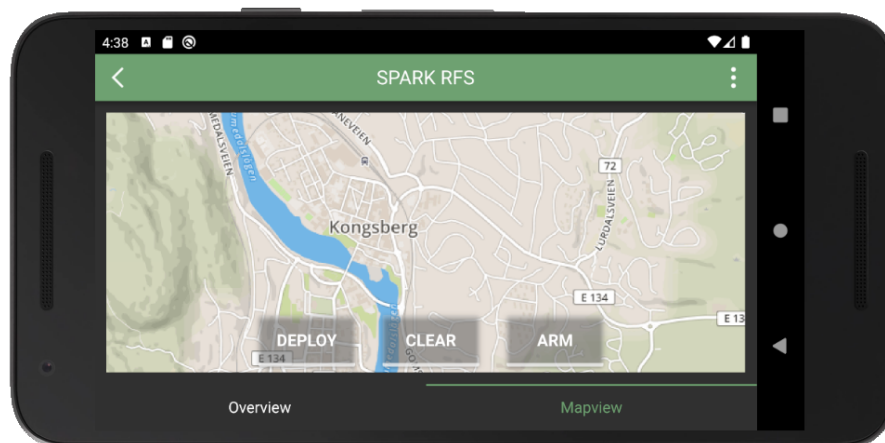


Figure 133: Map view (landscape)

### Network and communication

The network communication between the mobile application and the receiver is established through standard Internet Protocol (IP), where the receiver currently operates as a UDP server and the mobile application operates as a client in order to communicate with the receiver. This is purely for demonstration purposes, as the milestone of this prototype is to demonstrate that the mobile application successfully communicates with the receiver and that it is capable of changing its state from idle to deployed or armed, and to detonate the receiver as well.

The reason why the receiver is configured as a UDP server, instead of TCP, is simply because the client only needs to send a bitstream to the server in order to indicate that the state must be changed, and there is no need to confirm that the bitstream has been received by the server, since this is only for demonstration purposes and neither protocols will be used as standalone servers in the actual remote firing system.

The UDP server is simple and works perfect to give a demonstration of how the mobile application connects to the receiver. The server was developed on an Arduino MKR board, which is connected to a router to establish a local network connection. The Arduino code for the UDP server is included in the appendix.

As for the network communication on the client-side, the mobile application uses built-in methods in Qt Framework to establish a UDP interface communication using QUdpSocket [120], which allows the application to send and receive UDP datagrams over the local network. The following code snippet shows how the mobile application establishes a connection to the UDP server (full code is included in the appendix).

### Tactical mapview

As of this prototype we've established basic map functionality and the ability to add tactical markers representing deployed receivers. The markers has the ability to change their color (STANDBY = GREEN, ARMED = RED) which shall in future development change in accordance to the operative state of the receiver. We have so far managed to enable functionality of our waypoints using only 1 marker at the time, this will be revised and developed further to provide a solution for any given  $N$  markers available on screen.

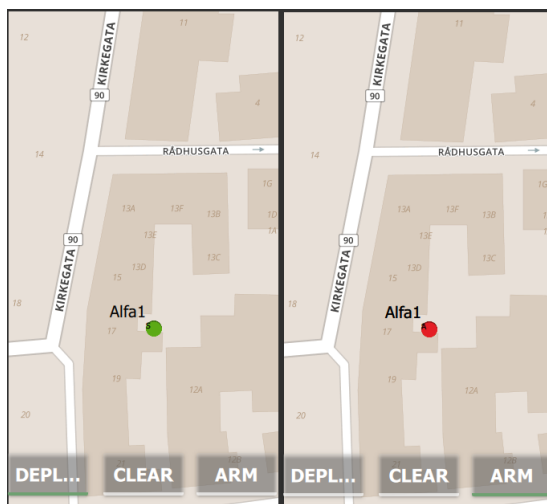


Figure 134: Tactical Map View

Per now this core functionality is in place:

- Persistent map view (Remembers location when leaving and entering map view)
- Ability to place 1 marker with a hold & touch gesture on screen
- Ability to change color of the markers in accordance to a deployed receivers state

Another future revision will be the deleting of a state instantaneously as a new one is set. Changing from a "Green" state to a "Red" state should draw a new overlay on top of the old one and then delete the old state, freeing unnecessary resources.

As of now we interact with the map using three main buttons; *Deploy*, *Clear* and *Arm*, which act as the EOD operators user interface. Take notice that this version is strictly for demonstration purposes to learn and build main functionality upon for further progress.

The user interface itself is *not* safe for operative use, and can easily lead to misplaced clicks and unintended actions.

### 5.5.2.1 Casing V2

The second version of the container unit is still quite large due to the components accessible at this moment. We have tried to implement as many physical customer- and system requirements as possible to get a, as mentioned earlier, proof of concept prototype.

For this version we still have chosen to use a 9V battery instead of CR123, as in prototype V1. A partition in the bottom part of the device Figure 137, between the battery- and circuit-board space will make sure that the components does not interfere with each other. The partition has a designated space for wires so that they can be gathered in one place and not move around, Figure 137 point 9.

The bottom part also holds one plasma igniter (the other is in the lid) and a suspension Figure 135 point 1 and 2. The suspension is to satisfy the system requirement nr 045, 113. For the LEDs, a support around the holes was made in the lid, this is to ensure that the LEDs does not exceeds the height of the lid, thus making them less visible to others than the operators, Figure 136 point 8. The lid also holds the on/off switch and 1 plasma igniter, Figure 136 point 5 and 6.

The NFC tag has a designated space in the lid, Figure 136 point 7. Here the tag can be clicked into place, and the spot is marked on the top of the lid Figure 139.

There is a storage lid that encloses the area with the plasma igniters and on/off switch. This is for three reasons: 1. It will be easier to make the device water resistant, 2. it will prevent an accidental power-on situation and 3. there are two holder for the plasma igniters in that lid Figure 138 point 10.

Different solutions for all components has been tried out, so drawing wise this is prototype V4.2. We are so lucky that one of our team member has a 3D printer at home. This gives us the opportunity to use the try and fail/succeed work method.



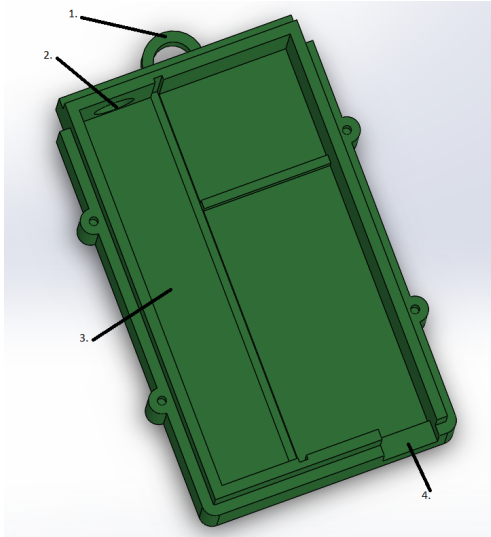


Figure 135: Bottom of device

- 1. Suspension (113, SR-045)
- 2. Plasma igniter 1
- 3. Battery compartment
- 4. RJ45

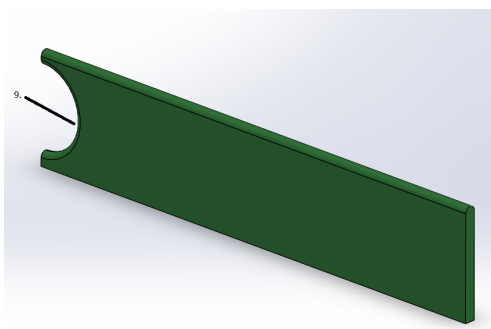


Figure 137: Battery- and board compartment partition

- 9. Wire gate

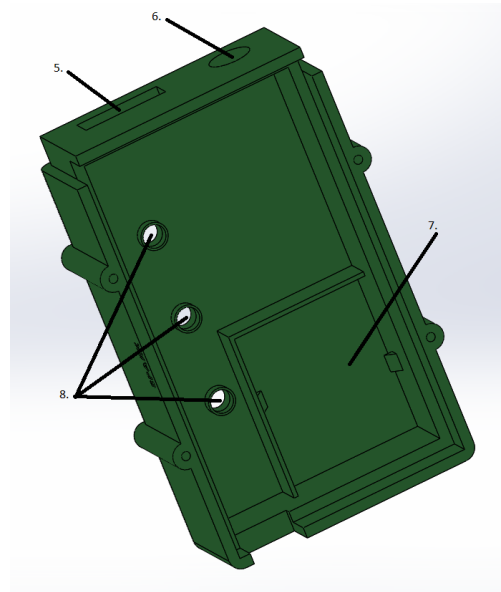


Figure 136: Lid with NFC space

- 5. On/Off switch
- 6. Plasma igniter 2
- 7. NFC tag compartment
- 8. Diodes for On, Armed and Fired (99, CR-009)

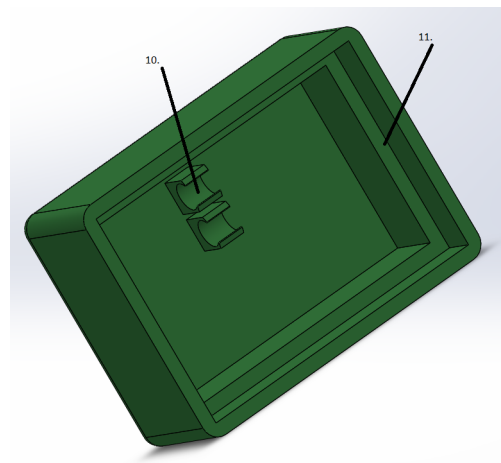


Figure 138: Storage lid

- 10. Plasma igniter storage
- 11. Gasket edge (71, CR-004)



Figure 139: Closed assembly

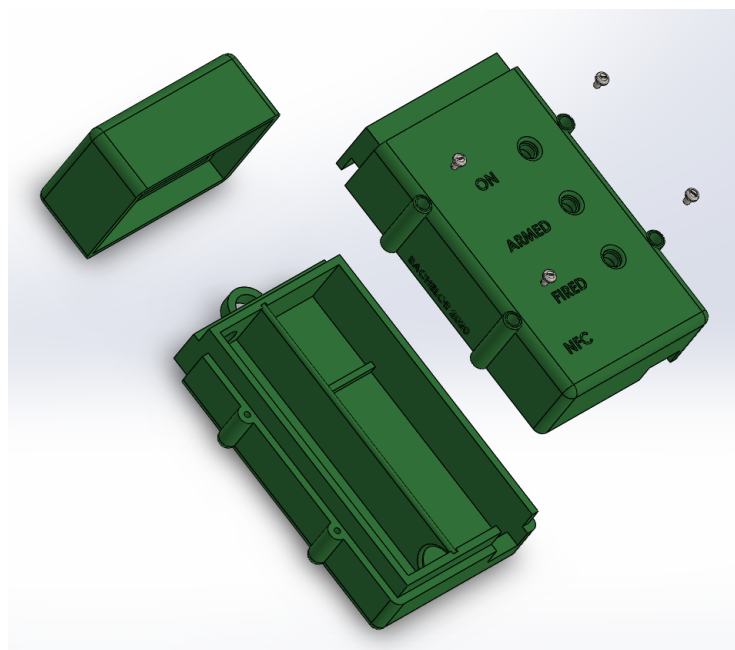


Figure 140: Open assembly

# SPARK

## 6 Conclusions

### Contents

6.0.1	Thesis Summary . . . . .	224
6.0.2	Evaluation . . . . .	225
6.0.3	Future Work . . . . .	226
6.0.4	Closing Thoughts . . . . .	227

### 6.0.1 Thesis Summary

This thesis served to provide an overview of a multi-disciplinary engineering team’s approach to solving the problems with today’s remote firing systems. First, in **Section 1**, the modern environment of warfare in the 21st century was discussed, including its ever-evolving complexity and focus on the individual soldier and their decision-making abilities; the degree of autonomy of a soldier in the lower echelon of military rank was determined to have vastly increased over the years, necessitating a modern solution to aid them.

In **Section 2**, the engineering team’s approach to project management was discussed, wherein the SCRUM project model was detailed, and how it was used to facilitate an AGILE workflow. Furthermore, Jira, Slack, Clockify and Google Drive were tools discussed as parts of the team’s daily workflow, wherein Jira was employed to facilitate SCRUM. The last part of this section addressed the risk of the project, and how it was handled throughout the entirety of the project’s duration.

**Section 3** detailed the problem with today’s solution, highlighting its weight and size, its use of two units (a receiver and transmitter), and its lack of a software solution. The team’s approach to requirements were also discussed, opting for a hierarchy wherein customer requirements were at the top, followed by derived system requirements, followed again by derived technical requirements. The validation and verification process was also addressed, where the team opted to use test cases, acceptance criteria, and frequent communication with the customer to ensure the solution aligned with their vision.

**Section 4** presented the proposal produced over the course of the project. In it, a three-pronged solution was proposed from all the disciplines. The software team presented a conceptual software model; the base functionality of the system was laid out, followed by a discovery of data, before consolidating these into a software architectural model structured in the MVC pattern. Furthermore, the software team expanded upon the base use case model to convey that there were more use cases on a lower level, and choosing the “Make Decision” use case in particular to expand upon further, as this was the most crucial of basic functionality. Another, more specific software architectural model was constructed to that effect. Finally, some of the technologies the team planned on using were addressed, including TCP/IP, software security barriers and security concerns, WiFi, and failover.

Mechanical presented an overview of the design challenges and important considerations when approaching the preliminary design. Background theory, material alternatives, ingress protection standards, size, weight and user friendliness/manageability are addressed to prepare for the implementation. Two different models were made and tested, both of which

had mostly the same features, but different material.

The electrical team presented a proposal for how the electrical system in the receiver could be designed. Critical components such as the plasma igniters, capacitors, step-up circuitry, battery, safety barriers were addressed, as well as crucial communication interfaces such as WiFi, RJ45, and NFC. Seeing as how one of the primary objectives of the project was to minimize the size and weight of the receiver, the circuitry had to be minimized whilst maintaining or surpassing the capabilities of existing RFS systems; to that effect, a custom PCB was detailed.

**Section 5** presented the partial implementation of the proposal discussed in the previous section. In it, a user-friendly application running on a Samsung S9, and two receivers (one 3D printed with PETG, the other nanotube-reinforced epoxy resin), and an Arduino successfully communicated over a UDP network using a router. Using either NFC or IP, the application could pair with several receivers and control these as expected (arm, disarm, detonate, etc.). A map and waypoint functionality had also been implemented, wherein an operator could navigate a map and place a waypoint to represent the location and state of a receiver. Upon switching the firing state of the receiver (on, armed, detonated), the LEDs switched accordingly.

### 6.0.2 Evaluation

We believe our proof-of-concept proves that it is possible to innovate today's remote firing system. In section 3.1, where we described the five objectives of the project, *objective 1* is stated to be the reduction of the size and weight of RFS systems. This was achieved by the mechanical team in section 4.2.2, where a proposal for the casing constructed out of nanotube-reinforced epoxy resin is presented, and in section 5.4.0.2 where the prototype casing that was actually produced is detailed. Furthermore, in section 5.3.6, the electrical team proposed a custom PCB wherein the size and weight were the driving factors of its design. Of course, by ridding the system entirely of the stand-alone transmitter (*objective 2*, detailed subsequently), the total weight of the system is further reduced.

*Objective 2* was to cut out the stand-alone transmitter in favor of digitalizing its functionality and use a mobile device instead. This has been achieved in section 5.2 with a mobile application running on Android, capable of receiver pairing via NFC or IP, and the controlling of these receivers (to arm it, disarm it, detonate it, and so on).

*Objective 3* was to develop a user-friendly software application capable of aiding the soldier

## 6. CONCLUSIONS

---

in their decision making. To a degree, this has been achieved - in terms of implementation, the aforementioned application can convey to the operator the location of the receiver, its current state, and the receiver is capable of sending warnings to the transmitter in certain conditions (for example, if the battery is low). This is detailed in the aforementioned section 5.2. However, in section 4.1.2, the software model accommodates far more than we ended up implemented (due to time constraints). Still, for a proof of concept, we believe we have sufficiently achieved this objective.

*Objective 4* was to develop training functionality inside the application. This was an ambitious goal that, in hindsight, may not have been the most appropriate for a proof of concept; such training software is not necessary to illustrate whether our Spark concept works or not. Consequently, we have not succeeded in this objective, but this was deliberate and planned, and we maintain that our proof of concept is feasible despite this.

Finally, *Objective 5* was more of an overarching objective of digitalizing the RFS system. Based on our achievements in objective 2 and 3, we believe this also renders this objective a success as a result.

### 6.0.3 Future Work

For the future, there are a number of improvements that can be made to the SPARK RFS in order to elevate it from a proof-of-concept to something that can safely and reliably be used in a real-life scenario. Specifically, in terms of *software*, we were not able to implement any manner of security (RSA or otherwise, for example) on the network; in a military scenario, it is absolutely critical that a system such as this is as safe and secure as it can be. Additionally, due to time constraints and inexperience working on a project of this size, our code in general is not as good as it could be, and a few parts of the front-end is hard-coded to facilitate a demonstration rather than usage in the field (notably the receiver frames). The software architecture proposed in section 4 carries a lot of value, but we were unable to truly capitalize on this as much as we would have wanted; given more time (and armed with the experience and knowledge we have now), elaborating further on this and tying it better together with the implementation would have been ideal. Our research into WiFi direct, Bluetooth and DDS/ROS could also be used to evolve the system further and transform it into a more ad hoc solution that doesn't depend on a router.

When it comes to the *electrical side*, the components and subsystems directly needs additional testing, measurement, and analysis in order to be able to conclude what's sufficient according to the customer's wishes, and what could stand to be improved. While the PCB constructed for the project is suitable for a proof-of-concept, we have not been able to

rigorously test it enough to be able to confidently determine its weaknesses.

Finally, for the *mechanical side*, it would be optimal to cast the nano casing instead of piecing it together with different parts. Testing revealed that the weakness lies in the joints, even with strong adherent. Furthermore, the casing could always stand to become even smaller, but this largely depends on the electrical components it houses, although a better means of lid attachment could also yield some size reduction. Due to the covid-19 situation at the time of writing this thesis, various means of testing have been challenging to perform (including ingress protection), and the casing could stand to benefit from further testing to determine weaknesses and areas of improvement. To combat potential condensation and corrosion in the interior of the casing, another alternative to the RayTech sealing gel from section [5.4.0.4](#), is to fill the casing with resin given that it doesn't need to be opened again. Here the test lies in the weight difference between gel and resin. Finally, testing of Faraday capabilities of single walled carbon nanotubes could be of interest, as this would help against electromagnetic fields.

#### 6.0.4 Closing Thoughts

Without a doubt, this project has been fun, educational, rewarding, but also incredibly challenging. Particularly in a multidisciplinary team, the covid-19 situation forced us to be inventive and creative in order to facilitate the crucial integration process, as well as the planning of interfaces across (and within) disciplines. Online drawing boards and diagram tools were paramount in ensuring we were all on the same page; words are often interpreted differently by each member, and so being able to visualize and convey your own understanding this way was important. Even then, testing was a challenge throughout the semester, particularly for the mechanical team who depended on the facilities provided by USN. Overall, we're satisfied with how we handled the situation, and we were in good spirits every morning when we conducted our daily meeting, despite the circumstances. We're incredibly thankful to both Spectac and USN for giving us the opportunity to work on a project like this; Spectac for the problem to solve, the equipment to do so, and the experience and knowledge to aid us on the journey, and USN for shaping us into budding engineers over three years, equipping us to professionally and intelligently handle problems such as Spectac's.

## 7 References

- [1] 3DInsider. *PETG Filament: Properties, How to Use, and Best Brands*. URL: <https://3dinsider.com/petg-filament/>. (accessed: 10.02.20).
- [2] J. H. Abawajy, M. Othman, and et al. *Proceedings of the International Conference on Data Engineering 2015*. Springer, 2015.
- [3] UNAMA (UN Assistance Mission in Afghanistan). *Quarterly Report On the Protection of Civilians in Armed Conflict: 1 January to 30 September 2017*. 2019. URL: [https://unama.unmissions.org/sites/default/files/unama\\_protection\\_of\\_civilians\\_in\\_armed\\_conflict\\_quarterly\\_report\\_1\\_january\\_to\\_30\\_september\\_2017\\_-\\_english.pdf](https://unama.unmissions.org/sites/default/files/unama_protection_of_civilians_in_armed_conflict_quarterly_report_1_january_to_30_september_2017_-_english.pdf). (accessed: 04.03.2020).
- [4] M. I. Ahmad et al. "Performance analysis of data distribution service on error prone channels". In: *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*. 2017, pp. 33–37. DOI: 10.1109/RISE.2017.8378120.
- [5] Wifi Alliance. *How far does a Wi-Fi Direct connection travel?* URL: <https://www.wi-fi.org/knowledge-center/faq/how-far-does-a-wi-fi-direct-connection-travel>. (accessed: 20.03.2020).
- [6] Wifi Alliance. *How fast is Wi-Fi Direct?* URL: <https://www.wi-fi.org/knowledge-center/faq/how-fast-is-wi-fi-direct>. (accessed: 20.03.2020).
- [7] Wifi Alliance. *Wi-Fi Direct*. URL: <https://www.wi-fi.org/discover-wi-fi/wi-fi-direct>. (accessed: 17.03.2020).
- [8] Android. *Create P2P connections with Wi-Fi Direct*. URL: <https://developer.android.com/training/connect-devices-wirelessly/wifi-direct>. (accessed: 20.03.2020).
- [9] D. Antonioli, N. O. Tippenhauer, and K. B. Rasmussen. *The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR*. 2019. URL: <https://www.usenix.org/conference/usenixsecurity19/presentation/antonioli>. (accessed: 22.03.2020).
- [10] Atlassian. *The #1 software development tool used by agile teams*. URL: <https://www.atlassian.com/software/jira>. (accessed: 23.01.20).
- [11] *ATmega328P, 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*. Microchip, 2015.
- [12] *Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V datasheet*. Atmel, 2014.
- [13] *ATWINC15x0-MR210xB IEEE® 802.11 b/g/n SmartConnect IoT Module*. Microchip, 2018.



- [14] *ATWINC15x0/ATWINC3400 Wi-Fi® Network Controller Software Programming Guide*. Microchip, 2019.
- [15] R. Ballantine et al. *Remote Initiation for the Remote Initiation of Explosive Charges*. 2006. URL: <https://patentimages.storage.googleapis.com/48/f5/60/5bd35851cf0556/US8134822.pdf>. (accessed: 05.03.2020). U.S. Patent 8,134,822 B2.
- [16] Ankit Banerjee. *Samsung Galaxy S9 and Galaxy S9 Plus problems and how to fix them*. 2019. URL: <https://www.androidauthority.com/samsung-galaxy-s9-plus-problems-fix-859473/>. (accessed: 05.04.2020).
- [17] G. Maarten Bonnema, Karel T. Veenliet, and Jan F. Broenink. *Systems Design and Engineering: Facilitating Multidisciplinary Development Projects*. CRC Press, 2016. ISBN: 9781498751261. URL: <https://www.crcpress.com/Systems-Design-and-Engineering-Facilitating-Multidisciplinary-Development/Bonnema-Veenliet-Broenink/p/book/9781498751261>.
- [18] G. Maarten Bonnema Karel Th. Veenliet Jan F. Broenink. *SYSTEMS DESIGN and ENGINEERING Facilitating Multidisciplinary Development Projects*. CRC Press Taylor & Francis Group, 2016.
- [19] M. Capek et al. “Optimal Planar Electric Dipole Antennas: Searching for antennas reaching the fundamental bounds on selected metrics”. In: *IEEE Antennas and Propagation Magazine* 61.4 (2019), pp. 19–29.
- [20] J. Edward Carryer, Matthew Ohline, and Thomas Kenny. *Introduction to Mechatronic Design*. Vol. 1. Pearson, December 2010.
- [21] *Wi-Fi Peer-to-Peer (P2P) Technical Specification Version 1.7*. Standard. Wi-Fi Alliance P2P Task Group, 2016.
- [22] Janus B. Cihlar and Joey R. Bray. “Induced Currents on Electric Detonators for Improvised Explosive Device Pre-Detonation”. In: *2011 IEEE International Symposium on Electromagnetic Compatibility* (2011). DOI: 978-1-4577-0811-4/11. (accessed: 15.04.20).
- [23] Janus B. Cihlar and Joey R. Bray. “Thermal Response Tester for Qualification of Detonators”. In: *2013 IEEE Radar Conference (RadarCon13)* (2013). DOI: 978-1-4673-5794-4/13. (accessed: 15.04.20).
- [24] Lexington Clayton R. Paul Univ. of Kentucky. *Transmission Lines in Digital and Analog Electronic Systems: Signal Integrity and Crosstalk*. URL: <https://ezproxy2.usn.no:2160/servlet/opac?bknumber=5732774>. (accessed: 14.04.20).
- [25] Cloudflare. *What Is Edge Computing?* URL: <https://www.cloudflare.com/learning/serverless/glossary/what-is-edge-computing/>. (accessed: 24.05.2020).

## 7. REFERENCES

- [26] P. Cope, J. Campbell, and T. Hayajneh. *An investigation of Bluetooth security vulnerabilities*. 2017. URL: <https://ieeexplore.ieee.org/document/7868416>. (accessed: 22.03.2020).
- [27] J. T. Costello, K. L. Stewart, and I. B. Stewart. *Inside the ‘Hurt Locker’: The Combined Effects of Explosive Ordnance Disposal and Chemical Protective Clothing on Physiological Tolerance Time in Extreme Environments*. 2015. URL: <https://academic.oup.com/annweh/article/59/7/922/2196094>. (accessed: 05.03.2020).
- [28] M. Carrillo D. M. Selfa and M. Del Rocio Boone. *A Database and Web Application Based on MVC Architecture*. 2006. URL: <https://ieeexplore.ieee.org/document/1604744>. (accessed: 22.05.2020).
- [29] U. Demir, C. Tapparello, and W. Heinzelman. “Maintaining Connectivity in Ad Hoc Networks Through WiFi Direct”. In: *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. Oct. 2017, pp. 308–312. DOI: 10.1109/MASS.2017.60.
- [30] Android Developers. *App Manifest Overview*. 2020. URL: <https://developer.android.com/guide/topics/manifest/manifest-intro>. (accessed: 04.04.2020).
- [31] Android Developers. *NFC basics*. URL: <https://developer.android.com/guide/topics/connectivity/nfc/nfc>. (accessed: 04.03.2020).
- [32] Analog Devices. *ADXL337*. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL337.pdf>. (accessed: 08.05.20).
- [33] UNODA (United Nations Office for Disarmament Affairs). *Improvised Explosive Devices (IEDs) Publication*. URL: <https://www.un.org/disarmament/convarms/ieds2/>. (accessed: 04.03.2020).
- [34] *DN2540N8 N-Channel Depletion-Mode Vertical DMOS FETs datasheet*. Supertex Inc, 2013.
- [35] Qt Official Documentation. *How to create a simple chat application*. URL: [wiki.qt.io/WIP-How\\_to\\_create\\_a\\_simple\\_chat\\_application](http://wiki.qt.io/WIP-How_to_create_a_simple_chat_application). (accessed: 24.05.2020).
- [36] Qt Official Documentation. *Signals and Slots*. URL: <https://doc.qt.io/qt-5/signalsandslots.html>. (accessed: 24.05.2020).
- [37] Miroslav Dulik. *Security in Military Cloud Computing Applications*. URL: [http://sm.aos.sk/images/dokumenty/archiv\\_cisel/1\\_2016/Article5.pdf](http://sm.aos.sk/images/dokumenty/archiv_cisel/1_2016/Article5.pdf). (accessed: 24.05.2020).
- [38] *Duracell Ultra 123 datasheet*. Duracell.
- [39] *ENC28J60 Datasheet*. Microchip, 2006.

- [40] *ENERGIZER 123 Product Datasheet*. Energizer.
- [41] Kevin R. Fall and W. Richard Stevens. *TCPIP Illustrated*. Vol. 1. Person Education, December 2010.
- [42] Felgo. *About Felgo*. URL: <https://felgo.com/qt-app-developers>. (accessed: 23.03.2020).
- [43] Fraunhofer FKIE. *Towards Verification of NATO Generic Vehicle Architecture-Based Systems*. URL: <https://d2vkrkwbbxbylk.cloudfront.net/sites/default/files/iccrts2016-66.pdf>. (accessed: 27.04.2020).
- [44] Thomas L. Floyd. *Electronic Devices Conventional Current Version*. Pearson, 2018.
- [45] DDS Foundation. *What is DDS?* URL: <https://www.dds-foundation.org/what-is-dds-3>. (accessed: 23.03.2020).
- [46] Eclipse Foundation. *Eclipse*. URL: <https://netbeans.org/>. (accessed: 24.05.2020).
- [47] Matthew Gast. *802.11® Wireless Networks: The Definitive Guide*. O’Reilly, 2012.
- [48] Florian H. Gojny et al. “Influence of different carbon nanotubes on the mechanical properties of epoxy matrix composites – A comparative study.” In: *Composites Science and Technology* 65 (2005). DOI: 10.1016/j.compscitech.2005.04.021. URL: <https://www.sciencedirect.com/>.
- [49] Arturo Guadalupi. *MKRETHShieldV2.0 schematic*. Arduino, 2018.
- [50] P. Henkel. “Reduction and optimization of almanac transmission for GNSS satellites”. In: *Proceedings ELMAR-2011*. 2011, pp. 329–332.
- [51] *HP 3D High Reusability PA 12*. HP Development Company, L.P., 2017.
- [52] IBM. *What is edge computing?* URL: <https://www.ibm.com/cloud/what-is-edge-computing>. (accessed: 24.05.2020).
- [53] *Introduction to SPI Interface*. Analog Devices, 2018.
- [54] *Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1)*. Standard. International Organization for Standardization, 2004.
- [55] John B. Jacobsen et al. “Climate-Protective Packaging”. In: *IEEE industrial electronics magazine* September 2014 (19.09.2014). DOI: 0.1109/MIE.2014.2330912. (accessed: 12.04.20).
- [56] W. Jung, H. Ahn, and Y. Ko. “Designing content-centric multi-hop networking over Wi-Fi Direct on smartphones”. In: *2014 IEEE Wireless Communications and Networking Conference (WCNC)*. Apr. 2014, pp. 2934–2939. DOI: 10.1109/WCNC.2014.6952920.

- [57] *JX0026D21BNL datasheet*. Pulse, 2019.
- [58] Serope Kalpakjian and Stephen R.Schmid. “Manufacturing Engineering and Technology”. In: Pearson Education Centre, 2013. Chap. 40.4.
- [59] J. C. Knight. “Safety critical systems: challenges and directions”. In: *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002*. 2002, pp. 547–550.
- [60] Gen. C. C. Krulak. *The Strategic Corporal: Leadership in the Three Block War*. 1999. URL: <https://apps.dtic.mil/dtic/tr/fulltext/u2/a399413.pdf>. (accessed: 04.03.2020).
- [61] Thomas L.Floyd. *Digital Fundamentals*. Vol. 11. Pearson, 2014.
- [62] Thomas L.Floyd. *Electronic Devices*. Vol. 10. Pearson, 2017.
- [63] *L1 Series 1.0 Watt*. American Power Design, 31.01.2014, Rev. 3.01.
- [64] J. H. Lee, M. Park, and S. C. Shah. “Wi-Fi direct based mobile ad hoc network”. In: *2017 2nd International Conference on Computer and Communication Systems (ICCCS)*. July 2017, pp. 116–120. DOI: 10.1109/CCOMS.2017.8075279.
- [65] G. Lewis et al. “Tactical Cloudlets: Moving Cloud Computing to the Edge”. In: *2014 IEEE Military Communications Conference*. 2014, pp. 1440–1446. DOI: 10.1109/MILCOM.2014.238.
- [66] *Lithium Battery, 2/3A Size datasheet*. GPBatteries.
- [67] K. Liu et al. “Development of Mobile Ad-hoc Networks over Wi-Fi Direct with off-the-shelf Android phones”. In: *2016 IEEE International Conference on Communications (ICC)*. May 2016, pp. 1–6. DOI: 10.1109/ICC.2016.7511190.
- [68] *LM3914 Dot/Bar Display Driver*. Texas Instruments, March 2013.
- [69] D.W. Lovell and DP. Baker. *The Strategic Corporal Revisited: Challenges for combatants in 21st-century warfare*. UCT Press, 2017.
- [70] *LT3757-3757A*. ANALOG DEVICES, 2008–2018.
- [71] T. Talty M. Collotta G. Pau and O. K. Tonguz. *Bluetooth 5: A Concrete Step Forward toward the IoT*. 2018. URL: <https://ieeexplore.ieee.org/document/8419192>. (accessed: 22.03.2020).
- [72] Lihua Ma and Shangli Zhou. “Positional Accuracy of Gps Satellite Almanac”. In: *Artificial Satellites* 49.4 (2014), pp. 225–231. URL: <https://content.sciendo.com/view/journals/arsa/49/4/article-p225.xml>.
- [73] E. Mackensen, M. Lai, and T. M. Wendt. *Performance analysis of an Bluetooth Low Energy sensor system*. 2012. URL: <https://ieeexplore.ieee.org/document/6377634>. (accessed: 22.03.2020).

## 7. REFERENCES

- [74] C. T. Mann and H. Fischer. *Recent Trends in Active-Duty Military Deaths*. 2019. URL: <https://crsreports.congress.gov/product/pdf/IF/IF10899>. (accessed: 05.03.2020).
- [75] Mapbox. *Mapbox pricing*. URL: <https://www.mapbox.com/pricing/>. (accessed: 15.03.2020).
- [76] Karen Mason. “Fiberglass multiaxials reinforce their market share.” In: *CompositesWorld Magazine* (2/1/2005). DOI: 10.3303/CET1977034. URL: <https://www.compositesworld.com/articles/fiberglass-multiaxials-reinforce-their-market-share>.
- [77] Michael Medoff and Rainer Faller. *Functional Safety - An IEC 61508 SIL 3 Compliant Development Process*. Exida, 2014.
- [78] Mehdi.G.Mousavi. *Composite*. 2015. (Teaching material at USN).
- [79] S. L. Meshram and P. D. Dorge. “Design and performance analysis of mobile Ad hoc network with reactive routing protocols”. In: *2017 International Conference on Communication and Signal Processing (ICCSP)*. Apr. 2017, pp. 0443–0447. DOI: 10.1109/ICCSP.2017.8286396.
- [80] *MFRC63102HN*. NXP, 26.02.2020, Rev. 4.6.
- [81] C. K. Miller. “Data distribution over IP in high error rate military radio environments”. In: *IEEE Military Communications Conference. Proceedings. MILCOM 98 (Cat. No.98CH36201)*. Vol. 3. 1998, 1067–1071 vol.3. DOI: 10.1109/MILCOM.1998.727011.
- [82] Neatbeans. *Neatbeans IDE*. URL: <https://netbeans.org/>. (accessed: 24.05.2020).
- [83] NEMA. *Degrees of Protection Provided by Enclosures (IP Code) (identical national adoption)*. URL: <https://www.nema.org/Standards/ComplimentaryDocuments/ANSI-IEC-60529.pdf>. (accessed: 21.01.20).
- [84] NGVA. *NATO Generic Vehicle Architecture*. URL: <https://www.natogva.org/>. (accessed: 26.04.2020).
- [85] *nRF24L01 Single Chip 2.4GHz Transceiver Product Specification*. Vol. 2.0. Mouser Electronics, July 2007.
- [86] NXP. *PN7150*. URL: <https://www.nxp.com/docs/en/data-sheet/PN7150.pdf>. (accessed: 20.05.20).
- [87] NXP. *PN7150*. URL: [PN7150%20Hardware%20Design%20Guide](https://www.nxp.com/docs/en/application-note/AN11755.pdf). (accessed: 20.05.20).
- [88] NXP. *PN7150 Antenna Design and Matching Guide*. URL: <https://www.nxp.com/docs/en/application-note/AN11755.pdf>. (accessed: 20.05.20).

## 7. REFERENCES

- [89] OCSiAl. *Single Wall Carbon Nanotubes*. URL: <https://ocsial.com/en/material-solutions/tuball/>. (accessed: 10.02.20).
- [90] OIRA. *Purpose of Risk Assessment*. URL: <https://oiraproject.eu/en/purpose-risk-assessment>. (accessed: 24.01.20).
- [91] Oracle. *Java 2 Platform, Enterprise Edition (J2EE) Overview*. URL: <https://www.oracle.com/java/technologies/appmodel.html>. (accessed: 24.05.2020).
- [92] D. Ota and M. Pradhan. “Modular verification and validation for NATO generic vehicle architecture-based land platforms”. In: *2018 International Conference on Military Communications and Information Systems (ICMCIS)*. 2018, pp. 1–7. DOI: 10.1109/ICMCIS.2018.8398715.
- [93] Stack Overflow. *Adding markers/places to Qt QML Maps?* URL: <https://stackoverflow.com/questions/49344638/adding-markers-places-to-qt-qml-maps>. (accessed: 16.03.2020).
- [94] Stack Overflow. *How to get NFC working on Android using QT 5.6*. URL: <https://stackoverflow.com/questions/33346378/how-to-get-nfc-working-on-android-using-qt-5-6/40479333>. (accessed: 04.03.2020).
- [95] Heinz P.Bloch. “Ingress Protection code explained”. In: *World Pumps* 2009.11 (2009). URL: <https://www.worldpumps.com/>. (accessed: 21.01.20).
- [96] Visual Paradigm. *Failure Mode and Effects Analysis (FMEA)*. URL: <https://online.visual-paradigm.com/de/tabular/templates/failure-mode-and-effects-analysis-fmea/>. (accessed: 20.01.20).
- [97] G. Pardo-Castellote. *OMG Data-Distribution Service: architectural overview*. 2003. URL: <https://ieeexplore.ieee.org/document/1203555>. (accessed: 24.03.2020).
- [98] *PETG Technical Data Sheet*. Standardized Distributed 3D printing, SD3D.
- [99] Alexis Pey. “Reviewing the Risk Acceptance Criteria in ATEX.” In: *CHEMICAL ENGINEERING TRANSACTIONS* 77 (2019), pp. 199–204. DOI: 10.3303/CET1977034.
- [100] M. Poorani and R. Kurunjimalar. “Design implementation of UART and SPI in single FGPA”. In: *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. 2016, pp. 1–5.
- [101] Protonex. *What do soldiers carry and what does it weigh?* URL: <http://www.ptxnomad.com/what-do-soldiers-carry-and-what-does-it-weigh/%7D>. (accessed: 21.01.20).
- [102] QT. *Adding OpenSSL Support for Android*. URL: <https://doc.qt.io/qt-5/android-openssl-support.html>. (accessed: 15.03.2020).



## 7. REFERENCES

- [103] QT. *Annotated URL Example*. URL: <https://doc.qt.io/qt-5/qtnfc-annotatedurl-example.html>. (accessed: 02.03.2020).
- [104] QT. *Creating Project Files*. URL: <https://doc.qt.io/qt-5/qmake-project-files.html>. (accessed: 02.03.2020).
- [105] QT. *QAbstractListModel Class*. URL: <https://doc.qt.io/qt-5/qabstractlistmodel.html>. (accessed: 16.03.2020).
- [106] QT. *QGeoCoordinate Class*. URL: <https://doc.qt.io/qt-5/qgeocoordinate.html>. (accessed: 16.03.2020).
- [107] QT. *QNearFieldTarget Class*. URL: <https://doc.qt.io/qt-5/qnearfieldtarget.html>. (accessed: 16.03.2020).
- [108] QT. *Qt Location*. URL: <https://doc.qt.io/qt-5/qtlocation-index.html>. (accessed: 15.03.2020).
- [109] QT. *Qt Location Open Street Map Plugin*. URL: <https://doc.qt.io/qt-5/location-plugin-osm.html>. (accessed: 15.03.2020).
- [110] QT. *QT NFC*. URL: <https://doc.qt.io/qt-5/qtnfc-index.html>. (accessed: 02.03.2020).
- [111] QT. *QT NFC Overview*. URL: <https://doc.qt.io/qt-5/qtnfc-overview.html>. (accessed: 02.03.2020).
- [112] QT. *Signals & Slots*. URL: <https://doc.qt.io/qt-5/signalsandslots.html>. (accessed: 26.02.2020).
- [113] Qt. *About Qt*. URL: [https://wiki.qt.io/About\\_Qt](https://wiki.qt.io/About_Qt). (accessed: 23.03.2020).
- [114] Qt. *Design, Develop & Deploy User Interfaces and Applications*. URL: <https://www.qt.io/product>. (accessed: 24.05.2020).
- [115] Qt. *Implementing Model/View/Controller*. URL: <https://doc.qt.io/archives/qq/qq10-mvc.html>. (accessed: 24.05.2020).
- [116] Qt. *Qt Documentation*. URL: <https://doc.qt.io/>. (accessed: 24.05.2020).
- [117] Qt. *Qt Features*. URL: <https://www.qt.io/product/features>. (accessed: 24.05.2020).
- [118] Qt. *Qt for Beginners*. URL: [https://wiki.qt.io/Qt\\_for\\_Beginners](https://wiki.qt.io/Qt_for_Beginners). (accessed: 24.05.2020).
- [119] Qt. *Qt QML*. URL: <https://doc.qt.io/qt-5/qtqml-index.html>. (accessed: 24.05.2020).
- [120] Qt. *QUdpSocket*. URL: <https://doc.qt.io/qt-5/qudpsocket.html>. (accessed: 23.03.2020).

## 7. REFERENCES

- [121] Y.Dilli Rao et al. “Radar Cross Section Modeling and Measurement of Electric Detonators”. In: *2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing* (2017). DOI: 978-1-5090-4778-9/17. (accessed: 15.04.20).
- [122] *RayTech Wondergel Invisible Data Sheet*. RayTech Cable Accessories Company. URL: <https://www.raytech.it/en/product/low-voltage/fillers/gel/wonder-gel>.
- [123] *RMCF / RMCP Series datasheet*. Stackpole Electronics, Inc., 2020.
- [124] Rubycon. *RADIAL LEAD ALUMINUM ELECTROLYTIC CAPACITORS*. URL: [http://www.rubycon.co.jp/en/catalog/e\\_pdfs/aluminum/e\\_BXW.pdf](http://www.rubycon.co.jp/en/catalog/e_pdfs/aluminum/e_BXW.pdf). (accessed: 20.05.20).
- [125] *S4N1RP datasheet*. Littelfuse, 2010.
- [126] K. V. S. S. S. Sairam, N. Gunasekaran, and S.R. Redd. *Bluetooth in wireless communication*. 2002. URL: <https://ieeexplore.ieee.org/document/1007414>. (accessed: 22.03.2020).
- [127] Ken Schwaber and Jeff Sutherland. *The Scrum Guide (November 2017)*. URL: <https://www.scrumguides.org/scrum-guide.html>. (accessed: 17.01.20).
- [128] UNMAS (United Nations Mine Action Service). *United Nations Improvised Explosive Device Disposal Standards*. 2018. URL: <https://reliefweb.int/sites/reliefweb.int/files/resources/UN%5C%20IEDD%5C%20Standards.pdf>. (accessed: 04.03.2020).
- [129] Cathleen Shamieh. *Systems Engineering FOR DUMmIES IBM LIMITED EDITION*. Wiley Publishing, 2011.
- [130] G. Simons and I. Chifu. *The Changing Face of Warfare in the 21st Century*. Routledge, 2018.
- [131] Spectac. *Frontpage*. URL: <https://www.spectac.no/>. (accessed: 06.01.20).
- [132] *STLM20 datasheet*. STMicroelectronics, 2010.
- [133] STMicroelectronics. *STWD100 Watchdog timer circuit*. URL: <https://www.st.com/content/ccc/resource/technical/document/datasheet/06/6a/b3/83/9a/c7/4f/22/CD00176077.pdf/files/CD00176077.pdf/jcr:content/translations/en.CD00176077.pdf>. (accessed: 20.05.20).
- [134] Michael Taylor. “Improved EN-60529 Dust Chamber for IP5x or IP6x Enclosure Testing.” In: *2007 IEEE Symposium on Product Compliance Engineering UN.UN* (2007), p. 6. DOI: 10.1109/PSES.2007.4378482.
- [135] GitHub user TMRh20. *RF24*. <https://github.com/nRF24>. 2014.



- [136] *Transmission Control Protocol*. RFC 793. Sept. 1981. DOI: 10.17487/RFC0793. URL: <https://rfc-editor.org/rfc/rfc793.txt>.
- [137] *W5500 Datasheet*. Vol. 1.0.9. WIZnet Co., Ltd., 2019.
- [138] H. Wadhwa and R. Aron. “Fog Computing with the Integration of Internet of Things: Architecture, Applications and Future Directions”. In: *2018 IEEE Intl Conf on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/Sustain-Com)*. 2018, pp. 987–994. DOI: 10.1109/BDCloud.2018.00144.
- [139] Mr. Suyog A. Wani and Prof. R.P.Chaudhari. *Ethernet Enabled Digital I/O Control in Embedded Systems*. International Conference on Computing, Electronics and Electrical Technologies, 2012.
- [140] Wireshark. *Wireshark Network Protocol Analyzer*. URL: <https://www.wireshark.org/>. (accessed: 24.05.2020).
- [141] Wireshark. *Wireshark Network Protocol Analyzer*. URL: <https://www.wireshark.org/>. (accessed: 24.05.2020).
- [142] Feifei Wang Ximing Liu and Bowen Wu. *A Remote Monitoring Method of Radio based on W5500*. Fifth International Conference on Instrumentation, Measurement, Computer, Communication, and Control, 2015.
- [143] K. Zaerens. “Enabling the Benefits of Cloud Computing in a Military Context”. In: *2011 IEEE Asia-Pacific Services Computing Conference*. 2011, pp. 166–173. DOI: 10.1109/APSCC.2011.42.

## 8 Appendices

### Contents

8.1	List of Requirements . . . . .	239
8.2	FMEA V7.0 . . . . .	319
8.3	Test reports . . . . .	333
8.4	Economy attachments . . . . .	352
8.5	Prototype 1.0 code . . . . .	360
8.6	Prototype 2.0 code . . . . .	362
8.7	Software Architectural Model . . . . .	366
8.8	Receiver IP Simulation . . . . .	369
8.9	Sequence Diagram Pairing . . . . .	372
8.10	Sequence Diagram Tactical Operations . . . . .	373
8.11	Sequence Diagram Heartbeat . . . . .	374
8.12	Material Test Report (ILSS and Tensile Properties) . . . . .	375
8.13	Material Test Results of Assembly . . . . .	432
8.14	Bill of Material . . . . .	434
8.15	Gantt diagram . . . . .	437
8.16	SPARK PCB Schematics . . . . .	440

## List of Requirements

<b>CR-001: Interaction between the system and Samsung mobile</b>	243
SR-001: Communication between system and operator	244
SW-001: Android operating system	245
SW-002: Network communication through Wi-Fi	245
EL-001: The receiver must have a Ethernet communication	246
EL-004: The receiver must support IEEE 802.11 connectivity	246
EL-005: The receiver hardware should support Wi-Fi Direct	247
EL-015: Equal or greater than 10 Mbits/s	247
EL-007: Logic level shifter ATWINC1500 threshold	248
EL-008: Logic level shifter ATmega2560 threshold	248
EL-009: ATmega2560 clock frequency	249
EL-010: W5500 oscillator frequency tolerance	249
SR-002: Different states of the receiver	250
SW-023: Show different receiver states	250
SR-003: Ability to arm the receiver	251
SW-024: Arming functionality	251
SR-004: Detonation of the receiver	252
SW-025: Detonation functionality	252
SR-005: Abort detonation	253
SW-026: Abort functionality	253
SR-006: Location of the receiver	254
SW-003: Fixed zoom position for map	254
SW-004: Persistent waypoint position	255
SW-005: Waypoint state compliant to receiver state	255
SW-009: Add Multiple Waypoints	256
SW-010: Change Multiple Waypoints	256
SR-007: Android operating system	257
SR-008: Locations of all deployed receivers	258
SW-027: Location of receiver	258
SR-009: Communicate with other receivers	259
SR-010: NFC authentication	260
SW-006: Handling of pairing requests	260
SW-007: Accepted NFC types	261
SW-008: NFC range	261
SR-011: Send device key	262
SR-012: Configure the receiver	263
SR-046: Software safety barriers	264

	SW-028: Safety barriers for arming . . . . .	265
	SW-029: Safety barriers for detonation . . . . .	265
	SR-047: Network Communication . . . . .	266
	SW-011: IP Network Communication . . . . .	266
	SW-012: IP Network Communication . . . . .	267
	SW-013: IP Network Communication . . . . .	267
	SW-014: IP Network Communication . . . . .	268
	SW-015: IP Network Communication . . . . .	268
	SW-016: IP Network Communication . . . . .	269
	SW-017: IP Network Communication . . . . .	269
	SW-018: IP Network Communication . . . . .	270
	SW-019: Detonate command Verification . . . . .	270
	SW-020: Arm command verification . . . . .	271
	SW-021: Disarm command verification . . . . .	271
	SW-022: System Communication via App . . . . .	272
	SR-048: Failover mechanisms . . . . .	273
<b>CR-002: Ignite two individual nonel tubes simultaneously</b>		274
	SR-013: Ignite two plasma igniters simultaneously . . . . .	275
	EL-002: Deliver energy . . . . .	275
	SR-014: Physical safety barrier . . . . .	276
	EL-012: Switch toggle self safety check . . . . .	276
	SR-015: Authentication method . . . . .	277
	SR-016: Independent arm functionality . . . . .	277
	SR-017: Abort instruction . . . . .	278
	EL-014: Discharge time when disarming . . . . .	278
	SR-018: Energy for each plasma igniter . . . . .	279
	EL-003: Accumulate energy . . . . .	279
	EL-011: Peak current through plasma igniter . . . . .	280
	SR-019: Status of the energy containers . . . . .	280
<b>CR-003: CR123 battery</b>		281
	SR-020: Compartment to fit CR123 . . . . .	282
	SR-021: Compensate for voltage change . . . . .	282
	SR-023: Battery change routine . . . . .	283
<b>CR-004: Water resistant</b>		284
	SR-024: Battery compartment cover . . . . .	285
	SR-025: Water resistance . . . . .	285
	ME-001: Physical moist barriers . . . . .	286
	ME-002: IP Standard . . . . .	286
	SR-026: Gaskets . . . . .	287

<b>CR-005: Robustness and lightweight receiver</b>	288
SR-027: Robust printable container	289
ME-003: Robust and printable	289
SR-028: Shock resistant container	290
ME-004: Shock resistance	290
SR-044: Device weight	291
ME-005: Total weight	291
<b>CR-006: Fit military pockets</b>	292
SR-029: Small designed	293
ME-006: Device size	293
SR-030: Ergonomically designed	294
ME-007: User friendly design	294
<b>CR-007: Preconfigure the system</b>	295
SR-031: Predefined location	296
SR-032: Auto-reconnect	296
SR-033: Safe to carry	297
<b>CR-008: Operate the system when I am stressed and in short of time</b>	298
SR-035: Handbook	299
SR-036: Simple UI	299
SR-037: Colors on user interface	300
SR-038: UI big buttons	300
SR-043: Pair receiver and app	301
<b>CR-009: Easily understand the arm-state</b>	302
SR-039: The receiver diodes shall be dimmed	303
SR-040: Arm led	303
<b>CR-010: On/off mechanism</b>	304
SR-041: Two-step on/off mechanism	305
<b>CR-011: Schedule delayed detonation</b>	306
SR-042: Schedule detonations	307
<b>CR-013: Connection through obstacles</b>	308
<b>CR-014: Connection in open landscape</b>	309
<b>CR-016: Replace plasma igniter</b>	310
<b>CR-017: Fixing possibilities</b>	311
SR-034: Fast and easy placement	312
EL-013: Switch toggle self safety check	312
ME-008: Fixing to any surface	313
SR-045: Hanging bracket	313
ME-009: Suspension	314
<b>CR-018: Battery status of receiver</b>	315

---

SR-022: Monitor the battery . . . . .	316
<b>CR-019: Use in all weather conditions . . . . .</b>	<b>317</b>
SR-049: Estimate battery capacity . . . . .	318
EL-006: The receiver must have a temperature sensor . . . . .	318

## Requirement CR-001:

REQ-1	Interaction between the system and Samsung mobile				
<b>ID:</b> CR-001	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want to be able to interact with the system through an in-kit Samsung mobile phone.				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> The application works on a Samsung device</li> <li><input checked="" type="checkbox"/> The application works on an Android operating system</li> <li><input checked="" type="checkbox"/> The application is able to connect to the receiver</li> <li><input type="checkbox"/> The application is able to configure the receiver</li> <li><input checked="" type="checkbox"/> The application is able to interact with the receiver</li> <li><input checked="" type="checkbox"/> The application is able to abort a detonation</li> <li><input checked="" type="checkbox"/> The application displays a map of receivers</li> <li><input type="checkbox"/> The connection between receiver and application is reliable</li> <li><input type="checkbox"/> The application can transmit a master code to another operator</li> </ul>				
<b>Relation</b>	<u>SR-001</u> , <u>SR-002</u> , <u>SR-003</u> , <u>SR-004</u> , <u>SR-005</u> , <u>SR-006</u> , <u>SR-007</u> , <u>SR-008</u> , <u>SR-009</u> , <u>SR-010</u> , <u>SR-011</u> , <u>SR-012</u> , <u>SR-032</u> , <u>SR-035</u> , <u>SR-036</u> , <u>SR-037</u> , <u>SR-038</u>				
<b>Risk ID</b>	RAT-007, RAT-011, RAT-023				
<b>Changelog</b>	<ul style="list-style-type: none"> <li>- 12.03.2020 (CR-001): Added more acceptance criterias</li> <li>- 14.03.2020 (SR-001): Renamed system requirement</li> </ul>				

**Requirement SR-001:**

REQ-2	Communication between system and operator				
<b>ID:</b> SR-001	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u> <u>CR-011</u>	<b>Priority</b> Highest	<b>Version</b> 1.1
<b>Description</b>	The system shall communicate with the operator through an Android mobile application.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> <u>SW-001</u> has been fulfilled <input type="checkbox"/> <u>SW-002</u> has been fulfilled <input type="checkbox"/> <u>EL-001</u> has been fulfilled <input type="checkbox"/> <u>EL-004</u> has been fulfilled <input type="checkbox"/> <u>EL-005</u> has been fulfilled <input type="checkbox"/> <u>EL-015</u> has been fulfilled <input type="checkbox"/> <u>EL-007</u> has been fulfilled <input type="checkbox"/> <u>EL-008</u> has been fulfilled <input type="checkbox"/> <u>EL-009</u> has been fulfilled <input type="checkbox"/> <u>EL-010</u> has been fulfilled				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-032				



**Requirement SW-001:**

REQ-3	Android operating system				
<b>ID:</b> SW-001	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The mobile application must work on an Android operating system.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-002:**

REQ-4	Network communication through Wi-Fi				
<b>ID:</b> SW-002	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u>	<b>Priority</b> Normal	<b>Version</b> 1.0
<b>Description</b>	The mobile application must be able to communicate with the receiver through a Wi-Fi connection.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement EL-001:**

REQ-5	The receiver must have a Ethernet communication				
<b>ID:</b> EL-001	<b>Date</b> 16.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The receiver must have a RJ45 interface that supports Ethernet (IEEE 802.3) communication				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement EL-004:**

REQ-6	The receiver must support IEEE 802.11 connectivity				
<b>ID:</b> EL-004	<b>Date</b> 03.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The receiver must support one of the IEEE 802.11 a/b/g/n/ac standards				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement EL-005:**

REQ-7	The receiver hardware should support Wi-Fi Direct				
<b>ID:</b> EL-005	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The receiver hardware should support Wi-Fi direct				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement EL-015:**

REQ-8	Equal or greater than 10 Mbits/s				
<b>ID:</b> EL-015	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The speed of the wired and wireless network communication must be equal to or greater than 10 Mbits/s				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement EL-007:**

REQ-9	Logic level shifter ATWINC1500 threshold				
<b>ID:</b> EL-007	<b>Date</b> 17.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The lower high threshold for the ATWINC1500 side of the logic level shifter must be between 2.7-3.3V				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement EL-008:**

REQ-10	Logic level shifter ATmega2560 threshold				
<b>ID:</b> EL-008	<b>Date</b> 17.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The lower high threshold for the ATmega2560 side of the logic level shifter must be between 4.2-5V				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement EL-009:**

REQ-11	ATmega2560 clock frequency				
<b>ID:</b> EL-009	<b>Date</b> 17.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The ATmega2560 must have a clock frequency of at least 16MHz				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement EL-010:**

REQ-12	W5500 oscillator frequency tolerance				
<b>ID:</b> EL-010	<b>Date</b> 03.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The external oscillator for the W5500 must be 25Mhz with a frequency tolerance of $\pm 30$ PPM				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-002:**

REQ-13	Different states of the receiver				
<b>ID:</b> SR-002	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u> <u>CR-009</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The mobile application shall show different modes/states of the receiver.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-033				

**Requirement SW-023:**

REQ-14	Show different receiver states				
<b>ID:</b> SW-023	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-002</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The overview page must show different states (deployed, armed, disarmed, etc.) for each receiver.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-003:**

REQ-15	Ability to arm the receiver				
<b>ID:</b> SR-003	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u> <u>CR-002</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The mobile application shall be able to arm the receiver.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-007, RAT-033				

**Requirement SW-024:**

REQ-16	Arming functionality				
<b>ID:</b> SW-024	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-003</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	There should be a button to arm a selected receiver.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-004:**

REQ-17	Detonation of the receiver				
<b>ID:</b> SR-004	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u> <u>CR-002</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The mobile application shall be able to detonate the receiver.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-021, RAT-033				

**Requirement SW-025:**

REQ-18	Detonation functionality				
<b>ID:</b> SW-025	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-004</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	There should be a button to detonate a selected receiver.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				



**Requirement SR-005:**

REQ-19	Abort detonation				
<b>ID:</b> SR-005	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The mobile application shall be able to abort the detonation of a receiver.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-033				

**Requirement SW-026:**

REQ-20	Abort functionality				
<b>ID:</b> SW-026	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-005</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	There should be a button to disarm/abort a selected receiver.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-006:**

REQ-21	Location of the receiver				
<b>ID:</b> SR-006	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The app shall be able to show location of the receiver on a map.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-034				

**Requirement SW-003:**

REQ-22	Fixed zoom position for map				
<b>ID:</b> SW-003	<b>Date</b> 18.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-006</u>	<b>Priority</b> Low	<b>Version</b> 1.0
<b>Description</b>	The map function shall remember its fixed position when entering and leaving the mapview of the application.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-004:**

REQ-23	Persistent waypoint position				
<b>ID:</b> SW-004	<b>Date</b> 18.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-006</u>	<b>Priority</b> Low	<b>Version</b> 1.0
<b>Description</b>	The waypoint shall remain at its deployed position when switching views in the application.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-005:**

REQ-24	Waypoint state compliant to receiver state				
<b>ID:</b> SW-005	<b>Date</b> 18.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-006</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	A deployed waypoint, shall at all times correspond to the receivers given state (STANDBY, ARMED, ERROR, DETONATED).				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-009:**

REQ-25	Add Multiple Waypoints				
<b>ID:</b> SW-009	<b>Date</b> 23.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-006</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	Should be able to deploy multiple waypoints on map				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-010:**

REQ-26	Change Multiple Waypoints				
<b>ID:</b> SW-010	<b>Date</b> 23.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-006</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	Should be able to change the state of a single waypoint, without altering the state of others in the list				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-007:**

REQ-27	Android operating system				
<b>ID:</b> SR-007	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The system's sender shall use an Android operating system to seamlessly integrate with current military technology.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-032				

**Requirement SR-008:**

REQ-28	Locations of all deployed receivers				
<b>ID:</b> SR-008	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The app shall display a list with all deployed receivers.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-006, RAT-011, RAT-012, RAT-013, RAT-014, RAT-015, RAT-018				

**Requirement SW-027:**

REQ-29	Location of receiver				
<b>ID:</b> SW-027	<b>Date</b> 23.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-008</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The map should show a waypoint for each deployed receiver				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-009:**

REQ-30	Communicate with other receivers				
<b>ID:</b> SR-009	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The app shall be able to communicate with other receivers in the network and take over control.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-006, RAT-011, RAT-012, RAT-013, RAT-014, RAT-015, RAT-018				

**Requirement SR-010:**

REQ-31	NFC authentication				
<b>ID:</b> SR-010	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The app shall be able to use NFC as authentication.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-006, RAT-018, RAT-027, RAI-004				

**Requirement SW-006:**

REQ-32	Handling of pairing requests				
<b>ID:</b> SW-006	<b>Date</b> 18.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-010</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	When in pairing mode, the app shall handle incoming pairing requests from receivers.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				



**Requirement SW-007:**

REQ-33	Accepted NFC types				
<b>ID:</b> SW-007	<b>Date</b> 18.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-010</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The app shall be able to read ISO/IEC14443-A, MIFARE, and ISO/IEC14443-B NFC types.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-008:**

REQ-34	NFC range				
<b>ID:</b> SW-008	<b>Date</b> 18.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-010</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The app shall be able to detect NFC tags within 3cm.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-011:**

REQ-35	Send device key				
<b>ID:</b> SR-011	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The app shall be able to send device key to another soldier.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-006, RAT-013, RAT-014, RAT-015, RAT-018, RAT-025, RAT-033				

**Requirement SR-012:**

REQ-36	Configure the receiver				
<b>ID:</b> SR-012	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u> <u>CR-007</u> <u>CR-008</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The app shall be able to configure the receiver wirelessly.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-006, RAT-013, RAT-014, RAT-015, RAT-018, RAT-025, RAT-027				

**Requirement SR-046:**

REQ-37	Software safety barriers				
<b>ID:</b> SR-046	<b>Date</b> 30.03.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	There shall be two software safety barriers to inhibit detonation or arming by accident.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-030				

**Requirement SW-028:**

REQ-38	Safety barriers for arming				
<b>ID:</b> SW-028	<b>Date</b> 18.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-046</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	There should be safety barriers to prevent the receiver from unintentionally arming itself.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-029:**

REQ-39	Safety barriers for detonation				
<b>ID:</b> SW-029	<b>Date</b> 18.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-046</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	There should be safety barriers to prevent the receiver from unintentionally detonating itself.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-047:**

REQ-40	Network Communication				
<b>ID:</b> SR-047	<b>Date</b> 30.03.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The App and Receiver devices should be able to use IP as an independent communication channel.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-038, RAT-039				

**Requirement SW-011:**

REQ-41	IP Network Communication				
<b>ID:</b> SW-011	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The app shall send a Pairing Response with necessary pairing data, when receiving a Pairing Request over IP.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-012:**

REQ-42	IP Network Communication				
<b>ID:</b> SW-012	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The App shall be able to receive and store updates on receivers status over IP.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-013:**

REQ-43	IP Network Communication				
<b>ID:</b> SW-013	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The App shall be able to send out an Standby command to a given receiver over IP.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-014:**

REQ-44	IP Network Communication				
<b>ID:</b> SW-014	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The App shall be able to send out a Arming command to a given receiver over IP.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-015:**

REQ-45	IP Network Communication				
<b>ID:</b> SW-015	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The App shall be able to send out a Detonation command to a given receiver over IP.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				



**Requirement SW-016:**

REQ-46	IP Network Communication				
<b>ID:</b> SW-016	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The App shall be able to handle multiple receiver connections simultaneously over IP.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-017:**

REQ-47	IP Network Communication				
<b>ID:</b> SW-017	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	Upon a Pairing Request sent to the App over IP the app shall create a connection and put it in a Pending State				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-018:**

REQ-48	IP Network Communication				
<b>ID:</b> SW-018	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	Upon a Pairing Acknowledge sent to the App over IP the App shall put the pre-existing pending connection in an active state, then create an instance of the Device to hold relevant data				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-019:**

REQ-49	Detonate command Verification				
<b>ID:</b> SW-019	<b>Date</b> 13.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	When a Detonate command is emitted from the app, the system will generate a warning if the Receiver does not acknowledge the command within a given timespace				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-020:**

REQ-50	Arm command verification				
<b>ID:</b> SW-020	<b>Date</b> 13.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	When an Arm command is emitted from the app, the system will generate a warning if the Receiver does not acknowledge the command within a given timespace				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-021:**

REQ-51	Disarm command verification				
<b>ID:</b> SW-021	<b>Date</b> 13.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	When a Disarm command is emitted from the app, the system will generate a warning if the Receiver does not acknowledge the command within a given timespace				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SW-022:**

REQ-52	System Communication via App				
<b>ID:</b> SW-022	<b>Date</b> 13.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-047</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The App should be able to receive and process heartbeat packets over IP from deployed Receiver devices, to provide the operator with updated context data.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-048:**

REQ-53	Failover mechanisms				
<b>ID:</b> SR-048	<b>Date</b> 30.03.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	There shall be an effective failover mechanisms to ensure control of all devices at all times.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-035				

**Requirement CR-002:**

REQ-54	Ignite two individual nonel tubes simultaneously				
<b>ID:</b> CR-002	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want to be able to safely use the receiver to ignite two individual nonel tubes simultaneously.				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Ignite two plasma igniter's simultaneously <input type="checkbox"/> Drain energy container when aborting detonation <input type="checkbox"/> Detonation barriers are tested and reliable <input type="checkbox"/> Barriers in terms of cyber security measures <input type="checkbox"/> Independent arm functionality for software and electrical subsystems				
<b>Relation</b>	<u>SR-003</u> , <u>SR-004</u> , <u>SR-013</u> , <u>SR-014</u> , <u>SR-015</u> , <u>SR-016</u> , <u>SR-017</u> , <u>SR-018</u> , <u>SR-019</u> , <u>SR-033</u> , <u>SR-0035</u>				
<b>Risk ID</b>	RAT-002, RAT-007, RAT-008, RAT-021, RAT-023				
<b>Changelog</b>	-				

**Requirement SR-013:**

REQ-55	Ignite two plasma igniters simultaneously				
<b>ID:</b> SR-013	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-002</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The system shall be able to ignite two plasma igniters simultaneously				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-002, RAT-007, RAT-008, RAT-021, RAT-023				

**Requirement EL-002:**

REQ-56	Deliver energy				
<b>ID:</b> EL-002	<b>Date</b> 18.02.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-013</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	Deliver 350V and 9J of energy to each plasma igniter				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-014:**

REQ-57	Physical safety barrier				
<b>ID:</b> SR-014	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-002</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The system shall have at least two physical safety barriers				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-029, RAT-030				

**Requirement EL-012:**

REQ-58	Switch toggle self safety check				
<b>ID:</b> EL-012	<b>Date</b> 05.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-014</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The discharge circuit needs to facilitate testing of the the discharge switches ability to toggle between OFF and ON state before charging the capacitors				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				



**Requirement SR-015:**

REQ-59	Authentication method				
<b>ID:</b> SR-015	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-002</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The system shall have a user authentication method that uses common cyber security measures				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-035, RAT-036				

**Requirement SR-016:**

REQ-60	Independent arm functionality				
<b>ID:</b> SR-016	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-002</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The system shall have an independent arm functionality for software and electrical subsystems				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-007, RAT-029, RAT-030				

**Requirement SR-017:**

REQ-61	Abort instruction				
<b>ID:</b> SR-017	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-002</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The system receiver should disarm safely if it receives an abort instruction				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> <u>EL-014</u> has been fulfilled				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-001, RAT-004, RAT-008, RAT-016, RAT-017, RAT-021, RAT-023, RAT-037				

**Requirement EL-014:**

REQ-62	Discharge time when disarming				
<b>ID:</b> EL-014	<b>Date</b> 17.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-017</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The discharge time when disarming must be no more than 20s				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-018:**

REQ-63	Energy for each plasma igniter				
<b>ID:</b> SR-018	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-002</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The system shall be able to accumulate the energy needed for each plasma igniter				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> <u>EL-001</u> has been fulfilled <input type="checkbox"/> <u>EL-002</u> has been fulfilled <input type="checkbox"/> <u>EL-011</u> has been fulfilled				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-004, RAT-007, RAT-008, RAT-010, RAT-011, RAT-016, RAT-017, RAT-023, RAT-036, RAT-037, RAE-001, RAE-002				

**Requirement EL-003:**

REQ-64	Accumulate energy				
<b>ID:</b> EL-003	<b>Date</b> 18.02.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-018</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	Accumulate 350V and 9J of energy for each capacitor				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement EL-011:**

REQ-65	Peak current through plasma igniter				
<b>ID:</b> EL-011	<b>Date</b> 17.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-018</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The discharge circuit must be able to allow at least 1A peak of current through the plasma igniter				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-019:**

REQ-66	Status of the energy containers				
<b>ID:</b> SR-019	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-002</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The system shall be able to sense the status of the energy containers				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-001, RAT-008, RAT-016, RAT-017				

**Requirement CR-003:**

REQ-67	CR123 battery				
<b>ID:</b> CR-003	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want to be able to power the receiver with a CR123 battery				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> CR123 powers the system <input type="checkbox"/> The system compensates for voltage change <input type="checkbox"/> The system monitors the battery status <input type="checkbox"/> No tools needed for battery change				
<b>Relation</b>	<u>SR-020</u> , <u>SR-021</u> , <u>SR-022</u> , <u>SR-023</u> , <u>SR-024</u>				
<b>Risk ID</b>	RAT-003, RAT-010, RAT-011, RAT-022, RAT-023, RAE-001, RAE-002, RAE-003				
<b>Changelog</b>	-				

**Requirement SR-020:**

REQ-68	Compartment to fit CR123				
<b>ID:</b> SR-020	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-003</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The system's receiver shall have a compartment to fit a single CR123 battery				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-009, RAT-010, RAT-011, RAT-020, RAT-022, RAT-023				

**Requirement SR-021:**

REQ-69	Compensate for voltage change				
<b>ID:</b> SR-021	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-003</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The system shall compensate for voltage change in battery				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-010, RAT-011, RAT-022, RAT-023				

**Requirement SR-023:**

REQ-70	Battery change routine				
<b>ID:</b> SR-023	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-003</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The system shall have a battery change routine without the need for tools				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-009, RAT-010, RAT-011, RAT-022, RAT-023				

**Requirement CR-004:**

REQ-71	Water resistant				
<b>ID:</b> CR-004	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want a water resistant receiver				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> The receiver has gaskets on every interface <input type="checkbox"/> The receiver fulfills IP65 standard <input type="checkbox"/> The receiver fulfills IP67 standard <input type="checkbox"/> The receiver fulfills IP68 standard				
<b>Relation</b>	<u>SR-024</u> <u>SR-025</u> <u>SR-026</u>				
<b>Risk ID</b>	RAT-004, RAT-008, RAT-012				
<b>Changelog</b>	-				



**Requirement SR-024:**

REQ-72	Battery compartment cover				
<b>ID:</b> SR-024	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-003</u> <u>CR-004</u> <u>CR-019</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The receiver shall have a water resistant battery compartment cover				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TCME-001-01				
<b>Risk ID</b>	RAT-008, RAT-012, RAT-016				

**Requirement SR-025:**

REQ-73	Water resistance				
<b>ID:</b> SR-025	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-004</u> <u>CR-019</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The receiver shall have no less than IP65 standard				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> <u>ME-001</u> has been fulfilled <input type="checkbox"/> <u>ME-002</u> has been fulfilled				
<b>Test ID</b>	TCME-002-01, TCME-002-02, TCME-002-03				
<b>Risk ID</b>	RAT-004, RAT-005, RAT-012				

**Requirement ME-001:**

REQ-74	Physical moist barriers				
<b>ID:</b> ME-001	<b>Date</b> 15.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-024</u> <u>SR-025</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The receiver must have gaskets on every interface.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement ME-002:**

REQ-75	IP Standard				
<b>ID:</b> ME-002	<b>Date</b> 15.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-024</u> <u>SR-025</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The device must be IP65 classified as a minimum				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-026:**

REQ-76	Gaskets				
<b>ID:</b> SR-026	<b>Date</b> 15.03.2020	<b>Status</b> Deleted	<b>Origin</b> <u>CR-004</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The receiver must have gaskets for every interface (this is converted into ME-001)				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TC-00X				
<b>Risk ID</b>	RAX-00X				

**Requirement CR-005:**

REQ-77	Robustness and lightweight receiver				
<b>ID:</b> CR-005	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want a solid lightweight receiver with a high degree of robustness				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> The receiver body is printable <input type="checkbox"/> The receiver is shock resistant <input type="checkbox"/> The receiver is lightweight				
<b>Relation</b>	<u>SR-027 SR-028</u>				
<b>Risk ID</b>	RAT-002, RAT-024, RAX-003, RAI-004, RAI-007				
<b>Changelog</b>	ASB: 15.03.20 Added Risk-ID to this CR and the coherent SRs				

**Requirement SR-027:**

REQ-78	Robust printable container				
<b>ID:</b> SR-027	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-005</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The system shall be protected by a robust printable container				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TCME-003-01				
<b>Risk ID</b>	RAT-002, RAT-024, RAX-003, RAI-007				

**Requirement ME-003:**

REQ-79	Robust and printable				
<b>ID:</b> ME-003	<b>Date</b> 15.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-027</u>	<b>Priority</b> Low	<b>Version</b> 1.0
<b>Description</b>	The device must be printable in 3D printer				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-028:**

REQ-80	Shock resistant container				
<b>ID:</b> SR-028	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-005</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The system shall be protected by a shock resistant container				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TCME-004-01				
<b>Risk ID</b>	RAT-002, RAT-024				

**Requirement ME-004:**

REQ-81	Shock resistance				
<b>ID:</b> ME-004	<b>Date</b> 15.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-028</u>	<b>Priority</b> Low	<b>Version</b> 1.0
<b>Description</b>	The device must withstand a shock wave of .....				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-044:**

REQ-82	Device weight				
<b>ID:</b> SR-044	<b>Date</b> 15.03.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-005</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The device shall be as lightweight as possible				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TCME-005-01				
<b>Risk ID</b>	RAT-020, RAI-004				

**Requirement ME-005:**

REQ-83	Total weight				
<b>ID:</b> ME-005	<b>Date</b> 15.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-044</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The device shall not weigh more than XX kg in total				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement CR-006:**

REQ-84	Fit military pockets				
<b>ID:</b> CR-006	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want the receiver to easily fit my pockets				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Relation</b>	<u>SR-029</u> <u>SR-030</u>				
<b>Risk ID</b>	RAX-003, RAI-004, RAI-005				
<b>Changelog</b>	ASB: 15.03.20 Added Risk-ID to this CR and the coherent SRs				



**Requirement SR-029:**

REQ-85	Small designed				
<b>ID:</b> SR-029	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-006</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The device shall be smaller than similar devices				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TCME-006-01				
<b>Risk ID</b>	RAT-020, RAX-003, RAI-004, RAI-005				

**Requirement ME-006:**

REQ-86	Device size				
<b>ID:</b> ME-006	<b>Date</b> 15.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-029</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The device must not be larger than 120 · 80 · 30 mm ( $L \cdot W \cdot H$ )				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-030:**

REQ-87	Ergonomically designed				
<b>ID:</b> SR-030	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-006</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The physical system shall be designed ergonomically				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TCME-007-01				
<b>Risk ID</b>	RAT-020, RAX-003, RAI-004, RAI-005				

**Requirement ME-007:**

REQ-88	User friendly design				
<b>ID:</b> ME-007	<b>Date</b> 15.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-030</u>	<b>Priority</b> Low	<b>Version</b> 1.0
<b>Description</b>	The device must be designed to be comfortable and manageable				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement CR-007:**

REQ-89	Preconfigure the system				
<b>ID:</b> CR-007	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want to be able to configure the system ready for use before spending time on object				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> The receiver auto-reconnects if connection is lost <input type="checkbox"/> The receiver remembers a predefined location <input type="checkbox"/> The receiver remembers preset configuration <input type="checkbox"/> The receiver is safe to carry with nonel tubes attached				
<b>Relation</b>	<u>SR-012</u> , <u>SR-031</u> , <u>SR-032</u> , <u>SR-033</u> , <u>SR-034</u> , <u>SR-042</u>				
<b>Risk ID</b>	RAT-012, RAT-018, RAT-024, RAT-025, RAT-026, RAT-027, RAT-001, RAT-007, RAT-008, RAT-029, RAT-030				
<b>Changelog</b>	BB: Added RAT-012, RAT-018, RAT-024, RAT-025, RAT-026, RAT-027, RAT-001, RAT-007, RAT-008, RAT-029, RAT-030 to risk.				

**Requirement SR-031:**

REQ-90	Predefined location				
<b>ID:</b> SR-031	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-007</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The system shall be able to be preconfigured with a predefined location				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-024, RAT-026				

**Requirement SR-032:**

REQ-91	Auto-reconnect				
<b>ID:</b> SR-032	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u> <u>CR-007</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The receiver shall auto-reconnect if the connection breaks				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-018, RAT-024,RAT-025, RAT-026, RAT-027				

**Requirement SR-033:**

REQ-92	Safe to carry				
<b>ID:</b> SR-033	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-002</u> <u>CR-007</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The receiver shall be safe to carry with nonel tubes attached without risk of delivering energy to plasma igniters unintentionally				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-001, RAT-007, RAT-008, RAT-029, RAT-030				

**Requirement CR-008:**

REQ-93	Operate the system when I am stressed and in short of time				
<b>ID:</b> CR-008	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want to be able to operate the system easily when I am stressed and in short of time				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> The system has usable handbook <input type="checkbox"/> The UI is simple and intuitive <input type="checkbox"/> The App is easy and intuitive to use <input type="checkbox"/> The App has clear colors to differentiate controls and settings				
<b>Relation</b>	<u>SR-012, SR-035, SR-036, SR-037, SR-038</u>				
<b>Risk ID</b>	RAT-018,RAT-027, RAT-015, RAT-007, RAT-008, RAT-003, RAT-009, RAT-019, RAT-020, RAT-005, RAT-028, RAX-003, RAI-004, RAI-003, RAI-002,RAT-001, RAT-006				
<b>Changelog</b>	BB: Added RAT-018,RAT-027, RAT-015,RAT-007,RAT-008, RAT-003, RAT-009, RAT-019, RAT-020, RAT-005, RAT-028, RAX-003, RAI-004, RAI-003, RAI-002,RAT-001, RAT-006 to risk.				

**Requirement SR-035:**

REQ-94	Handbook				
<b>ID:</b> SR-035	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u> <u>CR-002</u> <u>CR-008</u>	<b>Priority</b> Lowest	<b>Version</b> 1.0
<b>Description</b>	The system shall have a easy to use handbook				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAI-002, RAI-003, RAI-004				

**Requirement SR-036:**

REQ-95	Simple UI				
<b>ID:</b> SR-036	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-001</u> <u>CR-008</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The system shall have a simple and intuitive UI				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-009, RAT-019, RAT-020, RAT-028				

**Requirement SR-037:**

REQ-96	Colors on user interface				
ID: SR-037	Date	Status	Origin	Priority	Version
	24.02.2020	Active	<u>CR-001</u> <u>CR-008</u>	Medium	1.0
<b>Description</b>	The system shall use colors for enhancing contrast in the mobile app user interface				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-020				

**Requirement SR-038:**

REQ-97	UI big buttons				
ID: SR-038	Date	Status	Origin	Priority	Version
	24.02.2020	Active	<u>CR-008</u>	Medium	1.0
<b>Description</b>	The system shall have big buttons that is easily accessed even though the hands of the soldier is shaking.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-019, RAT-020				



**Requirement SR-043:**

REQ-98	Pair receiver and app				
<b>ID:</b> SR-043	<b>Date</b> 04.03.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-008</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The system shall easily and quickly pair receiver and app				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-007, RAT-027				

**Requirement CR-009:**

REQ-99	Easily understand the arm-state				
<b>ID:</b> CR-009	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want to easily understand the arm-state of the receiver				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> One of the receiver leds is dedicated to indicate armed status <input type="checkbox"/> The receiver leds are dimmed, not too bright for operative use				
<b>Relation</b>	<u>SR-002</u> , <u>SR-039</u> , <u>SR-040</u>				
<b>Risk ID</b>	RAT-005, RAT-020, RAT-028				
<b>Changelog</b>	BB: Added Risks Rat-005,020,028				

**Requirement SR-039:**

REQ-100	The receiver diodes shall be dimmed				
<b>ID:</b> SR-039	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-009</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The receiver diodes shall be dimmed				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-028				

**Requirement SR-040:**

REQ-101	Arm led				
<b>ID:</b> SR-040	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-009</u>	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	The system receiver shall employ one visible LED to show if the receiver is armed or not				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-005, RAT-020				

**Requirement CR-010:**

REQ-102	On/off mechanism				
<b>ID:</b> CR-010	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want the receiver to have a definite on/off mechanism				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> The receiver has a two step switch to turn on the receiver				
<b>Relation</b>	<u>SR-041</u>				
<b>Risk ID</b>	RAT-001, RAT-011				
<b>Changelog</b>	-				

**Requirement SR-041:**

REQ-103	Two-step on/off mechanism				
<b>ID:</b> SR-041	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-010</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The system shall have a two-step on/off mechanism to prevent accidental detonation				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-001				

**Requirement CR-011:**

REQ-104	Schedule delayed detonation				
ID: CR-011	Date	Status	Origin	Priority	Version
	24.02.2020	Active	SPECTAC	Low	1.0
<b>Description</b>	As a special forces soldier I want to be able to schedule delayed detonation until 60 minutes after receiver configuration				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> User interface that supports scheduling of delayed detonations <input type="checkbox"/> The user interface displays the time until detonation <input type="checkbox"/> The user interface has an panel to change the detonation scheme				
<b>Relation</b>	<u>SR-042</u>				
<b>Risk ID</b>	RAT-001, RAT-022, RAI-004				
<b>Changelog</b>	Tested and Verified				

**Requirement SR-042:**

REQ-105	Schedule detonations				
<b>ID:</b> SR-042	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-001</u> , <u>CR-007</u> , <u>CR-011</u>	<b>Priority</b> Low	<b>Version</b> 1.0
<b>Description</b>	The system shall be able to schedule detonations up until 60 min				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-001, RAT-022				

**Requirement CR-013:**

REQ-106	Connection through obstacles				
<b>ID:</b> CR-013	<b>Date</b> 24.02.2020	<b>Status</b> Passive	<b>Origin</b> SPECTAC	<b>Priority</b> Lowest	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want to be able to have a connection with the receiver through 2-3 wooden walls				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> The device can transmit through at least two wooden walls				
<b>Relation</b>	-				
<b>Risk ID</b>	RA-00X				
<b>Changelog</b>	MB: Set requirement to Passive since it is not properly defined				



**Requirement CR-014:**

REQ-107	Connection in open landscape				
<b>ID:</b> CR-014	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> Low	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want to be able to have a connection with the receiver at 200 meters in open landscape				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> The system transmits at 200 meters in open landscape				
<b>Relation</b>	-				
<b>Risk ID</b>	RA-00X				
<b>Changelog</b>	-				

**Requirement CR-016:**

REQ-108	Replace plasma igniter				
<b>ID:</b> CR-016	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> High	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want the plasma igniter to easily be replaceable without special tools				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> The plasma igniter can be replaced without special tools				
<b>Relation</b>	-				
<b>Risk ID</b>	RA-00X				
<b>Changelog</b>	-				

**Requirement CR-017:**

REQ-109	Fixing possibilities				
<b>ID:</b> CR-017	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want to attach the receiver to a surface.				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> The device has a hoop on top that can hold the total weight of the device. <input type="checkbox"/> The device has a solid adhesive solution on the back that can hold the total weight of the device.				
<b>Relation</b>	<u>SR-034</u> <u>SR-045</u>				
<b>Risk ID</b>	RAT-020, RAX-003, RAI-004, RAI-005				
<b>Changelog</b>	ASB: 16.03.20 Added acceptance criteria				

**Requirement SR-034:**

REQ-110	Fast and easy placement				
<b>ID:</b> SR-034	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-017</u>	<b>Priority</b> Low	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want the receiver to have a easy and manageable adhesive solution on the back.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TCME-008-01				
<b>Risk ID</b>	RAT-020, RAX-003, RAI-004, RAI-005				

**Requirement EL-013:**

REQ-111	Switch toggle self safety check				
<b>ID:</b> EL-013	<b>Date</b> 05.05.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-014</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The discharge circuit needs to facilitate testing of the the discharge switches ability to toggle between OFF and ON state before charging the capacitors				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement ME-008:**

REQ-112	Fixing to any surface				
<b>ID:</b> ME-008	<b>Date</b> 15.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-034</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The device attaches to any chosen surface				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement SR-045:**

REQ-113	Hanging bracket				
<b>ID:</b> SR-045	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-017</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want a hoop on the receiver.				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TCME-00X-01				
<b>Risk ID</b>	RAT-020, RAX-003, RAI-005, RAI-005				

**Requirement ME-009:**

REQ-114	Suspension				
<b>ID:</b> ME-009	<b>Date</b> 15.03.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-045</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The device can be hung by a wire or hook.				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

**Requirement CR-018:**

REQ-115	Battery status of receiver				
<b>ID:</b> CR-018	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> SPECTAC	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	As a special forces soldier I want to be able to know the battery status of the receiver				
<b>Type</b>	Customer requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Battery percentage is clearly visible to the operator on the receiver. <input type="checkbox"/> Battery percentage is clearly visible to the operator in the application. <input type="checkbox"/> The operator is notified in the app if the battery status of the receiver is critical.				
<b>Relation</b>	<u>SR-022</u>				
<b>Risk ID</b>	RAT-005, RAT-015, RAT-018, RAT-023				
<b>Changelog</b>	Tested and Verified				

**Requirement SR-022:**

REQ-116	Monitor the battery				
<b>ID:</b> SR-022	<b>Date</b> 24.02.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-003</u> <u>CR-018</u>	<b>Priority</b> Medium	<b>Version</b> 1.0
<b>Description</b>	The system shall monitor the status of the battery				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RAT-003, RAT-010, RAT-011, RAT-022, RAT-023				



**Requirement CR-019:**

REQ-117	Use in all weather conditions				
ID: CR-019	Date	Status	Origin	Priority	Version
	24.02.2020	Active	SPECTAC	Medium	1.0
Description	As a special forces soldier I want to be able to use the system in all weather conditions				
Type	Customer requirement				
Acceptance Criteria	<input type="checkbox"/> The system must work normally in temperature ranges from -40 to 60 degrees Celsius. <input type="checkbox"/> The system must work normally when subjected to heavy rain. <input type="checkbox"/> The system must work normally when covered with snow.				
Relation	<u>SR-024</u> , <u>SR-025</u>				
Risk ID	RAT-003, RAT-004, RAT-006, RAT-012, RAT-023, RAT-024, RAT-025				
Changelog	Tested and Verified				

**Requirement SR-049:**

REQ-118	Estimate battery capacity				
<b>ID:</b> SR-049	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>CR-019</u>	<b>Priority</b> Highest	<b>Version</b> 1.1
<b>Description</b>	The receiver must be able to know estimate its remaining battery capacity				
<b>Type</b>	System requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> <u>EL-006</u> has been fulfilled				
<b>Test ID</b>	TEST-ID				
<b>Risk ID</b>	RISK-ID				

**Requirement EL-006:**

REQ-119	The receiver must have a temperature sensor				
<b>ID:</b> EL-006	<b>Date</b> 24.04.2020	<b>Status</b> Active	<b>Origin</b> <u>SR-049</u>	<b>Priority</b> Highest	<b>Version</b> 1.0
<b>Description</b>	The receiver must have a temperature sensor with an accuracy of minimum +-5 degrees Celsius				
<b>Type</b>	Technical requirement				
<b>Acceptance Criteria</b>	<input type="checkbox"/> Tested and Verified				

## 8.2 FMEA V7.0

The risk process is an ongoing process throughout the entire project, and we will address these 4 steps in every sprint:

- Identification: locate, register and characterizing risks
- Analysing: comprehending the level and the nature of the risk
- Evaluation: to determine the severity and possible occurrence of the risk, and then prioritize it.
- Control/testing: to monitor, implement decisions, re-evaluate

Failure Mode and Effects Analysis (FMEA)

		Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy								
		Responsible: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20		(Rev.): Version 7.0						
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken		
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?		What can cause the failure?	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?		
<b>Technical Risks</b>												
22.01.20	RAT-001	SR-017, SR-019, SR-033, CR-010, CR-011	Premature detonation	Multiple casualties and/or death	10	No safety functionality in place for arming and detonating a receiver		Make sure all test cases are completed and approved. Make sure action is taken when test case is not approved	Development team	Electrical: Watchdog enabled trigger system with double signal logic. Discharge switches self test before charging capacitors. At least two HW barriers before detonation		Severity
22.01.20	RAT-002	SR-027, SR-028	Shock to receiver	Non functional product that could lead to loss of allied lifes	7	Insufficient material research, inadequate container unit design	TCME-004-01	Change device design, change materials	The mechanical engineer	Made alternative casing with epoxy matrix, enhanced with SWCNT and quadraxial fiberglass		Fatal
22.01.20	RAT-003	SR-008, SR-009, SR-010, SR-011, SR-012, SR-018, SR-020, SR-021, SR-022, SR-023, SR-036, CR-003, CR-018, CR-019	Device is non-responsive	Operator will not be able to use the system effectively	6	Faulty Wiring, Faulty connectors, Wrong Software Logic, Weak computation power	TCSW016-01, TCSW017-01, TCSW018-01, TCSW019-01, TCSW020-01, TCSW021-01	Data: Ensure that the connection info is saved, so that it is possible to re-connect to the same receiver again	Development team			5-6 Detrimental

Failure Mode and Effects Analysis (FMEA)

Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy								
Responsible: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20								
		(Rev.): Version 7.0								
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?		What can cause the failure?	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?
22.01.20	RAT-004	SR-017, SR-018, CR-004, CR-019	The device circuit board is shorted	System will not be in use	6	Water or moisture in the unit	TCME-002-01, TCME-002-02, TCME-002-03	Change device design, change materials, change gasket size	The mechanical engineer	All critical functionality is set to ground in default state
22.01.20	RAT-005	SR-025, SR-040, CR-018	LED lights are malfunctioning	System signalling will not be compliant to system protocol	4	Water or moisture in the unit. Physical stress to the LEDs	TCME-002-01, TCME-002-02, TCME-002-03	Redesign device, change materials, change gasket size	The mechanical engineer	led are covered with resin on top and mounted with gasket
22.01.20	RAT-006	SR-008, SR-009, SR-010, SR-011, SR-012, SR-043, CR-005, CR-014, CR-019	Insufficient connectivity between receiver and transmitter	Operators will not get real time relevant data, can be fatal in operative use	6	Network error	TCEL-001-01, TC SW016-01, TC SW017-01, TC SW018-01, TC SW019-01, TC SW020-01, TC SW021-01, TC SW012-01, TC SW011-01	Ensure that the connection is established between the sender and receiver before the mission	The computer engineers	Re-establish the connection between sender and receiver
22.01.20	RAT-007	SR-003, SR-013, SR-016, SR-018, SR-033, SR-043, CR-001, CR-002, CR-008	Arm function malfunction	Multiple casualties and/or death	8	Not having independent arm functions	TCEL-002-01, TCEL-003-01		Development team	Electrical: Watchdog enabled trigger system with dubbel signal logic

Failure Mode and Effects Analysis (FMEA)

Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy								
Responsible: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20								
		(Rev.): Version 7.0								
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?		What can cause the failure?	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?
22.01.20	RAT-008	SR-013, SR-017, SR-018, SR-019, SR-024, SR-033, CR-002, CR-004, CR-008	Accidental capacitor discharge	Multiple casualties and/or death	8	Circuit board malfunction	TCME-001-01, TCME-002-01, TCME-002-02, TCME-002-03	Redesign battery compartment	The electrical engineers	Electrical: Watchdog enabled trigger system with double signal logic. Discharge switches self test before charging capacitors. At least two HW barriers before detonation
22.01.20	RAT-009	SR-020, SR-023, SR-036	Operator unable to change batteries	Non functional product that could lead to loss of allied lifes	6	Inadequate design for battery compartment	TCME-00X-01		The mechanical engineer	Battery lid can be opened without the use of any tools
22.01.20	RAT-010	SR-018, SR-020, SR-021, SR-022, SR-023, CR-003	Device will not power on before operation	Non functional product	2	Device drains power when not in use. Battery is empty. Error ! circuit board		1: Development team need to make sure nothing drains power when device is turned off. 2: A user manual that states that all devices need a status check before operation. 3: Change battery.	The electrical engineers	Physical detachment of battery in power off mode

Failure Mode and Effects Analysis (FMEA)

Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20		(Rev.):		Version 7.0		
Responsible: Anne Synnøve Brendøy <td colspan="2"></td> <td colspan="2"></td> <td colspan="2"></td> <td colspan="2"></td>										
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken
22.01.20	RAT-011	SR-008, SR-009, SR-018, SR-020, SR-021, SR-022, SR-023, CR-001, CR-003, CR-010	Device will not power on during operation	Non functional product that could lead to loss of allied lifes	6	Device drains power when not in use. Battery is empty.	TCME-001-01, TCME-002-01, TCME-002-02, TCME-002-03	Redesign battery compartment, change gasket	The mechanical engineer	Gasket on every interface and edges sealed with Araldite
22.01.20	RAT-012	SR-025, CR-004, CR-019	Takes in water/moisture	Electronics and/or components may malfunction	6	Defective compartment/ container. Defect gasket.				
22.01.20	RAT-013	SR-008, SR-009, SR-011, SR-012	Software Memory Leak	System Crash	8	Extensive Heap usage without freeing memory after		Make sure to free memory during runtime, avoid dangling references	The computer engineers	Restart the mobile application if it crashes during runtime
22.01.20	RAT-014	SR-008, SR-009, SR-011, SR-012	Faulty Logic in Software	System Crash or Undefined Behavior	8	Wrong logic when programming		Perform tests to check the program logic and to ensure that it works as intended	The computer engineers	
22.01.20	RAT-015	SR-008, SR-009, SR-011, SR-012, CR-017, CR-018, CR-008	Network Error	No possible communication between subsystems or operators	6	Problem with NIC or Network Protocol. Carrier network not available	TCEL-001-01		The computer engineers	Two independent network carriers

Failure Mode and Effects Analysis (FMEA)

Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy								
Responsible: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20								
		(Rev.): Version 7.0								
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?		What can cause the failure?	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?
22.01.20	RAT-016	SR-017, SR-018, SR-019	Hardware short-circuit fault	Non functional product that could lead to loss of allied lites	8	Close spacing between transmission lines and/or accidental material between transmission lines, driving current		Recommended action to prevent or correct potential failure.	The electrical engineers	Sufficient PCB design
22.01.20	RAT-017	SR-017, SR-018, SR-019, CR-009	Hardware open-circuit fault	Non functional product that could lead to loss of allied lites	8	Fault in transmission lines on PCB/IC			The electrical and mechanical engineer	Sufficient PCB design
22.01.20	RAT-018	SR-009, SR-010, SR-011, SR-012, SR-032, CR-008, CR-014, CR-017, CR-018	The device freezes sporadically	No possible communication between subsystems or operators	6	Unstable Endpoint Connection (NFC, Ethernet)	TCSW006-01, TCSW006-02, TCSW006-03, TCSW006-04, TCSW016-01, TCSW017-01, TCSW018-01, TCSW019-01, TCSW020-01, TCSW021-01, TCSW012-01, TCSW011-01		The computer engineers	
22.01.20	RAT-019	SR-036, SR-038	App not well integrated into Android OS	May crash during operation/mission	7	Unsupported Android version or using old libraries	TCSW003-01, TCSW004-01, TCSW009-01, TCSW10-01	Ensure that the app is running on the latest Android version and to check that all libraries are supported	The computer engineers	



Failure Mode and Effects Analysis (FMEA)

Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy								
Responsible: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20								
		(Rev.): Version 7.0								
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?		What can cause the failure?	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?
22.01.20	RAT-020	CR-007, CR-016, SR-011, CR-009, SR-020, SR-037, SR-038, SR-040, CR-017	Bad user design	Accidental arming/detonation	8	Insufficient design regarding UI	TCME-007-01, TCME-008-01, TCSW003-01, TCSW004-01, TCSW009-01, TCSW10-01	Redesign UI	The electrical- and mechanical engineers	On/off switch is placed under a lid
23.01.20	RAT-021	SR-004, SR-026, SR-013, SR-017, CR-002, CR-020	Scheduled detonation will not be aborted	Multiple casualties and/or death	10	No possible communication between subsystems or operators	TCSW022-01, TCSW015-01, TCSW013-01, TCSW014-01	Keep sending heartbeat to ensure that there is a connection between sender and receiver at all times	The computer engineers	Re-establish the connection between sender and receiver
23.01.20	RAT-022	SR-020, SR-021, SR-022, SR-023, CR-003, CR-011	Device will not be configured before operation	Non functional product that could lead to loss of allied lifes	4	App not well integrated into Android OS			The computer engineers	Reset the device
13.02.20	RAT-023	SR-013, SR-017, SR-018, SR-020, SR-021, SR-022, SR-023, CR-001, CR-002, CR-003, CR-005, CR-018, CR-019	Nonel tubes are not ignited	Non functional product that could lead to loss of allied lifes	6		TCEL-002-01, TCEL-003-01		Development team	2 separate plasma igniter systems implemented
13.02.20	RAT-024	SR-028, SR-031, SR-032, CR-004 CR-019	Corrupted Storage	Preconfigured Settings will be lost	2	Faulty/Defect Storage Component from sources such as heat, water, dust, sand, cold	TCME-001-01, TCME-002-01, TCME-002-02, TCME-002-03, TCME-004-01		Development team	Casing is sealed with gaskets at interfaces and with Araldite on edges and corners

Failure Mode and Effects Analysis (FMEA)

Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy								
Responsible: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20								
		(Rev.): Version 7.0								
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?		What can cause the failure?	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?
13.02.20	RAT-025	CR-007, SR-032, CR-008, SR-011, SR-012, SR-032, CR-019	NIC defect	Not able to talk via IP over Network Communication	4	Faulty/Defect NIC Component from sources such as heat, water, dust, sand, cold	TCEL-001-01	Ensure that the connection into computer is saved, so that it is possible to re-connect to the same receiver again	Development team	Reconnect the sender and receiver
13.02.20	RAT-026	SR-031, SR-032	Device(s) will not reconnect	Will lose communication channel between subsystems (Receiver and Sender)	4	Wrong logic in software	TCEL-001-01, TCSW011-01, TCSW012-01, TCSW016-01, TCSW017-01, TCSW018-01, TCSW019-01, TCSW020-01, TCSW021-01	The electrical connection into computer is saved, so that it is possible to re-connect to the same receiver again	The electrical and computer engineers	Electrical: sufficient antenna design and use of standard communication protocol MIFARE
13.02.20	RAT-027	SR-043, CR-008, SR-010, SR-012, SR-032, SR-043	NFC defect	Receiver and transmitter are unable to establish a connection	2	Faulty/defect NFC chip drivers, either in receiver or mobile device	TCSW006-01, TCSW006-02, TCSW006-03, TCSW006-04,	Dimming the LEDs, redesign device to conceal the LEDs better	The electrical-mechanical engineers	Adjustable pin configuration for LEDs and placement is countersunk in the lid
15.03.20	RAT-028	SR-036, SR-039	Too visible LEDs	Receiver is too visible to the eye, and could alert hostiles in the area.	3	LEDs are given too much energy / unsuitable LED choice				

Failure Mode and Effects Analysis (FMEA)

Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20		(Rev.): Version 7.0				
Responsible: Anne Synnøve Brendøy										
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?		What can cause the failure?	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?
15.03.20	RAT-029	SR-033, SR-014, SR-016	Failure of Hardware Safety Barrier	Fatal Effect, Casualties	10	Malfunction in electrical circuit of hardware safety barrier		Ensure that all safety barriers logic is properly implemented, in terms of software and hardware	The electrical engineers	Electrical: Watchdog enabled trigger system with double signal logic. Discharge switches self test before charging capacitors. At least two HW barriers before detonation
15.03.20	RAT-030	SR-033, SR-014, SR-016	Failure of Software Safety Barrier	Fatal Effect, Casualties	10	Malfunction in software logic of software safety barrier			The computer engineers	Disable arming and detonation functionality if an error has been detected in the safety barriers
15.03.20	RAT-031									
15.03.20	RAT-032	SR-001, SR-007	Android operating system version used to develop app is outdated	Will not be able to install or use the application as intended	4	Bad communication between the development team and customer. Must agree upon an operating system standard beforehand			Development team	All details regarding operating system is agreed upon with the customer

Failure Mode and Effects Analysis (FMEA)

Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy								
Responsible: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20								
		(Rev.): Version 7.0								
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?		What can cause the failure?	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?
15.03.20	RAT-033	SR-002, SR-003, SR-004, SR-005, SR-011	Latency between receiver and mobile application	The application will not show real-time modes/states of the receiver if there is latency present	4	Bad network coverage between the receiver and app	TCSW022-01, TCSW015-01, TCSW013-01, TCSW014-01, TCSW016-01, TCSW017-01, TCSW018-01, TCSW019-01, TCSW020-01, TCSW021-01, TCSW012-01, TCSW011-01	Ensure that heartbeats are sent frequently to update the receiver status and fetch new updated data as much as possible	The computer engineers	Reconnect the sender and receiver
15.03.20	RAT-034	SR-006	The mobile app does not show a map	The operator will not be able to see deployed receivers on a map	4	If there is no GPS available on the mobile phone	TCSW004-01, TCSW003-01, TCSW009-01, TCSW010-0	Implement proper security mechanisms	The computer engineers	Reconnect the sender and receiver
15.03.20	RAT-035	SR-015	The mobile app gets reverse engineered	Valuable information gets lost	4	Insufficient cyber security measures			The computer engineers	
15.03.20	RAT-036	SR-015, SR-018	The receiver gets reverse engineered	Valuable information gets lost	4	Papers gets prematurely published			Development team	NDA
19.03.20	RAT-037	SR-017, SR-018	Insufficient isolation of high voltage circuitry and components	Personnel injury or death	4	Insufficient component research and implementation		Find more suitable components, redo implementation	The electrical engineers	Sufficient PCB layout and design
22.05.20	RAT-038	SR-047	Exploitation of IP interface	Execution of malicious code (RCE attempt)	8	Not enough resources or planning spent on endpoint security	TCSW015-01, TCSW013-01, TCSW014-01		Software Engineers	

Failure Mode and Effects Analysis (FMEA)

Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy								
Responsible: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20								
		(Rev.): Version 7.0								
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?		What can cause the failure?	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?
	RAT-039		Crashing of IP service	DDOS attack exceeding capacity for incoming server requests	8	Not enough resources or planning spent on endpoint security	TCSW015-01, TCSW013-01, TCSW014-01	Sufficient component research and thinking about power consumption throughout the design	Software Engineers	Always keep in mind power consumption when development
	22.05.20		Duration of operation time of receiver is to short	End-user not satisfied	4	Insufficient component research, no better components available			The electrical engineers	
	22.05.20	RAT-040								
<b>Environmental Risk</b>										
	RAE-001	SR-018, CR-003	Hazardous components	Expensive methods for disposal	4	Insufficient component research, no better components available		Research if new components are on the market, develop new components	The electrical- and mechanical engineers	Electrical: Use standard components and avoid certine capacitors
	RAE-002	SR-018, CR-003	Hazardous components	Negative reputation due to large impact on flora and fauna in the area of operation	4	Insufficient component research, no better components available		Research if new components are on the market, develop new components	The electrical- and mechanical engineers	Electrical: Use standard components and avoid certine capacitors. Mechanical: PETG for a disposable device



Failure Mode and Effects Analysis (FMEA)

Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy								
Responsible: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20								
		(Rev.): Version 7.0								
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)	Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?		What can cause the failure?	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?
	RAI-001		The collaboration in the development team are dysfunctional	Product will not be sufficiently developed, failed bachelor	8	Bad communication . Insufficient leadership, general attitude		Group leader takes action or a meeting with internal advisor if necessary	All team members	Morning meeting every work day + aftermeeting when needed
	RAI-002	SR-035	Internal dependencies	Prototype will not be finished	3	Workflow are interrupt due to interlocked assignments between team members		Improve sprint planning	Scrum master and group leader	Sufficient sprint planning and execution
	RAI-003	SR-035	Prototype will not be finished	Lower grade on project	3	Necessary components can not be delivered or can not be delivered on time		Improve planning to order components earlier, find different suppliers og components	All team members	Theoretical analysis and modelling to get proof of concept
	RAI-004	All CR	End-user not satisfied		5		TCME-007-01	Redesign system	All team members	
	RAI-005	All CR	Customer not satisfied		5			Redesign system	All team members	
	RAI-006		Raw material prices increase	Higher production cost	4	Economy collapse		Not something the development team can controll	None	Not something the development team can controll
	RAI-007	SR-027	Device is not very printable	Higher production cost	4	Unnecessary complex physical design	TCME-003-01	Redesign device	The mechanical engineer	Design made a simple as possible

**Failure Mode and Effects Analysis (FMEA)**

		Process/Product Name: SPARK RFS		Prepared By: Anne Synnøve Brendøy															
		Responsible: Anne Synnøve Brendøy		FMEA Date (Orig.): 20.01.20		(Rev.):		Version 7.0											
Risk raised date	Risk Number	Coherent to requirement nr.	Potential Failure Mode	Potential Failure Effects	SEVERITY (1 - 10)		Potential Causes	Current Controls	Action Recommended	Responsible	Actions Taken								
Date the risk proposition was raised	Assign a number to the risk in question	The requirement ID of the requirement that are affected by this risk	What can go wrong?	What is the impact on the customer or team if this failure is not prevented or corrected?	3		Being exposed to the virus from other people in public	Test specification ID	Recommended action to prevent or correct potential failure.	Responsible to ensure action is taken.	What actions were completed (and when) with respect to the RPN?								
	RAI-008		One of the team gets hit by Covid-19	Loss of temporarily manpower, not enough work gets done	3		Being exposed to the virus from other people in public		Follow government regulations and advice. Keep distance	All team members	Home office, online meetings, voluntary quarantine								
	RAI-009		Most/All of the team gets hit by Covid-19	Significant loss of manpower, will not meet milestones	6		Being exposed to the virus from other people in public		Follow government regulations and advice. Keep distance	All team members	Home office, online meetings, voluntary quarantine								



**List of Test Reports**

1	TCEL-001-01	334
2	TCEL-002-01	334
3	TCEL-003-01	335
4	TCSW003-01	336
5	TCSW004-01	336
6	TCSW009-01	337
7	TCSW010-01	337
8	TCSW011-01	338
9	TCSW012-01	338
10	TCSW006-01	339
11	TCSW006-02	339
12	TCSW006-03	340
13	TCSW006-04	340
14	TCSW015-01	341
15	TCSW013-01	341
16	TCSW014-01	342
17	TCSW022-01	342
18	TCSW016-01	343
19	TCSW017-01	343
20	TCSW018-01	344
21	TCSW019-01	344
22	TCSW020-01	345
23	TCSW021-01	345
24	TCME-001-01	346
25	TCME-002-01	346
26	TCME-002-02	347
27	TCME-002-03	347
28	TCME-003-01	348
29	TCME-004-01	348
30	TCME-005-01	349
31	TCME-006-01	349
32	TCME-007-01	350
33	TCME-008-01	350
34	TCME-00X-01	351

**TEST REPORT: TCEL-001-01**

**Test Responsible:** MB

**DATE:** 22.03.20

**Objective:**

Determine if the receiver supports Ethernet 803.3 through its RJ45 interface.

**Method:**

Do the following steps sequentially: 1) Turn on the receiver. 2) Connect a PC to the router via WIFI and start to continuously ping the IP address of the receiver. 3) Connect a 10 meter CAT5 cable to a router and the other end to the receiver.

**Acceptance Criteria :**

The PC must get a reply to its ping request within 10 seconds of connecting the receiver to the router.

**Test Results:**

T.B.D

**Conclusion:**

T.B.D

**TEST REPORT: TCEL-002-01**

**Test Responsible:** MB

**DATE:** 22.03.20

**Objective:**

Determine if the receiver capacitors can deliver 350V and 9J of energy to both plasma igniters

**Method:**

Do the following steps sequentially: 1) Find a safe location where you will not be disturbed 2) Configure the receiver ready for operation. 4) Connect a voltmeter to each of the capacitors 5) Arm the receiver from a distance of 1 meter

**Acceptance Criteria :**

The plasma igniters must make a spark that lasts for the amount of time required for a nonel fuse to ignite.

**Test Results:**

T.B.D

**Conclusion:**

T.B.D

**TEST REPORT: TCEL-003-01**

**Test Responsible:** MB

**DATE:** 22.03.20

---

**Objective:**

Determine if the receiver capacitors can accumulate 350V and 9J of energy

**Method:**

Do the following steps sequentially: 1) Find a safe location where you will not be disturbed  
2) Configure the receiver ready for operation. 4) Connect a voltmeter to each of the capacitors 5) Arm the receiver from a distance of 1 meter

**Acceptance Criteria :**

The capacitors voltage must be within  $\pm 5\%$  of 350V

**Test Results:**

T.B.D

**Conclusion:**

T.B.D

## TEST REPORT: TCSW003-01

Test Responsible: BB

DATE: 23.03.2020

---

### Objective:

The map function shall remember its fixed position when entering and leaving the mapview of the application

### Method:

Will Verify this by using the app, swiping back and forth between views 50 times

### Acceptance Criteria :

Should be able to keep its position after at least 50 swipes back and forth

### Test Results:

After 50 swipes, the map still remember its fixed position

### Conclusion:

The test case was compliant to its acceptance criteria, and therefore successful

## TEST REPORT: TCSW004-01

Test Responsible: BB

DATE: 23.03.2020

---

### Objective:

The waypoint shall remain at its deployed position when switching views in the application

### Method:

Will Verify this by using the app, deploy multiple waypoints, swiping back and forth between views 50 times

### Acceptance Criteria :

Should be able to remember its position after atleast 50 swipes

Should be able to do the above using multiple waypoints;

### Test Results:

After 50 swipes, the waypoints are still in place

### Conclusion:

The test case was compliant to its acceptance criteria, and therefore successful

## TEST REPORT: TCSW009-01

Test Responsible: BB

DATE: 23.03.2020

---

**Objective:**

Should be able to deploy multiple waypoints on map

**Method:**

Will Verify this by using the app, adding multiple waypoints, and see over time if they alter or change.

**Acceptance Criteria :**

Be able to place unique waypoints on map without interference.

**Test Results:**

After placing up to 10 waypoints, the function works as expected.

**Conclusion:**

The test case was compliant to its acceptance criteria, and therefore successful

## TEST REPORT: TCSW010-01

Test Responsible: BB

DATE: 23.03.2020

---

**Objective:**

Should be able to deploy multiple waypoints on map

**Method:**

Will Verify this by using the app, adding multiple waypoints, and see over time if they alter or change.

**Acceptance Criteria :**

Each waypoint should be able to change state isolated

Each change of a waypoint state should not affect or alter another waypoints state;

**Test Results:**

The action is only able to change the first initial waypoint deployed, while the rest upon deployment, sets in the same state of which the first one is currently in.

**Conclusion:**

The test case shows faults in the handling of waypoint object and must be revised. Test case not compliant to requirement, therefore not successful

**TEST REPORT: TCSW011-01**

**Test Responsible:** BB

**DATE:** 25.04.2020

---

**Objective:**

The app shall send a Pairing Response with necessary pairing data, when receiving a Pairing Request over IP.

**Method:**

Will verify packets according to protocol using Wireshark tool and a Python test client script to initiate pairing handshake.

**Acceptance Criteria :**

One packet should be received with a header according to protocol as sent from Server endpoint;

**Test Results:**

Packet was sent and received intact 10 out of 10 times.. Wireshark raw hex dump showed the header contents as according to the protocol

**Conclusion:**

Test was successful

**TEST REPORT: TCSW012-01**

**Test Responsible:** BB

**DATE:** 25.04.2020

---

**Objective:**

The App shall be able to receive and store updates on receivers status over IP.

**Method:**

Will verify packets according to protocol using a Python test client script to send out Heartbeat packets. The updates will be 5 different 8-bit values for testing.

**Acceptance Criteria :**

Packets should be parsed correctly and sent to the correct receiver instance.;

**Test Results:**

Packet was sent and received intact. Each value was split correctly into a 8-bit value from the given offset according to the Header. The values was sent to the correct receiver instance holding the correct system ID. 10 out of 10 times successful.

**Conclusion:**

Test was successful

## TEST REPORT: TCSW006-01

Test Responsible: KA

DATE: 12.05.2020

---

### Objective:

Determine if receiver appears in the "overview" tab of the application when scanning an NFC tag.

### Method:

- 1) Start application, enter "overview" tab.
- 2) Bring NFC tag to the back of transmitter.
- 3) Close application.
- 4) Repeat steps 1-3 10 times.

### Acceptance Criteria :

The receiver should appear in the "overview" tab. Upon scanning of the same tag several times, receiver should simply remain in the "overview" tab with no change.

### Test Results:

Receiver appears in the overview tab as expected.

### Conclusion:

NFC functionality behaves as expected when performing a simple tag read.

## TEST REPORT: TCSW006-02

Test Responsible: KA

DATE: 12.05.2020

---

### Objective:

Determine if receiver remains in the "overview" tab when switching tabs.

### Method:

- 1) Start application, enter "overview" tab.
- 2) Bring NFC tag to the back of transmitter.
- 3) Go back to application home
- 4) Go back into the "overview" tab
- 5) Repeat steps 1-4 2 times.

### Acceptance Criteria :

The receiver should remain in the "overview" tab despite switching in and out of the tab.

### Test Results:

Receiver remains in the "overview" tab as expected.

### Conclusion:

NFC functionality behaves as expected when switching views in the application. This verifies that the front-end does not control the NFC object.

## TEST REPORT: TCSW006-03

Test Responsible: KA

DATE: 12.05.2020

---

### Objective:

Determine if application is capable of reading NFC tags after it has been minimized (sent to background).

### Method:

1) Start application, enter "overview" tab. 2) Bring NFC tag to the back of transmitter. 3) Minimize application, then maximize again. 4) Go back into the "overview" tab 5) Attempt another NFC tag read. 6) Repeat steps 1-5 10 times.

### Acceptance Criteria :

The application should still be able to read NFC tags.

### Test Results:

Application is still able to read NFC tags after being minimized.

### Conclusion:

NFC functionality behaves as expected after minimizing the application, being able to read tags as usual.

## TEST REPORT: TCSW006-04

Test Responsible: KA

DATE: 23.05.2020

---

### Objective:

Determine if application is capable of reading NFC tags after transmitter has been locked and unlocked.

### Method:

1) Start application, enter "overview" tab. 2) Bring NFC tag to the back of transmitter. 3) Verify application can read NFC tag to begin with. 4) Lock phone, then unlock again. 5) Attempt another NFC tag read. 6) Repeat steps 1-5 10 times.

### Acceptance Criteria :

The application should still be able to read NFC tags.

### Test Results:

Application is still able to read NFC tags after being locked and unlocked.

### Conclusion:

NFC functionality behaves as expected after application has been locked and unlocked, verifying that the intent filters of the application works correctly.



**TEST REPORT: TCSW015-01**

**Test Responsible:** BB

**DATE:** 23.05.2020

---

**Objective:**

Verify the content of a Detonation Packet.

**Method:**

Initiate a Pairing Sequence and emit a Detonation Packet from the App 10 times

**Acceptance Criteria :**

Content should be compliant to SPARK networking protocol.

**Test Results:**

Packet content matches packet command 10/10 times

**Conclusion:**

Packet Content should always be as defined in the SPARK Networking Protocol, test success

**TEST REPORT: TCSW013-01**

**Test Responsible:** BB

**DATE:** 23.05.2020

---

**Objective:**

Verify the content of a Standby Packet.

**Method:**

Initiate a Pairing Sequence and emit a Standby Packet from the App 10 times

**Acceptance Criteria :**

Content should be compliant to SPARK networking protocol.

**Test Results:**

Packet content matches packet command 10/10 times

**Conclusion:**

Packet Content should always be as defined in the SPARK Networking Protocol, test success

## TEST REPORT: TCSW014-01

Test Responsible: BB

DATE: 23.05.2020

---

**Objective:**

Verify the content of an Arm Packet.

**Method:**

Initiate a Pairing Sequence and emit an Arm Packet from the App 10 times

**Acceptance Criteria :**

Content should be compliant to SPARK networking protocol.

**Test Results:**

Packet content matches packet command 10/10 times

**Conclusion:**

Packet Content should always be as defined in the SPARK Networking Protocol, test success

## TEST REPORT: TCSW022-01

Test Responsible: BB

DATE: 23.05.2020

---

**Objective:**

Verify the application ability to read, parse and update heartbeat packets.

**Method:**

Initiate a Pairing Sequence and emit a Heartbeat Packet from the simulated Receiver 10 times

**Acceptance Criteria :**

Content should be compliant to SPARK networking protocol.

**Test Results:**

Packet content matches packet command 10/10 times

**Conclusion:**

Packet Content should always be as defined in the SPARK Networking Protocol, test success

## TEST REPORT: TCSW016-01

Test Responsible: BB

DATE: 23.05.2020

---

**Objective:**

Verify ability to handle multiple connections.

**Method:**

Initiate a pairing sequence and emit heartbeat packets from 2 devices

**Acceptance Criteria :**

Updated heartbeats should be assigned to their correct instantiated device.

**Test Results:**

Correct devices are updated based on the Receiver ID packet field matching Identifier value in the App

**Conclusion:**

A device instance should always and only receive packets meant for them alone, test success

## TEST REPORT: TCSW017-01

Test Responsible: BB

DATE: 23.05.2020

---

**Objective:**

Verify handling of Pairing Process

**Method:**

Initiate a Pairing Sequence and check connection status in the App 10 times

**Acceptance Criteria :**

Content should be compliant to SPARK networking protocol.

**Test Results:**

Connection status was put in Pending 10/10 times

**Conclusion:**

Connection status should always be as defined in the SPARK Networking Protocol, test success

## TEST REPORT: TCSW018-01

Test Responsible: BB

DATE: 23.05.2020

---

**Objective:**

Verify handling of Pairing Process

**Method:**

Initiate a Pairing Sequence and check connection status in the App 10 times

**Acceptance Criteria :**

Content should be compliant to SPARK networking protocol.

**Test Results:**

Connection status was put in Active state 10/10 times

**Conclusion:**

Connection status should always be as defined in the SPARK Networking Protocol, test success

## TEST REPORT: TCSW019-01

Test Responsible: BB

DATE: 23.05.2020

---

**Objective:**

Verify the application ability to notify operator of unacknowledged packet.

**Method:**

Initiate a Pairing Sequence and emit a Detonation Packet from App 10 times, then not acknowledge it from the simulated Receiver.

**Acceptance Criteria :**

App should notify Operator.

**Test Results:**

Warning was generated 10/10 times

**Conclusion:**

Expected behavior, test success

**TEST REPORT: TCSW020-01**

**Test Responsible:** BB

**DATE:** 23.05.2020

---

**Objective:**

Verify the application ability to notify operator of unacknowledged packet.

**Method:**

Initiate a Pairing Sequence and emit an Arm Packet from App 10 times, then not acknowledge it from the simulated Receiver.

**Acceptance Criteria :**

App should notify Operator.

**Test Results:**

Warning was generated 10/10 times

**Conclusion:**

Expected behavior, test success

**TEST REPORT: TCSW021-01**

**Test Responsible:** BB

**DATE:** 23.05.2020

---

**Objective:**

Verify the application ability to notify operator of unacknowledged packet.

**Method:**

Initiate a Pairing Sequence and emit a Standby Packet from App 10 times, then not acknowledge it from the simulated Receiver.

**Acceptance Criteria :**

App should notify Operator.

**Test Results:**

Warning was generated 10/10 times

**Conclusion:**

Expected behavior, test success

**TEST REPORT: TCME-001-01**

**Test Responsible:** ASB

**DATE:** 22.03.20

---

**Objective:**

Determine if the container unit parts all have gaskets on every interface and that all gaskets are undamaged.

**Method:**

Visual inspection

**Acceptance Criteria :**

The container unit should have gaskets without any damage on every interface.

**Test Results:**

1. The container unit has gaskets on every interface
2. The gaskets are undamaged. **Approved when both 1 and 2 are checked.**
3. The container unit does not have gaskets on every interface. **Not approved**
4. One or more gaskets are damaged **Not approved**

**Conclusion:**

Damage or No damage to electrical components inside container unit

**TEST REPORT: TCME-002-01**

**Test Responsible:** ASB

**DATE:** 22.03.20

---

**Objective:**

Determine if the container unit holds IP65 classification

**Method:**

Expose the container unit to a water jet with 6,3mm nozzle and up to 12,5l/m( $\pm 5\%$ ) for at least 3 minutes from any angle

**Acceptance Criteria :**

The container unit should be dust tight, have no entrance point for solid objects and have protection from liquids.

**Test Results:**

1. The container unit does keep water and foreign objects out. **Approved**
2. The container unit does not keep water, dust and foreign objects out.

**Not approved**

**Conclusion:**

Damage or No damage to electrical components inside container unit

**TEST REPORT: TCME-002-02**

**Test Responsible:** ASB

**DATE:** 22.03.20

---

**Objective:**

Determine if the container unit holds IP67 classification

**Method:**

Submerge container unit into water with the immersion depth of 1m for 30 minutes

**Acceptance Criteria :**

The container unit should be dust tight, have no entrance point for solid objects and have protection from liquids.

**Test Results:**

1. The container unit does keep water and foreign objects out. **Approved**
2. The container unit does not keep water, dust and foreign objects out.

**Not approved**

**Conclusion:**

Damage or No damage to electrical components inside container unit

**TEST REPORT: TCME-002-03**

**Test Responsible:** ASB

**DATE:** 22.03.20

---

**Objective:**

Determine if the container unit holds IP68 classification

**Method:**

Submerge container unit into water with continuous immersion depth for 30-60 minutes

**Acceptance Criteria :**

The container unit should be dust tight, have no entrance point for solid objects and have protection from liquids.

**Test Results:**

1. The container unit does keep water and foreign objects out. **Approved**
2. The container unit does not keep water, dust and foreign objects out.

**Not approved**

**Conclusion:**

Damage or No damage to electrical components inside container unit.

**TEST REPORT: TCME-003-01**

**Test Responsible:** ASB

**DATE:** 22.03.20

---

**Objective:**

Determine if the container unit is printable in 3D printer

**Method:**

Make a test specimen based on drawings made with CAD software

**Acceptance Criteria :**

The container unit should have the correct measures and tolerances after printing as it was drawn with. Visual inspection and measuring.

**Test Results:**

1. The container unit does maintain the measures and tolerances after printing. **Approved**
2. The container unit does not maintain the measures and tolerances after printing.

**Not approved**

**Conclusion:**

The container unit is or is not 3D printable

**TEST REPORT: TCME-004-01**

**Test Responsible:** ASB

**DATE:** 22.03.20

---

**Objective:**

Determine if the container unit is shock resistant

**Method:**

Expose device to equivalent shock as of an explosion, drop and direct hit.

**Acceptance Criteria :**

The container unit should maintain IP classification Visual inspection and redo test TCME-002-01/02/03

**Test Results:**

1. The container unit does maintain IP classification. **Approved**
2. The container unit does not maintain IP classification. **Not approved**

**Conclusion:**

Damage or No damage to electrical components inside container unit.



## TEST REPORT: TCME-005-01

Test Responsible: ASB

DATE: 22.03.20

### Objective:

Determine if the system has the required weight after assembly

### Method:

Make a complete assembly of the entire system.

### Acceptance Criteria :

The system as a whole should be within the required maximum weight. Visual inspection and weigh in.

### Test Results:

1. The system is within the required maximum weight limit. **Approved**
2. The system is not within the required maximum weight limit. **Not approved**

### Conclusion:

The system is or is not within the required weight

## TEST REPORT: TCME-006-01

Test Responsible: ASB

DATE: 22.03.20

### Objective:

The container unit must not exceed required outwards measurements

### Method:

Make a complete assembly of the container unit.

### Acceptance Criteria :

The complete container unit should be within these measures: 120 · 80 · 30 mm ( $L \cdot W \cdot H$ ). Visual inspection and measuring.

### Test Results:

1. The container is within the required measures. **Approved**
2. The container unit is not within the required measures. **Not approved**

### Conclusion:

The complete container unit is or is not smaller or equal to required outwards measurements

## TEST REPORT: TCME-007-01

Test Responsible: ASB

DATE: 22.03.20

### Objective:

The device must have an intuitive and user friendly design.

### Method:

Make a complete assembly of the entire system.

### Acceptance Criteria :

Visual inspection and hands-on testing by product owner or end user

### Test Results:

1. The device has an intuitive and user friendly design. **Approved**
2. The device has not an intuitive and user friendly design. **Not approved**

### Conclusion:

The device does or does not satisfy as an intuitive and user friendly device.

## TEST REPORT: TCME-008-01

Test Responsible: ASB

DATE: 22.03.20

### Objective:

The container unit must be able to be attached to any chosen surface and stay attached.

### Method:

Make a complete assembly of the entire system and attache it to a surface of wood, metal, glass and plastic

### Acceptance Criteria :

The device must stay attached for a minimum of 60 minutes.

### Test Results:

1. The device does stay attached to the chosen surface within the required time-limit.  
**Approved**
2. The device does not stay attached to the chosen surface within the required time-limit.  
**Not approved**

### Conclusion:

The device meets or meets not the expectations according to the adhesion requirement.

## TEST REPORT: TCME-00X-01

Test Responsible: ASB

DATE: 22.03.20

---

### Objective:

Determine if the requirements function is adequate to satisfy end user or product owner.

### Method:

Hands-on functionality test

### Acceptance Criteria :

The end-user must be able to do what the requirement specifies.

### Test Results:

1. The functionality is satisfactory according to the requirement. **Approved** 2. The functionality is not what the requirement describes. **Not approved**

### Conclusion:

The functionality is or is not fulfilled according to the requirement specification.

## **8.4 Economy attachments**



Product name	Product number	Link to product	Price per unit	Quantity	Result
Adafruit HUZZAH32 ESP32 Feather Board	141838	<a href="https://www.digitalimpu">https://www.digitalimpu</a>	329	1	329
Arduino mega 2560	110-38-920	<a href="https://www.elfadistrel">https://www.elfadistrel</a>	402,5	1	402,5
Raspberry Pi 4 Model B, 4GB RAM	3051887	<a href="https://www.komplett.no">https://www.komplett.no</a>	729	2	1458
Keystone Electronics 1050	36-1050-ND	<a href="https://www.digikey.no/c">https://www.digikey.no/c</a>	25,45	3	76,35
Keystone Electronics 232	36-232-ND	<a href="https://www.digikey.no/c">https://www.digikey.no/c</a>	4,51	1	4,51
25 pack Jumper Wire, Male to Male	2762508	<a href="https://no.farnell.com/bu">https://no.farnell.com/bu</a>	40	1	40
10 pack Jumper Wire, Male to Female	1471-1231-ND	<a href="https://www.digikey.no/c">https://www.digikey.no/c</a>	29,3	3	87,9
40 stk Jumper Wire, Female to Female	128499	<a href="https://www.digitalimpu">https://www.digitalimpu</a>	50	1	50
Arduino MKR ZERO	301-01-961	<a href="https://www.elfadistrel">https://www.elfadistrel</a>	270	2	540
Arduino MKR ETH SHIELD	87196	<a href="https://www.kiell.com/nc">https://www.kiell.com/nc</a>	400	2	800
ARDUINO MKR1000 WIFI WITH HEADERS MOUNTED	301-01-959	<a href="https://www.elfadistrel">https://www.elfadistrel</a>	358,75	2	717,5
Samsung Galaxy S9	N/A	Elkj�p outlet	4867	1	4867
K�GWERKS GALAXY S9 KIT	N/A	<a href="https://kagwerks.com/cof">https://kagwerks.com/cof</a>	2950	1	2950
				inc. Mva	12322,76
			<b>Total (NOK)</b>	exl. mva	9242,07

Figure 142: Purchase list for prototype v2

1407-PhoneHouse Grønland  
Tøyengt 2  
0191 Oslo

NO 947 054 600 MVA  
Foretaksregisteret

BANK: 6005 06 33198  
TLF: 210 021 21  
PhoneHouse.gronland@elkjop.no

Grønland  
Unik ID: V2IVAI87

**Kopi**  
**Utl.dok 1407XYYY0Y5**

Spectac AS

c/o Total Innovasjon Raufossvegen  
2821 GJØVIK

Ordrenr.: XX6VXZ5 28.02.20 (3) 1-17:16

Selger..: LOTTE K

Telefon.: 90189861

E-post..: pd@spectac.no

**Levering:**

Spectac AS

c/o Total Innovasjon Raufossvegen 4  
2821 GJØVIK

Telefon.: 90189861

Kontaktreferanse.. Petter Aamodt Dahlby

Telefon..... 45872322

Leveringsinfo..... Sirajuddin Asjad

Varekode	Beskrivelse	Ldato	Pris	Ant	Beløp
73368	OUTLET-SAMS9BK-B01 SAMSUNG SM-G960FZKDNEE Smartph Outlet - demo/retur/lagertømming Garantinr 201982817298/356204100069713 Hentes av Sirajuddin for Spectac AS. Overføres til side 2		3893,60	1	3.893,60'

1407-PhoneHouse Grønland  
Tøyengt 2  
0191 Oslo

NO 947 054 600 MVA  
Foretaksregisteret

BANK: 6005 06 33198  
TLF: 210 021 21  
PhoneHouse.gronland@elkjop.no

Grønland  
Unik ID: V2IVAI87

**Kopi**  
**Utl.dok 1407XYYY0Y5**

Spectac AS

c/o Total Innovasjon Raufossvegen  
2821 GJØVIK

-----  
Ordrenr.: XX6VXZ5 28.02.20 (3) 1-17:16  
Selger.: LOTTE K  
Telefon.: 90189861

E-post.: pd@spectac.no

**Levering:**

Spectac AS

c/o Total Innovasjon Raufossvegen 4  
2821 GJØVIK

Telefon.: 90189861

Kontaktreferanse.. Petter Aamodt Dahlby

Telefon..... 45872322

Leveringsinfo..... Sirajuddin Asjad

Varekode	Beskrivelse	Ldato	Pris	Ant	Beløp
	Overført fra side 1				3.893,60
	Sum eksklusive mva				3.893,60
	Mva				973,40
	TOTAL		NOK		4.867,00
	B2B FAKTURA SAP (97966)				4.867,00
		Grunnlag	MVA		
	' : 25% mva	3.893,60	973,40		

Viaplay kode: UC4LM6

Besøk: [viaplay.no/kampanje](http://viaplay.no/kampanje)

Siste innløsningsdato: 15.05.20

En verdikode pr kunde. Gyldig kun nye Viaplay-konti

Alt om 50 dager åpent kjøp på [elkjop.no/kundefordeler](http://elkjop.no/kundefordeler).

Produktet må være i samme stand som da du kjøpte det, rengjort godt og uten riper eller andre bruksmerker.

Alt tilbehør som fulgte med må returneres sammen med produktet.

Vi setter pris på om du har tatt vare på originalemballasjen.

Unntak fra 50 dager åpent kjøp:

Mobiltelefoner, produkter med abonnement, programvare, spill eller film med brutt emballasje.

Epoq-kjøkken og spesialbestilte kjøkkendeler.

Hygieneprodukter (in-ear hodetelefoner, ørepropper, hud- og kroppspleieprodukter og lignende)

Om du har fått en gave med byttelapp, vil du få tilgodelapp eller et gavekort som kan brukes i alle Elkjøps varehus.





Adresse: Domeneshop AS  
Christian Krohgs gate 16  
0186 Oslo  
Telefon: (+47) 22 94 33 33  
Fax: (+47) 22 94 33 34  
Epost: post@domeneshop.no  
Foretaksregisteret: NO 976 769 678 MVA

Sirajuddin Asjad  
Sirajuddin Asjad  
Galterudveien 8  
3034 Drammen  
Norway

Epost: sirasjad@gmail.com

**FAKTURA 3143500**

Fakturadato 25.01.2020

Forfallsdato 10.02.2020

PIN-kode 03292923

Beskrivelse	Periode	Antall	Pris	Sum
Domenerregistrering spark-rfs.no	24.01.2020 - 24.01.2021	1	120,00	120,00
Epost for spark-rfs.no	24.01.2020 - 24.01.2021	1	80,00	80,00
Sum mva-pliktig: 160,00	Sum mva-fritt: 0,00	25% mva: 40,00	<b>Totalt (NOK):</b>	<b>200,00</b>

**BETALT**

Fakturaen er betalt med MasterCard 25.01.2020.

ELKJØP STORMARKED DRAMMEN  
Bjørnstjerne Bjørnsonsgate 80, 3044 Drammen  
Tlf: 210 02 121 www.elkjop.no

NO 947 054 600 MVA  
Foretaksregisteret  
Bank: 6005 06 33198

Drammen  
Unik ID: SUUH5L8T

**Kopi**  
**FAKTURA 10534396074**

Spectac AS  
Raufossvegen 40  
2821 GJØVIK

Ordrenr.: 3509661 18.01.20 (3) 0-12:44  
Selger.: SIRAJUDDIN SIRAJUDDIN  
Telefon.: 45872322

E-post.: pd@spectac.no

Varekode	Beskrivelse	Ldato	Pris	Ant	Beløp
22150	AC CB241 24/FHD/IPS/60 ACER UM.QB1EE.018 Computer mon Garantinr 201996628624		1249,00	1	1.249,00'
22150	AC CB241 24/FHD/IPS/60 ACER UM.QB1EE.018 Computer mon Garantinr 201996853102		1249,00	1	1.249,00'
14583	HAMA HDMI CABLE BLACK 0.75M HAMA X1122215 Cable Pris redusert Prisgarant 100,00		199,00	1	99,00'
14583	HAMA HDMI CABLE BLACK 0.75M HAMA X1122215 Cable Pris redusert Prisgarant 100,00		199,00	1	99,00'
LHDM DVI16	1.8m DVI to HDMI cable with bl Pris redusert Prisgarant 180,00		279,00	1	99,00'
LHDM DVI16	1.8m DVI to HDMI cable with bl Pris redusert Prisgarant 180,00		279,00	1	99,00'
LHDM DVI16	1.8m DVI to HDMI cable with bl Pris redusert Prisgarant 180,00		279,00	1	99,00'
42468	ADX A01 Gaming keyboard ADX 266251 Keyboard Pris redusert Prisgarant 150,00		299,00	1	149,00'
42468	ADX A01 Gaming keyboard ADX 266251 Keyboard Pris redusert Prisgarant 150,00		299,00	1	149,00'
42468	ADX A01 Gaming keyboard ADX 266251 Keyboard Pris redusert Prisgarant 150,00		299,00	1	149,00'
42468	ADX A01 Gaming keyboard ADX 266251 Keyboard Pris redusert Prisgarant 150,00		299,00	1	149,00'
FK	***** Referanse: Petter Dahlby Faktura sendes til pd@spectac.no Overføres til side 2				0,00
					3.589,00



ELKJØP STORMARKED DRAMMEN  
Bjørnstjerne Bjørnsonsgate 80, 3044 Drammen  
Tlf: 210 02 121 www.elkjop.no

NO 947 054 600 MVA  
Foretaksregisteret  
Bank: 6005 06 33198

Drammen  
Unik ID: SUUH5L8T

**Kopi**  
**FAKTURA 10534396074**

Spectac AS  
Raufossvegen 40  
2821 GJØVIK

-----  
Ordrenr.: 3509661 18.01.20 (3) 0-12:44  
Selger..: SIRAJUDDIN SIRAJUDDIN  
Telefon.: 45872322

E-post..: pd@spectac.no

Varekode	Beskrivelse	Ldato	Pris	Ant	Beløp
	Overført fra side 1				3.589,00
	Sum rabatt 1.340,00				
	TOTAL (Mva inkl. med NOK 717,80)		NOK		3.589,00

Forfallsdato: 15.02.20

KundeID (KID): **044000043960747** Bankkonto: **6005 06 33198**

Faktura sendt i e-post 18.01.20

	Grunnlag	MVA
' : 25% mva	2.871,20	717,80

Varene forblir selgers eiendom til de er fullt betalt.

Alt om 50 dager åpent kjøp på [elkjop.no/kundefordeler](http://elkjop.no/kundefordeler).

Produktet må være i samme stand som da du kjøpte det, rengjort godt og uten riper eller andre bruksmerker.

Alt tilbehør som fulgte med må returneres sammen med produktet.

Vi setter pris på om du har tatt vare på originalemballasjen.

Unntak fra 50 dager åpent kjøp:

Mobiltelefoner, produkter med abonnement, programvare, spill eller film med brutt emballasje.

Epoq-kjøkken og spesialbestilte kjøkkendeler.

Hygieneprodukter (in-ear hodetelefoner, ørepropper, hud- og kroppspfleieprodukter og lignende)

Om du har fått en gave med byttelapp, vil du få tilgodelapp eller et gavekort som kan brukes i alle Elkjøps varehus.



## 8.5 Prototype 1.0 code

**CENSORED**

## **8.6 Prototype 2.0 code**

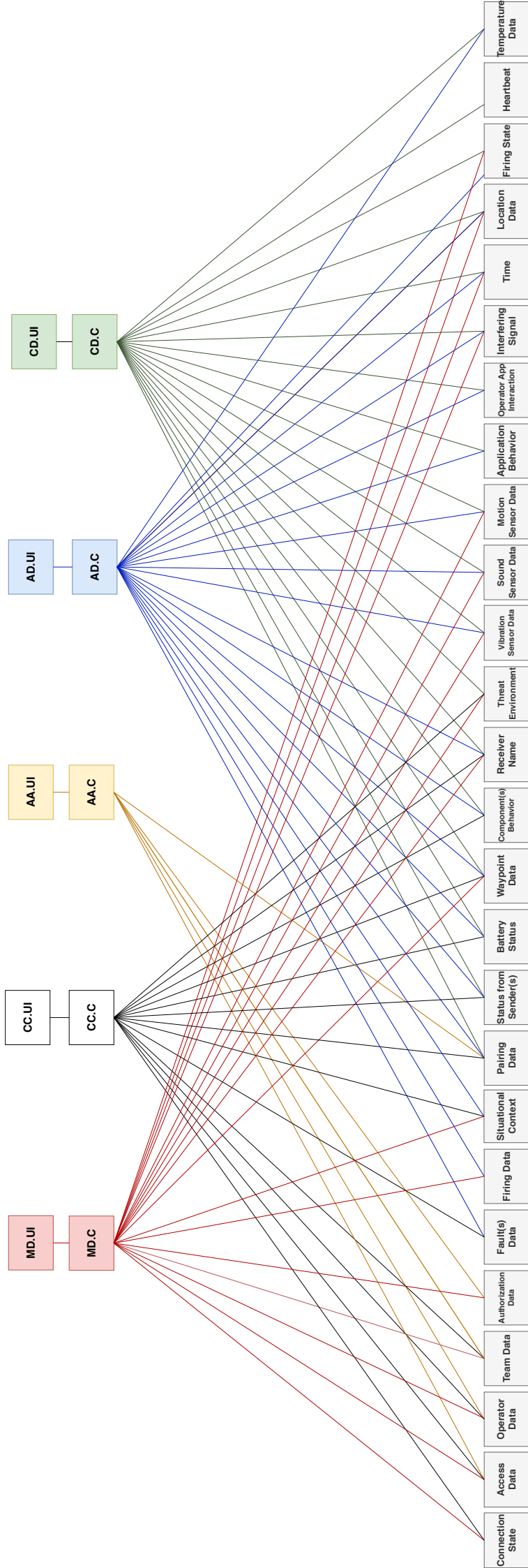
**CENSORED**

Near Field Communication (NFC) ([nfc.hpp](#))



**CENSORED**

## 8.7 Software Architectural Model





## 8.8 Receiver IP Simulation

```
1  #Receiver IP Simulation
2  import sys
3  import socket
4
5  #TCP_IP = "127.0.0.1"
6  TCP_IP = "192.168.1.195"
7  #TCP_IP = "192.168.1.6"
8  TCP_PORT = 1234
9  BUFFER_SIZE=1024
10 pairRequest= bytearray(10)
11 pairAck=bytearray(10)
12 heartbeat=bytearray(13)
13 ack=bytearray(10)
14
15 pairRequest[0]=0xff #start of packet
16 pairRequest[1]=0xff #start of packet
17 pairRequest[2]=0x00 #len payload
18 pairRequest[3]=0x00 #len payload
19 pairRequest[4]=0x03 #senderID1
20
21 pairRequest[5]=0xff #recipientID
22 pairRequest[6]=0x01 #DataCategory
23 pairRequest[7]=0x02 #DataID
24
25 pairAck[0]=0xff#start of packet
26 pairAck[1]=0xff#start of packet
27 pairAck[2]=0x00#len payload
28 pairAck[3]=0x00#len payload
29 pairAck[4]=0x03#senderID
30 pairAck[5]=0xff#recipientID
31 pairAck[6]=0x01#DataCategory
32 pairAck[7]=0x01#DataID
33
34 heartbeat[0]=0xff#start of packet
35 heartbeat[1]=0xff#start of packet
36 heartbeat[2]=0x00#len payload
37 heartbeat[3]=0x05#len payload
38 heartbeat[4]=0x03#senderID
39 heartbeat[5]=0xff#recipientID
```

```
40 heartbeat[6]=0x03#DataCategory
41 heartbeat[7]=0x02#DataID
42 heartbeat[8]=0x0f#Payload
43 heartbeat[9]=0x0f#Payload
44 heartbeat[10]=0x0f#Payload
45 heartbeat[11]=0x0f#Payload
46 heartbeat[12]=0x0f#Payload
47
48 ack[0]=0xff#start of packet
49 ack[1]=0xff#start of packet
50 ack[2]=0x00#len payload
51 ack[3]=0x00#len payload
52 ack[4]=0x03#senderID
53 ack[5]=0xff#recipientID
54 ack[6]=0x02#DataCategory
55 ack[7]=0x01#DataID
56
57
58
59 print(TCP_IP, " id:3")
60
61 sock=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
62 sock.connect((TCP_IP,TCP_PORT))
63 while True:
64     print("1 = pairReq, 2 = pairAck, 3 = heartbeat, 4 = tacOpsAck")
65     x=input()
66     if(x=="1"):
67         sock.send(pairRequest)
68         dataMsg=sock.recv(BUFFER_SIZE)
69         print("Received: ",dataMsg)
70     elif(x=="2"):
71         sock.send(pairAck)
72     elif(x=="3"):
73         sock.send(heartbeat)
74         dataMsg=sock.recv(BUFFER_SIZE)
75         print("Received: ",dataMsg)
76     elif(x=="4"):
77         sock.send(ack)
78         dataMsg = sock.recv(BUFFER_SIZE)
79         print("Received: ", dataMsg)
80     else:
```

```
81     dataMsg = sock.recv(BUFFER_SIZE)
82     print("Received: ", dataMsg)
```

### 8.9 Sequence Diagram Pairing

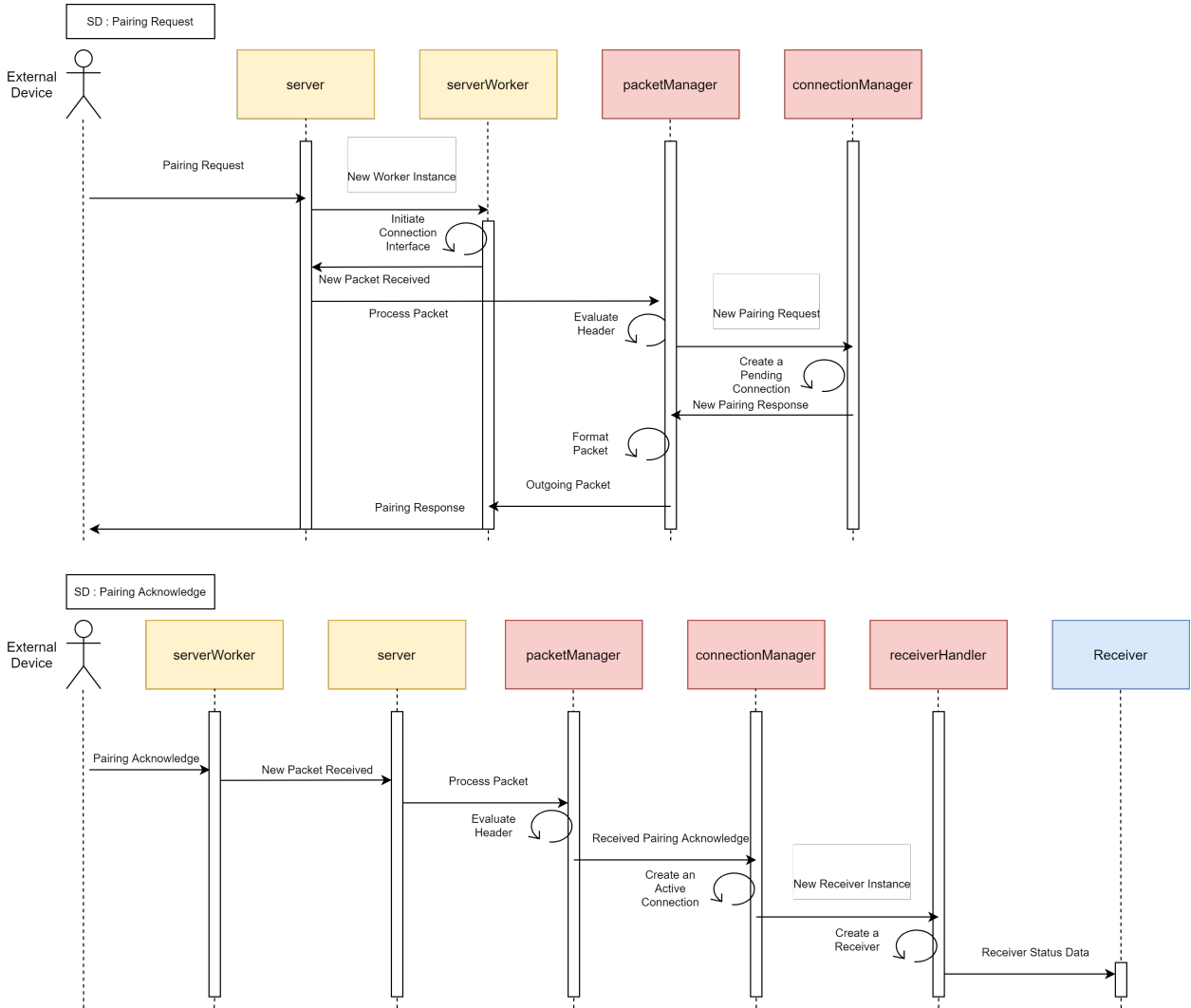


Figure 143: Pairing Static Process



### 8.10 Sequence Diagram Tactical Operations

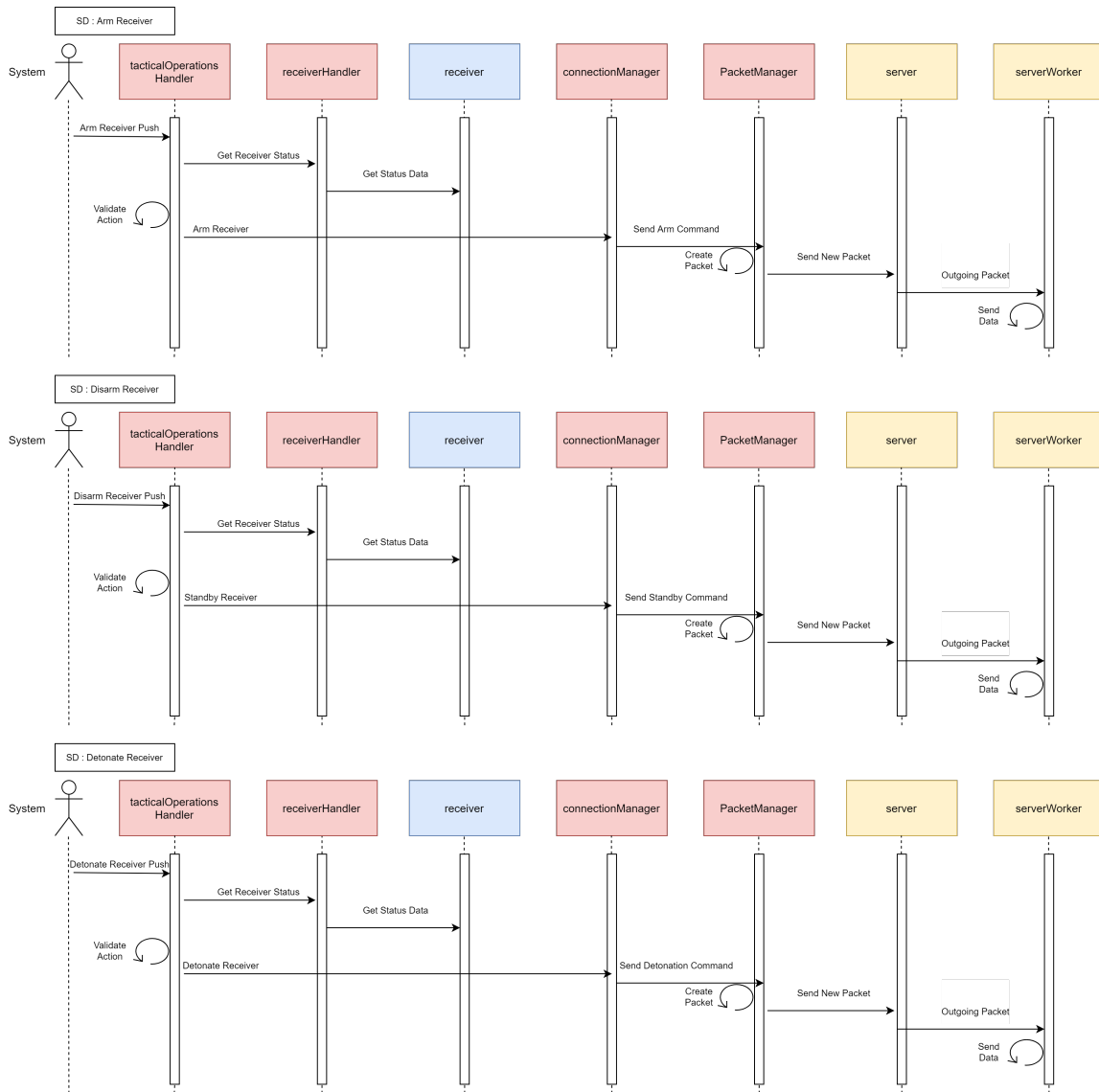


Figure 144: Tactical Operation Commands

### 8.11 Sequence Diagram Heartbeat

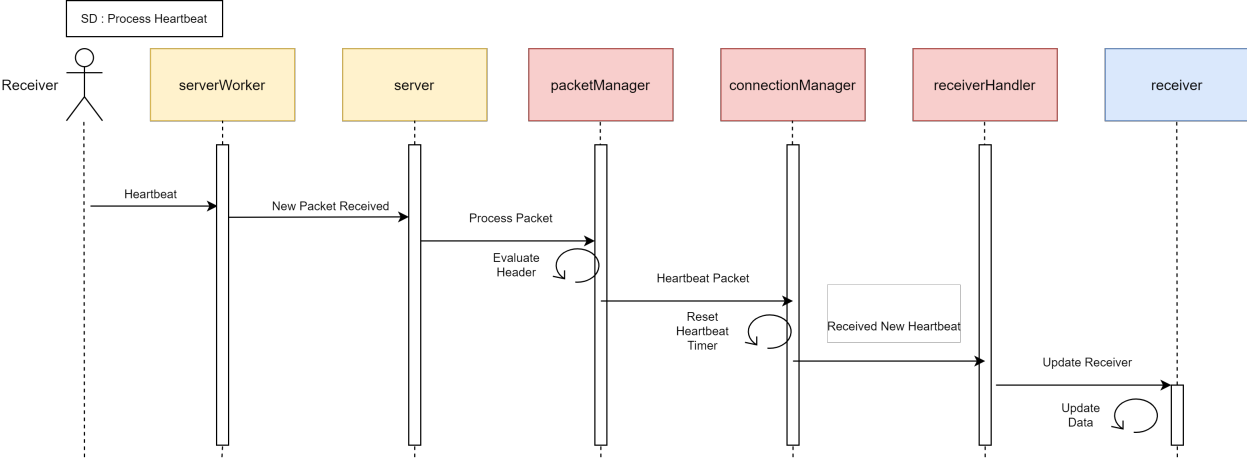


Figure 145: Processing a Heartbeat

## **8.12 Material Test Report (ILSS and Tensile Properties)**

Hasbergs vei 36

3616 KONGSBERG

NORWAY

---

# TEST REPORT

## SHORT-BEAM STRENGTH AND ULTIMATE TENSILE STRENGTH OF MODIFIED COMPOSITE MATERIAL

Anne Synnøve Brendøy

Bachelor project 2020, SPARK RFS

USN Mechanical Engineering Student

**19.05.2020**

Kåre Særen

Laboratory Manager

**TEST REQUESTED:**

Determine the mechanical properties (short beam strength and ultimate tensile strength) of modified composite material which is to be used in a bachelor project by Spark RFS.

Test laminates size (600x400) mm market with test numbers.

<b>MANUFACTURER LAMINATE</b>	Client <input checked="" type="checkbox"/> FTO/USN <input type="checkbox"/>
<b>CERTIFICATION OF CALIBRATION</b>	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/> EXP. OF CERTIFICATION: 26.11.2021
<b>VERIFICATION OF TEST RESULTS ACCORDING TO MATERIAL SEPCS</b>	<div style="border: 1px solid black; padding: 5px;">           Remarks:            SPARK RFS            Bachelor Project 2020         </div>
Client <input checked="" type="checkbox"/> FTO/USN <input type="checkbox"/>	
<b>COSOLIDATION METHOD:</b>	<b>MANUFACTURER SPECIMENS:</b>
Batch ID: Method: All Hand Lay-Up	Client <input checked="" type="checkbox"/> FTO/USN <input type="checkbox"/> <b>MACHINING METHODS SPECIMENS:</b> <div style="border: 1px solid black; padding: 5px;">Milled with tungsten carbide tooling</div>

**PLY ORIENTATION AND STACKING SEQUENCE OF THE LAMINATE**

BATCH ID:	LAY-UP:	NOTATION:
All batches	3 ply	(0,45,-45,90) <sub>3</sub>

## TEST REPORT SUMMARY:

### Materials and Methods:

Materials used were TUBALL MATRIX 201 (nanotubes), SvaPox 110 (epoxy resin), TL-1 (curing agent) and quadaxial fiberglass fabric (850g/m<sup>2</sup>). 75g nanotubes were dispersed in 20kg resin (0.3% ratio). OCSiAL has documentation on how to make sure the nanotubes are dispersed evenly.

In all laminate samples the pre-dispersed TUBALL/SvaPox and TL-1 are mixed 2 min by hand.

This laminate was produced 178 days after the nanotubes first got dispersed in SvaPox resin.

The plate specimens have been tested according to **ASTM D2344/ D2344M-13 Standard Test Method for Short-Beam Strength of Polymer Matrix Composite Materials and Their Laminates (ILSS)** and **ASTM D3039/ D3039m-00 Standard Test Method for Tensile Properties of Polymer Matrix Composite Materials** at USN facility.

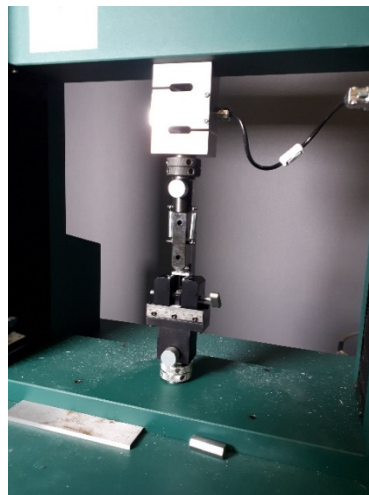
The ILSS (ASTM D2344) and The Tensile properties (ASTM D3039) tests were carried out using Tinius Olsen Testing Machine H25KL.

The test data were analyzed and recorded through the machine computer control system. The fracture of the test specimens was evaluated through visual inspection and through load-extension curves.

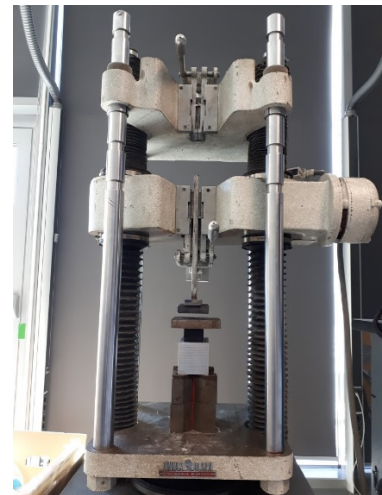
The assembly test was carried out on Tinius Olsen Company Super-L 300



H25KL



H25KL



Super-L 300

## PROCESS:

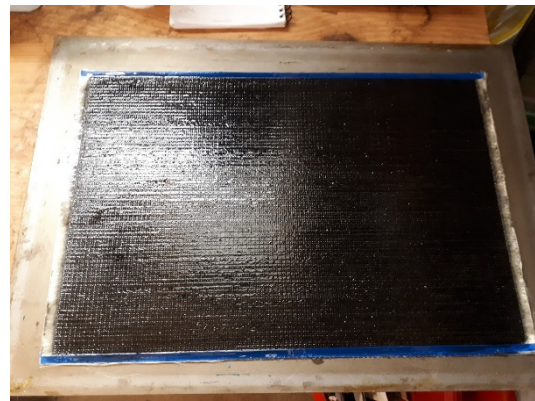
A metal plate was cleaned with PMC and prepared with 3 layers of Trennmittel T1-1 release with 20 minutes hardening between each release layer.

The nanotube enhanced epoxy resin and curing agent were mixed for 2 min by hand.

For these laminates it was used a quadaxial fiberglass fabric in 3 layers with nanotube enhanced epoxy resin, mixed with a curing agent, in between each layer. To ensure that the fabric was soaked thoroughly the two top layers were each rolled with an aluminum finned roller before continuing to the next layer.

The laminates were hardened in room temperature and post-cured in an oven at 60°C for four hours.

Each specimen was milled out with a tungsten carbide tool.



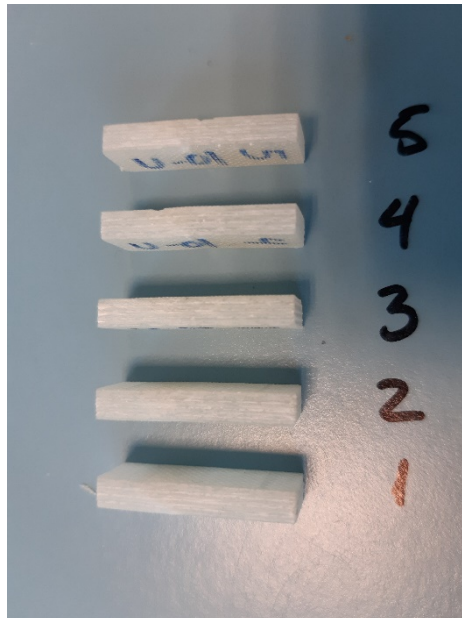




**SPECIMEN PROPERTIES ILSS:**

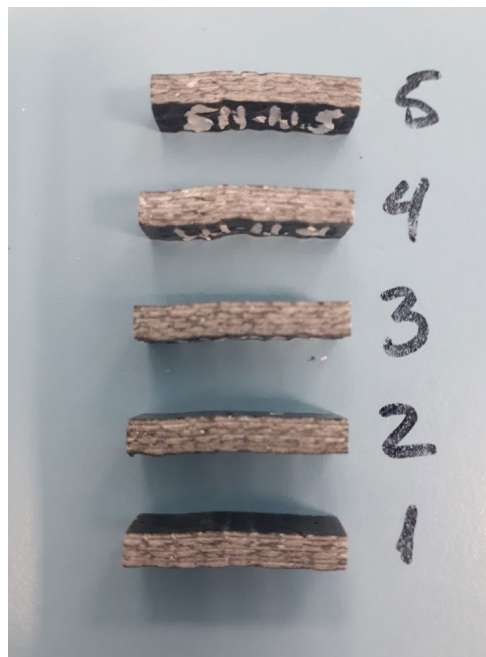
Batch ID:		SU-1.0i							
				Density:	Fabric	0,00256 g/mm <sup>3</sup>	Weight Fabric :	850 g/m <sup>2</sup>	
					Matrix	0,0011 g/mm <sup>3</sup>		0,00085 g/mm <sup>2</sup>	
Specimen ID:	Weight [g]	Length [mm]	Width [mm]	Thickness [mm]	Weight Fabric [%]	Volum Fabric [%]	SBS (1-5) [MPa]		
SU-1.0i-01	1,62	27,24	9,34	4,22	53,40 %	31,47 %	29,2		
SU-1.0i-02	1,65	27,33	9,34	4,19	52,60 %	31,70 %	27,5		
SU-1.0i-03	1,6	27,25	9,29	4,09	53,79 %	32,47 %	29,7		
SU-1.0i-04	1,57	27,13	9,26	4,12	54,41 %	32,24 %	28		
SU-1.0i-05	1,6	27,18	9,26	4,1	53,48 %	32,39 %	29,2		
<b>Average:</b>	1,608	27,226	9,298	4,144	53,54 %	32,05 %	28,72		

After ILSS test:



Batch ID:		SN-1.1i							
				Density:	Fabric	0,00256 g/mm <sup>3</sup>	Weight Fabric :	850 g/m <sup>2</sup>	
					Matrix	0,0011 g/mm <sup>3</sup>		0,00085 g/mm <sup>2</sup>	
Specimen ID:	Weight [g]	Length [mm]	Width [mm]	Thickness [mm]	Weight Fabric [%]	Volum Fabric [%]	SBS (1-5) [MPa]		
SN-1.1i-01	0,77	21,33	7,33	3,43	69,04 %	38,72 %	32,6		
SN-1.1i-02	0,8	21,24	7,29	3,55	65,81 %	37,41 %	33,4		
SN-1.1i-03	0,78	21,22	7,2	3,54	66,60 %	37,52 %	32,7		
SN-1.1i-04	0,76	21,24	7,24	3,44	68,80 %	38,61 %	31,6		
SN-1.1i-05	0,76	21,27	7,19	3,28	68,42 %	40,49 %	36,2		
<b>Average:</b>	0,774	21,26	7,25	3,448	67,73 %	38,55 %	33,3		

After ILSS test:



Batch ID:		SN-1.2i								
		Density:		Fabric	0,00256	g/mm <sup>3</sup>	Weight Fabric :	850	g/m <sup>2</sup>	
				Matrix	0,0011	g/mm <sup>3</sup>		0,00085	g/mm <sup>2</sup>	
Specimen ID:	Weight [g]	Length [mm]	Width [mm]	Thickness [mm]	Weight Fabric [%]	Volum Fabric [%]	SBS (1-5) [MPa]			
SN-1.2i-01	0,75	21,14	7,18	3,09	68,81 %	42,98 %	36,2			
SN-1.2i-02	0,72	21,11	7,15	3,13	71,28 %	42,43 %	33,7			
SN-1.2i-03	0,75	21,14	7,19	3,15	68,91 %	42,16 %	34,8			
SN-1.2i-04	0,74	21,17	7,22	3,21	70,23 %	41,37 %	31			
SN-1.2i-05	0,73	21,18	7,18	3,18	70,83 %	41,76 %	27,5			
<b>Average:</b>	0,738	21,148	7,184	3,152	70,01 %	42,14 %	32,64			

After ILSS test:



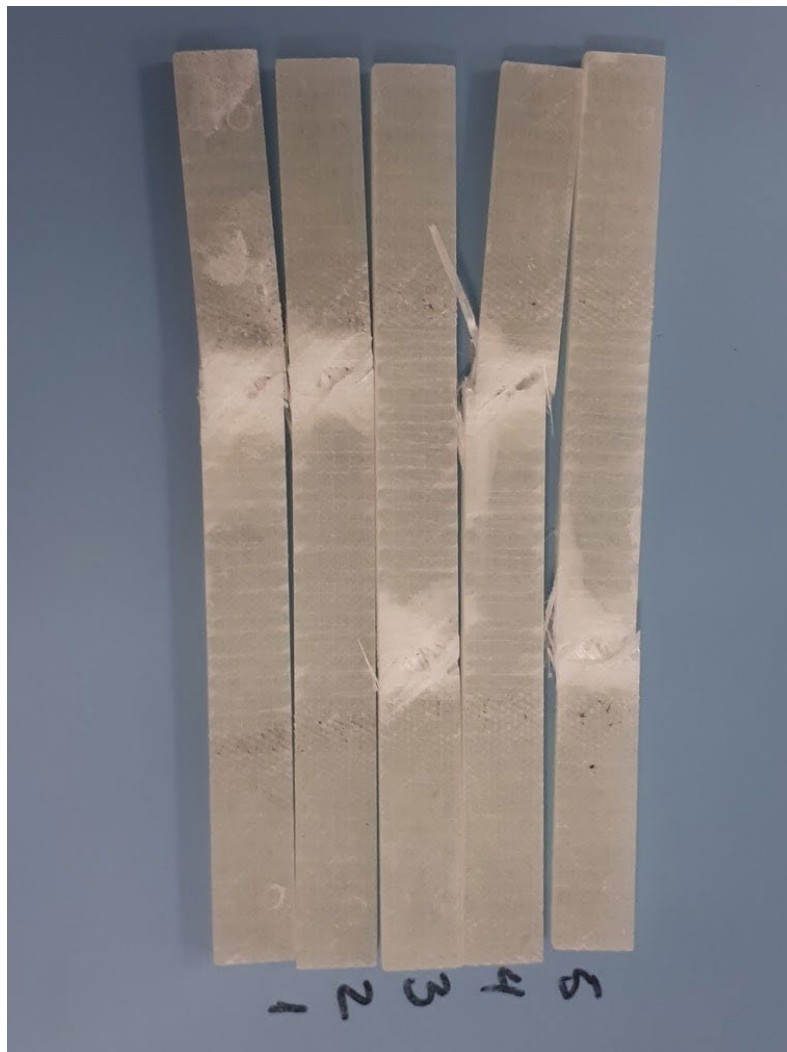
Batch ID:		SN-1.3i							
		Density:		Fabric	0,00256 g/mm <sup>3</sup>	Weight Fabric :	850 g/m <sup>2</sup>		
				Matrix	0,0011 g/mm <sup>3</sup>		0,00085 g/mm <sup>2</sup>		
Specimen ID:	Weight [g]	Length [mm]	Width [mm]	Thickness [mm]	Weight Fabric [%]	Volum Fabric [%]	SBS (1-5) [MPa]		
SN-1.3i-01	0,78	21,22	7,27	3,26	67,25 %	40,74 %	32,4		
SN-1.3i-02	0,73	21,16	7,16	3,14	70,56 %	42,30 %	32,8		
SN-1.3i-03	0,74	21,15	7,15	3,04	69,48 %	43,69 %	37,4		
SN-1.3i-04	0,72	21,12	7,13	3,04	71,11 %	43,69 %	40,9		
SN-1.3i-05	0,71	21,17	7,19	3,03	72,89 %	43,83 %	34		
<b>Average:</b>	0,736	21,164	7,18	3,102	70,26 %	42,85 %	35,5		

After ILSS test:



Batch ID:		SU-2.0s					
			Density:	Fabric	0,00256	g/mm <sup>3</sup>	
				Matrix	0,0011	g/mm <sup>3</sup>	
Specimen ID:	Weight [g]	Length [mm]	Width [mm]	Thickness [mm]	Weight Fabric [%]	Volum Fabric [%]	UTS [MPa]
SU-2.0-01	11,75	140,01	13,1566667	4,122	53,30 %	32,22 %	221
SU-2.0-02	11,83	139,87	13,1466667	4,182	52,85 %	31,76 %	225
SU-2.0-03	11,65	139,98	13,1533333	4,152	53,73 %	31,99 %	220
SU-2.0-04	11,85	139,99	13,1466667	4,208	52,80 %	31,56 %	230
SU-2.0-05	11,79	140,03	13,1333333	4,152	53,03 %	31,99 %	232
<b>Average:</b>	11,774	139,976	13,1473333	4,1632	53,15 %	31,90 %	225,6

After Stress-Strain test:



Batch ID:		SN-2.1s					
			Density:	Fabric	0,00256	g/mm <sup>3</sup>	
				Matrix	0,0011	g/mm <sup>3</sup>	
Specimen ID:	Weight [g]	Length [mm]	Width [mm]	Thickness [mm]	Weight Fabric [%]	Volum Fabric [%]	UTS [MPa]
SN-2.1-01	12,59	170,24	15,26	3,41	70,16 %	38,95 %	198
SN-2.1-02	13,1	170,27	15,2966667	3,48	67,60 %	38,16 %	206
SN-2.1-03	12,53	170,25	15,25	3,334	70,45 %	39,84 %	211
SN-2.1-04	12,24	170,23	15,2533333	3,536	72,13 %	37,56 %	198
SN-2.1-05	13,03	170,25	15,26	3,49	67,79 %	38,06 %	207
<b>Average:</b>	12,698	170,248	15,264	3,45	69,63 %	38,51 %	204

After Stress-Strain test:





Batch ID:		SN-2.2s					
		Density:		Fabric	0,00256	g/mm <sup>3</sup>	
				Matrix	0,0011	g/mm <sup>3</sup>	
Specimen ID:	Weight [g]	Length [mm]	Width [mm]	Thickness [mm]	Weight Fabric [%]	Volum Fabric [%]	UTS [MPa]
SN-2.2-01	12,51	170,19	15,2	3,249	70,31 %	40,88 %	231
SN-2.2-02	12,77	170,23	15,22	3,32	68,98 %	40,00 %	234
SN-2.2-03	12,52	170,21	15,1933333	3,31	70,23 %	40,12 %	229
SN-2.2-04	12,75	170,21	15,2366667	3,29	69,16 %	40,37 %	221
SN-2.2-05	12,52	170,25	15,2866667	3,338	70,68 %	39,79 %	231
<b>Average:</b>	12,614	170,218	15,2273333	3,3014	69,87 %	40,23 %	229,2

After Stress-Strain test:



Batch ID:		SN-2.3s					
			Density:	Fabric	0,00256	g/mm <sup>3</sup>	
				Matrix	0,0011	g/mm <sup>3</sup>	
Specimen ID:	Weight [g]	Length [mm]	Width [mm]	Thickness [mm]	Weight Fabric [%]	Volum Fabric [%]	UTS [MPa]
SN-2.3-01	12,24	170,13	15,2366667	3,098	72,01 %	42,87 %	212
SN-2.3-02	12,19	170,15	15,1966667	3,066	72,12 %	43,32 %	209
SN-2.3-03	12,17	170,11	15,23	3,108	72,38 %	42,73 %	208
SN-2.3-04	12,19	170,11	15,2566667	3,094	72,39 %	42,93 %	216
SN-2.3-05	12,22	170,14	15,23	3,148	72,10 %	42,19 %	198
<b>Average:</b>	12,202	170,128	15,23	3,1028	72,20 %	42,81 %	208,6

After Stress-Strain test:

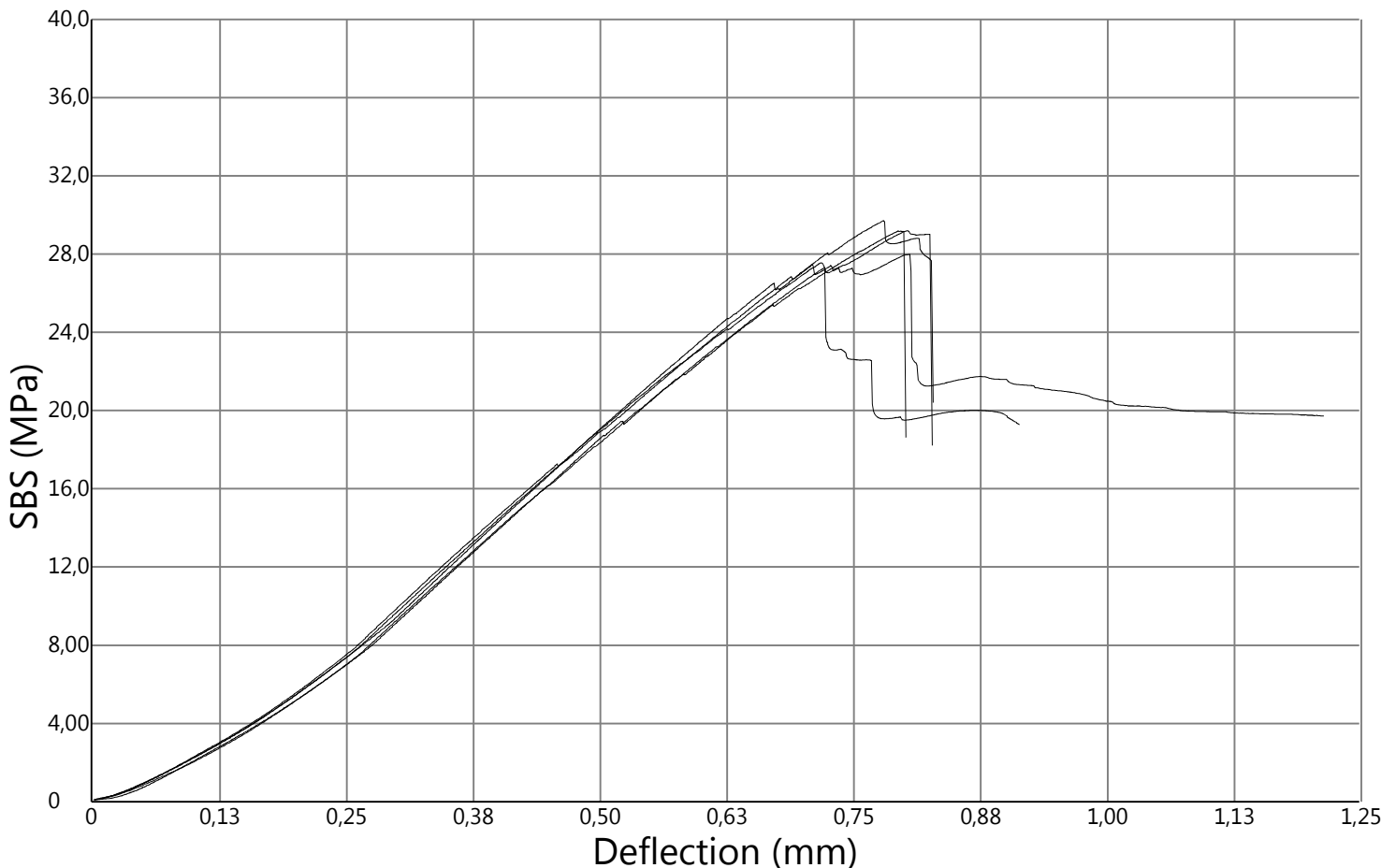




Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Kåre Særen  
 Date Tested: 04 des, 2019  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 19,0 °C  
 Humidity (Entry): 24,0 %  
 Batch ID.: U-01

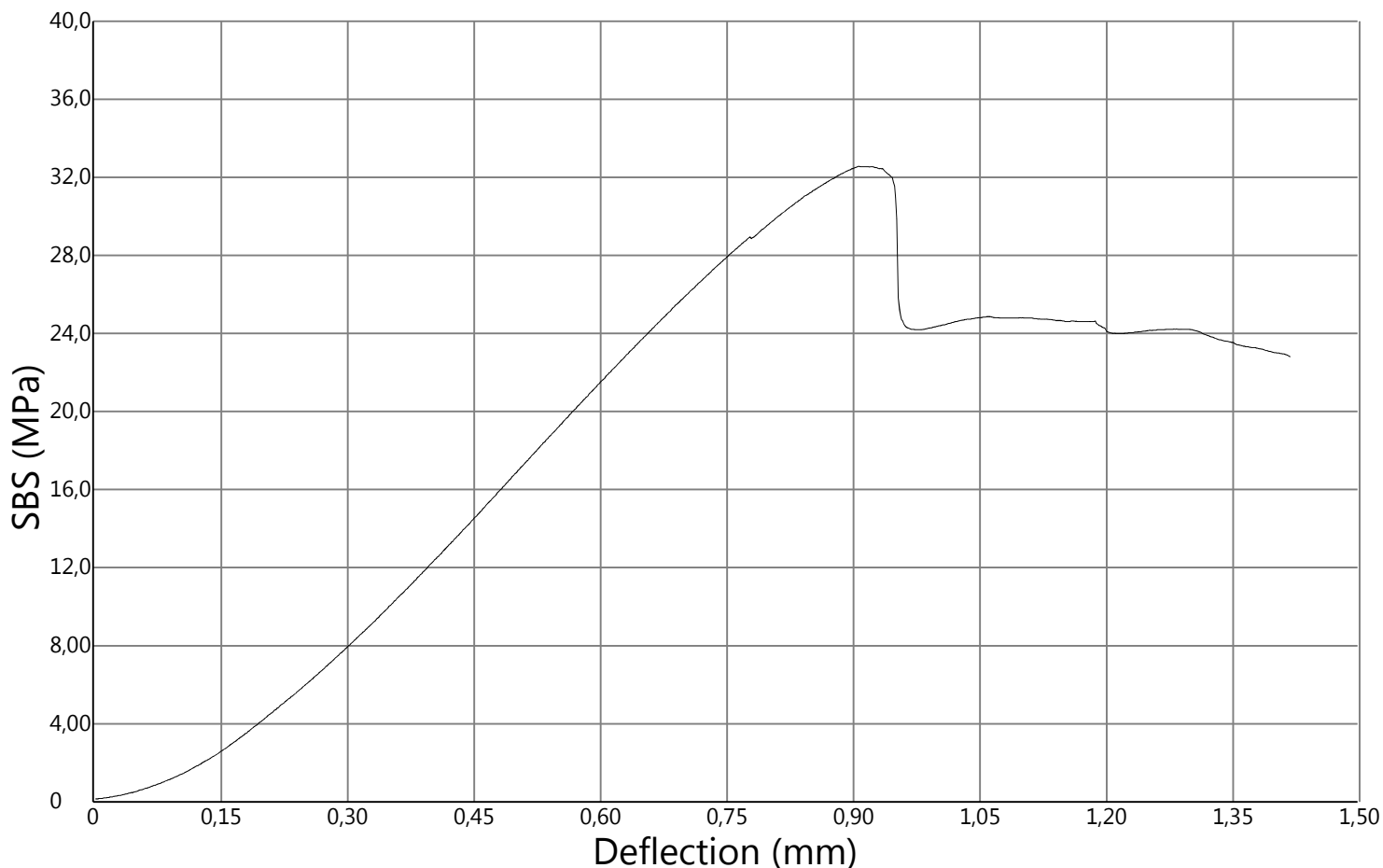
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Short-Beam Strength MPa	Failure mode
1	9,34	4,22	1 534	1 533	29,2	Interlaminar Shear
2	9,34	4,19	1 438	1 006	27,5	Interlaminar Shear
3	9,29	4,09	1 505	1 501	29,7	Interlaminar Shear
4	9,26	4,12	1 423	1 003	28,0	Interlaminar Shear
5	9,26	4,10	1 477	1 468	29,2	Interlaminar Shear
Average			1 475	1 302	28,7	
SD			45,8		0,91	
CoV			3,11		3,17	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.1i

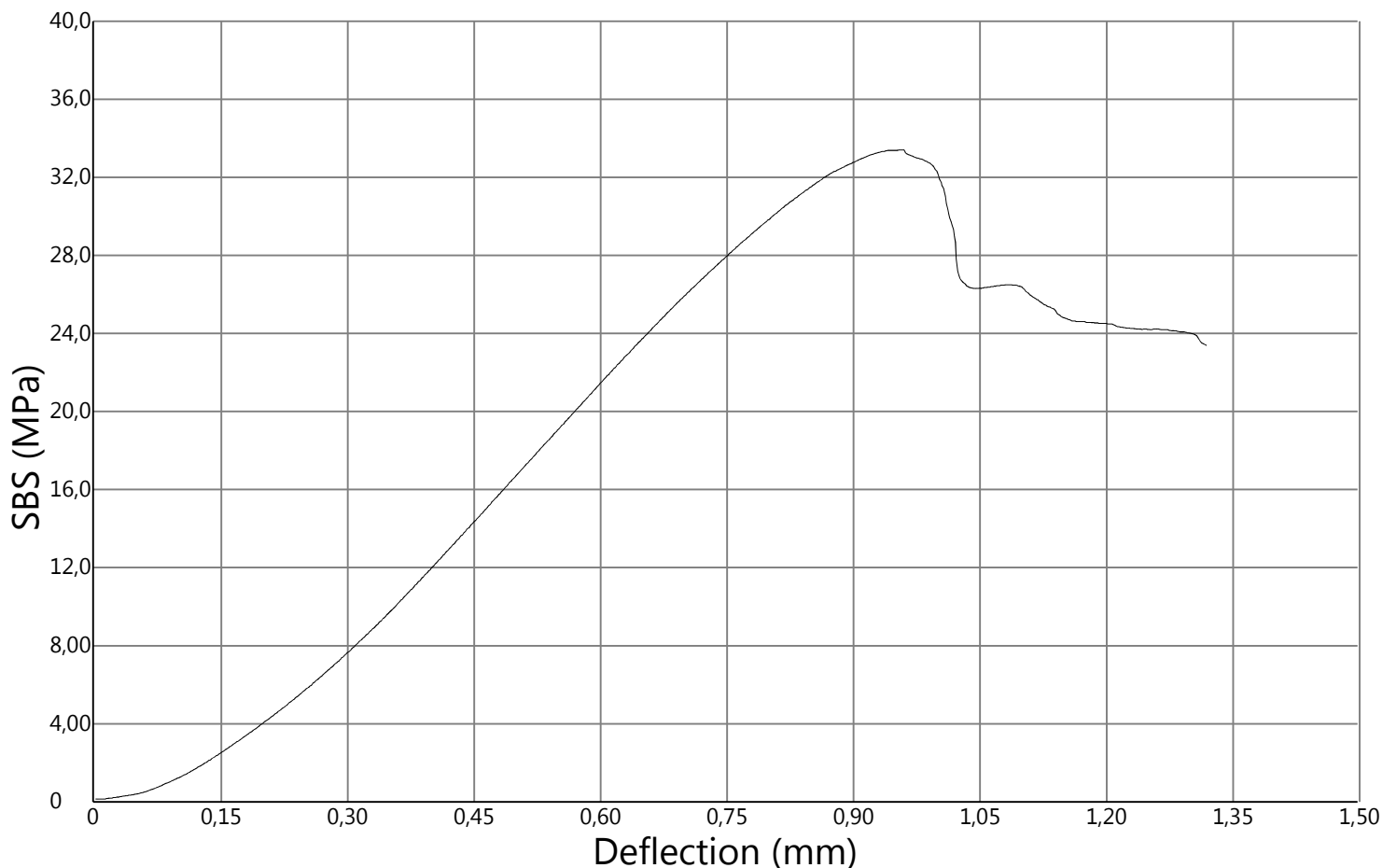
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
1	7,33	3,43	1 092	764	1,42	32,6	ILS
Average	7,33	3,43	1 092	764	1,42	32,6	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.1i

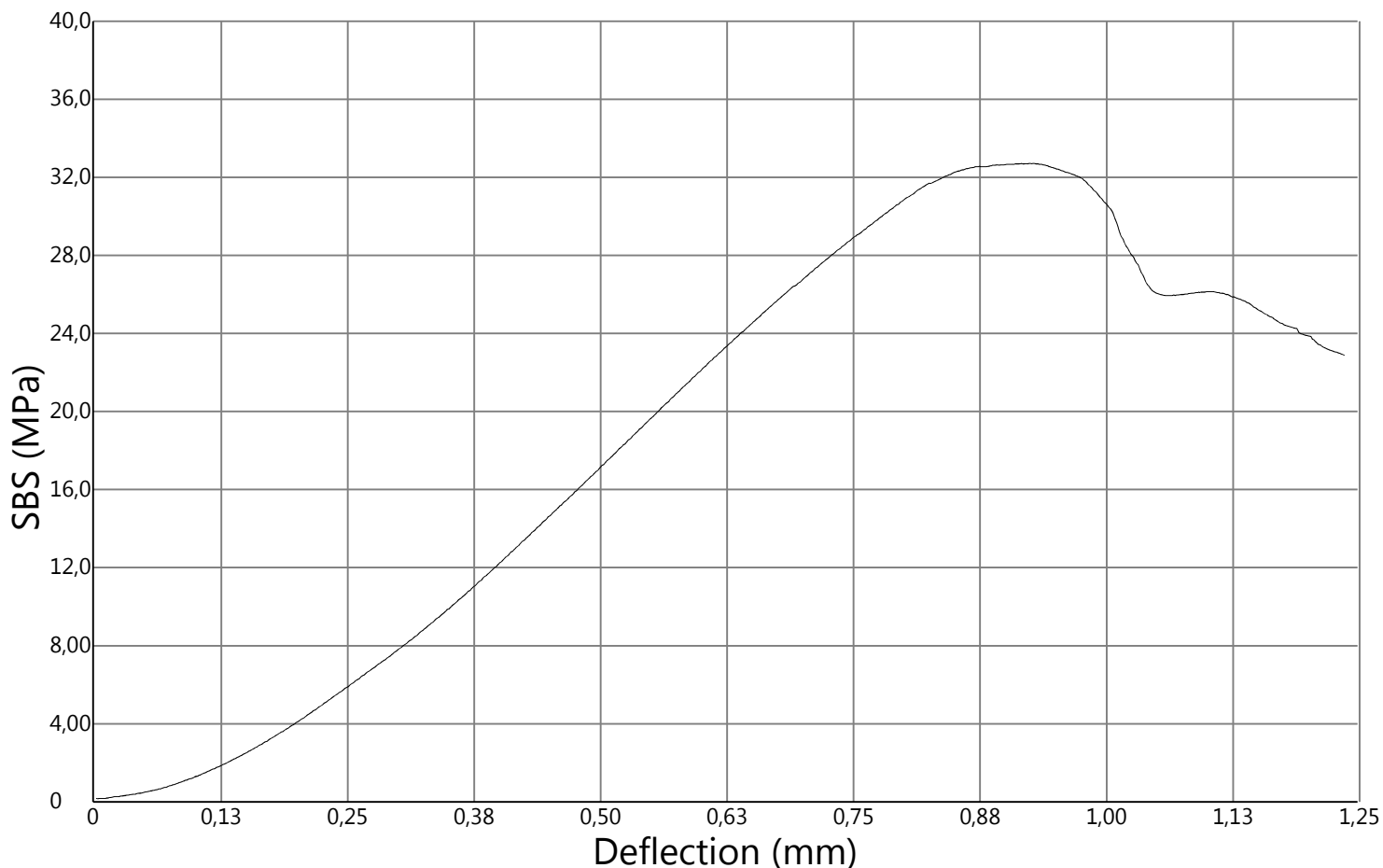
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
2	7,29	3,55	1 153	807	1,32	33,4	ILS
Average	7,29	3,55	1 153	807	1,32	33,4	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.1i

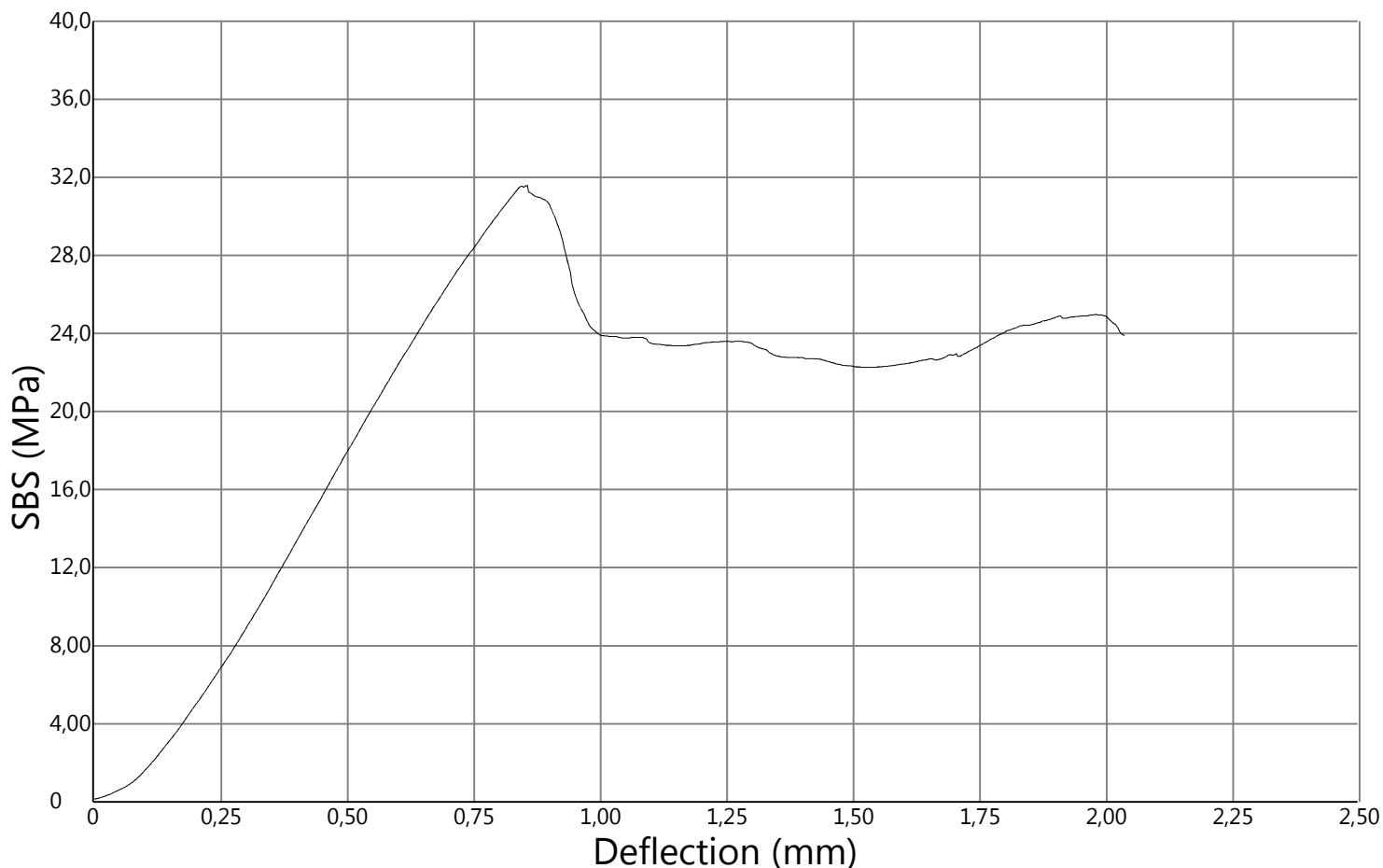
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
3	7,20	3,54	1 112	778	1,24	32,7	ILS
Average	7,20	3,54	1 112	778	1,24	32,7	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.1i

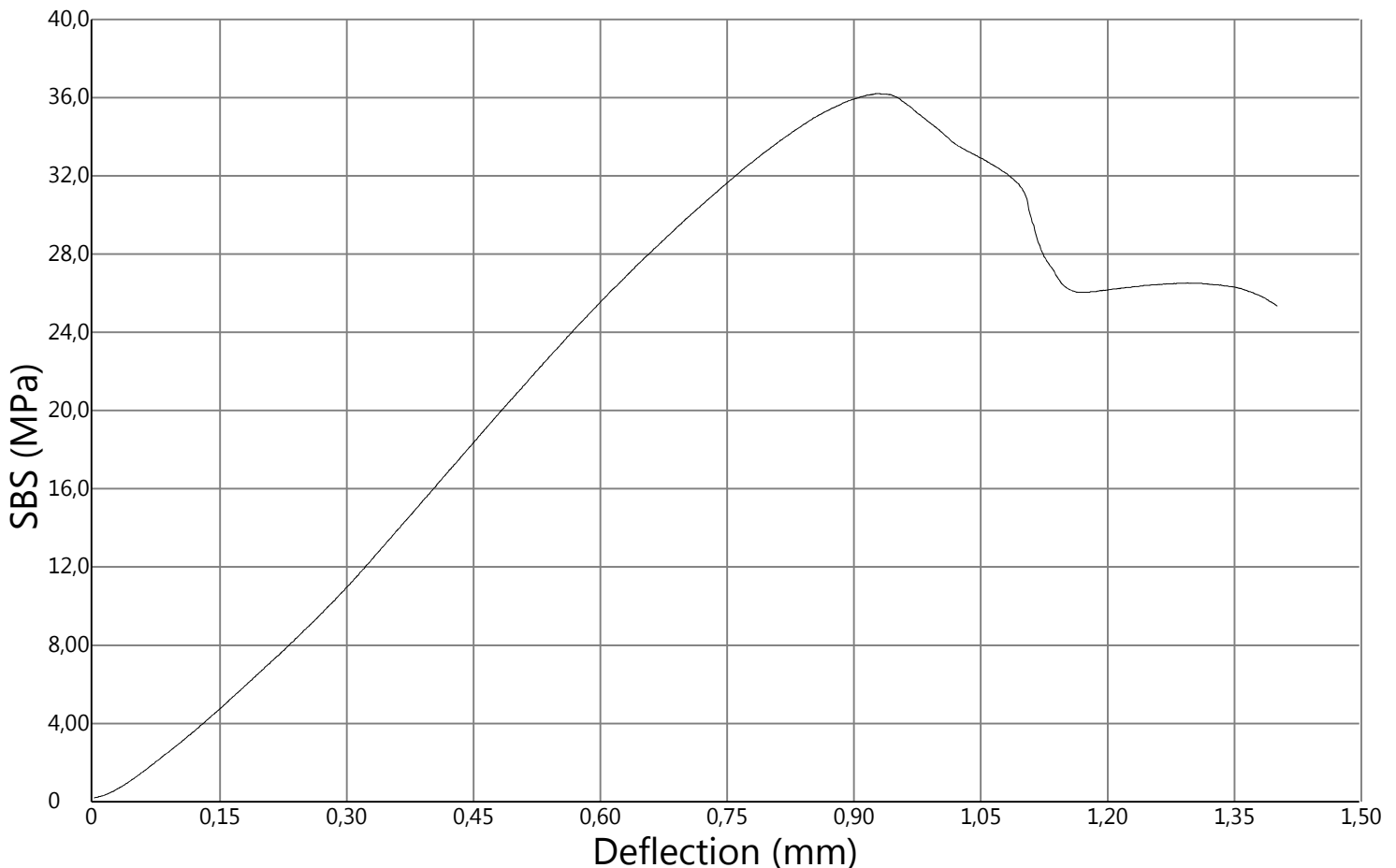
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
4	7,24	3,44	1 048	793	2,04	31,6	ILS
Average	7,24	3,44	1 048	793	2,04	31,6	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.1i

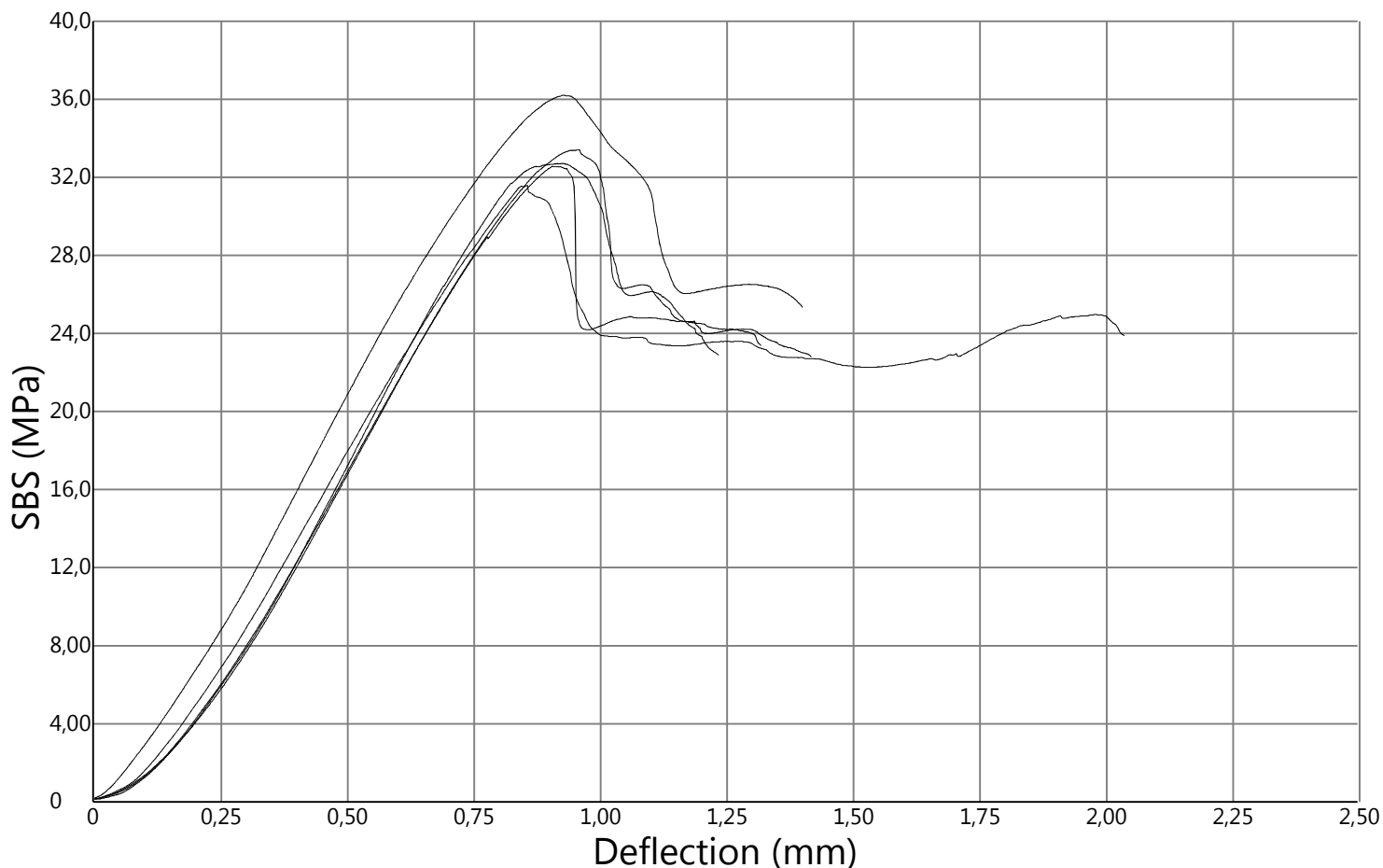
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
5	7,19	3,28	1 138	797	1,40	36,2	ILS
Average	7,19	3,28	1 138	797	1,40	36,2	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.1i

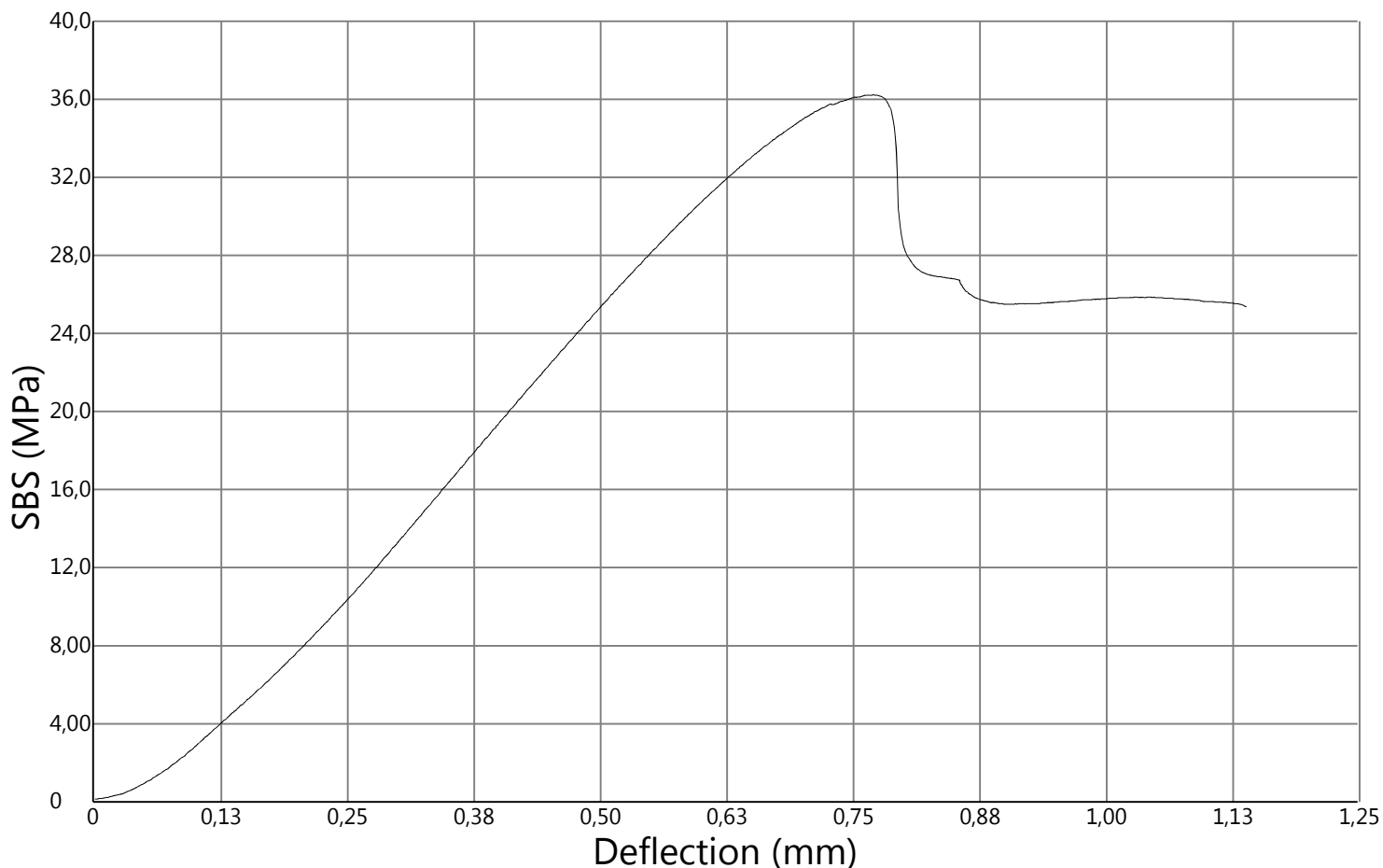
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
1	7,33	3,43	1 092	764	1,42	32,6	ILS
2	7,29	3,55	1 153	807	1,32	33,4	ILS
3	7,20	3,54	1 112	778	1,24	32,7	ILS
4	7,24	3,44	1 048	793	2,04	31,6	ILS
5	7,19	3,28	1 138	797	1,40	36,2	ILS
Average	7,25	3,45	1 108	788	1,48	33,3	
SD			41,0		0,32	1,75	
CoV			3,70			5,27	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.2i

Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
1	7,18	3,09	1 072	750	1,14	36,2	ILS
Average	7,18	3,09	1 072	750	1,14	36,2	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	

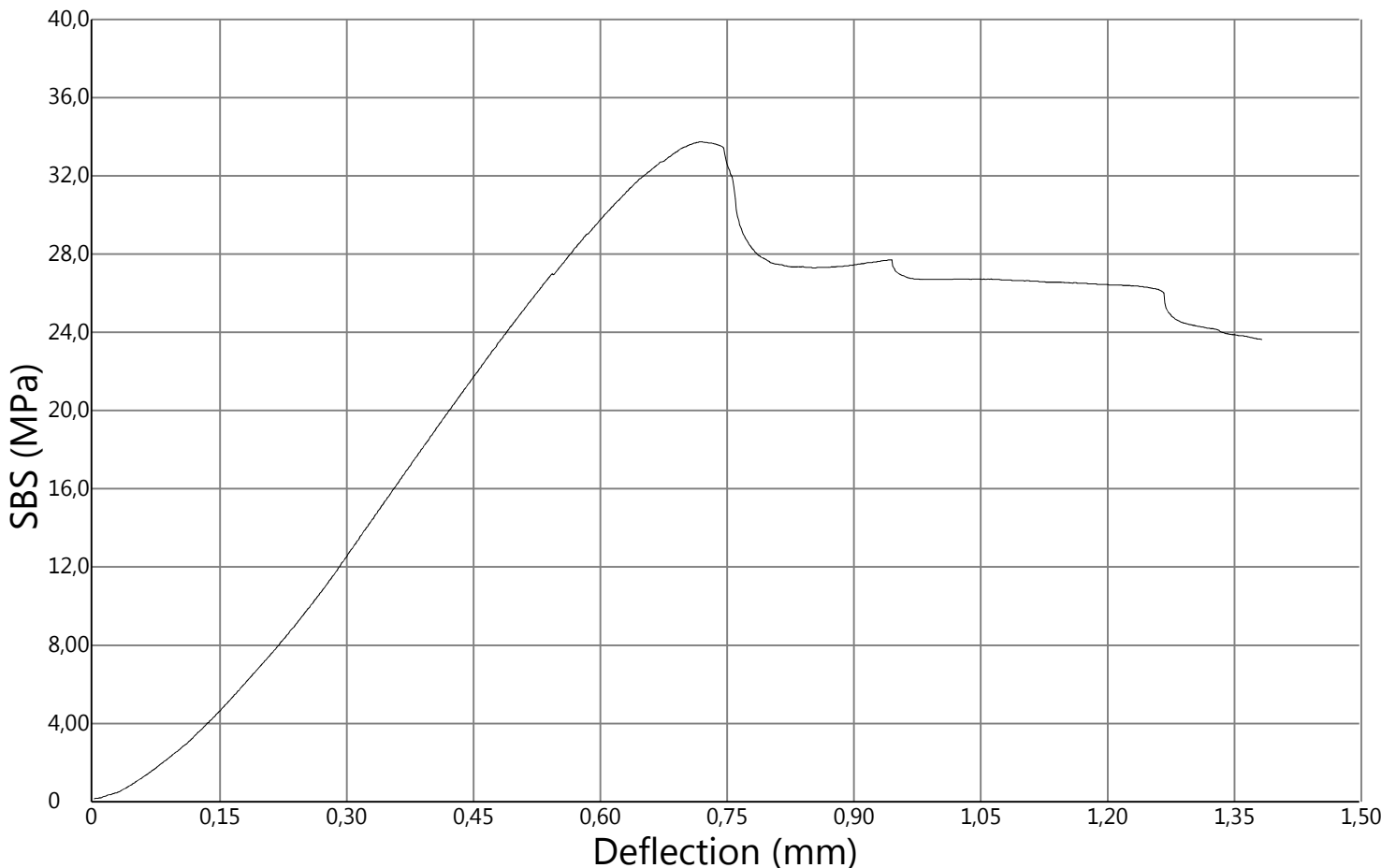




Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.2i

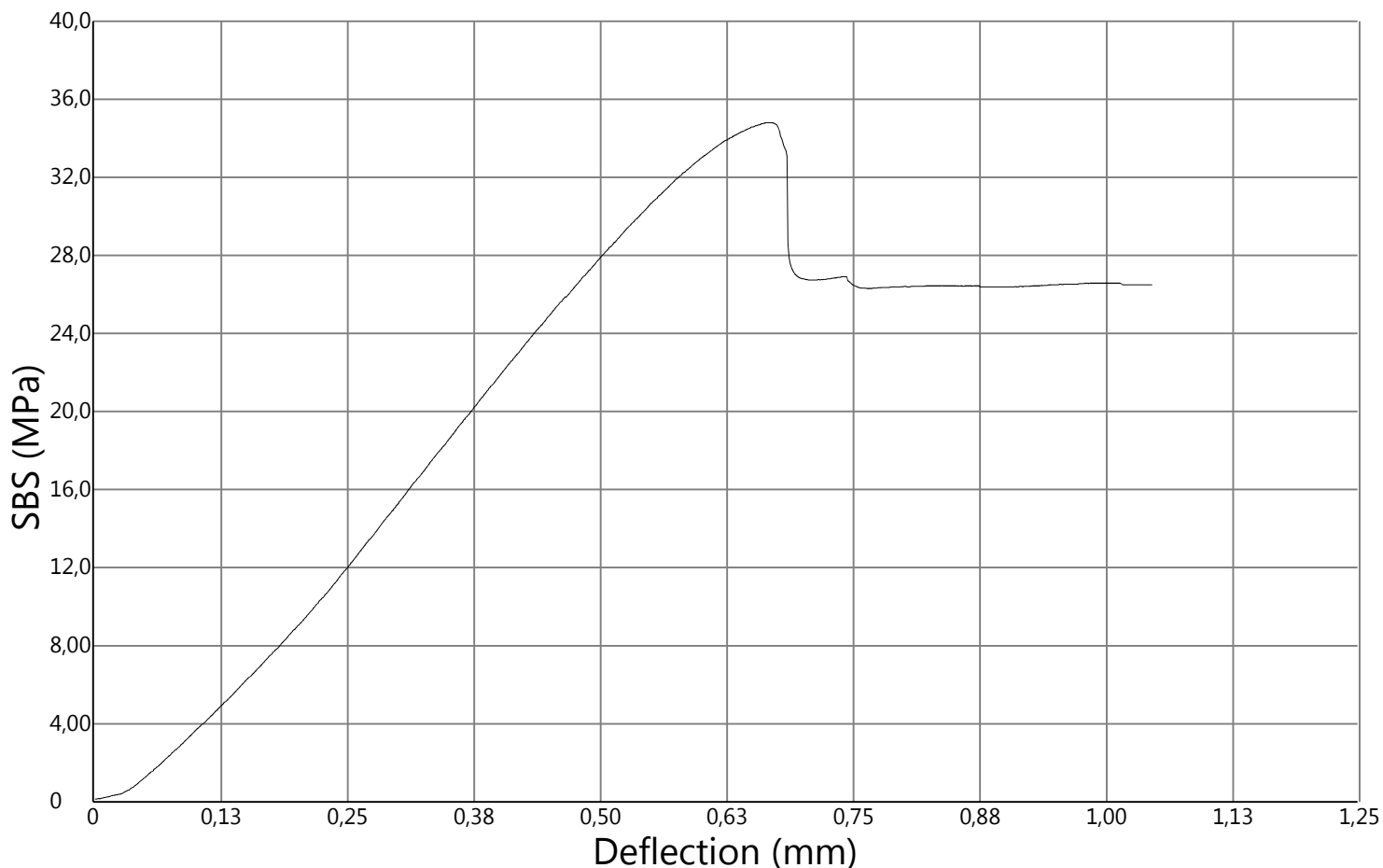
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
2	7,15	3,13	1 007	705	1,39	33,7	ILS
Average	7,15	3,13	1 007	705	1,39	33,7	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.2i

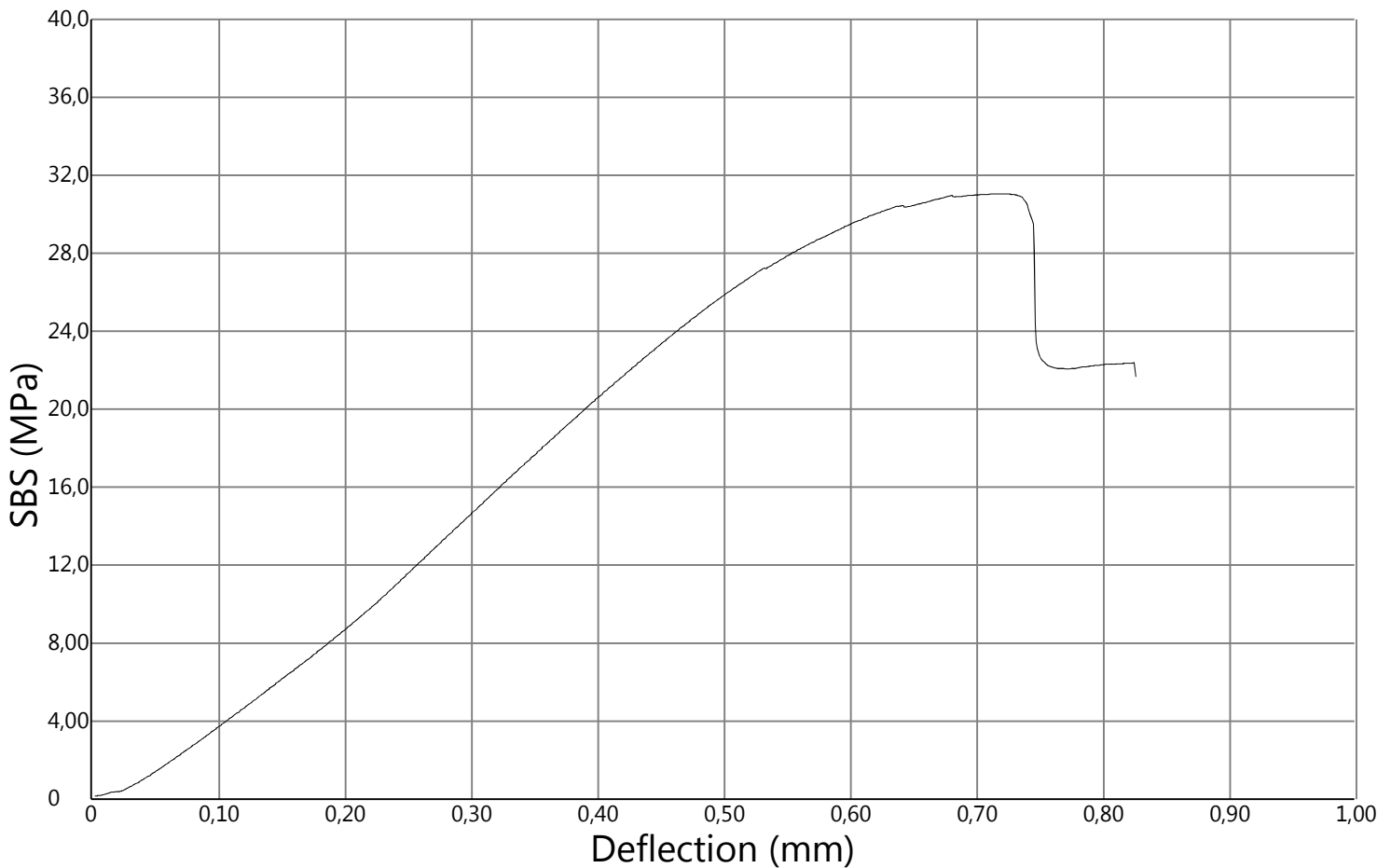
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
3	7,19	3,15	1 051	800	1,05	34,8	ILS
Average	7,19	3,15	1 051	800	1,05	34,8	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.2i

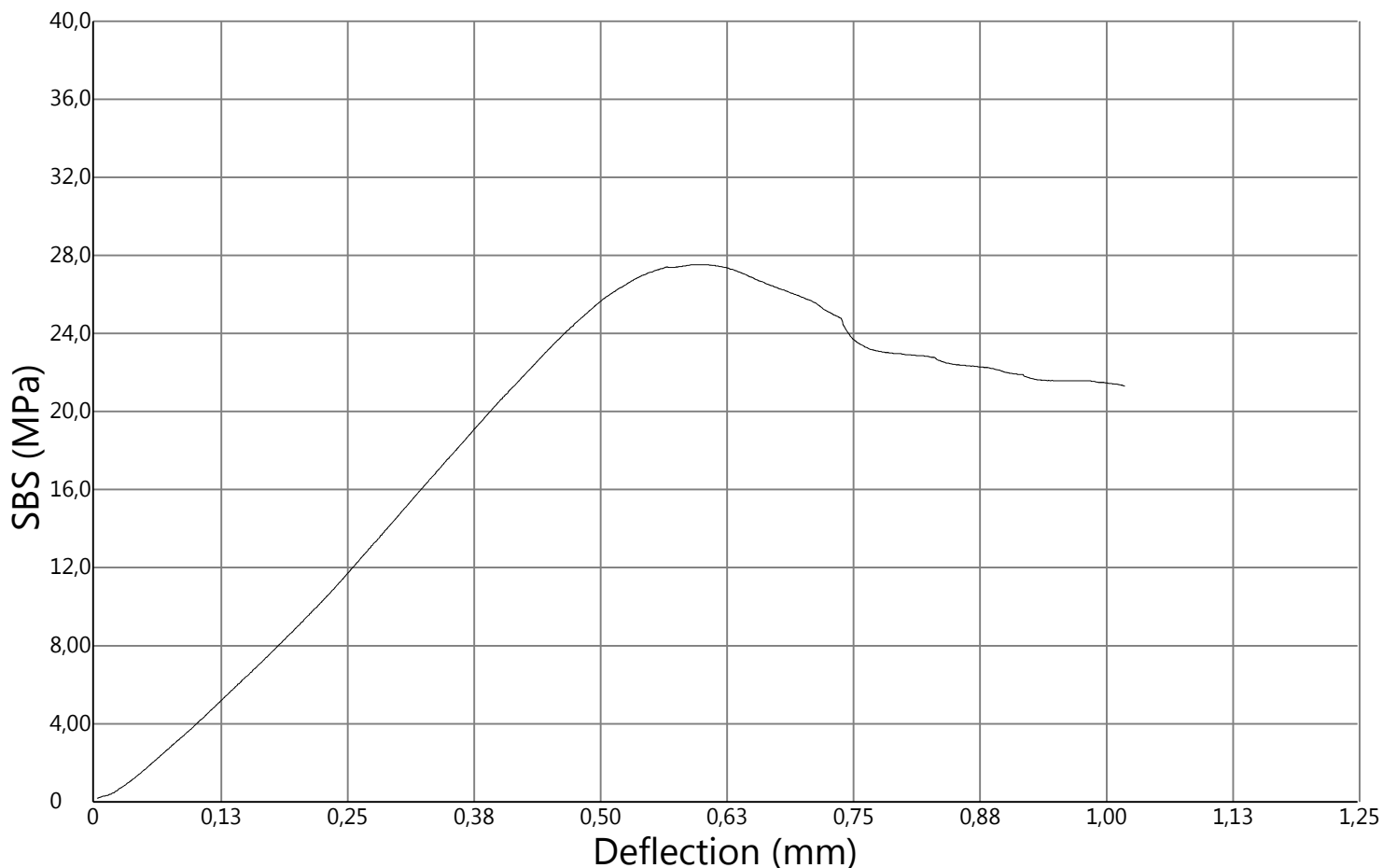
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
4	7,22	3,21	959	911	0,75	31,0	ILS
Average	7,22	3,21	959	911	0,75	31,0	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.2i

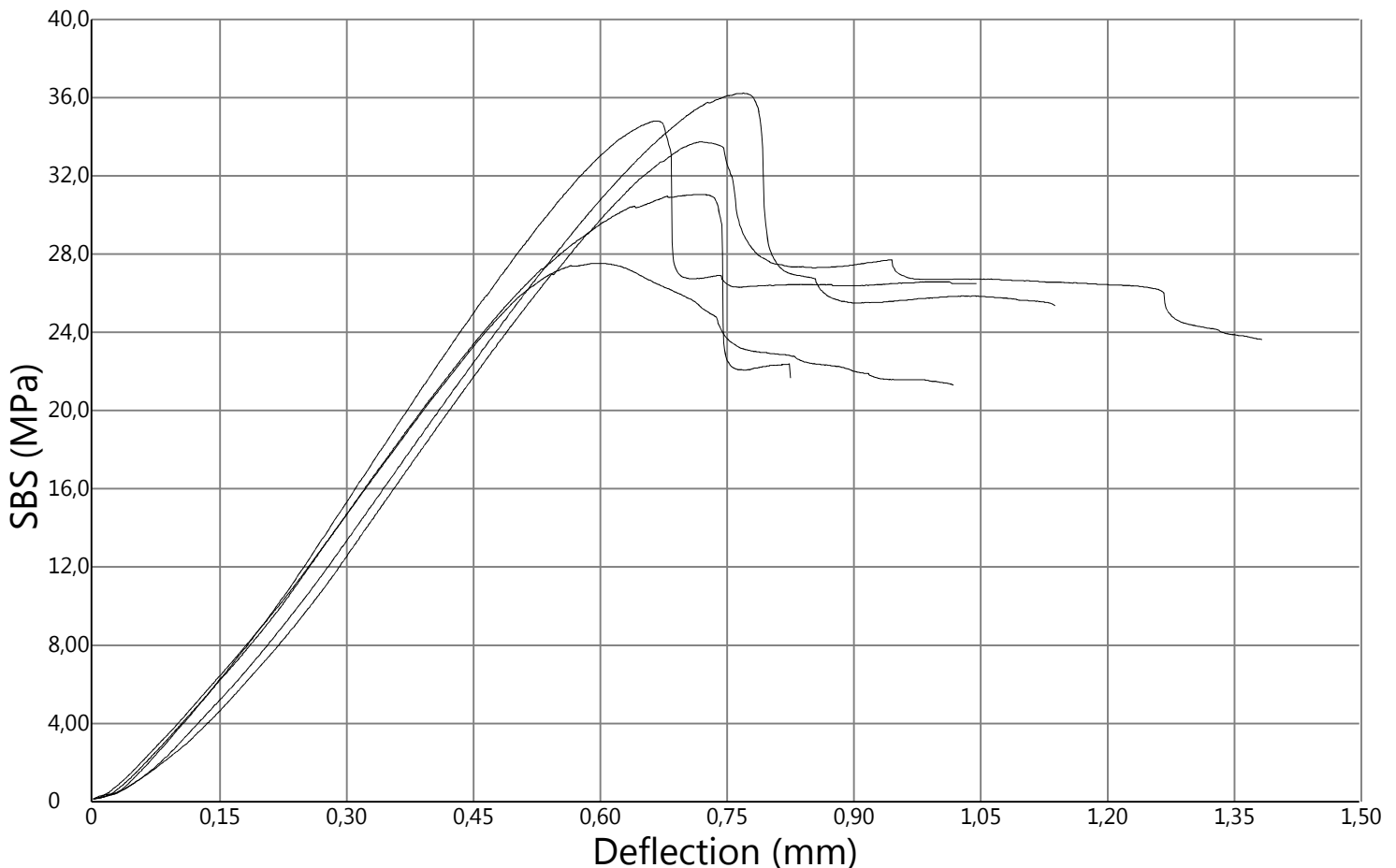
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
5	7,18	3,18	838	648	1,02	27,5	ILS
Average	7,18	3,18	838	648	1,02	27,5	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.2i

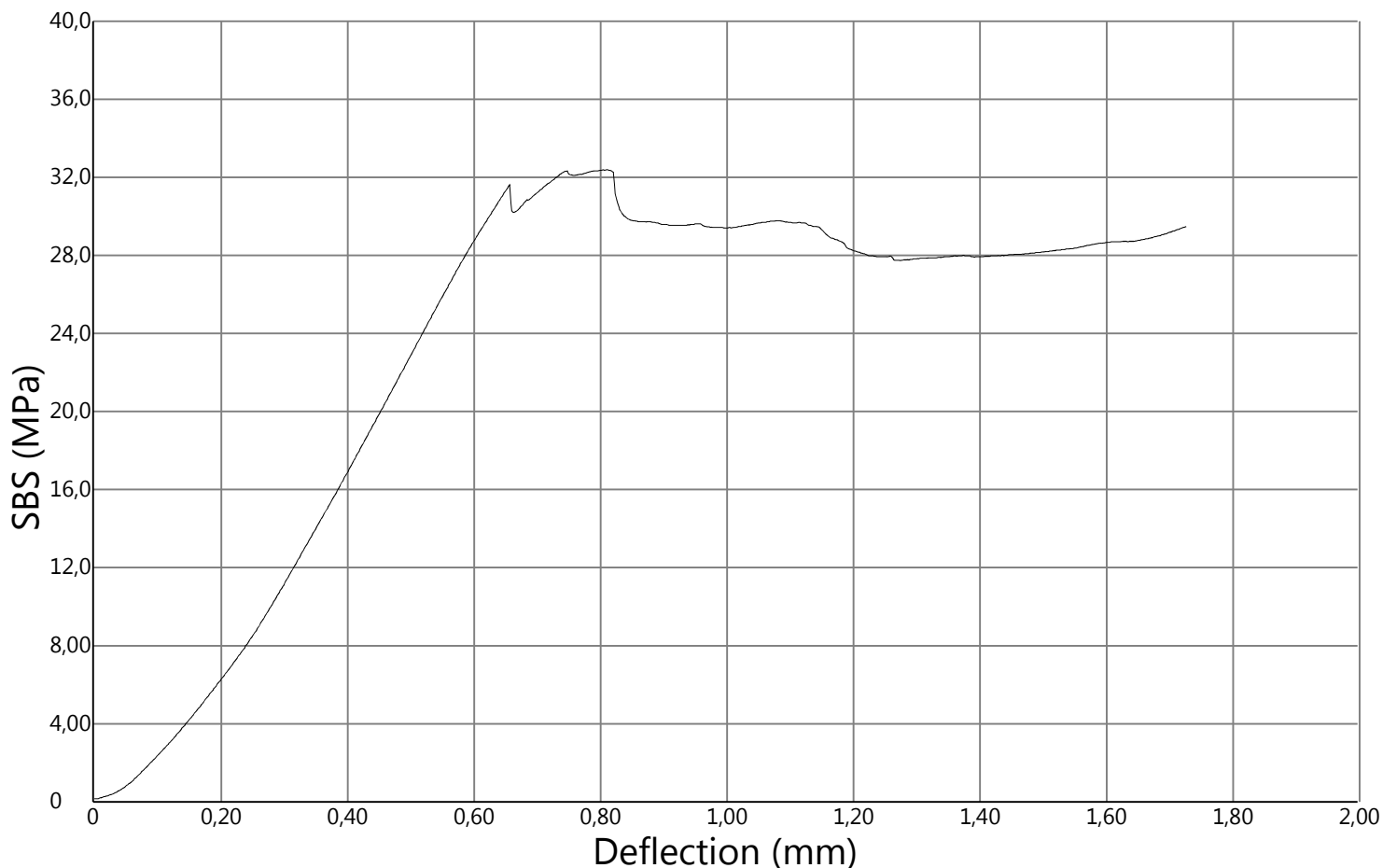
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
1	7,18	3,09	1 072	750	1,14	36,2	ILS
2	7,15	3,13	1 007	705	1,39	33,7	ILS
3	7,19	3,15	1 051	800	1,05	34,8	ILS
4	7,22	3,21	959	911	0,75	31,0	ILS
5	7,18	3,18	838	648	1,02	27,5	ILS
Average	7,18	3,15	985	763	1,07	32,7	
SD			93,2		0,23	3,45	
CoV			9,46			10,6	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.3i

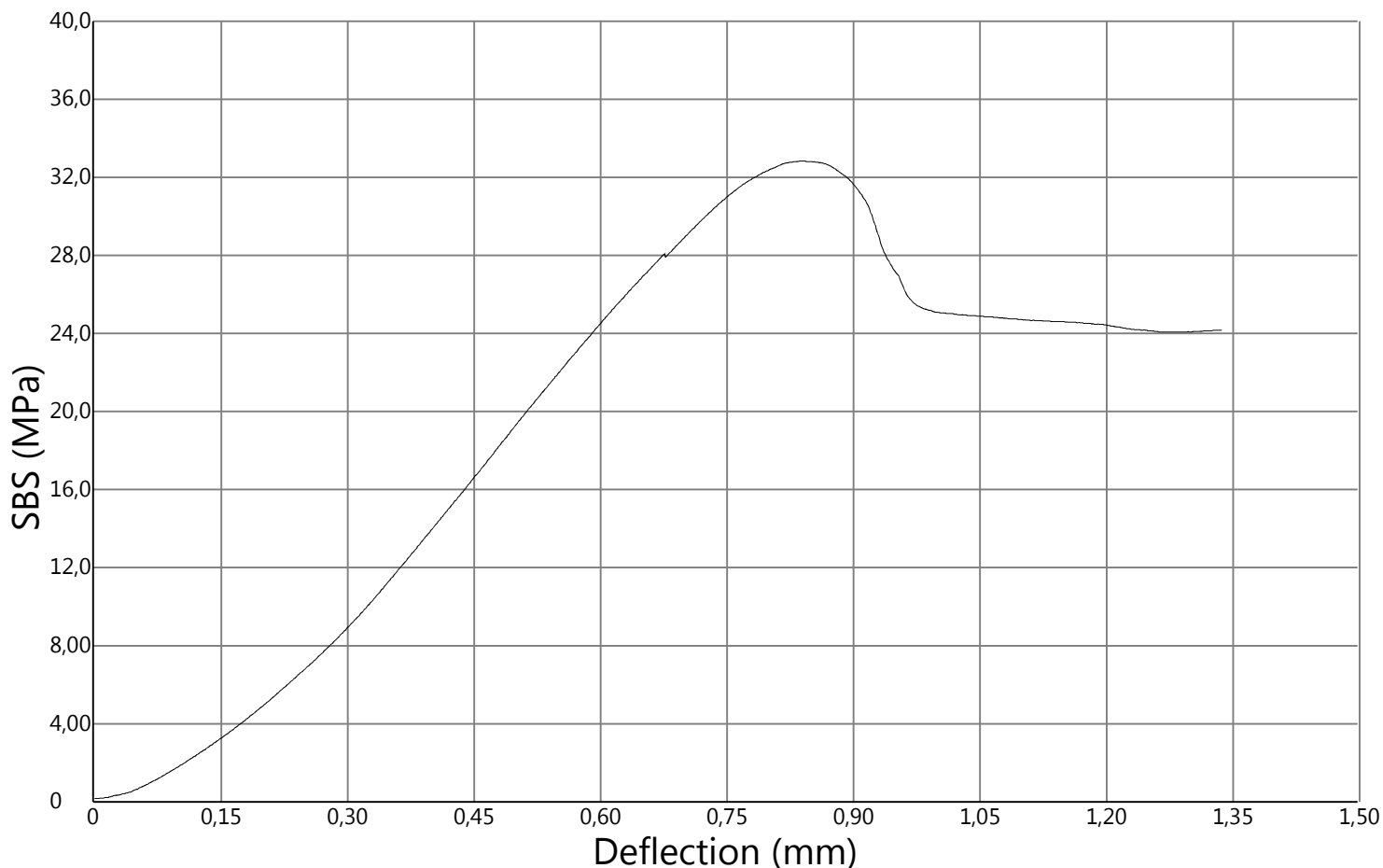
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
1	7,27	3,26	1 023	932	1,73	32,4	ILS
Average	7,27	3,26	1 023	932	1,73	32,4	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.3i

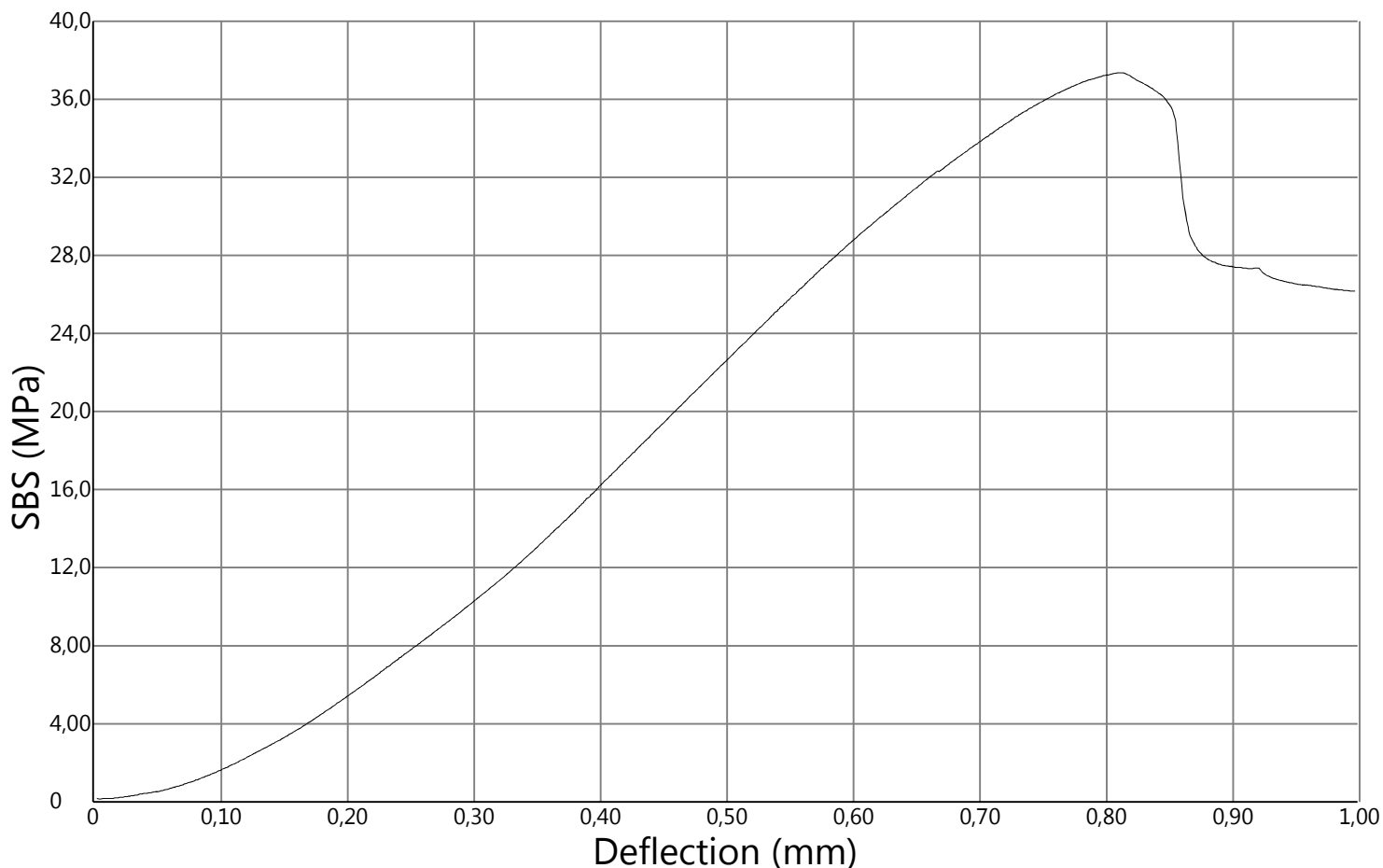
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
2	7,16	3,14	984	724	1,34	32,8	ILS
Average	7,16	3,14	984	724	1,34	32,8	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.3i

Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
3	7,15	3,04	1 083	758	1,00	37,4	ILS
Average	7,15	3,04	1 083	758	1,00	37,4	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	

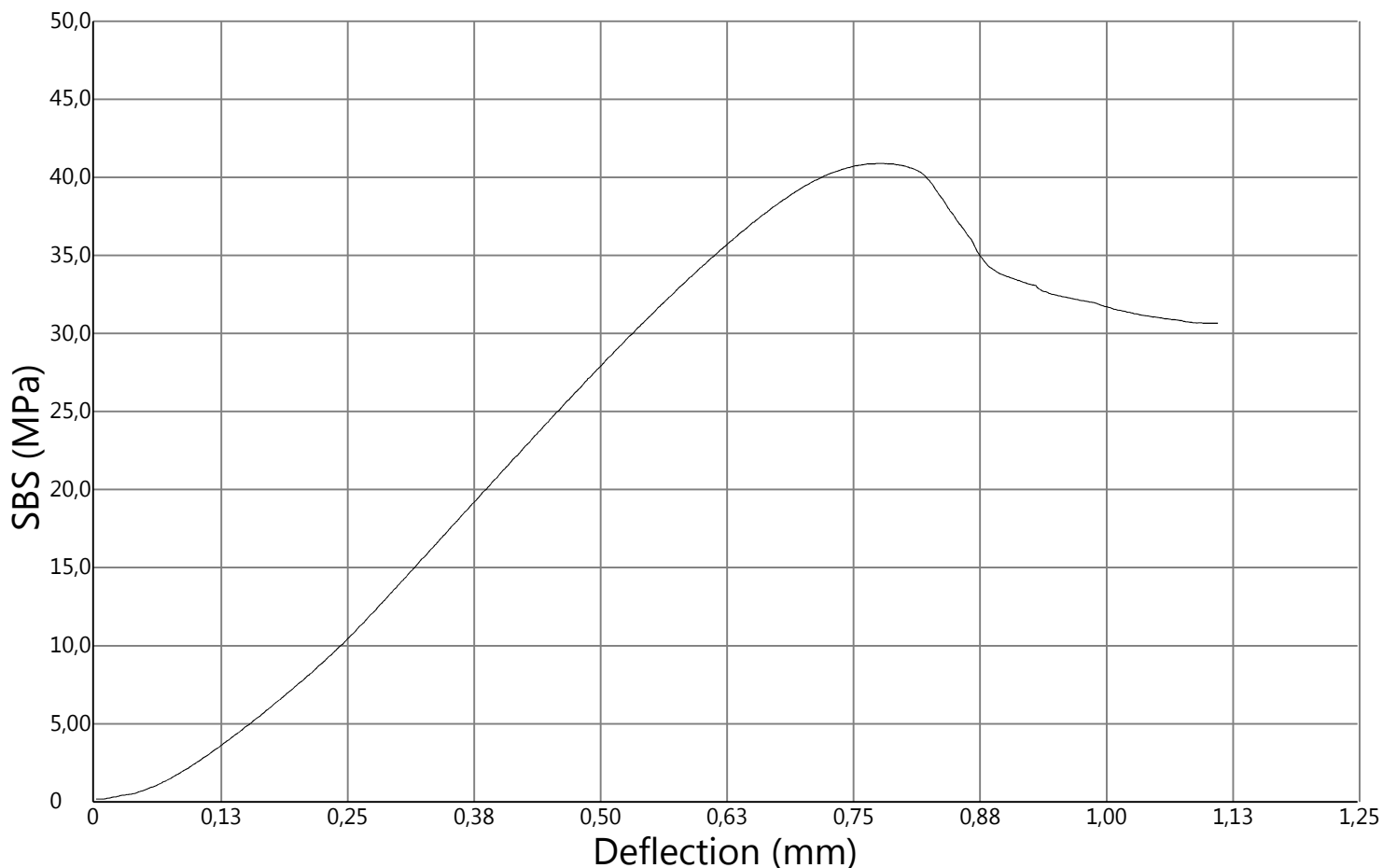




Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.3i

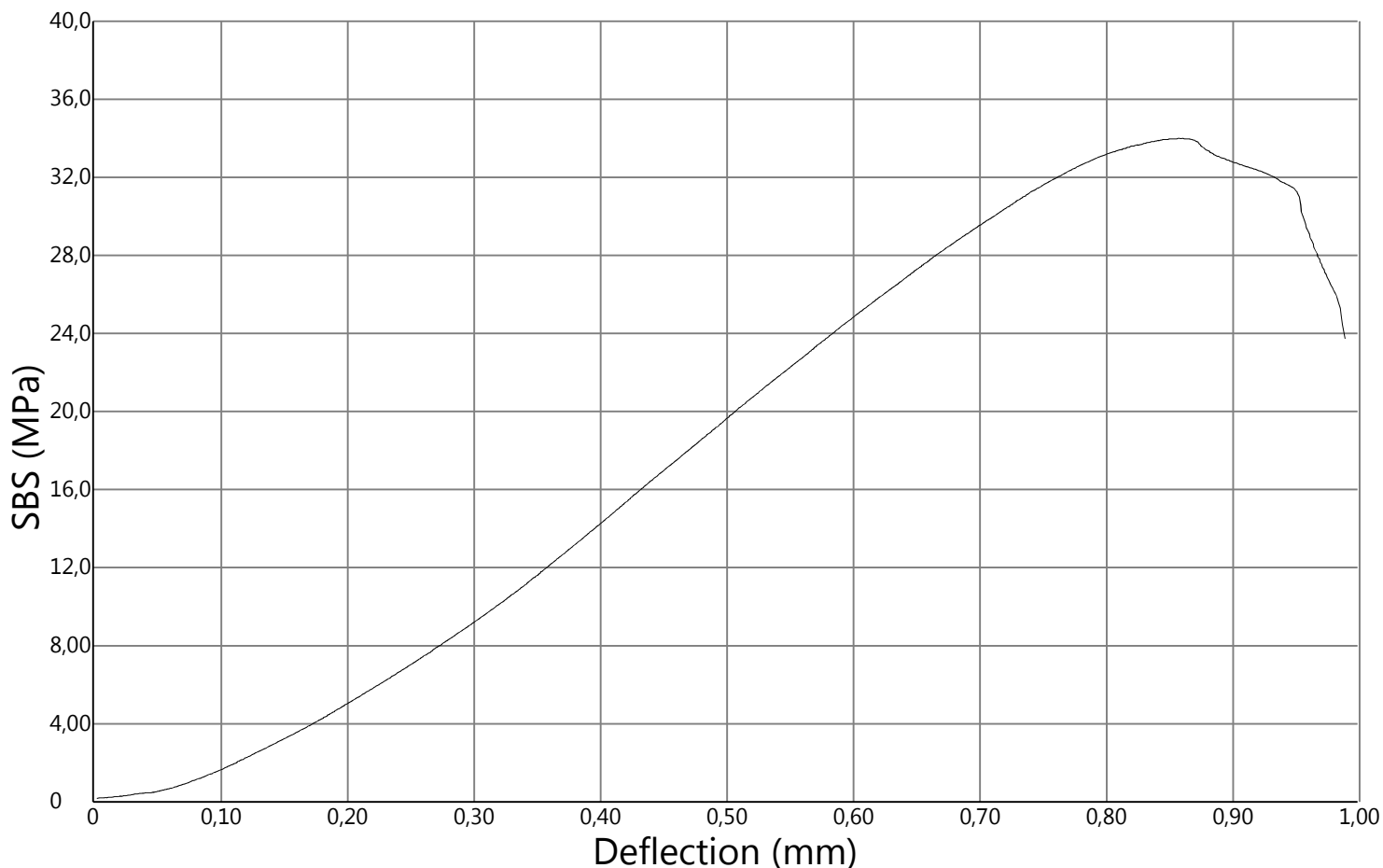
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
4	7,13	3,04	1 182	886	1,11	40,9	ILS
Average	7,13	3,04	1 182	886	1,11	40,9	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.3i

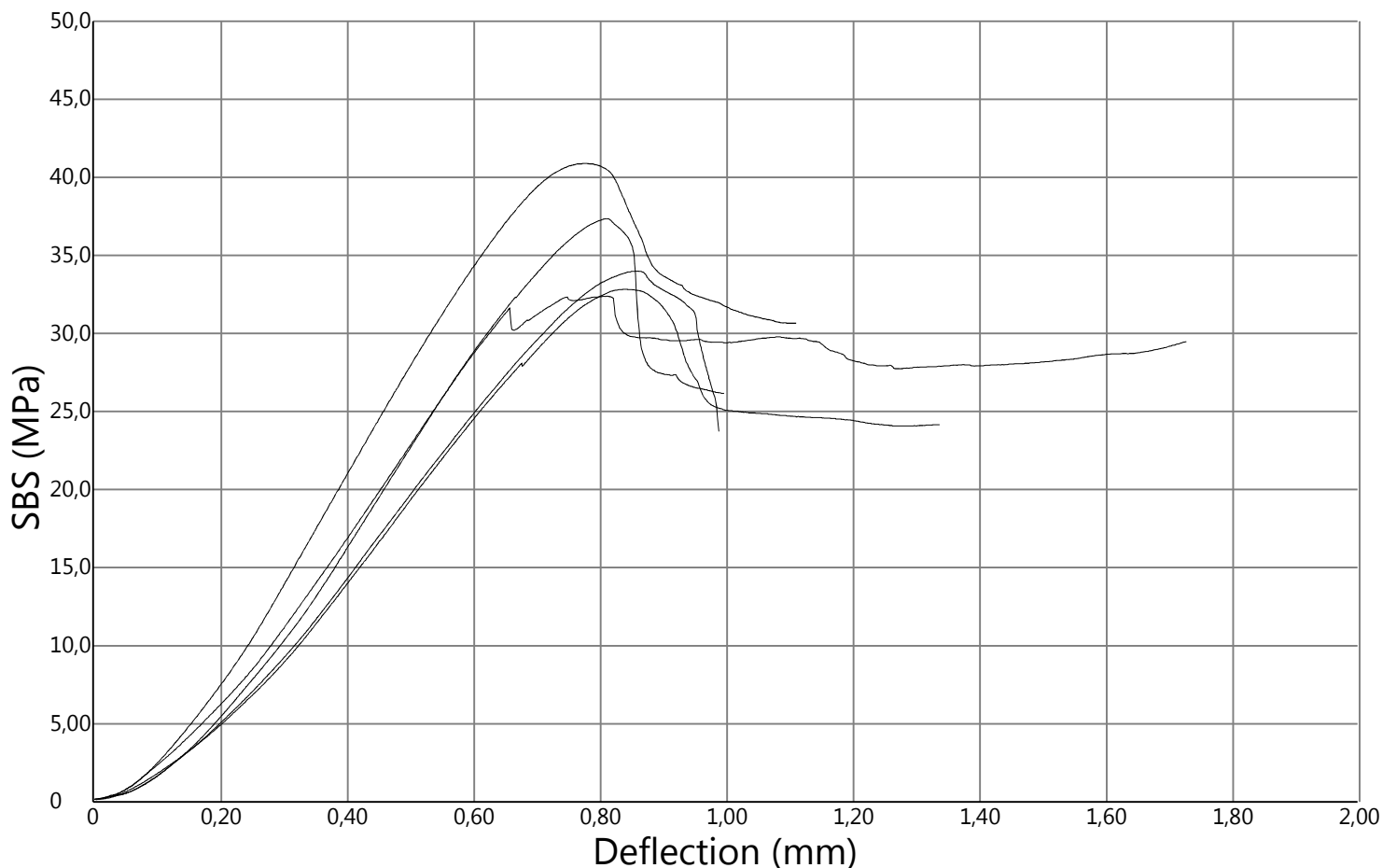
Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
5	7,19	3,03	988	900	0,95	34,0	ILS
Average	7,19	3,03	988	900	0,95	34,0	
SD			N/A		N/A	N/A	
CoV			N/A			N/A	



Short-Beam Strength of Polymer Matrix  
 Composite Materials and their Laminates

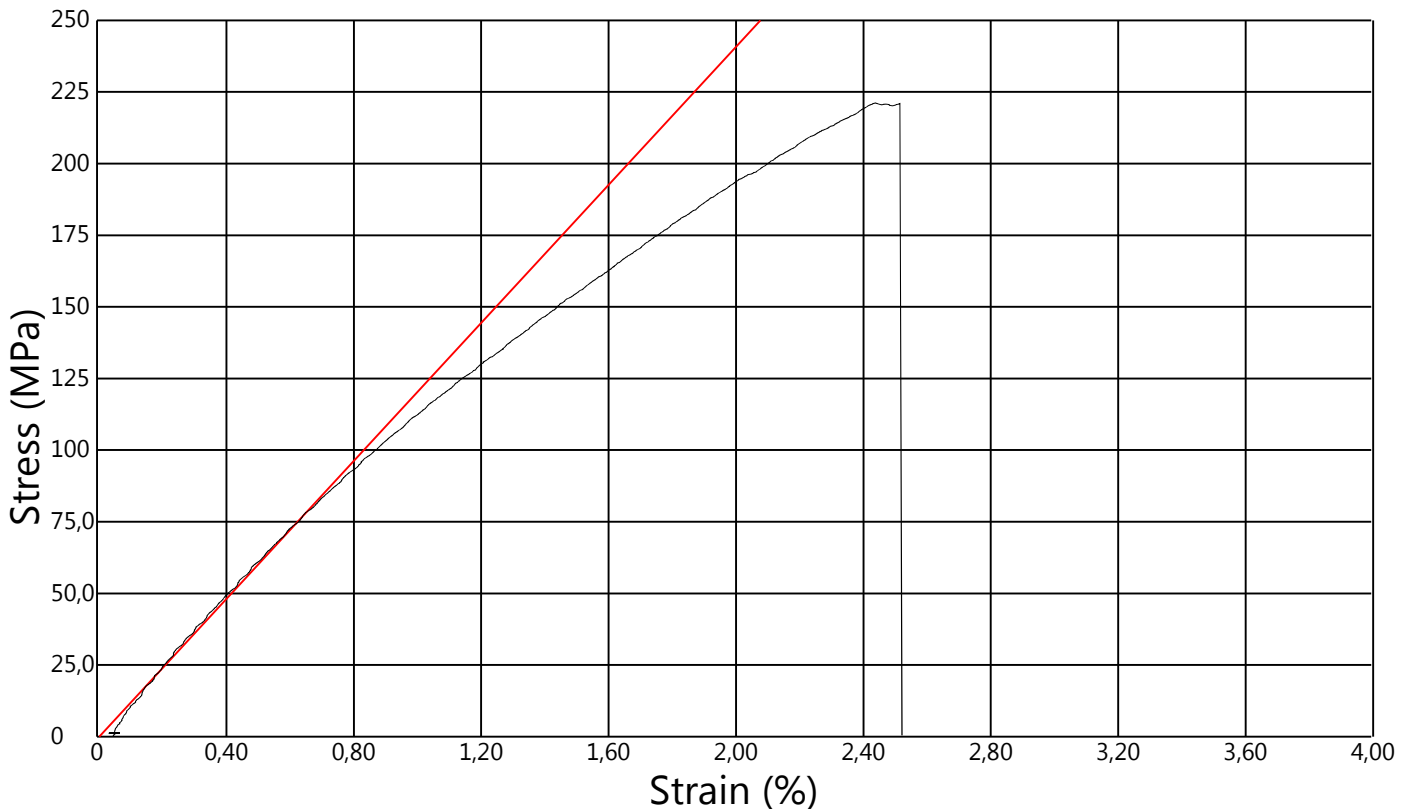
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Date Tested: 19 mai, 2020  
 Test Speed: 1,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Batch ID.: SN-1.3i

Specimen No.	Width (b) mm	Thickness (h) mm	Maximum Load N	Final Load N	Break Distance mm	Short-Beam Strength MPa	Failure mode
1	7,27	3,26	1 023	932	1,73	32,4	ILS
2	7,16	3,14	984	724	1,34	32,8	ILS
3	7,15	3,04	1 083	758	1,00	37,4	ILS
4	7,13	3,04	1 182	886	1,11	40,9	ILS
5	7,19	3,03	988	900	0,95	34,0	ILS
Average	7,18	3,10	1 052	840	1,23	35,5	
SD			82,7		0,32	3,59	
CoV			7,86			10,1	



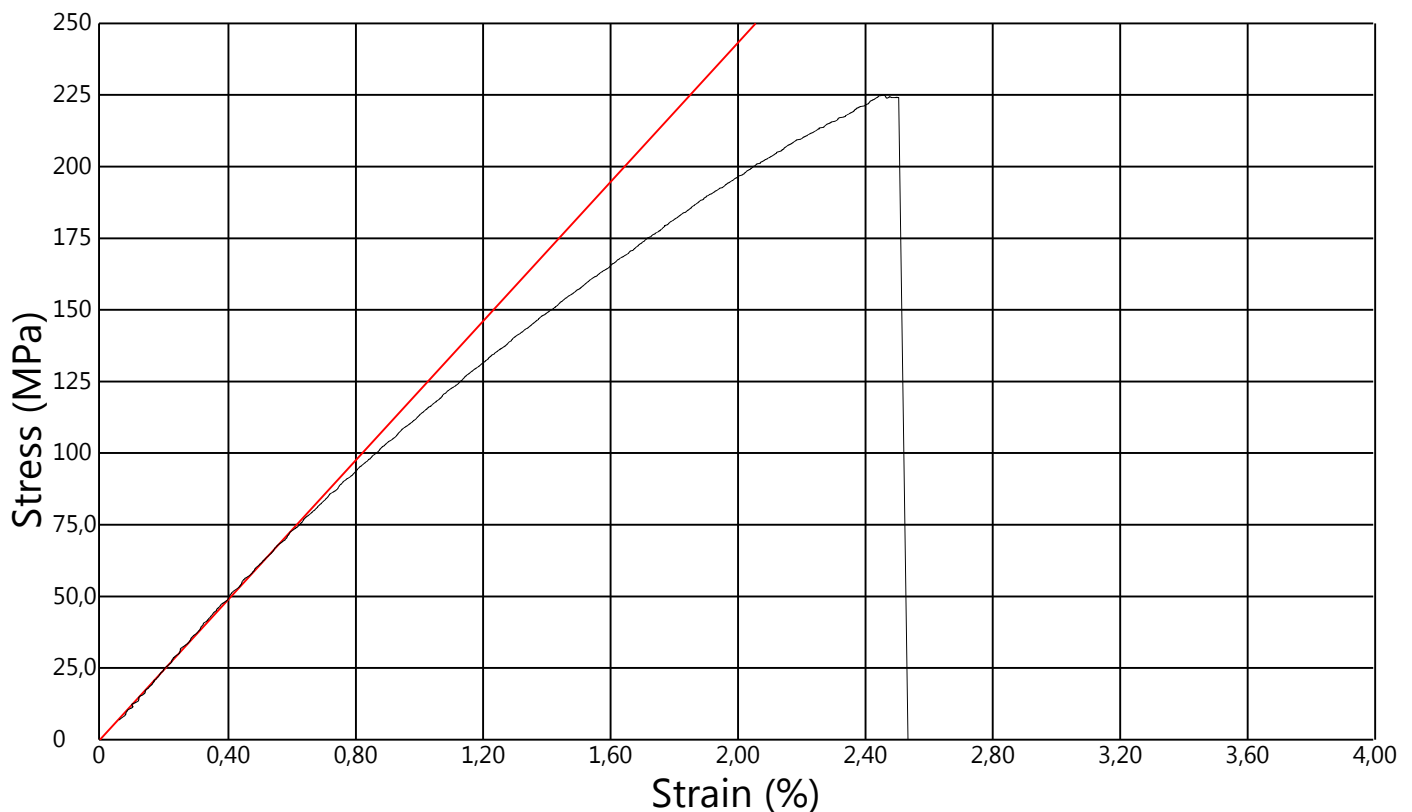
Operator: Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SU-2.0s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
1	13,2	4,12	12000	221	2,44	LAB	12,0
Average	13,2	4,12	12000	221	2,44		12,0
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



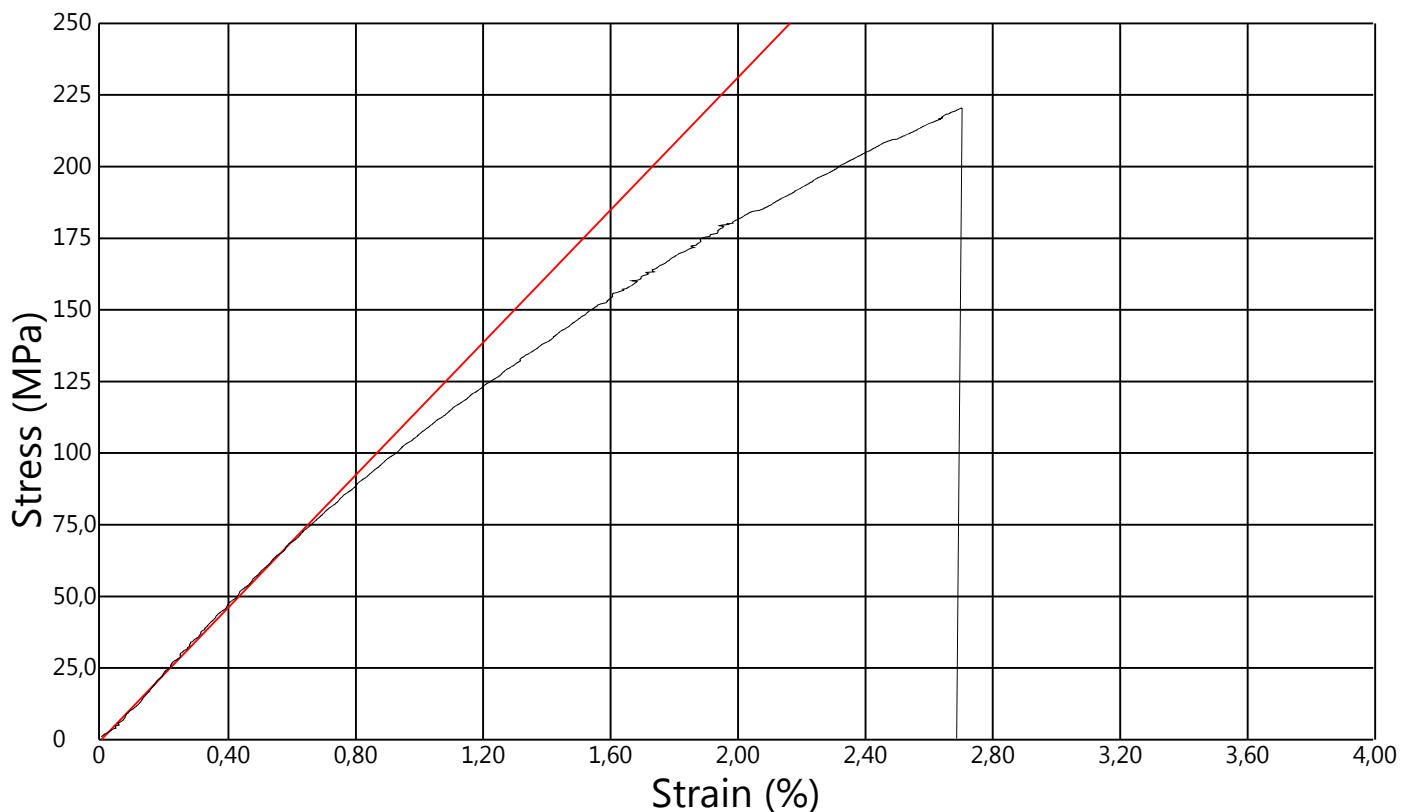
Operator: Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SU-2.0s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
2	13,1	4,18	12400	225	2,47	LAB	12,1
Average	13,1	4,18	12400	225	2,47		12,1
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



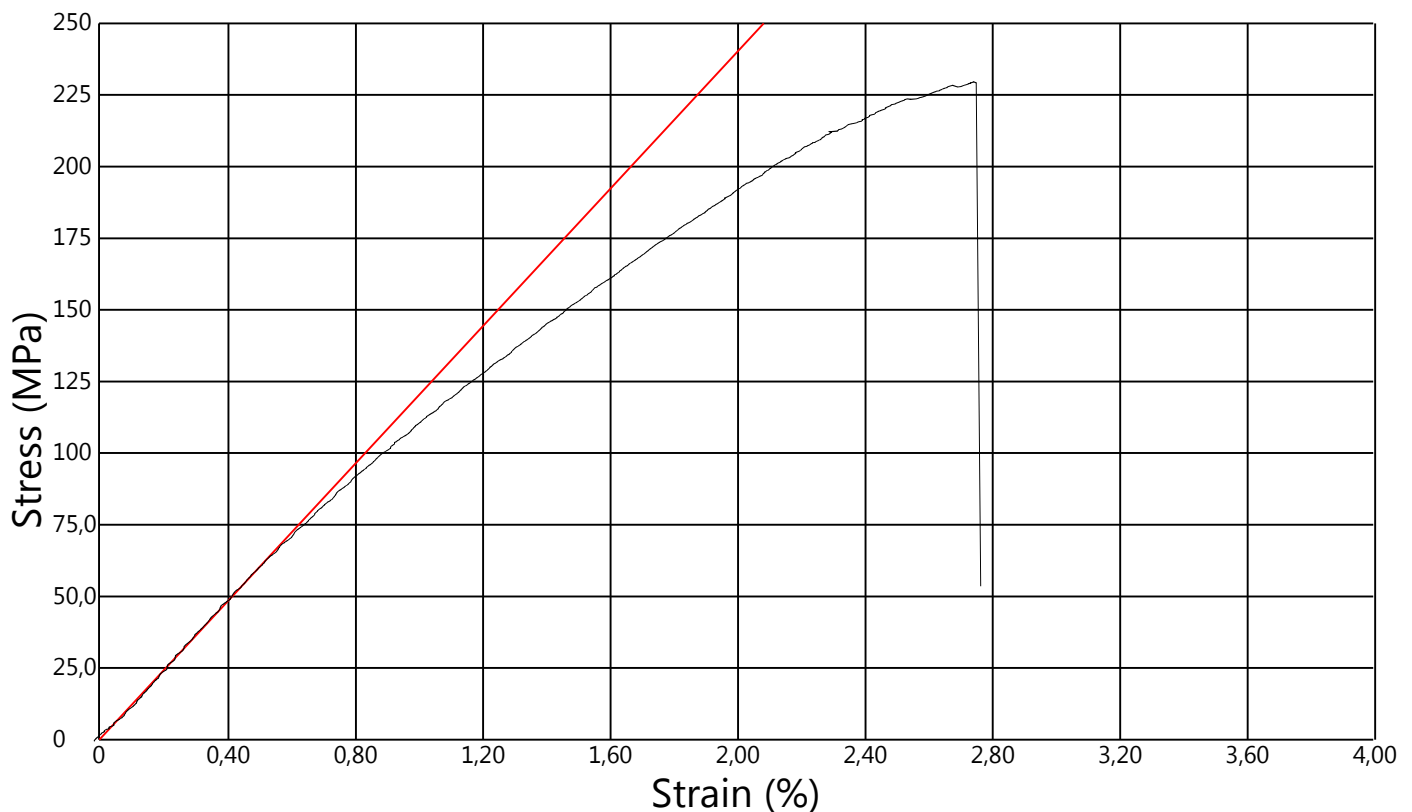
Operator: Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SU-2.0s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
3	13,2	4,15	12000	220	2,71	LAT	11,6
Average	13,2	4,15	12000	220	2,71		11,6
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



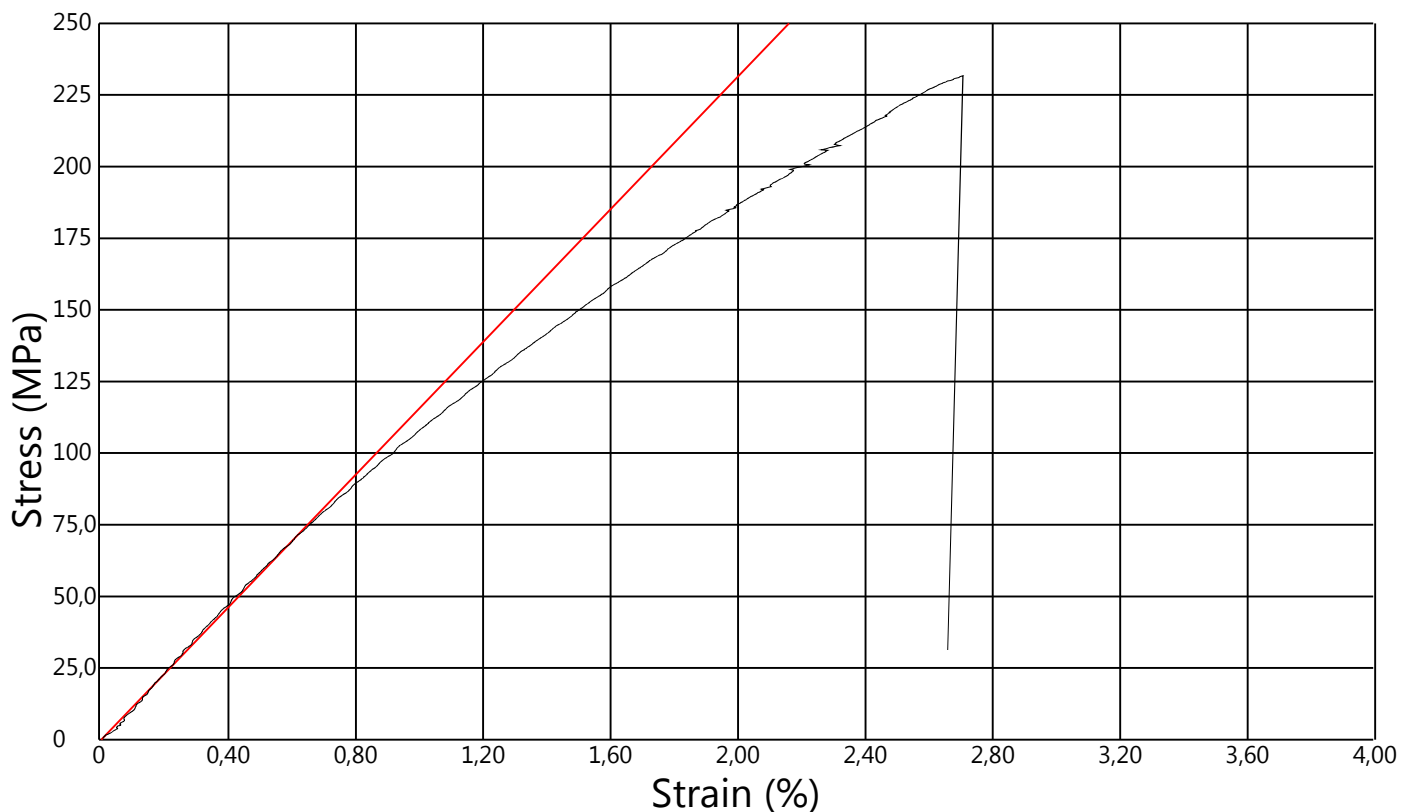
Operator: Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SU-2.0s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
4	13,1	4,20	12700	230	2,75	LAB	12,0
Average	13,1	4,20	12700	230	2,75		12,0
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



Operator: Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SU-2.0s

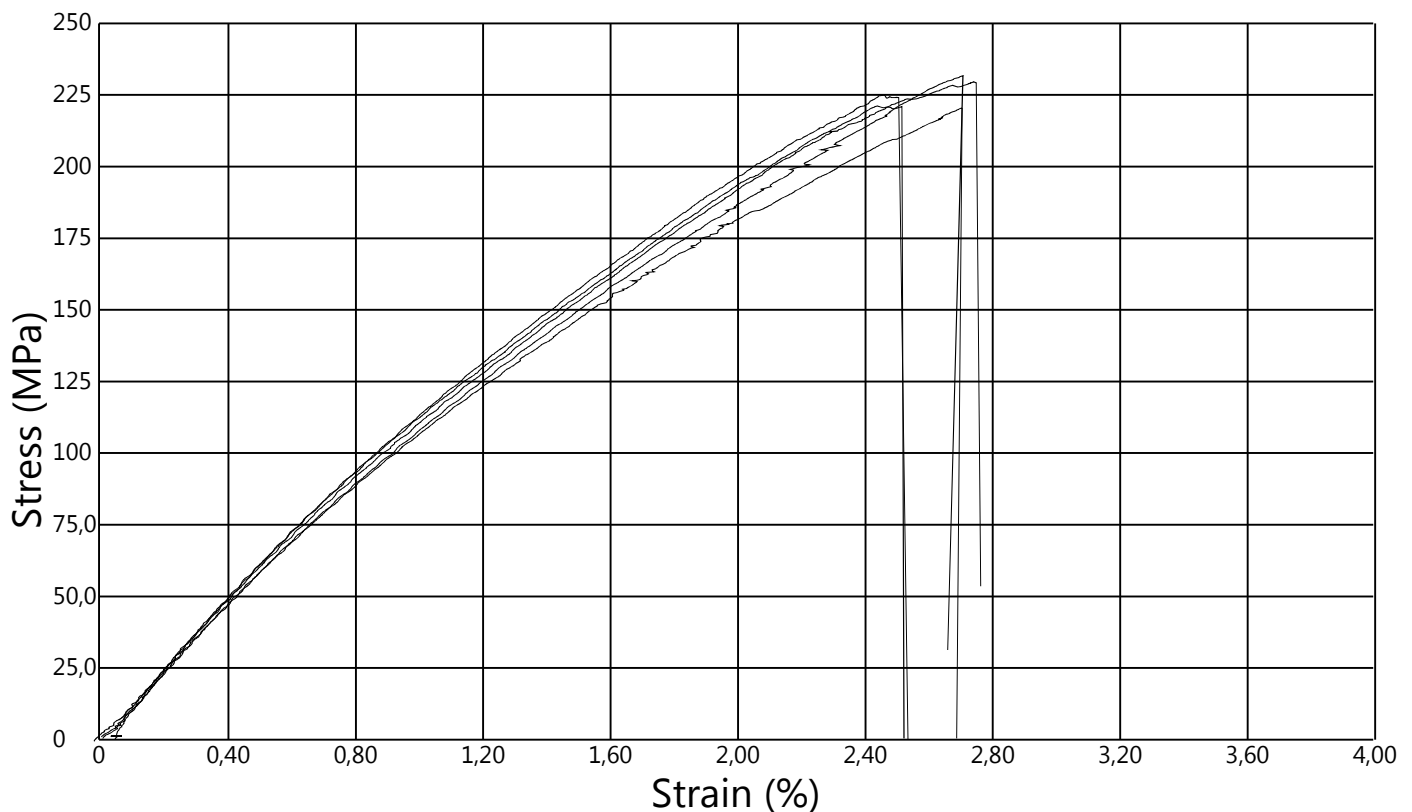
Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
5	13,1	4,15	12600	232	2,71	LAT	11,6
Average	13,1	4,15	12600	232	2,71		11,6
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A





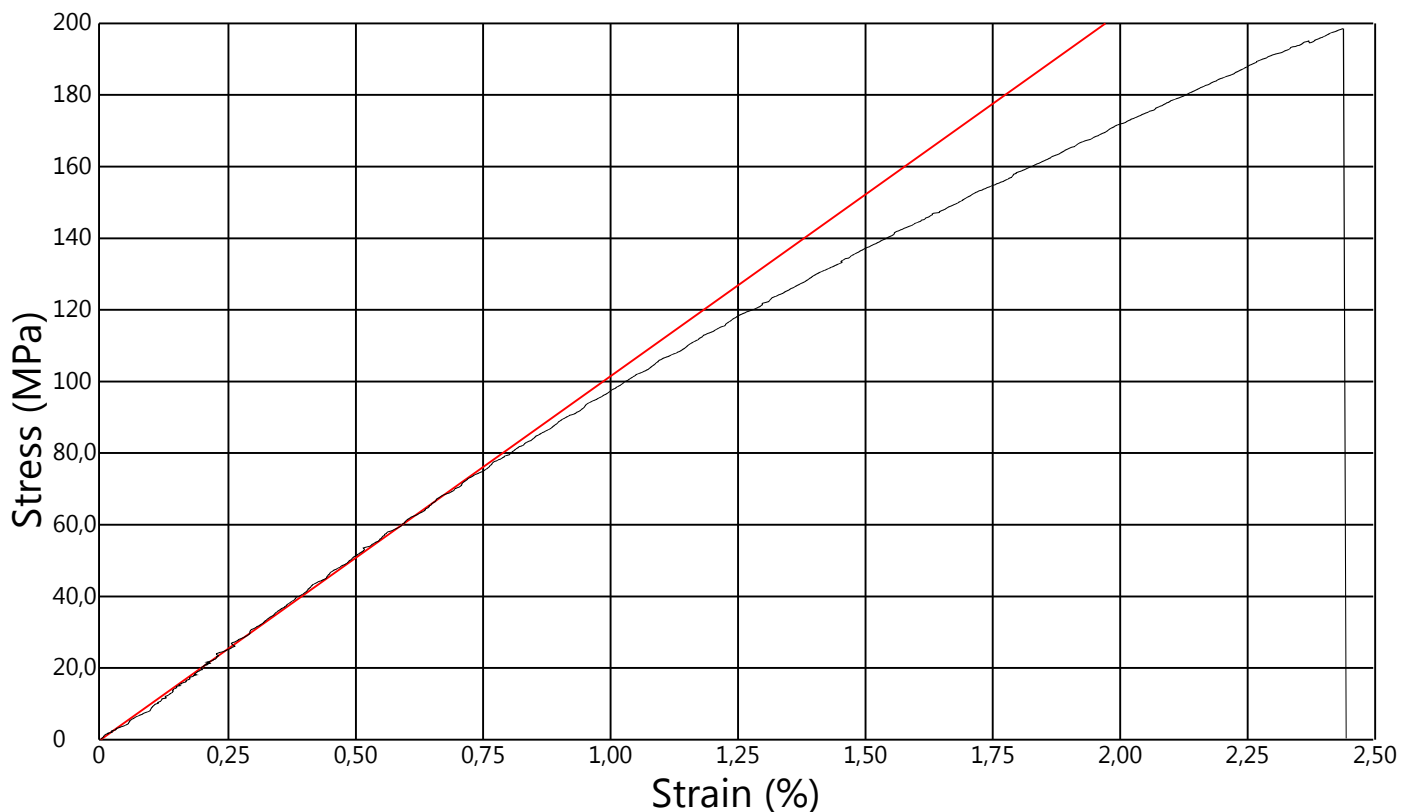
Operator: Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SU-2.0s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
1	13,2	4,12	12000	221	2,44	LAB	12,0
2	13,1	4,18	12400	225	2,47	LAB	12,1
3	13,2	4,15	12000	220	2,71	LAT	11,6
4	13,1	4,20	12700	230	2,75	LAB	12,0
5	13,1	4,15	12600	232	2,71	LAT	11,6
Average	13,1	4,16	12300	226	2,61		11,9
SD			322	4,99	0,147		0,276
CoV		0,741	2,61	2,21	5,62		2,33



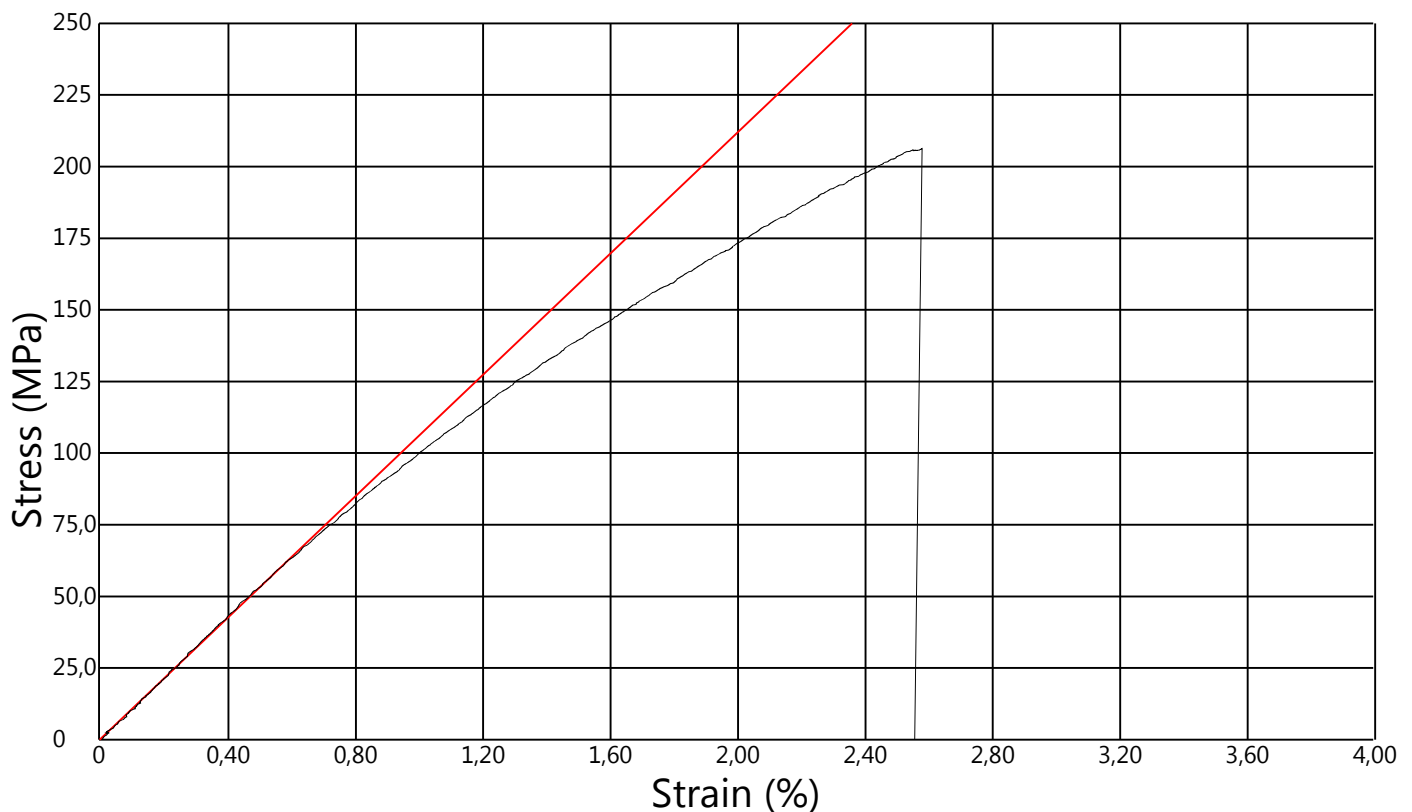
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.1s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
1	15,3	3,41	10300	198	2,44	LAB	10,1
Average	15,3	3,41	10300	198	2,44		10,1
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



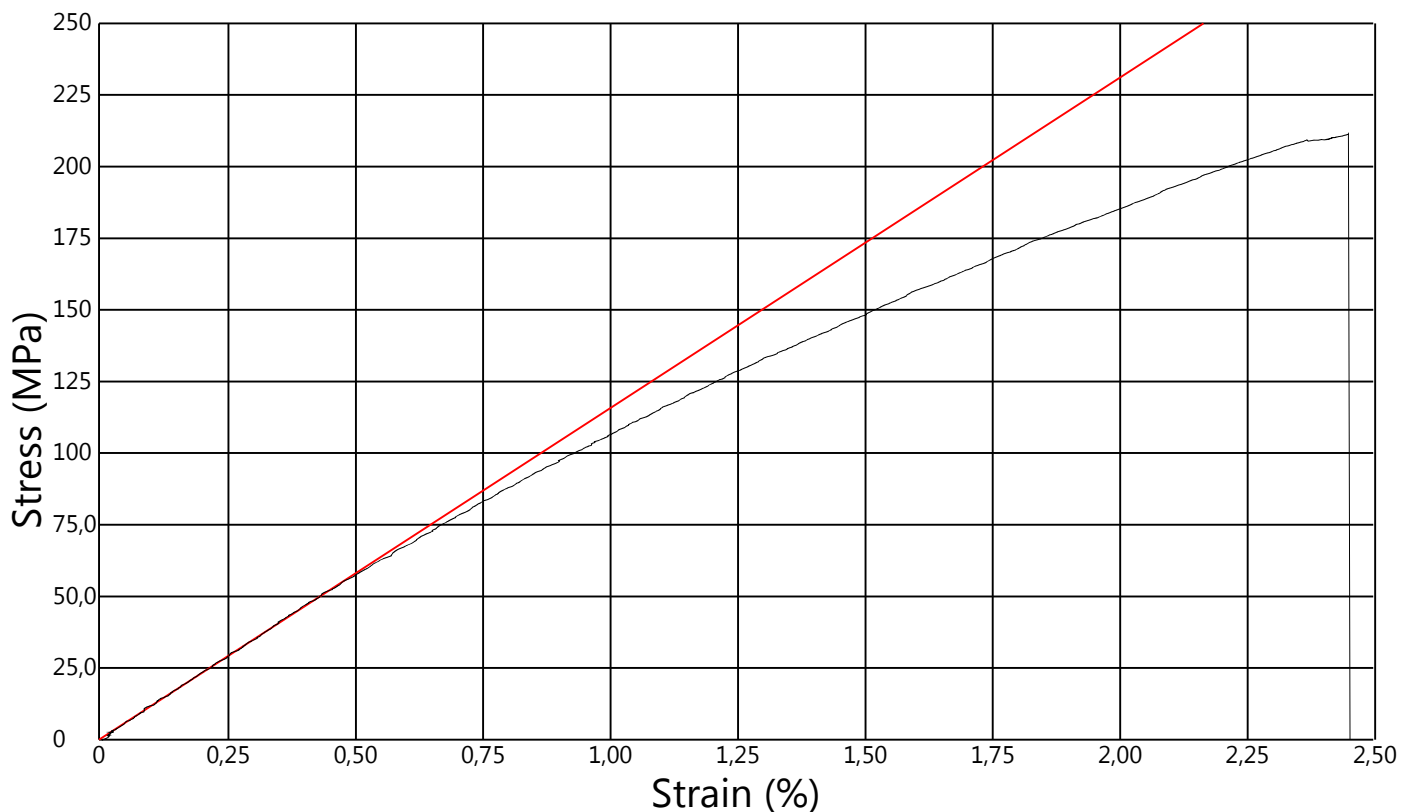
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.1s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
2	15,3	3,48	11000	206	2,58	LAB	10,6
Average	15,3	3,48	11000	206	2,58		10,6
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



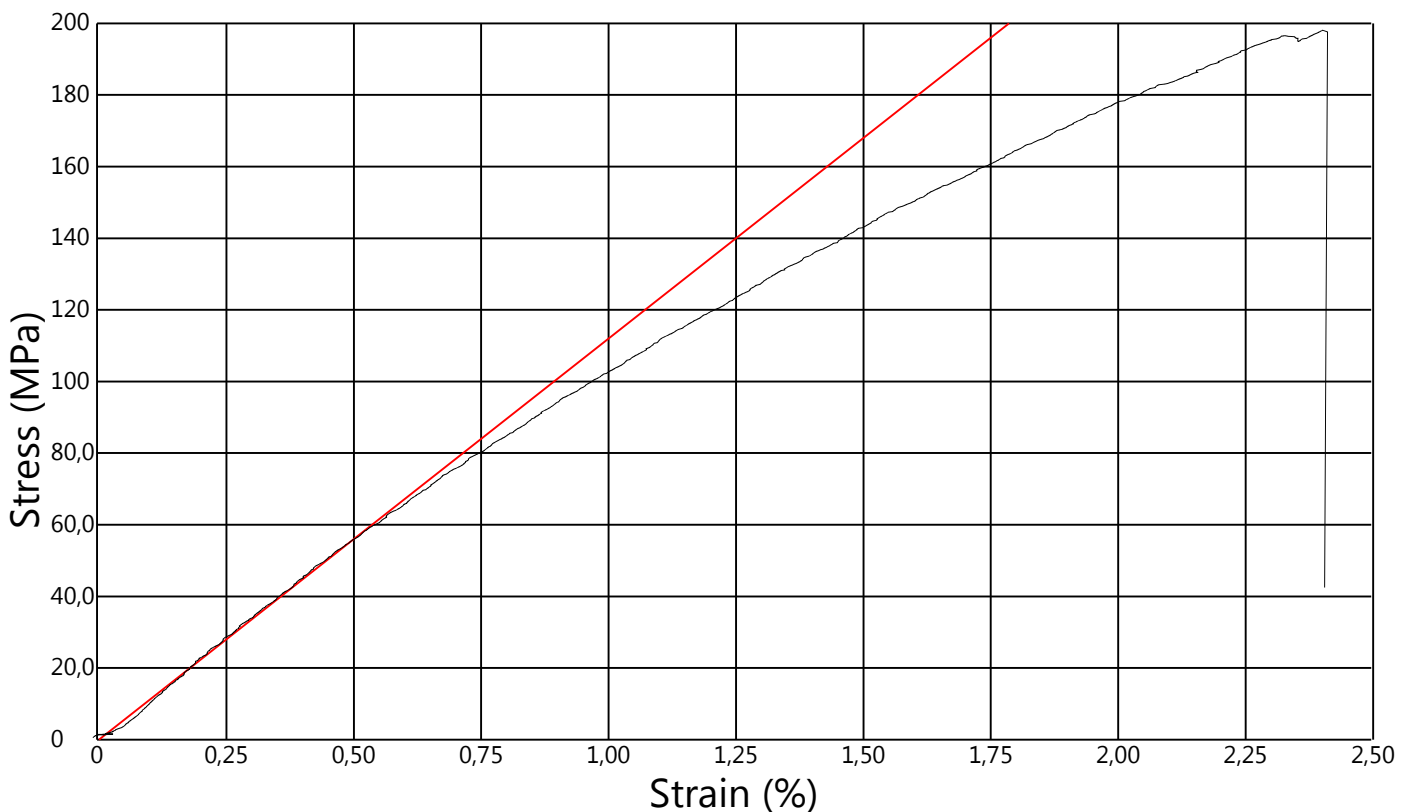
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.1s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
3	15,3	3,33	10700	211	2,45	LAB	11,5
Average	15,3	3,33	10700	211	2,45		11,5
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



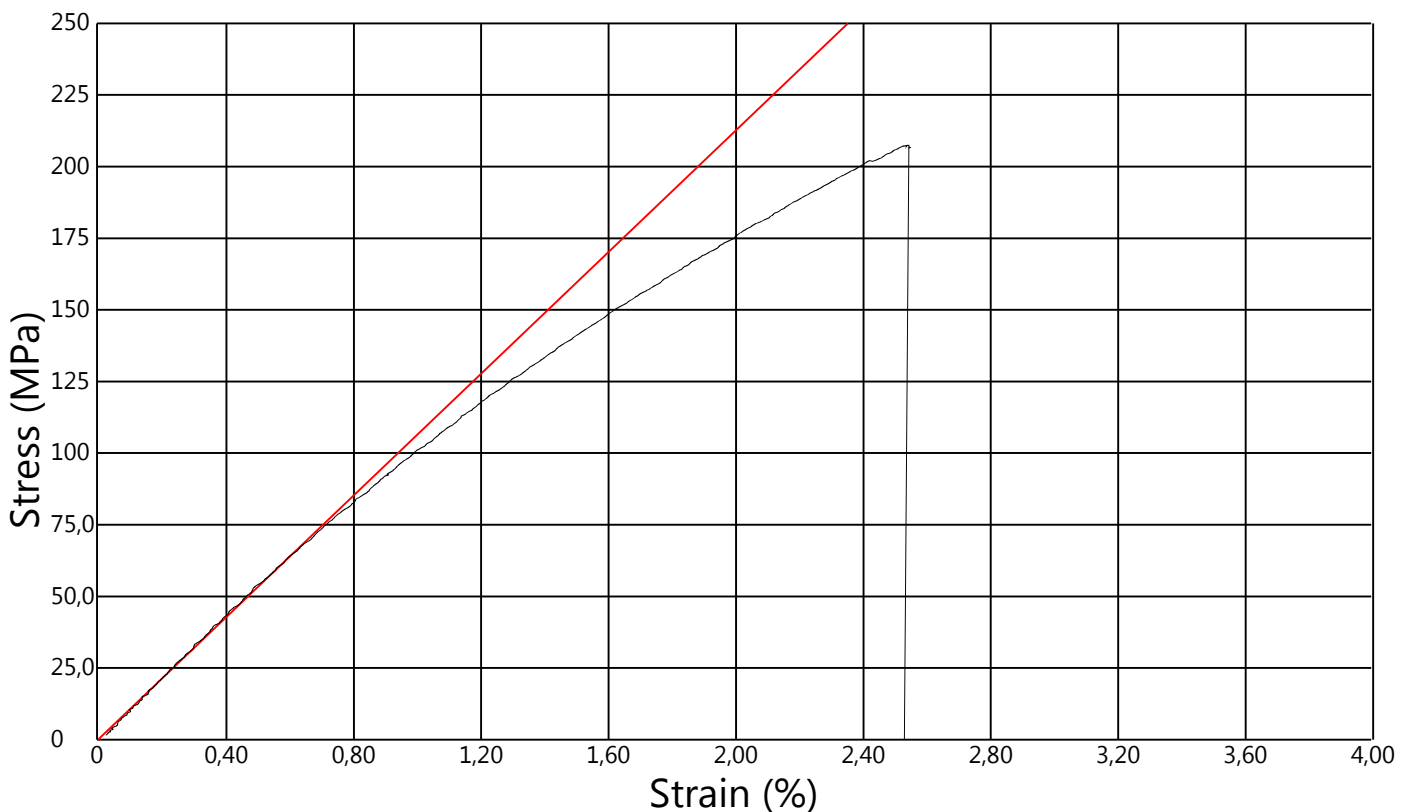
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.1s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
4	15,3	3,54	10700	198	2,41	LAB	11,2
Average	15,3	3,54	10700	198	2,41		11,2
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



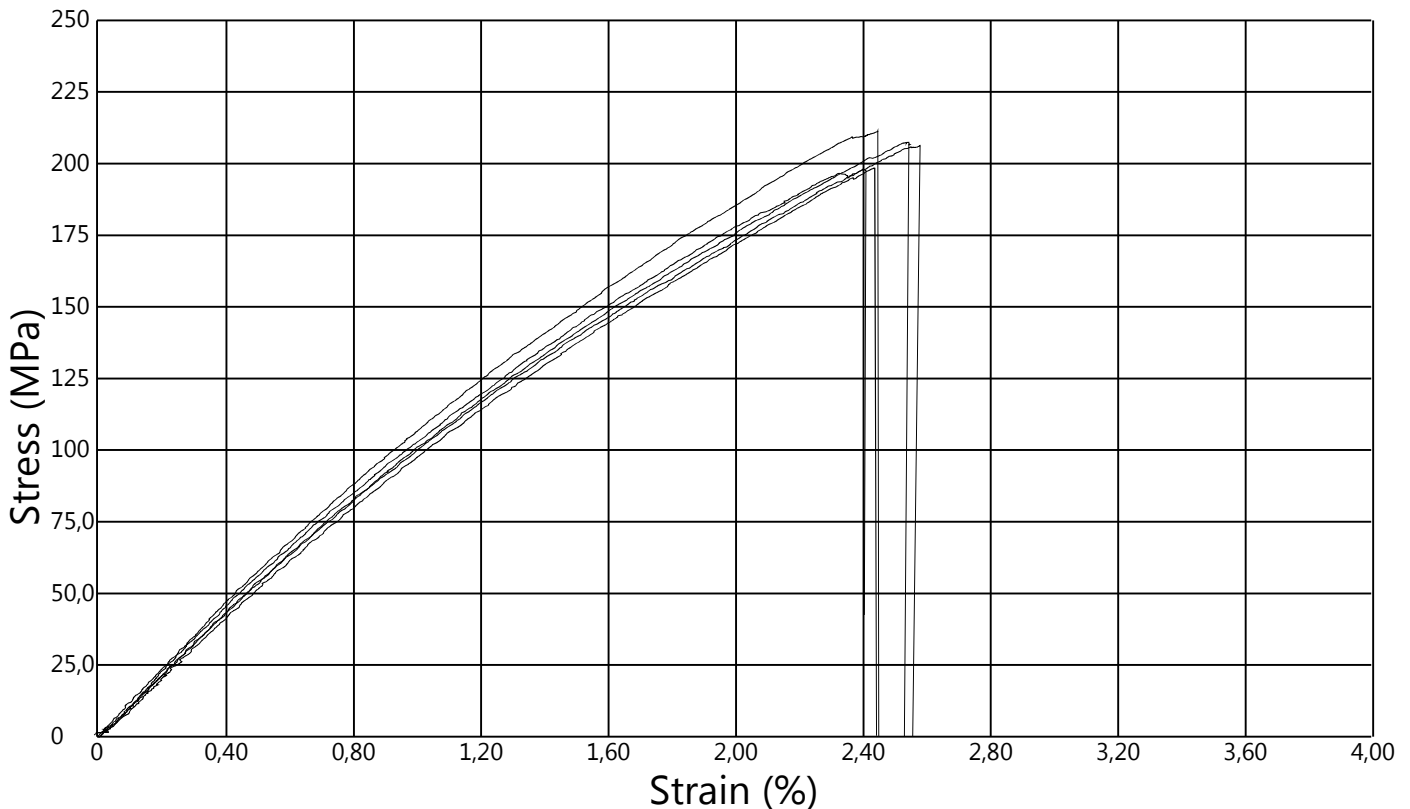
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.1s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
5	15,3	3,49	11000	207	2,55	LAB	10,6
Average	15,3	3,49	11000	207	2,55		10,6
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



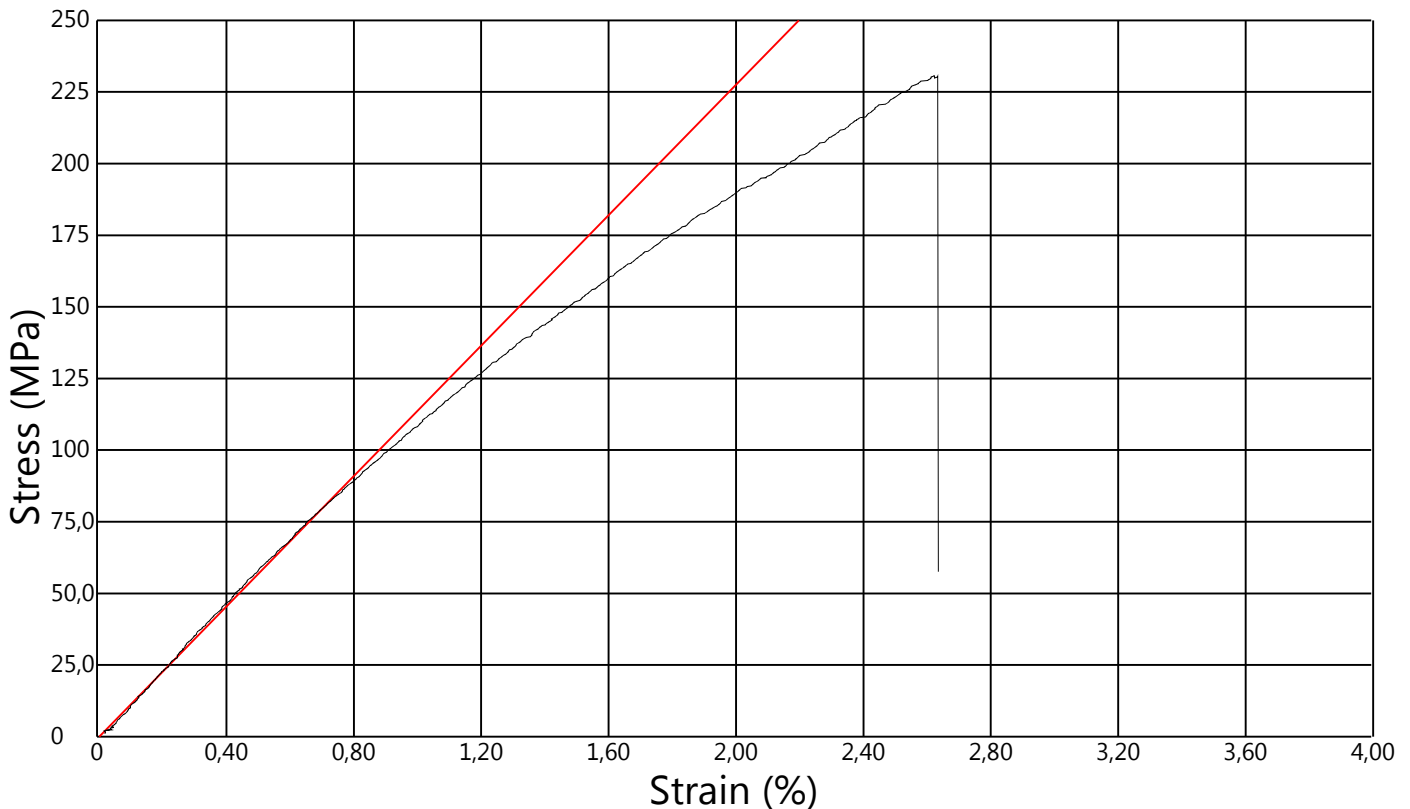
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.1s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
1	15,3	3,41	10300	198	2,44	LAB	10,1
2	15,3	3,48	11000	206	2,58	LAB	10,6
3	15,3	3,33	10700	211	2,45	LAB	11,5
4	15,3	3,54	10700	198	2,41	LAB	11,2
5	15,3	3,49	11000	207	2,55	LAB	10,6
Average	15,3	3,45	10800	204	2,49		10,8
SD			284	5,85	0,0757		0,550
CoV		2,36	2,64	2,87	3,05		5,09



Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

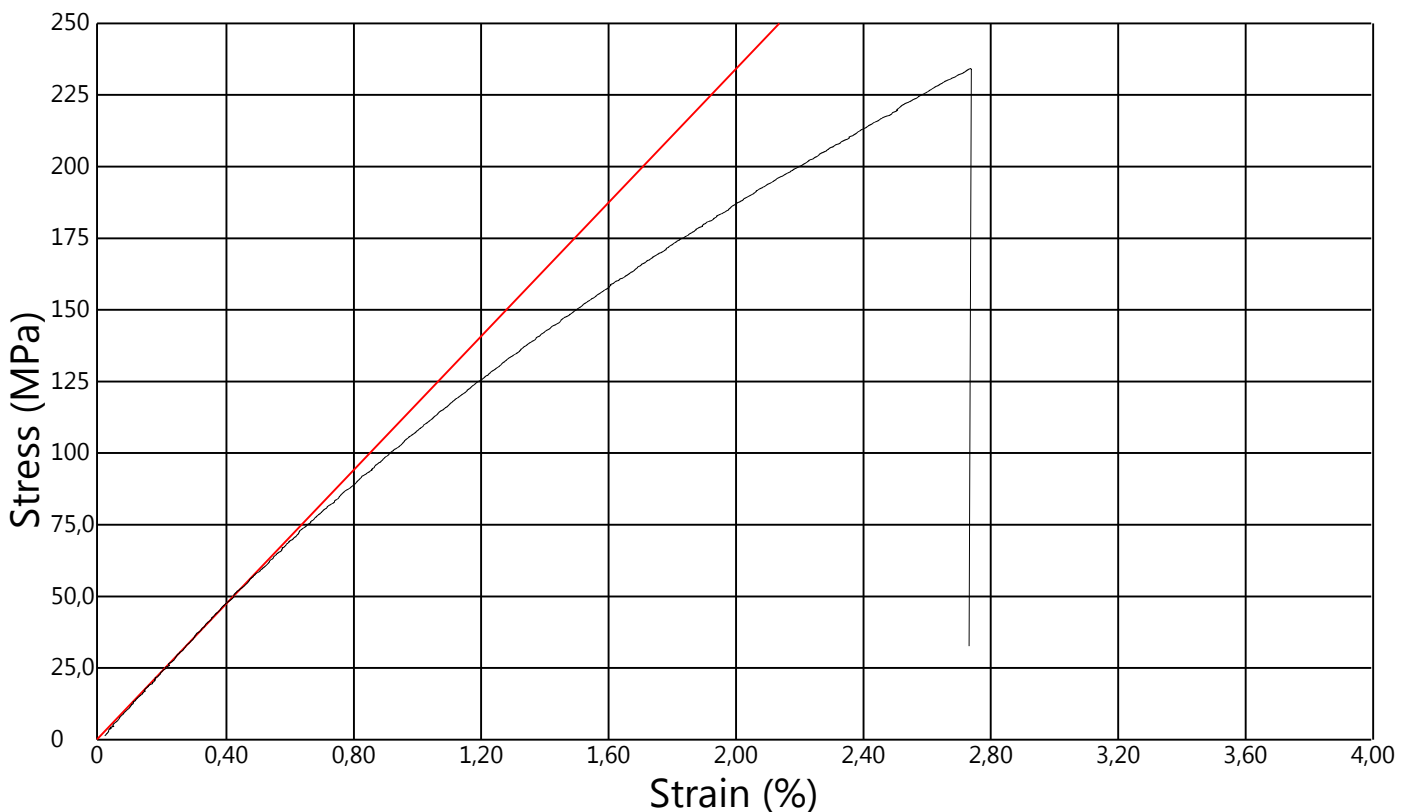
Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
1	15,2	3,25	11400	231	2,63	LAT	11,4
Average	15,2	3,25	11400	231	2,63		11,4
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A





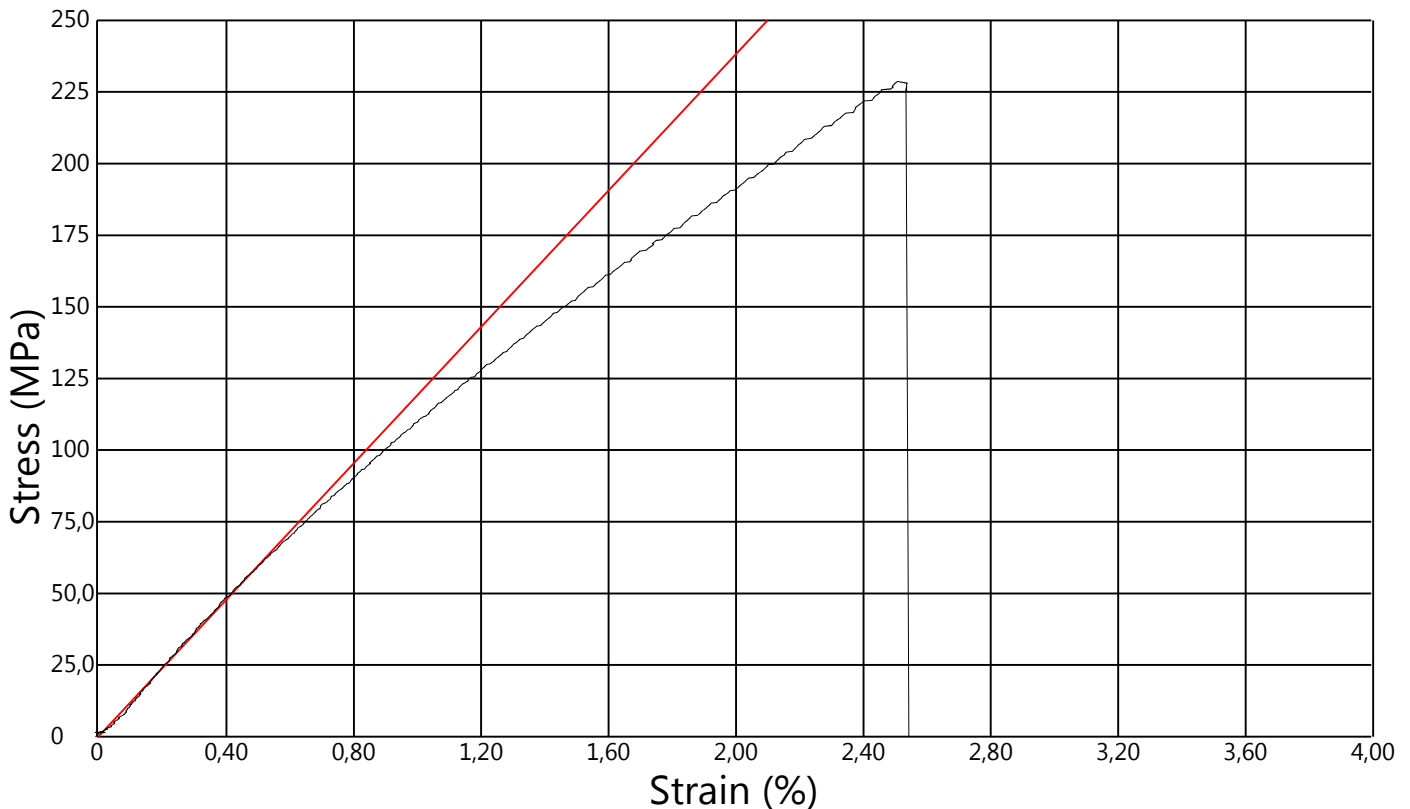
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
2	15,2	3,32	11800	234	2,74	LAT	11,7
Average	15,2	3,32	11800	234	2,74		11,7
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



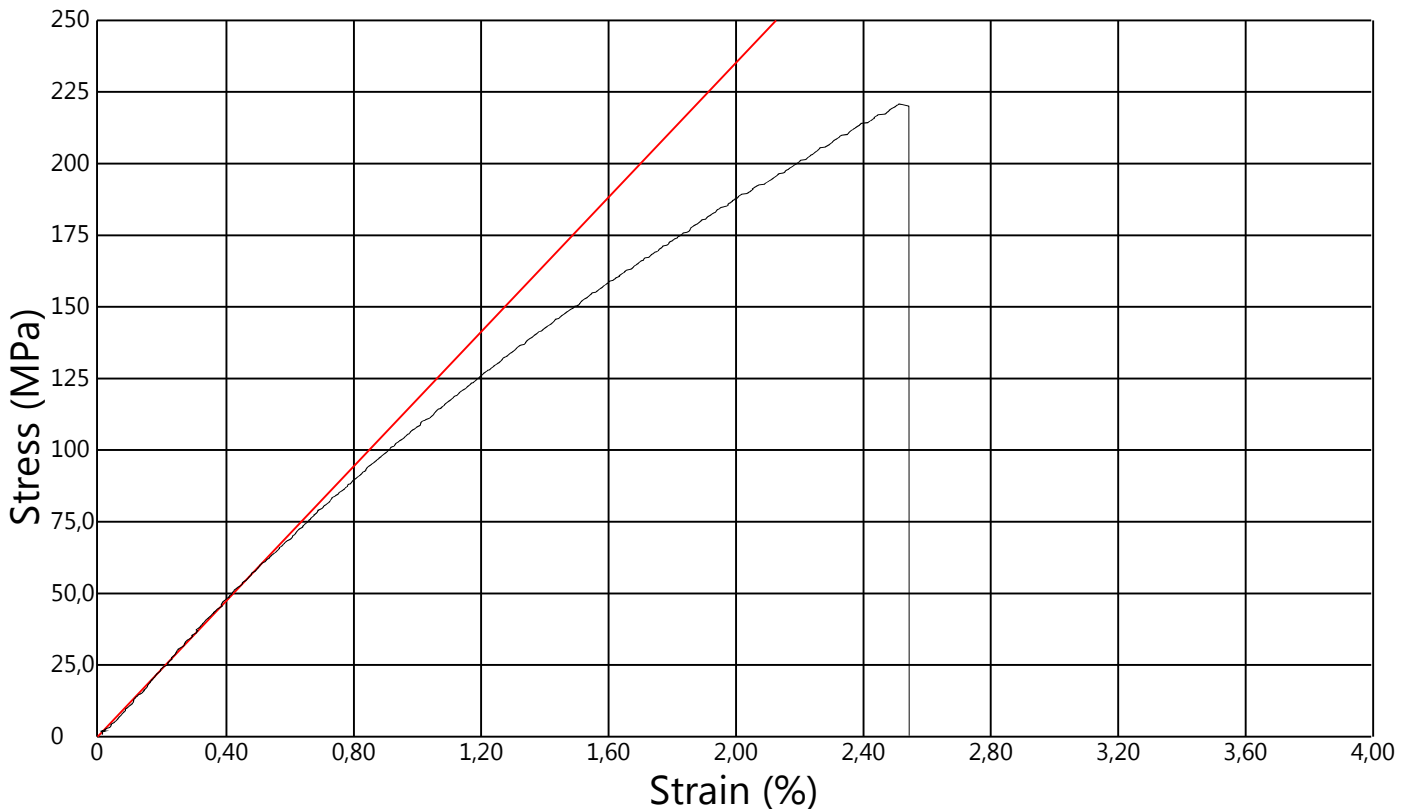
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
3	15,2	3,31	11500	229	2,51	AAT	11,9
Average	15,2	3,31	11500	229	2,51		11,9
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



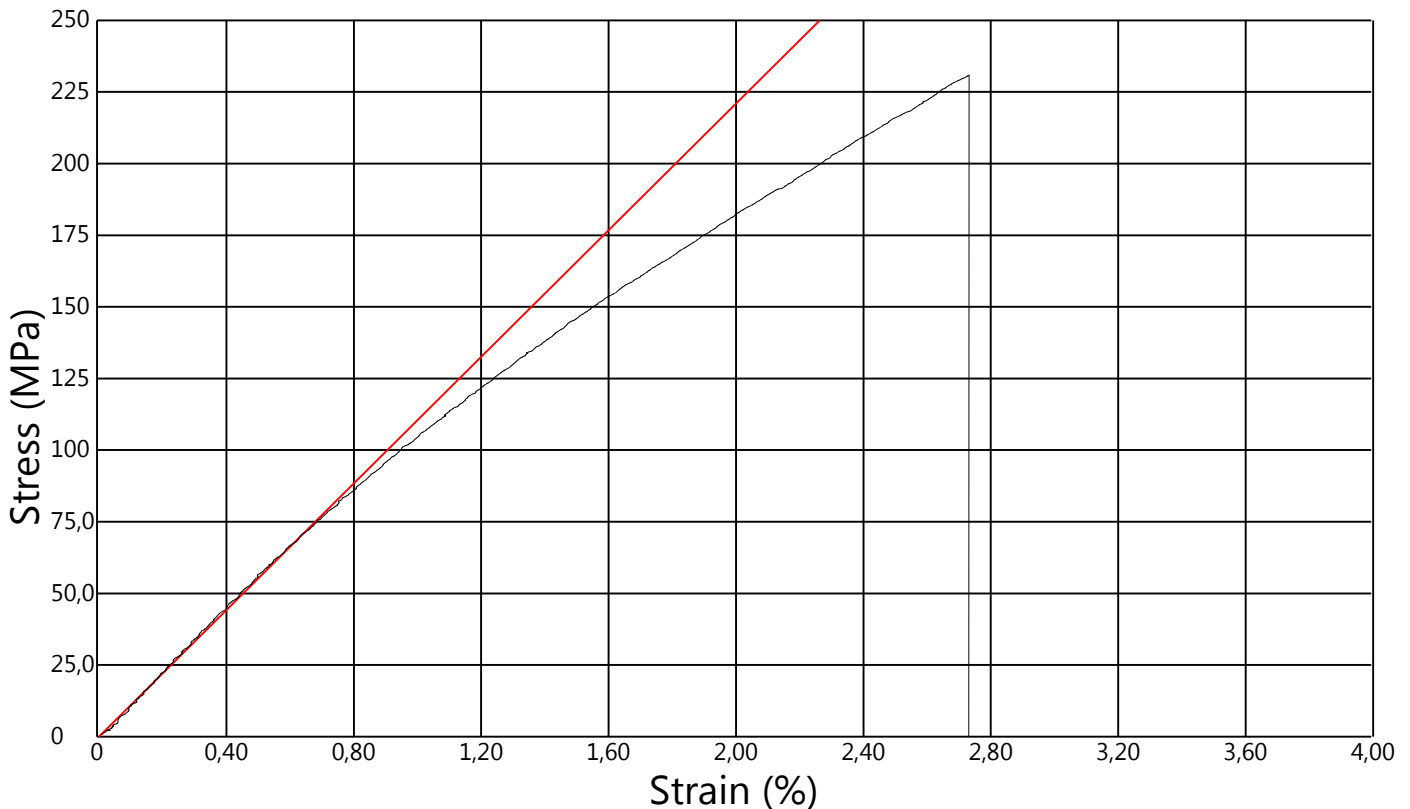
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
4	15,2	3,29	11100	221	2,52	LAB	11,7
Average	15,2	3,29	11100	221	2,52		11,7
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



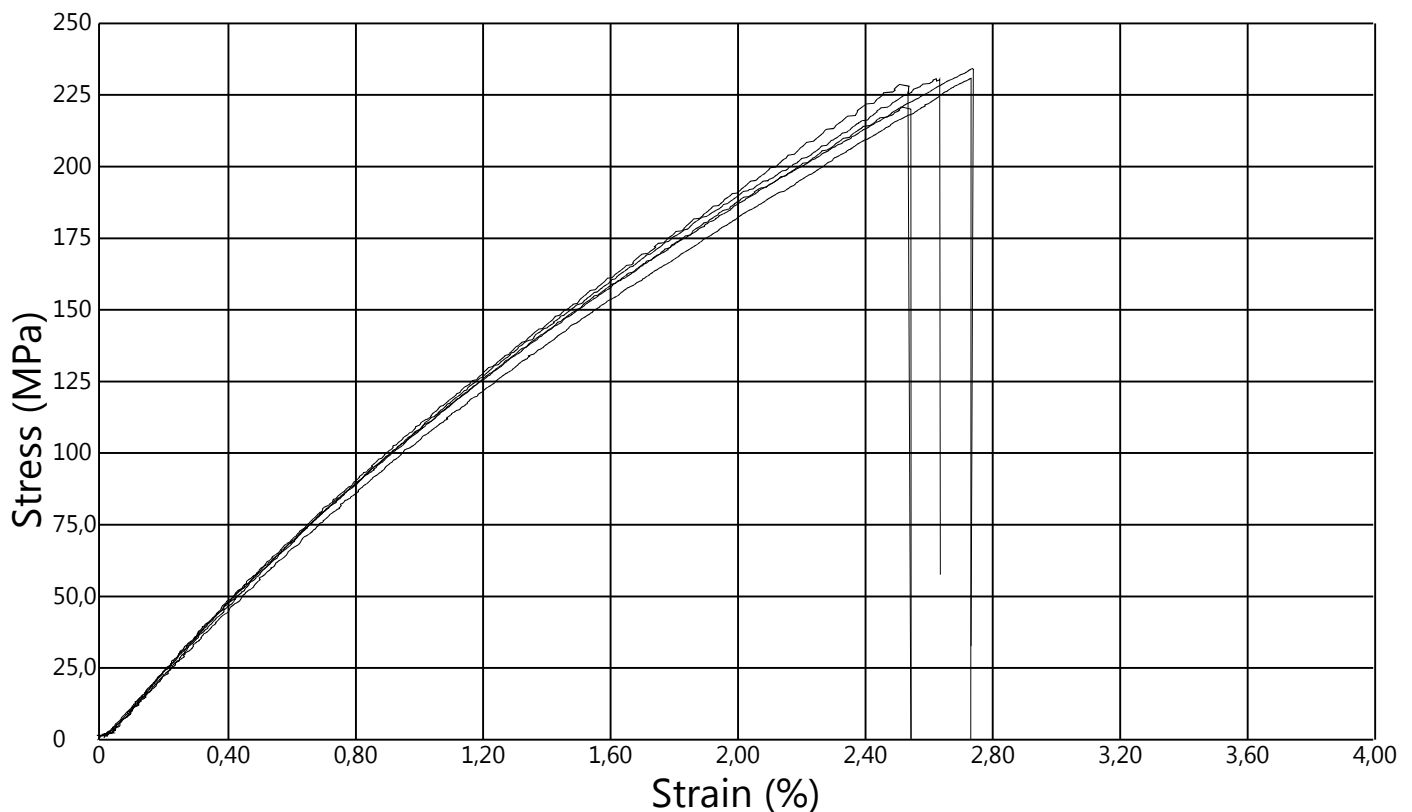
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
5	15,3	3,34	11800	231	2,74	LGM	11,1
Average	15,3	3,34	11800	231	2,74		11,1
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



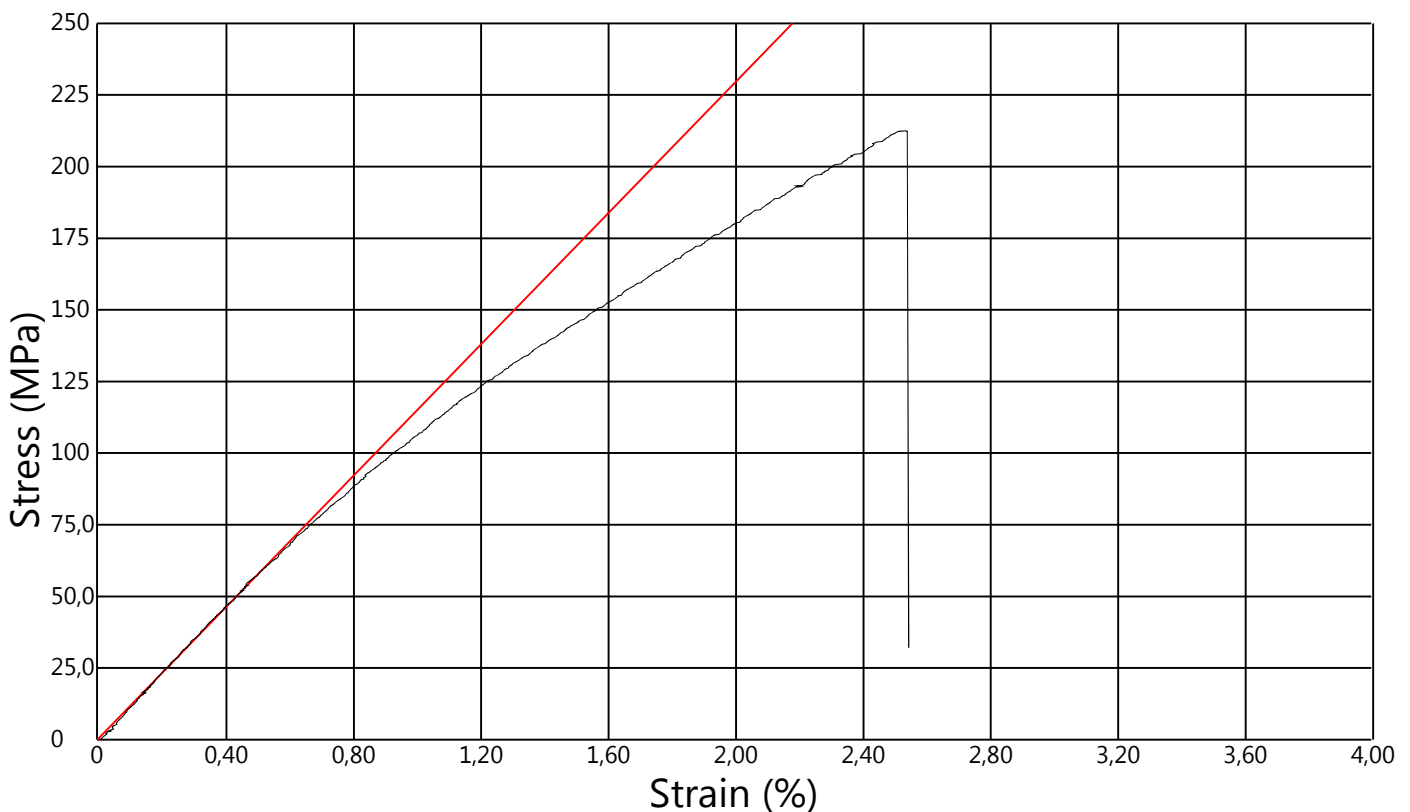
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
1	15,2	3,25	11400	231	2,63	LAT	11,4
2	15,2	3,32	11800	234	2,74	LAT	11,7
3	15,2	3,31	11500	229	2,51	AAT	11,9
4	15,2	3,29	11100	221	2,52	LAB	11,7
5	15,3	3,34	11800	231	2,74	LGM	11,1
Average	15,2	3,30	11500	229	2,63		11,5
SD			312	5,02	0,112		0,334
CoV		1,04	2,71	2,19	4,27		2,89



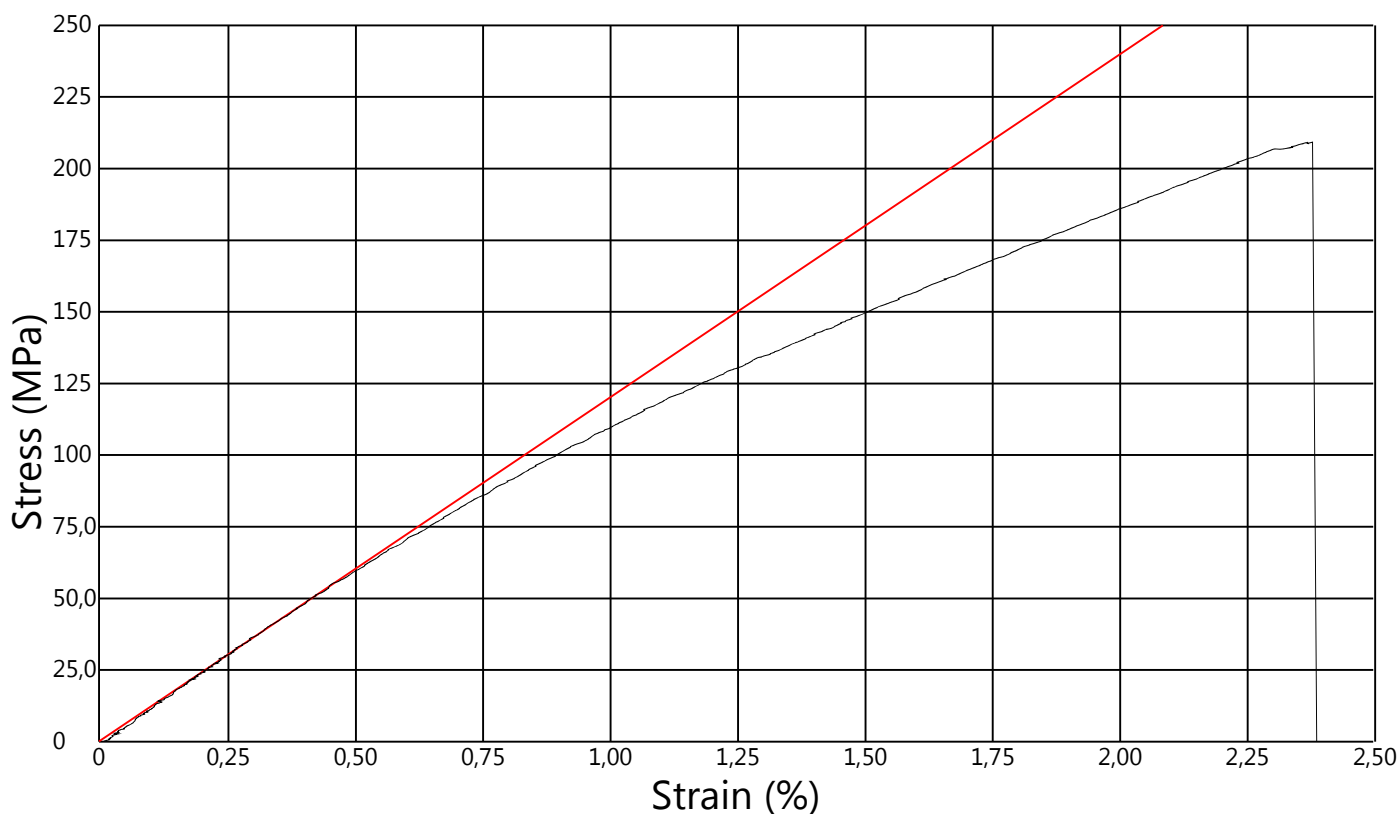
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
1	15,2	3,10	10000	212	2,54	LAB	11,5
Average	15,2	3,10	10000	212	2,54		11,5
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



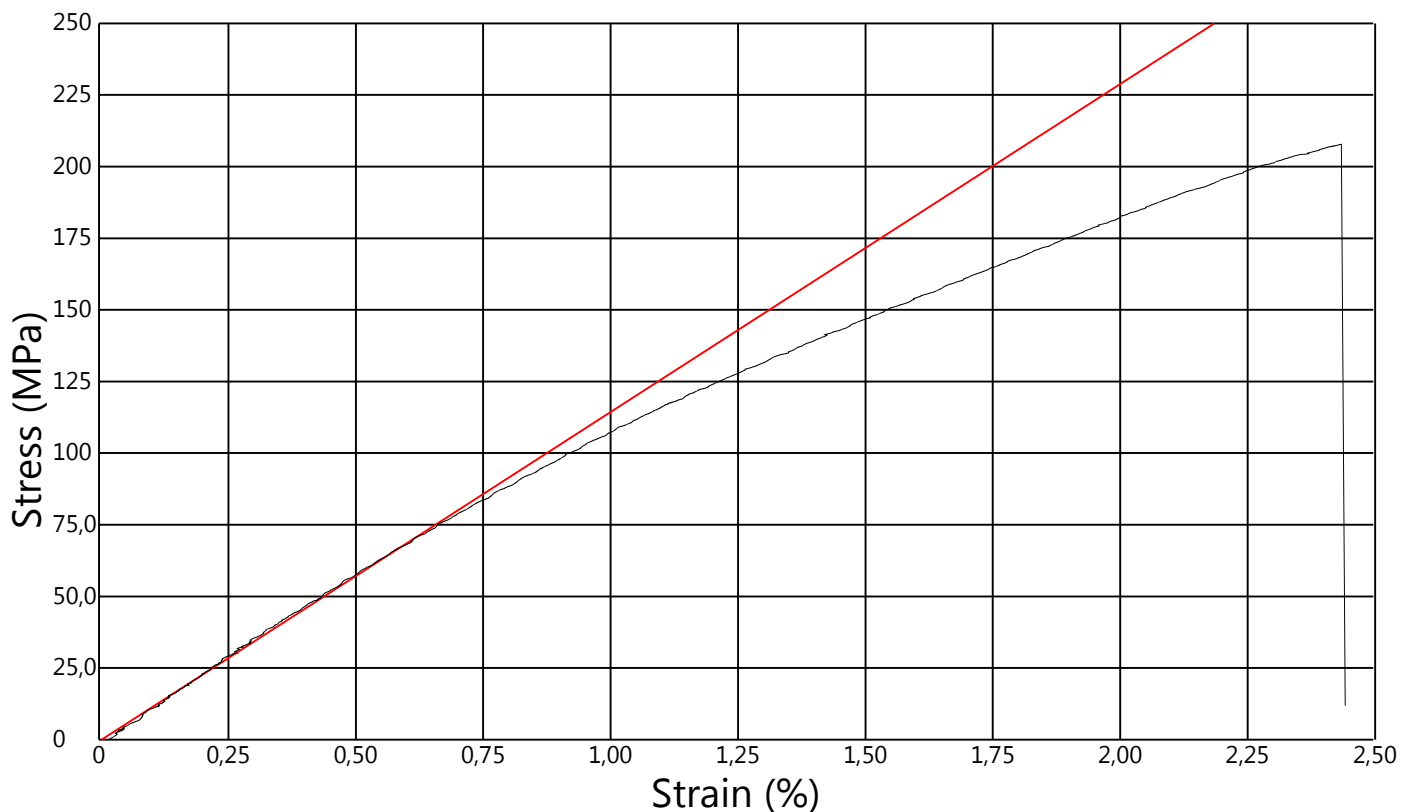
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
2	15,2	3,06	9730	209	2,38	LGT	12,0
Average	15,2	3,06	9730	209	2,38		12,0
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

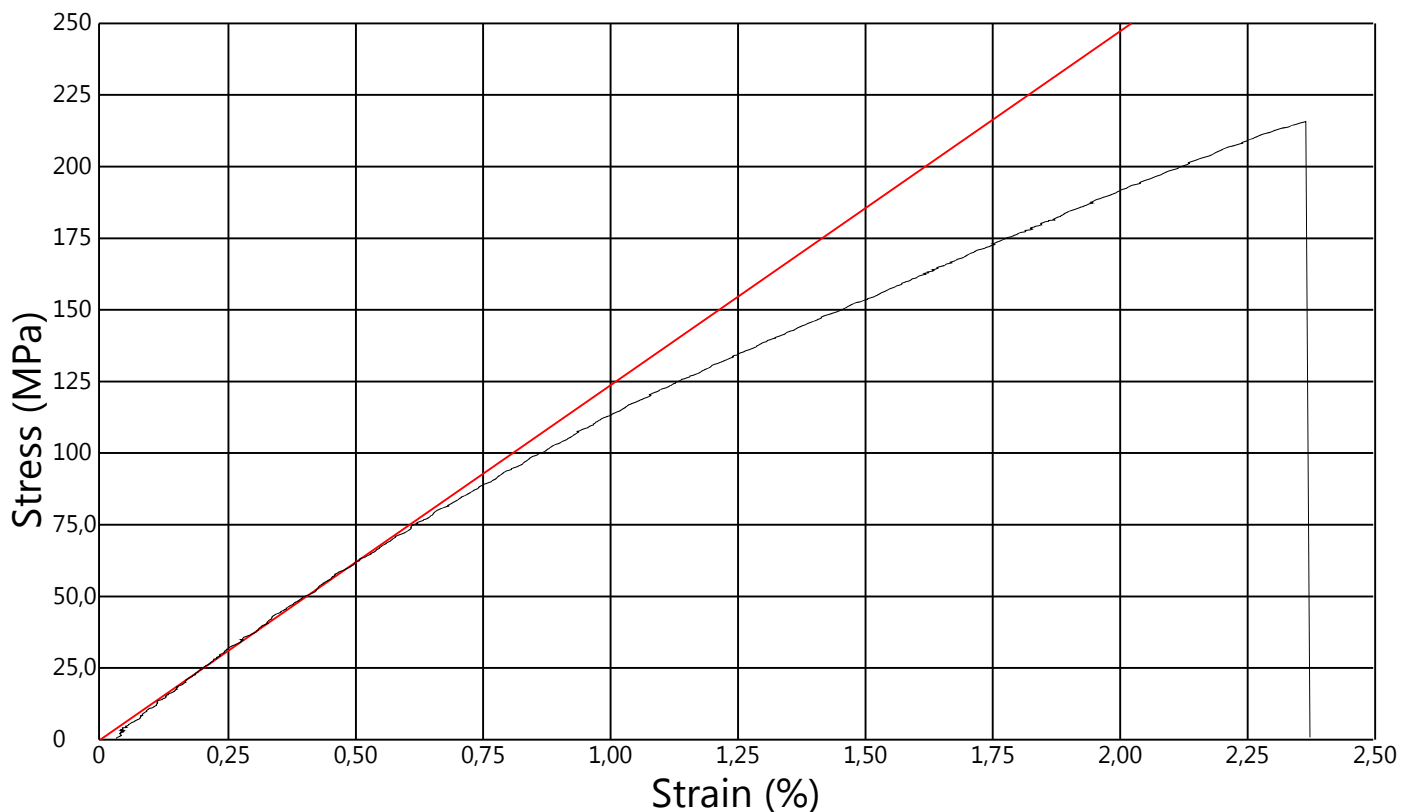
Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
3	15,2	3,10	9810	208	2,44	LGM	11,5
Average	15,2	3,10	9810	208	2,44		11,5
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A





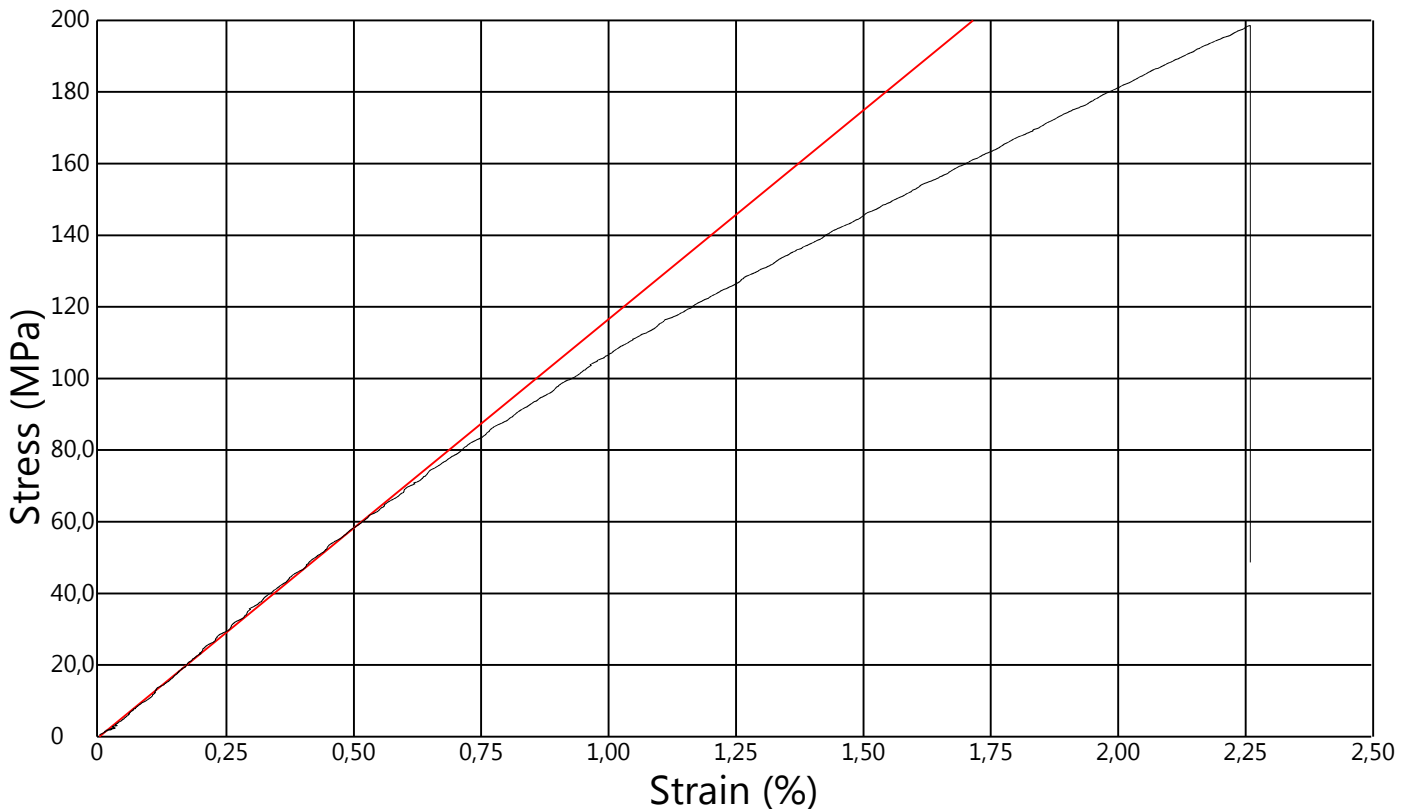
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
4	15,3	3,09	10200	216	2,37	LGM	12,4
Average	15,3	3,09	10200	216	2,37		12,4
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



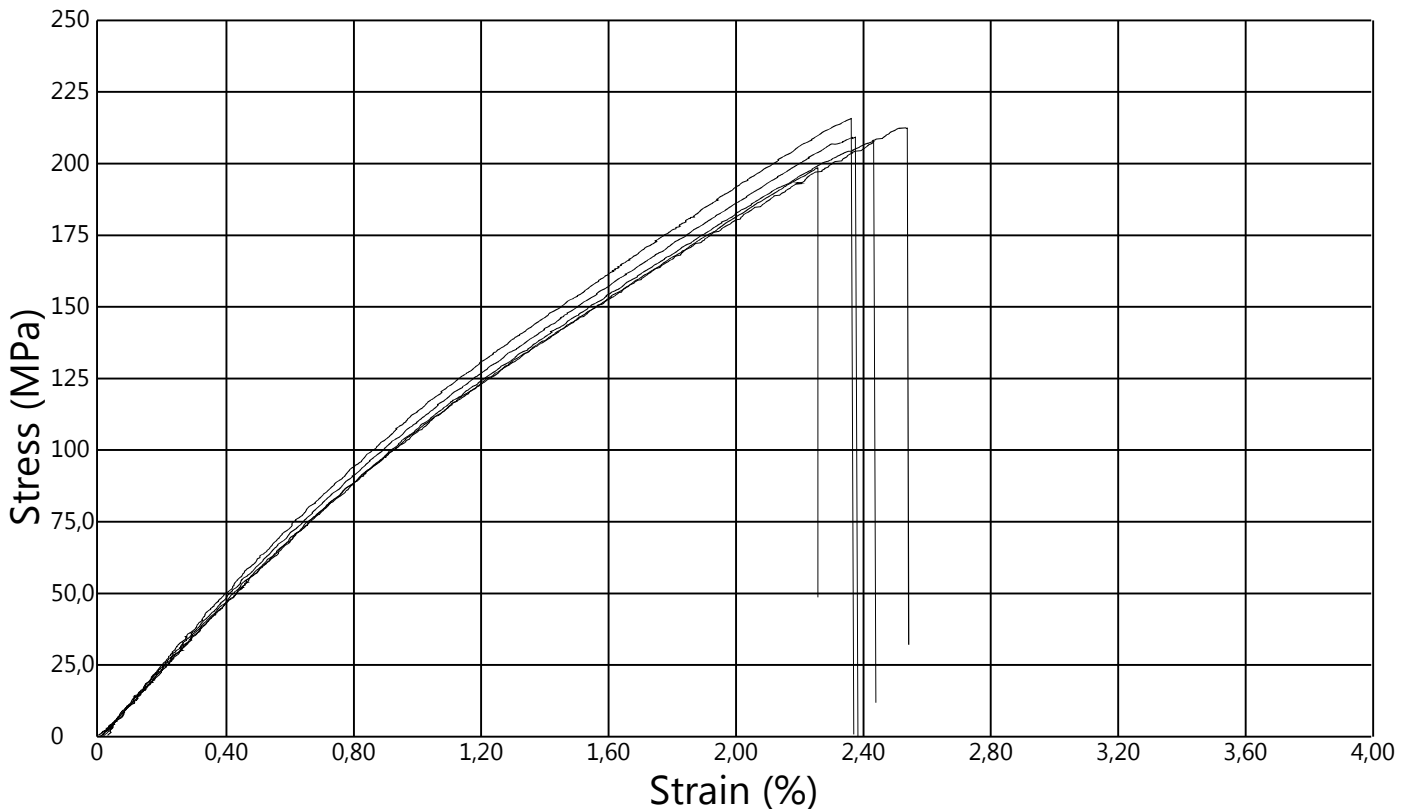
Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
5	15,2	3,14	9490	198	2,26	LGM	11,7
Average	15,2	3,14	9490	198	2,26		11,7
SD			N/A	N/A	N/A		N/A
CoV		N/A	N/A	N/A	N/A		N/A



Operator: Anne Synnøve Brendøy og Kåre Særen  
 Method Name: ASTM D 3039  
 Date: 15.05.2020  
 Speed: 2,00 mm/min  
 Temperature (Entry): 26,0 °C  
 Humidity (Entry): 10,0 %  
 Preload: N/F  
 Batch ID.: SN-2.2s

Specimen No.	Width mm	Thickness mm	Ultimate Force N	Ultimate Stress MPa	Ultimate Strain %	Failure mode	Modulus GPa
1	15,2	3,10	10000	212	2,54	LAB	11,5
2	15,2	3,06	9730	209	2,38	LGT	12,0
3	15,2	3,10	9810	208	2,44	LGM	11,5
4	15,3	3,09	10200	216	2,37	LGM	12,4
5	15,2	3,14	9490	198	2,26	LGM	11,7
Average	15,2	3,10	9850	209	2,40		11,8
SD			263	6,50	0,102		0,387
CoV		0,924	2,67	3,11	4,25		3,28



### 8.13 Material Test Results of Assembly

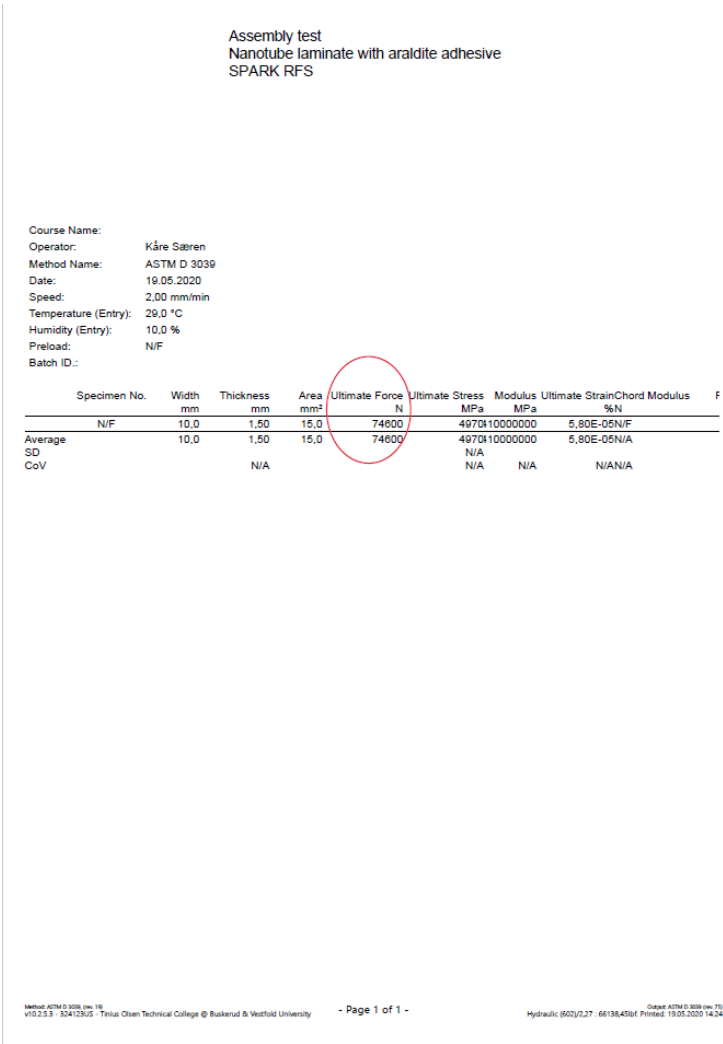


Figure 146: Test of Nano Assembly

Assembly test PETG  
SPARK RFS

Course Name:  
Operator: Kåre Sørensen  
Method Name: ASTM D 3039  
Date: 19.05.2020  
Speed: 2.00 mm/min  
Temperature (Entry): 29.0 °C  
Humidity (Entry): 10.0 %  
Preload: N/F  
Batch ID.:

Specimen No.	Width mm	Thickness mm	Area mm <sup>2</sup>	Ultimate Force N	Ultimate Stress MPa	Modulus MPa	Ultimate StrainChord %N	Modulus F
N/F	10,0	1,50	15,0	28100	1870190000000		-7,15E-05N/F	
Average	10,0	1,50	15,0	28100	1870190000000		-7,15E-05N/A	
SD					N/A			
CoV		N/A			N/A	N/A	N/A	N/A

Figure 147: Test og PETG Assembly

## **8.14 Bill of Material**

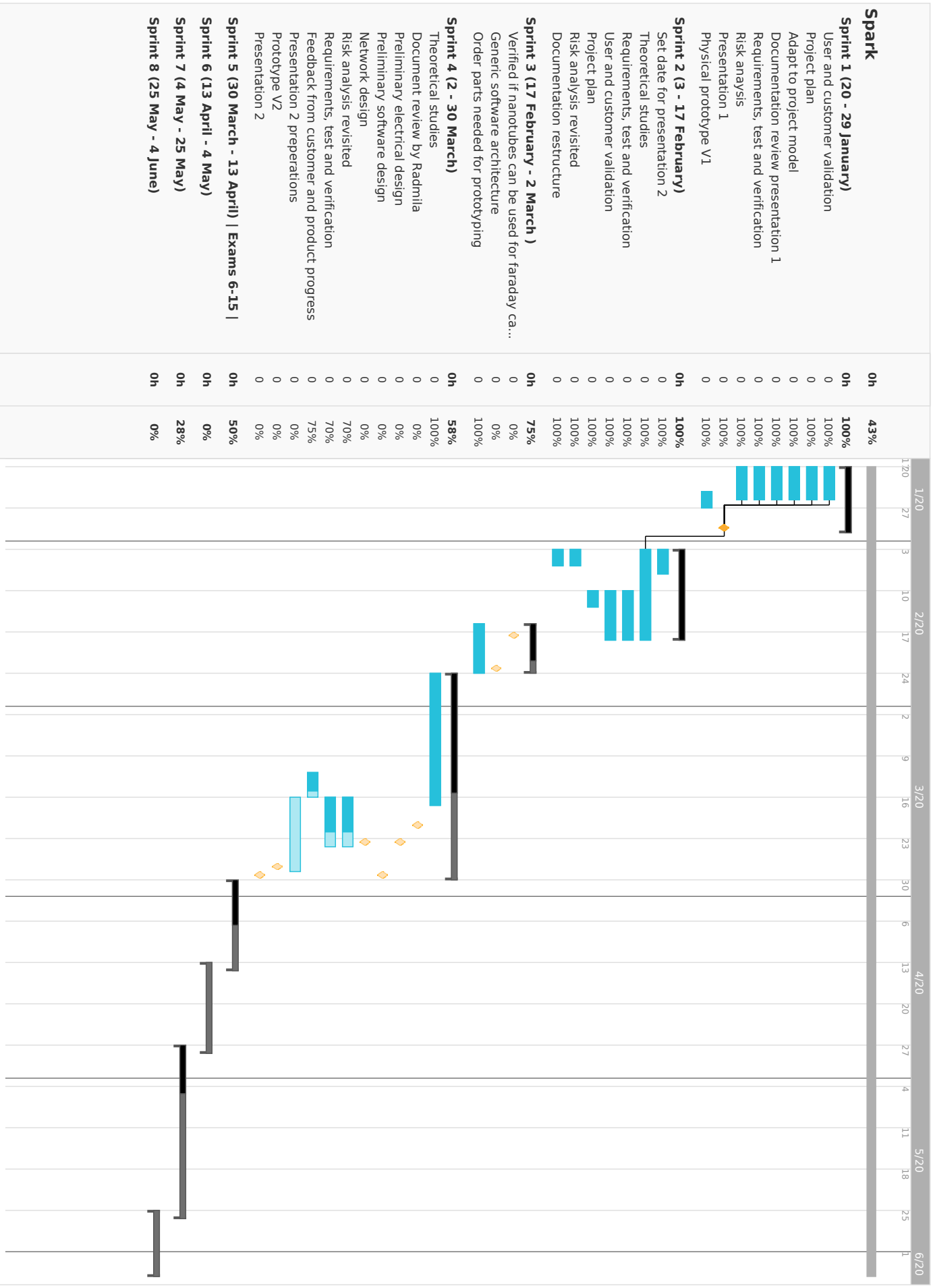
- 1: Casing V3.1. 3D Printed
- 2: Casing V3.2. Nano assembly

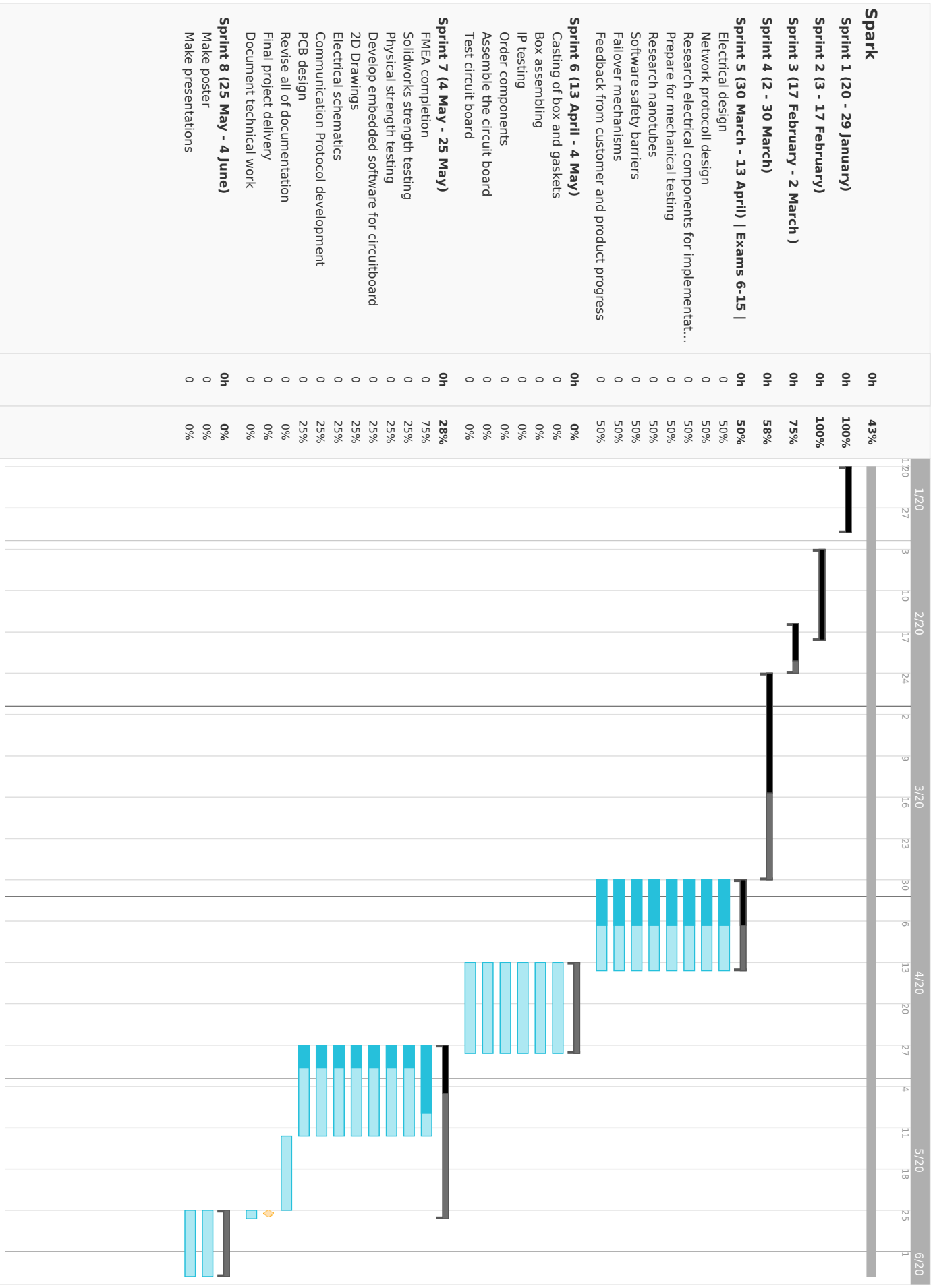
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Bottom device V7.3		1
2	Battery lid		1
3	Lid device V7.3		1
4	Storage lid V7.3		1
5	Battery compartment2		1
6	LEDb		1
7	LEDgr		1
8	LEDre		1
9	Rivet		6
10	B18.6.7M - M3 x 0.5 x 5 Type I Cross Recessed FHMS --5N		6
11	Gasket lid V7.3		1
12	Gasket storage V7.3		1

PART NUMBER	QTY.	DESCRIPTION	ITEM NO.
Bottom nano7.4	1		1
Longside spark7.4	1		2
	1		3
Short side bottom7.4	1		4
Short side top7.4	1		5
Battery compartment2	1		6
Battery lid	1		7
Lid device V7.3	1		8
Gasket lid V7.3	1		9
Gasket storage V7.3	1		10
Short side top lid7.4	1		11
Sidewall lid7.4	2		12
top and bottom lid7.4	2		13
Lid edge7.4 4stk	4		14
B18.6.7M - M3 x 0.5 x 5 Type I Cross Recessed FHMS --5N	6		15
Rivet M3	6		16
LEDb	1		17
LEDgr	1		18
LEDre	1		19



## 8.15 Gantt diagram





## **8.16 SPARK PCB Schematics**

**CENSORED**

**CENSORED**

**CENSORED**

**CENSORED**



**CENSORED**

**CENSORED**

**CENSORED**

**CENSORED**

**CENSORED**

**CENSORED**

**CENSORED**

**CENSORED**



**CENSORED**

**CENSORED**

**CENSORED**

**CENSORED**