Wireless Solution for Marine Engine Control Systems

A study in viable wireless technologies for marine applications

Master's thesis in Systems, Control & Mechatronics and Wireless, Photonics & Space Engineering

Fredrik Andersson
Zrean Tofiq

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's thesis 2017:EX015

Wireless Solution for Marine Engine Control Systems

A study in viable wireless technologies for marine applications

Fredrik Andersson
Zrean Tofiq

Department of Electrical Engineering
Division of Communication and Antenna Systems
Chalmers University of Technology
Gothenburg, Sweden 2017

Wireless Solution for Marine Engine Control Systems
A study in viable wireless technologies for marine applications
Fredrik Andersson
Zrean Tofiq

Supervisors: Jens Samsioe, Volvo Penta, and Morteza Esmaeili Tavana, Department of Electrical Engineering
Examiner: Giuseppe Durisi, Department of Electrical Engineering

Master's Thesis 2017:EX015
Department of Electrical Engineering
Division of Communication and Antenna Systems
Chalmers University of Technology
SE-412 96 Gothenburg

Cover: A ship's wheel with a banner of the five technologies most relevant to this report.

Typeset in LaTeX
Gothenburg, Sweden 2017

Wireless Solution for Marine Engine Control Systems
A study in viable wireless technologies for marine applications
Fredrik Andersson
Zrean Tofiq
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Volvo Penta designs engines, electronic systems, and transmission systems for marine vessels. One of the systems they have created is a drive-by-wire system used for docking marine vessels. This system operates in the form of a joystick, which is currently placed in a number of stationary positions on the vessels. Therefore the driver must remain close to one of these positions when its joystick is used for steering. By exchanging the wired joystick for a wireless system, the driver may freely move around on the boat.

By theoretically evaluating the specifications for various wireless solutions Wi-Fi, XBee, SmartMesh IP, and WirelessHart were chosen for the practical tests. They all meet the power, range, security and speed requirements for this application. With the use of Arduinos, range tests, latency tests, and join time tests were made. Results from these tests in combination with conclusions from the survey were then used to reach a final conclusion of the best available wireless standard for this application.

The only technology that truly fulfilled all the specifications was the XBee 2.4 GHz modules. But to increase security and achieve redundancy the final system consists of both XBee 2.4 GHz and XBee 868 MHz modules. This setup is resistant to interference, malicious attacks and signal jamming. By using two different frequency bands any external attacker is forced to perform two attacks against completely different networks before gaining access to the vessel. This solution was also a good combination between the speed and latency of the 2.4 GHz XBee, and the reliability and penetration of the 868 Mhz XBee.

Keywords: wireless, boats, marine, xbee, wi-fi, smartmesh, wirelesshart.

Acknowledgements

This project was made possible thanks to Volvo Penta's interest in masters thesis at Chalmers University of Technology. At Chalmers we want to thank our examiner Giuseppe Durisi and our supervisor Morteza Esmaeili Tavana for their interest and help in providing information and grading the project. We acknowledge Knut Åkesson and Lars Ulander, who are the directors of the masters programs in System, Control and Mechatronics, and Wireless, Photonics and Space Engineering. Volvo Penta has provided us with office space, equipment and coworkers, all of which has been invaluable for our progress. We also want to thank those at Volvo Penta who have provided us with information and assistance: Lars Johansson, Fredrik Jägersvärd, Juha Korpelainen, Klas Karlsson, and Luca Macellari. Further acknowledgements are directed to C-PAC and specifically Niklas Lange and David Nydahl for their expertise and help with testing. Mats Hellberg at Linear Technologies has been a really big help with equipment and advice for testing.

Lastly we want to offer a big thank you to the person who has helped us the most through this project: Jens Samsioe. Without his assistance and directions this thesis would never have been possible. He has helped us with everything from large purchases and important appointments to small advise and car-pooling. We wish him the best of luck and a happy future with both his family and his career.

Contents

List of Figures

List of Tables

1

Introduction

Wireless solutions are becoming more popular every day. Within the past couple of years virtually every electronic device on the market had wireless functionalities added to it. In this study several wireless solutions for controlling marine vessels will be compared.

1.1 Background

This project was performed at Volvo Penta, who designs electronic systems as well as transmission systems for marine vessels and industrial applications. One such system is used for precision control of boats. This system operates in the form of a joystick by which the operator uses only one hand to dock the boat to the harbour. However, this system is currently attached to a stationary point on the boat which locks the user in place during operation. As seen in figure 1.1, the position of the joystick may not always be optimal and can force the driver to stand in a position where their field of view is blocked.

By implementing a wireless solution for this joystick and its dedicated control system, the boat could be controlled from any point in its vicinity. The wireless solution should not heavily impact the integrity and reliability of the system when compared to the present wired system.

1.2 Purpose

The main purpose of this project is to investigate the possibility of a wireless solution which is robust, then implementing it on a test rig and doing a detailed validation of the system. The project will therefore contain a literature survey, a design stage, and a validation/testing stage.

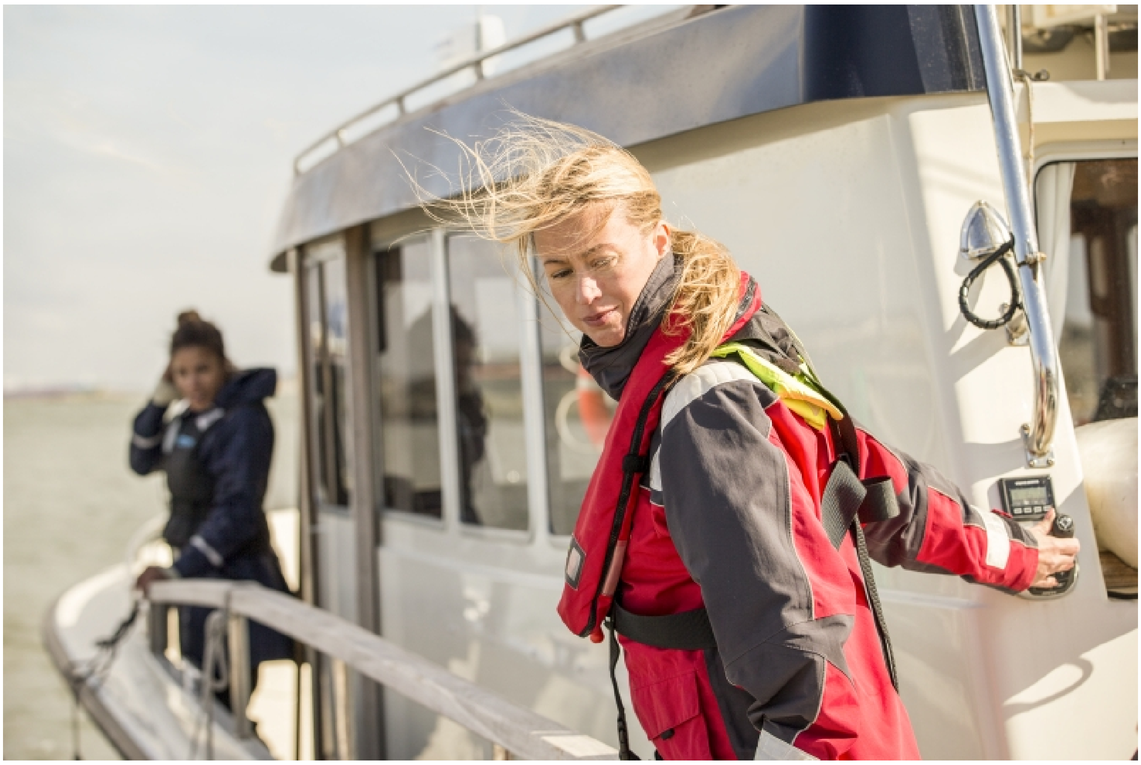

Figure 1.1: Today's wired joystick.

1.3 Boundaries

Since the project concerns the signal between the joystick and control circuit, minimal adjustments are to be made on these systems. The components and subsystems necessary for the wireless solution should be added and implemented on the current systems without altering their reliability and characteristics. Although, the cost should be taken into account, it is not the main objective of this project. The cost should be kept at a minimum as long as it does not jeopardize all other demands of the system. Furthermore, a possible result of this project might be that a robust solution is not possible at a reasonable product price.

One important aspect of the wireless system is its range, which should be enough to give a reliable connection anywhere from the boat. However, there are no specifications or requirements stating that this range needs to be greater or that the system should communicate with any other device.

The power consumption of the joystick needs to be taken into account. It is important for the joystick, since it will be a battery-powered device. How this device is powered and charged is not within the scope of this project.

Even though the signal is never meant to be lost, a part of the project is still dedicated to investigate and implement safe options in case it happens. The timing and activation of these safety measures will also be evaluated, i.e. after how much lag the signal is considered lost.

1.4 Specifications

During this project, the technologies that are considered for testing will be evaluated given a list of specifications. These specifications are demands which must be met by the technology for it to be a viable option for a future product. All relevant specifications are presented below. Most of these parameters can be found in the data sheets of the modules or in technical reports, but the ones that cannot be found online will be tested and verified.

1.4.1 Range

The target boats have a maximum length of about 33 meters (100 feet). As a result, if the access point is placed in the middle of the boat, the minimum range needs to be about 17 meters. However, we cannot ensure that the access point will be placed there by the boat manufacturers. Therefore, a minimum range of 35 meters is required.

1.4.2 Bit-rate

According to the current specifications of the joystick there is a total of 7 values that needs to be sent over the wireless network. The joystick seen in Figure 1.2 has two potentiometers, one HALL sensor, and 4 buttons. Each potentiometer, and the HALL sensor is connected to a 13-bit analog-to-digital converter (ADC), and each button only has a On and Off state. Thus, these values can all be encoded into 43 bits. A usual safety measure is to have a sequence counter, so that the receiver knows in what order packets are supposed to arrive. This counter is assumed to be 8 bits. That leads to a total of 51 bits. By rounding up to the nearest multiple of 8 we get 56 bits, or 7 bytes, per packet.

The current joystick specification sets the period of the message to be about 20 ms. Hence, we shall asume a packet rate of 50 Hz. By using the formula $Bit - rate = bits/packet * packetrate$, we can find that the needed data rate for the payload is $56 * 50 = 2800\ bits/s$.

Beside the payload, each packet contains overhead (i.e., header and trailer), which have different size for different wireless protocols. We assume the payload consists of

40% of each packet. Hence, the minimum needed bit-rate is $2800/0.4 = 7000\ bits/s$, which is $7\ kbits/s$.

Figure 1.2: Volvo Penta joystick.

1.4.3 Latency

When dealing with slow mechanical systems like boats, latency is not as crucial. Even if it takes one tenth of a second before the boat reacts, the driver will hardly feel it. However, if the system lags by seconds or more, it can cause serious harm to both property and people. Therefore, lower latency is preferred, and any wireless solution with more than 100 ms of delay will be considered unsafe.

1.4.4 Join time

The join time of this system is one of the most important parameters to consider. The join time is the time it takes for the joystick to rejoin the network after the signal has been lost. If the signal is lost, the optimal choice for marine vessels would be to activate a digital anchor and keep the boat still. Although since a digital anchor is not an option on all boats, the recommendation is to put the boat in neutral if the connection is lost, and to deem any technology with a join time of more than 3 seconds unsafe.

1.4.5 Security & reliability

Security of a system is defined by its network's ability to prevent malicious attacks and its robustness. The ability to prevent external attacks is usually provided as long as the standard has built-in encryption methods. Robustness of a system is not to be measured, but is a design target. The system is to never loose connection under any circumstances. In case of the improbable happening, the systems needs to be able to handle any situation safely.

1.4.6 Power consumption

The power consumption of a module is primarily important for the joystick as the access point uses the boat as a power source. As for the joystick, the power consumption will determine how big the battery needs to be. The battery size will determine the size, weight, and the time in between charges. There is no specific target we need to reach here, although a module with lower power consumption is preferred.

2
Theory

Before any of the standards are tested in practice, they need to be theoretically evaluated to keep the cost of the project to a minimum. After evaluating the specifications of each standard, they will be compared with each other. The top candidates will be benchmarked on a boat.

After each standard has been evaluated according to its specification, it will be compared with all the other available standards.

The parameters that are evaluated theoretically are listed as:

- Range
- Bit-rate
- Security
- Power consumption
- Cost

Numbers for join time and latency will be obtained through measurements made on the standards that pass the theoretical stage. Bit-rate and signal strength will be verified as well, since these parameters can be affected by custom software or the environment.

2.1 Wi-fi

Wi-fi is a well known standard that has been refined over the years. The latest Wi-fi standard (IEEE 802.11ac) can achieve a data-rate of 1.3 Gbit/s in ideal conditions [1]. Wi-fi also uses a 256-bit advanced encryption standard (AES) for enhanced security; compared to most other standards who uses 128-bits. All these specifications makes Wi-fi an ideal candidate for marine applications.

Most Wi-fi systems have a transmission power of almost 20 dBm, which equals 100 mW. By assuming purely isotropic antennas on both transmitter and using Frii's equation 2.1 it is possible to calculate the maximum theoretical range of a typical Wi-Fi chip.

$$P_r = \frac{P_t G_t G_r \lambda^2}{4\pi R} \tag{2.1}$$

P_r is the received power, P_t is the transmitted power, G_t and G_r are the transmitter's and receiver's antenna gains, λ is the wavelength and R is the range. $G_t = G_r = 1$ Since the antennas are assumed to be purely isotropic, and λ can be calculated from the frequency. For a typical Wi-Fi chip the receiver sensitivity is around -72 dBm [2]. By setting the received power to the receiver sensitivity and solving for range, a maximum theoretical range of 390 m is calculated.

One downside to using Wi-fi is that the peripheral hardware is highly power consuming. Another downside is the need for an access point (i.e. router), since the number of nodes in the network is increased; which in turn increases the total cost of the system. In the absence of a router, one can choose to make use of the Wi-fi Direct protocol or it's predecessor Ad-hoc. One drawback with Ad-hoc is that it severely limits the bandwidth of the connection [3]. Wi-fi direct addressed this issue, but since the hardware is so power hungry a more efficient solution is preferred.

2.1.1 Wi-fi Ad-hoc

Wi-fi Ad-hoc mode is basically the same as Wi-fi, but only with its core functionalities activated. The bare-bone of the specification allows for a dynamic peer-to-peer (P2P) network to be set up. There is usually no built-in security on the network layer. Instead it is usually implemented on the application layer. Ad-hoc networks are usually used for networks that require dynamic capability. Since Ad-hoc networks usually have no encryption, the number of packets that need to be sent before actual data is small. It is easy for devices to join and leave networks one the fly.

Since Wi-fi Ad-hoc is based on the bare-bone specifications of Wi-fi, it allows for a lot of freedom when setting up the protocol. However, it misses some good, but nonessential, functions for our application such as frequency hopping.

2.1.2 Wi-fi Direct

Wi-fi Direct is not an IEEE standard as it is a group of specifications set by the Wi-fi Alliance. It allows for devices to connect to each other without the need of a network coordinator such as a router. Wi-fi Direct devices negotiate their roles in the connection process. Usually one device takes the role of an access point (AP) while the other devices connect to the AP. One can argue that Wi-fi Direct is not true P2P as all the devices on the network are not equally privileged. But for all intents and purposes in this study it does not matter if the network is truly a P2P network.

One of the advantages of Wi-fi Direct is that it builds on top of already existing Wi-fi standard instead of rolling back to the bare-bones. Wi-fi Direct has built-in security and has more or less the same pros as normal Wi-fi.

A drawback with Wi-fi Direct would be the fact that since it build on top of the general Wi-fi specifications. It requires a lot of packets to be sent back and forth for authentication purposes. Essentially, it will take longer time to connect to a Wi-fi Direct network compared to Ad-hoc.

2.2 XBee

XBee can use either the ZigBee or the Digimesh protocol. Both have a layer-architecture; i.e. the protocol is split into four layers: physical, medium access control, network, and application [4]. The physical layer and the medium access control layer uses the IEEE 802.15.4 standard and all necessary software is handled by the top two layers, network and application.

The XBee modules used in this project, which uses the Digimesh protocol, is developed for networks of short range and low data rate applications. In spite of this, it provides the longest range among all of the technologies discussed before. For instance, the ranges are about 750 m for 2.4 GHz and 40 km for 868 MHz, with receiver sensitivities of -100 dBm and -112 dBm respectively [5] [6]. Moreover, it offers low cost and low power communication between a large number of nodes.

Most protocols running on the XBee standard uses encryption keys for communication between different devices. Hence, data between layers is always trusted. When broadcasting, XBee uses a 128-bit network key which is known by all devices in the system [5]. There are often multiple network keys stored on each device, making it possible for the system to switch the active key. On the application layer, there is also a possibility of secure unicast by using a 128-bit link key.

XBee systems use a direct-sequence spread spectrum (DSSS) [5] to insert noise with a known distribution into the signal which makes it less sensitive to white noise and interference. Therefore its signal is not easily jammed.

2.2.1 Operating frequencies

XBee operates in several different frequencies, where the three most common are 868 MHz, 900 MHz and 2.4 GHz. The choice of frequency is generally a trade-off between data-rate and range. Since the theoretical range is proportional to the wavelength according to Frii's equation [7] a higher frequency means shorter range. A higher frequency also makes it easier to have a large bandwidth. Therefore, one can reach higher data rates with a larger bandwidth, since the maximum sampling rate is two times the bandwidth.

A lower frequency also reduces the power consumed by the transceiver since the efficiency of its transistors drops with increasing frequency. On top of this, components designed for higher frequencies are generally more expensive than those designed for low frequencies. Moreover, higher frequency does not have the same penetration through materials as low frequencies do, which in turn means that higher frequencies have a higher tendency to bounce on objects without suffering as high loss.

2.3 WirelessHART

WirelessHART is basically the HART automation protocol, but wireless. The HART protocol is a fieldbus network standard developed in 1980 [8]. Both HART and its wireless version are designed for communication between sensors, actuators, and their control systems. WirelessHART uses radio modules compatible with the IEEE-802.15.4 standard for wireless communication, which is the same standard as XBee. It also uses DSSS, frequency hopping and 128-bit AES encryption. At the frequency of 2.4 GHz, it can reach data rates of 256 kbit/s on each of its 16 channels.

The data link layer (DL) is time synchronised in order to avoid crosstalk. This means that it uses time division multiple access (TDMA) to give each transmission 10 ms of communication time [9]. Since the TDMA protocol is used for transmission and reception, it takes 20 ms to transmit one packet and receive its acknowledgement. For sensors, which are monitored a couple of times per minute, this is not an issue. However, for this project it could cause a bit of unwanted delay between the wireless access point (AP) and the joystick. There also exists a process known as blacklisting in the data link layer, which can blacklist one channel if transmissions on that channel are deemed too poor. Channels in the blacklist can then be disabled by the user.

For security, a combination of cyclic redundancy check (CRC) and message integrity code (MIC) is used in the MAC-layer [8]. There are different security keys used for different scenarios, for example there is one network key used by the devices on the network to generate MICs and a join key to authenticate a joining device.

2.3.1 SmartMesh IP

There is also a technology called SmartMesh IP, developed by Linear Technologies, which is similar to WirelessHART [10]. The SmartMesh IP products run on a system-on-chip that manages large mesh networks with low power consumption and high reliability [10]. The main difference with WirelessHART being a more user-friendly software and the use of IPv6 addresses [11]. Just like WirelessHART, Smartmesh IP uses time division to schedule transmission; so two nodes (also known as motes) cannot transmit at the same time. It also uses frequency agility [12] to change channel if one frequency is too noisy and thereby interferes with the transmission [10].

Smartmesh IP also offers low power consumption since the motes usually transmit for a short time, and it uses a MAC engine instead of a micro-controller, which further reduces power consumption.

2.3.2 Bluetooth

The two newest Bluetooth protocols on the market at the writing of this survey are Bluetooth 4.2 and Bluetooth low energy (BLE). Bluetooth 4.2 can send packets of 256 bytes at a theoretical data rate of 800 kb/s [13]. BLE on the other hand has a higher data rate, up to 1 Mbit/s, and a smaller packet size of 27 bytes [14]. The latest Bluetooth technology, Bluetooth 5, has just been released and is not yet available on the consumer market. Bluetooth 5 can transfer up to 2 Mbit/s and has a range of about 240 m, while still consuming the same amount of power as BLE; 30-40 mW while transmitting [14, 15]. The data packets of Bluetooth 5 have a capable size of 255 bytes, resulting in 1024 packets being sent each second at a transmission speed of 2 Mbit/s. Bluetooth 5 also supports up to 37 broadcasting channels and is using frequency hopping to make jamming and hacking more difficult.

2.4 Evaluated and dismissed standards

In addition to the above mentioned standards, there were three other standards that were considered, but eliminated due to the specifications being unsuitable for a marine application.

2.4.1 Symphony Link

Symphony Link [16] is a Low Power Wide Area Network (LPWAN) created by Link Labs. The purpose of this network is to achieve maximum range with minimum power consumption. Most of the specifications are more or less like Wi-fi. However, the architecture is using mesh networking to achieve its range. Since the modules are low powered, the advertised range will require more than two nodes in the network. LPWAN networks are usually better suited for a large amount of wireless sensors or gadgets that need to reside on the same network. Since this study focuses on the P2P functionality of a network, LPWAN was discarded as a candidate.

2.4.2 Wi-fi HaLow

Wi-fi HaLow [17], also known as IEEE 802.11ah, is a new standard specified by the Wi-fi Alliance in 2016. HaLow uses sub-GHz frequencies to operate and is meant to replace other low power technologies such as Bluetooth. HaLow supports the concept of Internet of Things (IoT) and is therefore compatible with what this study is trying to achieve. Unfortunately, since the technology just came out, it is not possible to buy a HaLow compatible module on the consumer market. And since new technologies always need some time to mature, HaLow was not considered a good candidate.

2.4.3 ISA100

The International Society of Automation (ISA) has developed a standard called ISA100.11a which is build on the IEEE 802.15.4 technology [9]. ISA100 shares a lot of properties with WirelessHART in its low power low data rate design. However, ISA100 is built to be more flexible than WirelessHART. Where WirelessHart mainly uses TDMA, ISA100.11a uses a hybrid between TDMA and carrier sense multiple access with collision avoidance (CSMA/CA) [18]. The physical properties of ISA100.11a are similar to those of XBee which it is based upon. It uses 15 channels, each with 2 MHz bandwidth, in the 2.4 GHz band which it alternates between using frequency hopping. The ISA100 also uses the modulation scheme offset quadrature phase shift keying (O-QPSK) and can reach bit rates of 250 kbit/s.

ISA100.11a was intended to be tested in this project but the modules are quite difficult to find without going through ISA. Unfortunately the ISA would not respond to our attempts at collaboration, meaning that testing this technology proved impossible.

2.5 Comparison

Bluetooth 5 has improved greatly since Bluetooth 4. The optimization and redesign choices made to Bluetooth 5 has made it a top candidate. Using Wi-fi as a reference to compare to; Bluetooth 5 has better power consumption, better range, tighter security, and is cheaper. The only downside being it's bit rate, which is lower than Wi-fi.

XBee is quite similar to Bluetooth, but it is newer and has better specifications. XBee has basically the same security levels as Wi-fi, and the bit rate is a bit slower, but other than that it performs better in all categories.

Both Wi-Fi HaLow and Wi-Fi have very similar values, the only difference being that Wi-Fi HaLow has lower power consumption and better range at the expense of data-rate. Since, as stated in the section 2.4.2, Wi-Fi HaLow is not yet available for testing, it may not be used for the result of this project.

Symphony Link and Bluetooth 4.0 are the two final choices. Symphony Link requires too many nodes in the network to achieve its maximum capabilities. Additionally, it is not designed for P2P applications link the joystick. Therefore, Symphony Link is dismissed. Bluetooth 4.0 was dismissed due to its poor performance relative to the other technologies and due to its lack of robustness.

2.5.1 The pugh matrix

After comparing the technologies mentioned in sections 2.1-2.3, a list of specifications needs to be set up. In table 2.1, all of the specifications are listed together with a weight, representing the importance of that criteria in this study. This type of table is called a pugh matrix [19]. Since Wi-fi is among one of the oldest technologies on the list and is known to work well, it is used as a reference for the other technologies to be compared against.

If a technology has a parameter that is better than the reference it is given a score of +1, if it is worse it will receive -1, and if they are equally good the score will be 0. The total score is then calculated by multiplying the score for each parameter with it's weight. Finally the scores for all the parameters are added together.

According to the pugh matrix, Bluetooth 5.0 is the preferred choice for this project. Because of the great improvement of Bluetooth 5 compared to Bluetooth 4, it is a viable option for this project. Unfortuantely there is no way of testing this technology due to its recent release. Thus, it may be a good option for future products or upgrades when it may be thoroughly tested.

Table 2.1: Pugh matrix showing how each standard compares to Wi-fi.

	Weight	Ref - Wi-fi	Bluetooth 4	Bluetooth 5	XBee	Symphony Link	Wi-fi HaLow	WirelessHART
Max bit rate	5	0	−1	−1	−1	0	−1	−1
Power consumption	8	0	1	1	1	−1	1	1
Theoretical Range	6	0	−1	1	1	1	1	1
Built-in security	10	0	−1	1	0	0	0	1
Cost	7	0	0	1	1	−1	0	−1
Total		0	−13	26	16	−9	9	12

The second choice in the Pugh matrix is XBee. Its ability to operate in mesh networks further strengthens it since the decision of one or several access points is yet to be made. Next is WirelessHART, and therefore also SmartMesh IP, at the third highest value, this standard has similar hardware as XBee but is slightly more refined for robustness in noisy environments. They offer better security than XBee but are at the same time a much more expensive alternative.

3
Methods

To test the different standards in a marine environment, a test rig had to be made for each technology. To make sure that the tests were fair, the same micro-controllers and software structure were used for each test. Only the wireless components were changed between tests. The tests were designed to measure the signal strength, latency, and join time of each standard.

3.1 Test rig

First and foremost a hardware rig was set up to run the tests. In this case, it consists of three Arduinos, two ESP8266 Wi-fi modules, six XBee modules, and a starter kit for SmartMesh WirelessHART and SmartMesh IP. There are two types of XBee modules in the rig. One of them runs on the 2.4 GHz band and the other on the 868 MHz band. In addition to these modules, a DIY joystick[1] was used to simulate the actual Volvo Penta joystick. At the end of the project this DIY was replaced with a real Volvo Penta joystick, for demonstration purposes.

3.1.1 The Arduino

An Arduino Mega 2560 was used together with two CAN bus shields, one of which is seen in figure 3.1, to act as the access point for the joystick. By using different shield configurations, each wireless technology was tested individually together with the AP and the joystick.

Since the actual hardware of the AP and the joystick are not part of this study, their cost and performance will not be evaluated. The Arduinos and the joystick are simply used to evaluate the wireless standards.

[1] `https://www.sparkfun.com/products/14051`

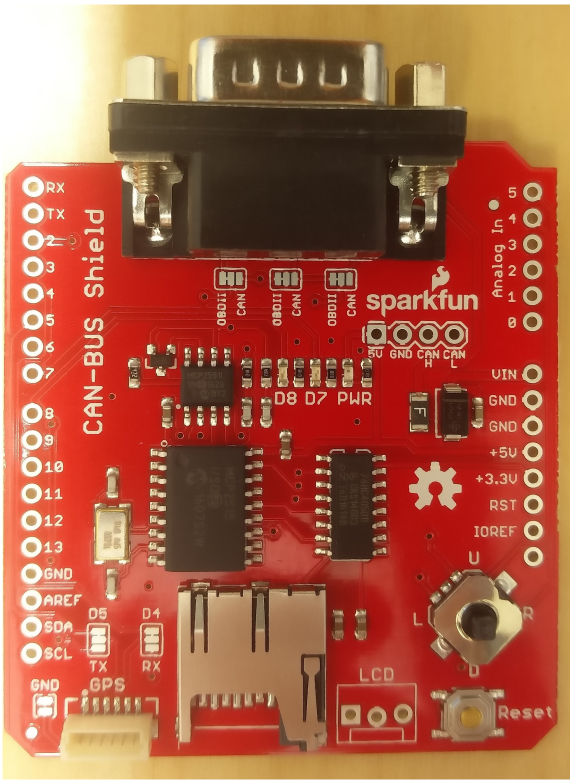

Figure 3.1: The CAN bus shield used together with the Arduino to interface with the boat's CAN network.

3.1.2 Wi-Fi

To see if Wi-Fi is a viable option, the ESP8266 Wi-Fi module, as seen in figure 3.2, was used. The ESP8266 is a relatively new piece of hardware and has gained a lot of popularity lately for its low cost and excellent performance. It was chosen for its large amount of community support and its reasonably low power consumption compared to other Wi-Fi modules.

The ESP8266 is actually a full system on chip (SoC). It features a 32-bit processor that handles the entire TCP/IP protocol stack. The module is reprogrammable, with about 80% of its processing power free for user applications. It is also possible to use the model simply as a serial-to-wireless interface and send data from the Arduino. Since it is simpler to work with the Arduino, the ESP8266 was only used as a serial-to-wireless interface.

3.1.3 XBee

The 868 MHz XBees were ordered in the form of a development kit which contained three Xbee modules and three development boards. The 2.4 GHz modules were however ordered separately and connected by the use of XBee shields, one for each

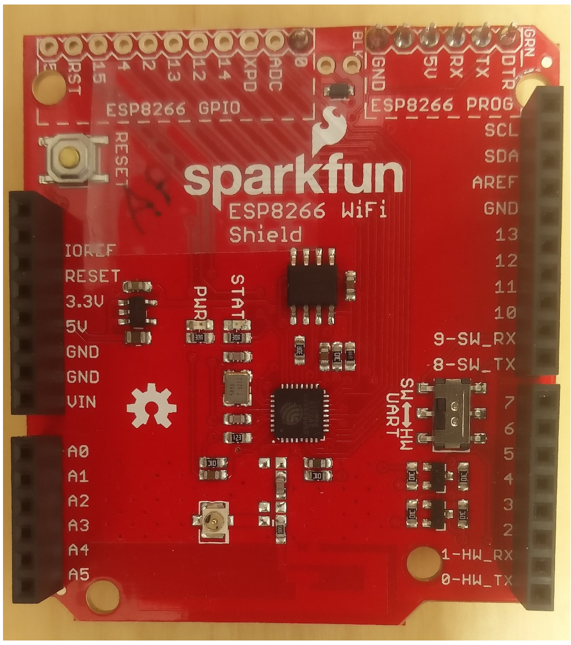

Figure 3.2: The ESP8266 Wi-fi module.

Arduino. Figure 3.3 shows the both the 868 MHz XBee and the 2.4 GHz XBee connected to their respective boards and antennas.

The XBees run with the default parameters in place with some exceptions. Encryption was activated on all the modules, and they were set up as a point-to-pont type network with packetization set to 0. Hence, the modules are essentially streaming the bytes over the network as they arrive.

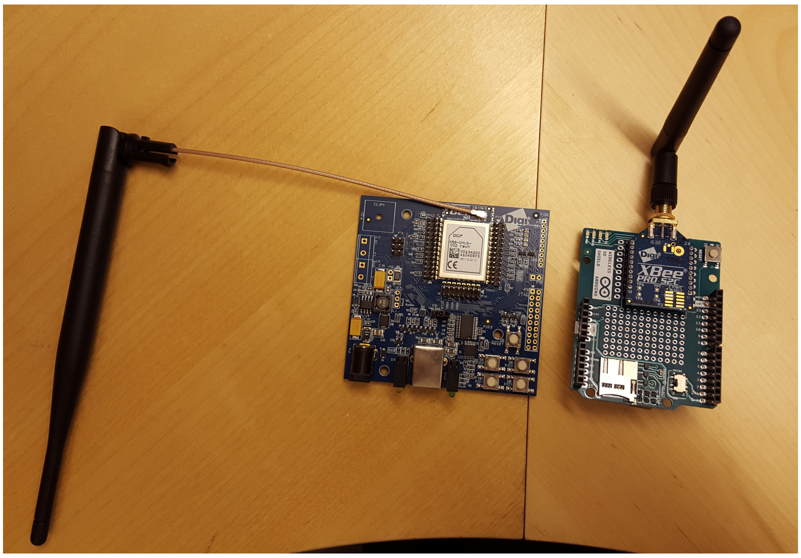

Figure 3.3: 868 MHz and 2.4 GHz XBees connected to their boards and antennas.

3.1.4 SmartMesh WirelessHART & SmartMesh IP

Both of these technologies are made for sensing networks, which only transmit at certain time frames, thus they do not need low latency. Therefore, they must be set to backbone or fastpipe mode, in which the latency is low and the powered motes are constantly listening [20]. Naturally, their ultra-low power consumption is nullified. However, due to the demands on delay and refresh rate in this project, this is a necessity.

These devices are compatible with, can be controlled, and monitored by the Arduino Mega. Nonetheless, the already developed software did not work as intended and the massive work to write new code for communication between the motes and the Arduino was deemed too large. As a result, all necessary tests were done using the command line interface (CLI) or through the application programming interface (API).

The motes came pre-programmed and configured as Smartmesh IP. Therefore, they had to be re-programmed to be used for Smartmesh WirelessHART. This was done using software for the Eterna Serial Programmer, which is a script that erases or loads the mote's flash when called with specific parameters.

3.2 Tests

Testing was performed with each of the specifications in mind: bit rate, range, latency, join time, and security. The range aspect was split into several parts, both the actual range of the systems and the relative positions of the transceivers.

The latency of the system shows how much overhead is included in the protocol. If there is a lot of overhead data that needs to be sent back and forth, or if there are too many layers in the protocol stack, the performance of the system could be affected.

The join time of the system is probably the most important attribute since it directly relates to how fast one can regain control of the system after connection has been lost.

The security of the system will be evaluated according to the specifications of each protocol, but no actual penetration testing will be done on the system.

3.2.1 Bit rate

As mentioned in section 1.4.2 the estimated data rate needed is 7 kbps. Since the slowest protocol considered has a theoretical speed of 80 kbps, as a result, we can assume that, no matter the protocol, the bit rate will be enough. Therefore, the bit rate will not be measured.

3.2.2 Signal strength map

Through measuring the signal strength, or received power, for various positions of the joystick and its relation to the AP, signal strength maps could be generated. When performing these tests the components were also placed in a manner to test the penetration of the technology and to see how it handled obstructions.

Due to the attenuation in air being different during heavy rain, the signal strength was also tested during poor weather conditions. Regardless of the size of the boat, maximum range tests were also performed. Partly to see the range margin of error on large boats and partly to see the impact in bit-rates as range increased.

For all the standards evaluated in this project there is a unit called received signal strength indication (RSSI), which may be read by the microprocessor. This value, which is generally below 0, has a linear relation to the received power level in dBm [21]. However, it also has a offset in relation to the actual power level, and this offset varies between manufacturers and chip sets. Therefore, the produced signal strength map will not be relative to received power levels but rather to the highest RSSI value obtained with that technology.

3.2.3 Latency

A simple latency test program was made for the Arduino as seen in appendix A.2. This check comprised of measuring the time on the transmitting Arduino between before transmission and after reception of the acknowledgement. It was then estimated that this time would be twice the time for the original message to be transferred and interpreted.

Both SmartMesh IP and SmartMesh WirelessHART had a function called trace stats, which showed the time for transmission and reception, as well as the difference. Through this function it was possible to determine the latency of the respective technologies. At its normal operation, the latency for these two technologies is very high, but by setting them into backbone mode they may still reach rather low latencies.

Further latency tests were also done indoors to determine the relation between latency and range. For these tests, an old factory building was used. The room is roughly 250x150 meters, with concrete walls for the signal to bounce on. This in combination with line-of-sight transmission made for the most ideal practical case.

3.2.4 Join time

When signal is lost, the network node should automatically reconnect. This procedure has what is known as a join time; i.e. the time it takes for the node to rejoin the network. For WirelessHART and SmartMesh IP, this time was measured ten times.

Join time was also measured for the XBee devices, both at 868 MHz and at 2.4 GHz. These tests were performed at four different distances and repeated five times to get an average value. This was to conclude if the join time was independent of the range of transmission. The ranges used were: Approximately 0 m, 5 m, 15 m and 30 m. When performing these tests, the joystick-unit was left on while the access point unit was placed a distance away in an office environment. The Access Point was then turned off and on and the time between boot and first message received was extracted. These tests were also done in the same manner for the Wi-fi technology.

To make sure that the results are statistically valid a stress test was performed to refine promising results, seen in appendix A.1. The Arduino turns the wireless module on and measures the time until the first message arrives, and then turns it back off again. This is done 1000 times or until the communication link fails. Then the values can be plotted to a histogram to show any trends.

3.2.5 Security & reliability

The security of the system will be analyzed in two parts. First, the network needs to be protected from external attacks. The second part consists of a failure-mode and effects analysis (FMEA), which describes what potential states the system could enter, and how potentially dangerous states are prevented.

Physical penetration testing is not performed. Instead, recommendations are provided, together with the FMEA, as to how known security flaws can be blocked. Finding unknown security flaws in these protocols are out of the scope of this project.

3.2.5.1 Failure modes & effects

There are essentially two types of failures in this system. Either there is no signal and therefore no control, or there is a signal but the data is somehow corrupted and causes unwanted control. There are numerous causes to such failures, and not all of them can be prevented. However, by using redundant parts in the system, single points of failure will be avoided.

Corrupted messages and alike are hard to detect, but can be prevented by the use of HMAC [22] algorithms to authenticate and validate the messages as described in [23, 24].

4

Results

This chapter presents the results from the testing. Whenever one of the technologies failed to show satisfying results during testing it was discarded and not used in the following tests. As a result some technologies will not be mentioned in certain tests. Each test was done in a rough manner, i.e. sample times and test points were chosen in a sparse manner, to get a better overview of the range of values expected. If promising results were observed, more thorough testing was done.

4.1 Range

The range of each technology was tested in an indoor environment to see if it was able to reach a total distance of about 33 meters, which is the largest boat Volvo Penta works with. The maximum range was determined by continuously sending data from the joystick to the access point and then observe the range at which no more packets were received. For those technologies with mesh capabilities, the range is the total distance for each hop, as the total distance of the network is, in theory, indefinite. The results for each protocol can be observed in table 4.1.

Table 4.1: Approximate range of each protocol. The > symbol means we ran out of space.

Protocol	Approximate Range (Meters)
XBee Digimesh 2.4GHz	>250
XBee Digimesh 868MHz	>250
Wi-fi	100
SmartMesh IP	130
SmartMesh WirelessHART	130

The motes of SmartMesh WirelessHART have the same hardware as the motes of SmartMesh IP, meaning that for range their values should be the similar [25]. Therefore the range of SmartMesh WirelessHART was not tested, but assumed to be approximately the same as for SmartMesh IP, which is already far beyond what is necessary.

Some measurements were also done with the XBee modules on a PTA80 boat to study the signal strength concerning the boats topology. This was done by placing the access point on a fix location, by the drivers seat, and noting the RSSI value from several locations on the boat. The results from these measurements was then used to create the heatmaps shown in figures 4.1 and 4.2.

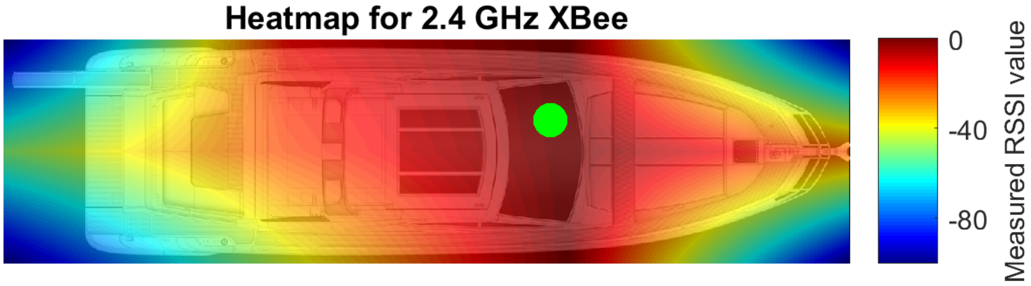

Figure 4.1: Heatmap of the RSSI-value for the 2.4 GHz module for different positions on the boat.

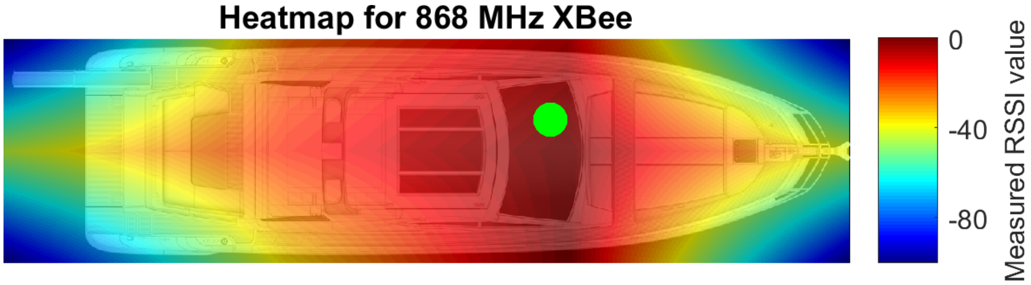

Figure 4.2: Join time measurements for XBee 868 MHz measured in milliseconds.

4.2 Latency

The latency of each technology is presented in the subsections below. They denote the average time it took for each package, out of 100, of 1 byte each to reach its destination.

4.2.1 XBee

Xbee 2.4 GHz, which uses the Digimesh protocol, had an average latency of 52 ms for message and acknowledgement. The 868 MHz version had a latency of 155 ms for two-way communication. Therefore this communication test showed that the 2.4 GHz has a delay of around 26 ms as seen in table 4.2, and the 868 MHz has a delay of 78 ms as seen in table 4.3, for single-way communication. The distance between the joystick and the access point was varied between 0-30 m and it was concluded that the distance had a slight affect on the latency.

Table 4.2: Approximate latency of the 2.4 GHz XBee module varied over distance. Time is measured for one way communication.

Distance (m)	Avg. Time (ms)	Packet loss (%)
0	24	0
5	24	0
15	25	0
30	31	0

Table 4.3: Approximate latency for one-way communication of the 868 MHz XBee technology for distances between 0 m and 30 m.

Distance (m)	Avg. Time (ms)	Packet loss (%)
0	75	0
5	75	0
15	76	0
30	85	0

4.2.2 Wi-fi

Wi-fi performance is quite close to that of the 2.4 GHz XBee. The average time it takes for a message to be sent is about 30 ms as seen in table 4.4.

Table 4.4: Approximate latency of the ESP8266 Wi-fi module varied over distance. Time is measured for one way communication.

Distance (m)	Avg. Time (ms)	Packet loss (%)
0	24	0
5	24	0
15	25	0
30	31	0

4.2.3 Smartmesh IP & WirelessHART

Latency tests for SmartMesh IP showed that in backbone mode the upstream latency usually altered between 36 ms and 14 ms. However, for longer range there occurred latency spikes where one transmission would yield a latency around 50 ms or even 100 ms, then the latency would return to normal for the next transmission. These latency spikes also became more common the greater the range. Even though the latency went up slightly, no packet drops were observed for any of the tests. Table

Table 4.5: Latency for transmission from mote to manager for SmartMesh IP.

Distance (m)	Avg. Time (ms)	Packet loss (%)
0	23.8	0
5	22.7	0
15	31.9	0
30	47.7	0

4.5 shows the average from 20 measurements for each of 4 distances, and their respective packet drop ratios.

For SmartMesh WirelessHART the mote was pinged 100 times for each distance. Both backbone and fastpipe mode were activated for these tests, i.e. the system was set for this type of application. The results from these tests are presented in 4.6

Table 4.6: Latency for communication between manager and mote for SmartMesh WirelessHART.

Distance (m)	Avg. Time (ms)	Packet loss (%)
0	108	0
5	103	0
15	112	0
30	130	0

4.3 Join time

The join time turned out to be parameter that varied the most between the technologies. Since the join time is highly critical for our application most standards were discarded in this step.

4.3.1 SmartMesh IP

The results from the join-time measurements for SmartMesh IP are presented in table 4.7 below. These measurements were taken when the join bandwidth was at 100 %.

These join time tests were also briefly done for WirelessHART. Here the join duty cycle was also set to 100 % and the time was measured manually to get an estimation of the join time before creating a script to measure it more precisely. Because the average join time for 10 tests were as high as 31.8 s, with a minimum of 19 s, no further testing was deemed necessary.

Table 4.7: Highest, lowest and average join times of SmartMesh IP measured in milliseconds.

Join Time (ms)		
Max	Min	Average
15150	9756	12680

4.3.2 Wi-fi

Following this were the join time tests done with Wi-fi. Since five measurements were taken from each distance, their respective max, min and average values were compiled into table 4.8.

Table 4.8: Maximum, minimum and average join times of W-Fi.

Distance (m)	Time (ms)					Avg. Time
0	5573	4231	4603	7242	5199	5369.6
5	4603	5395	5005	2727	7044	4954.8
15	3521	3661	2585	4842	2544	3430.6
30	5114	5115	5451	5229	5215	5224.8

4.3.3 XBee

Since XBee was the most promising technology in terms of join time, a larger number of tests were performed. After 1000 tests at a distance of 0 m, the histograms in figure 4.3 and figure 4.4 was made. As can be seen in these figures, the 2.4 GHz XBee had a join time of below half a second most of the times with a few join times above one or two seconds. The 868 MHz XBee was however very consistent with almost all samples around 7.5 s.

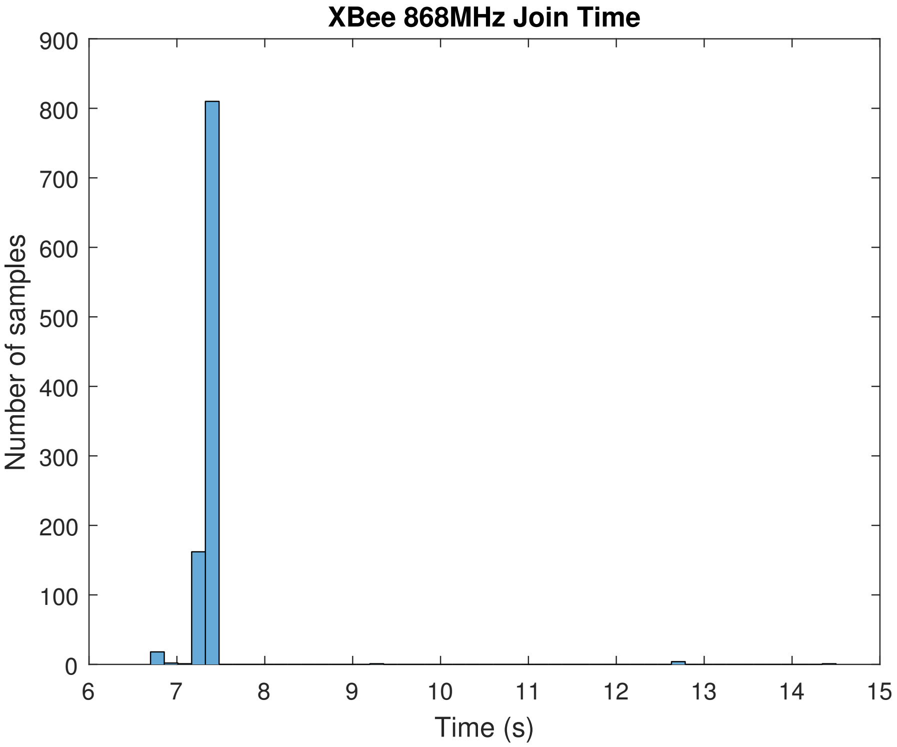

Figure 4.3: Join time measurements for XBee 868 MHz measured in seconds.

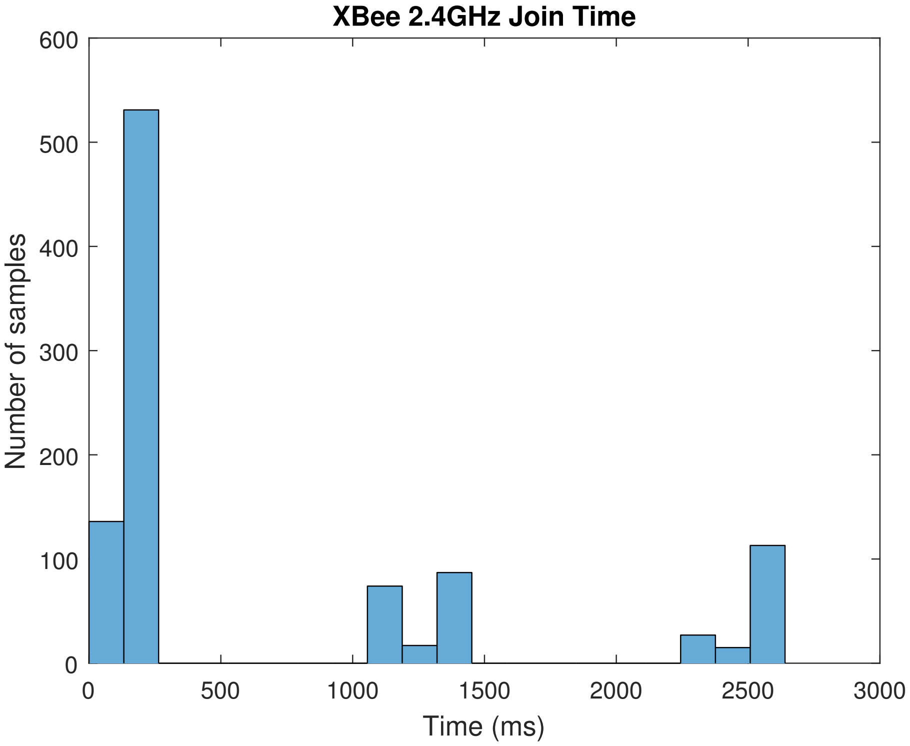

Figure 4.4: Join time measurements for XBee 2.4 GHz measured in milliseconds.

4.4 Security & reliability

To analyze the systems security an FMEA was composed as seen in table B.2. All the system failure modes are listed together with a rating, which shows how dangerous that fault mode is. The current solution for each fault mode is also listed together with potential action needed to be taken by the boat crew. To fully understand the ratings in the FMEA refer to table B.1.

While testing the robustness of the system the joystick was toggled on and off 1000 times. The network did not fail once and therefore it is assumed that the modules will be able to join the network at all times while in range.

4.5 Live testing on boat

When a final design had been made it was time to test the system in its right environment; on a boat, on the sea. So the access point was connected to the CAN-bus through its CAN-transceivers and the boat was wirelessly controlled. In these tests the 2.4 GHz XBee and the 868 MHz XBee were used as a dual-band solution since they showed the best performance in earlier testing. It was also concluded, both from this test and several other tests, that the bit-rate of both devices were enough for this application.

The 2.4 GHz XBee showed very promising performance with high stability and low latency. However, the 868 MHz XBee had a very long delay of 4 s. This delay was overcome by slowing down the transmissions to sending at 25 Hz, i.e. every 40 ms.

5

Discussion

This chapter discusses the results from testing the various technologies, it then does a comparison of them to determine the best technology for the application. Alongside this, security aspects and positioning of the access point is also evaluated.

5.1 Join time

Generally the join times for most technologies were worse than expected. This is the biggest reason behind the final choice of technology.

5.1.1 Smartmesh IP & Smartmesh WirelessHART

While these technologies are very similar and show great results when it comes to robustness, range and latency, they have a very long join time. This means that it is very unlikely that connection will be lost between the joystick and the access point; but if it is lost it will take time before the connection is formed again. This could have severe consequences in the remote chance that the connection is lost. In other words, it is always more ideal to run to the wired controls than to wait for the connection to be re-established.

5.1.2 Wi-fi

From the results of the join time tests it is possible to deem Wi-fi too slow for our application. If connection is lost, it could according to the measurements take anywhere between 2.6 s and 7.2 s to join the network again. While 2.6 s is an allowed delay for reestablishing connection, 7.2 s is far to long time. When increasing the range of transmission, the average first decreased only to increase again. This is likely due to random elements in the join process such as when in the joysticks search cycle the Access Point was powered on. Since it the average join time both decreases and increases with distance, it is likely that the average join time is independent of range.

Further study of the join time table, table 4.8, shows a tendency that suggests less difference between highest and lowest join time for increased range. While the join process is not affected by range, the received power is. Since the transmitter is sending at maximum RF-power at a close distance, it might reach the non-linear region of the receivers amplifiers. This could then theoretically distort the signal and increasing the bit-error-rate resulting in retries which may increase the join time slightly. This could also explain the increased minimum join time, since the receiver scans the frequency spectrum for a signal, and highly non-linear signal with high power has a wider spectrum than a linear signal with low power.

5.1.3 868 MHz XBee

For the XBee histogram in figure 4.3, almost all join times are between 7 s and 8 s. Thus the 868 MHz XBee is very stable although being somewhat slow. Unfortunately, 7 s is too long time the 868 MHz XBee to be used as a standalone solution, but its stability makes it a good backup system.

5.1.4 2.4 GHz XBee

The 2.4 GHz XBee showed promising results in the join time tests since it had an average below one second for all measured distances, according to figure 4.4. Even though the maximum join time was above one second, and some cases higher than 2.5 s, it was not often. As the histogram suggests, in more than half of the cases the join time will stay below 250 ms. Furthermore, only 1/10th of the times will the join time be as high as 2.5 seconds; which is still better than all the competing technologies. There are also almost equally many times when the system will reconnect in less than 100 ms.

5.2 Range

The required range, which was specified in the introduction, was set to 30 m. For line-of-sight communication, all of the tested standards passed this. Not a single one of them had problem transmitting over 100 m. Not only does this mean that the connection will reach far but it also makes it possible to reach all of the necessary places on the boat from one access point.

Studying of the heatmaps in figures 4.1 and 4.2 shows a high RSSI value even on the edges of the boat. Therefore the antenna for the access point may be placed almost anywhere on the vessel, except for inside a grounded box, and the system would still be operational. It is however still preferred to place this antenna in an open area, such as on the roof or inside the helm.

It is important to remember that the antennas used were dipoles with a gain of 2.2 dBi. Therefore this high range makes it possible to use smaller and less optimal antennas and still get a good signal across the entire boat.

5.3 Latency

For safe use, it was stated that the latency had to remain better than 100 ms. While all of the tested technologies except SmartMesh WirelessHART passed this, they were not equal.

5.3.1 SmartMesh IP

When used in backbone mode, the SmartMesh IP technology reaches very low latencies. Latency spikes were also detected at greater distances, most likely as a result of retransmissions. This is not a problem for the wireless systems as long as these latency peaks are few and far in between. It is also very likely that with better and more optimized software these peaks may be removed and the latency kept low.

5.3.2 Wi-Fi

Wi-Fi fared very well concerning latency, and typically stayed around 20-30 ms in average. In this regard, Wi-Fi is a solid option for this application.

5.3.3 XBee

Just like the other tested 2.4 GHz technologies, 2.4 GHz XBee showed good performance in terms of latency while the 868 MHz XBee did not perform as well, but still yielded a latency below 100 ms. Therefore the low frequency XBee is not as good in this regard as the high frequency one. Even though latency is one of the less important parameters, it is still desirable to have as low latency as possible.

The 868 MHz XBee could not transmit fast enough for the a refresh-rate of 50 Hz.

Meaning that when this technology is used, the system would have to be slowed down to adapt to its speed. Why the link behaves in this way is unclear, but some theories have been developed: The 868 MHz band has a demand from the EU which states it must have less than <10% duty cycle for each hour of operation. Therefore it is possible that the bitrate becomes higher than the devices are allowed to handle. Too high bitrate should not present itself in this manner unless the XBees were programmed to handle it like this. Another and more likely theory is that the units have too low sampling rate and therefore causes inter-symbol-interference (ISI) for high transmission-rates. This ISI in turn could trigger the built-in collision-avoidance which then delays the transmission.

5.4 Comparison

It is possible to do a complete comparison and reach a conclusion about the optimal technology for this application. The first and easiest parameter to compare is the range of the technologies. While these were different in a line of sight environment, all of them had a greater range than needed. That being said, the XBee devices are more likely to still meet the demands with a more compact antenna, with less gain, than Wi-fi and Smartmesh IP. The heatmaps from the measurements taken on the boat with the XBee technologies further strengthens this argument since the RSSI value remains very high on all parts of the boat.

Next is the security of the networks, which is one of the most important aspects. While all of the devices are prone to jamming, their encryptions are very strong. The best technologies when it comes to security are the Smartmesh technologies since they do a hardware encryption and therefore never have to openly send the key. This combined with 128-bit encryption, DSSS and frequency hopping makes the Smartmesh technologies extremely hard to hack. Wi-fi may have a higher encryption, but the encryption key is slightly more possible to get a hold of. Worst of the safe systems are XBee, which as stated is possible to hack using the right tools. If an in-house solution is made where the encryption is made on hardware level, all of these technologies would be extremely secure, the only threat then would be jamming.

Cost is always a relevant parameter both for the customer and for the producer. The most expensive technology is SmartMesh WirelessHART, followed by SmartMesh IP. This is due to them being complete system-on-chip (SoC) devices with a lot of functionality. Next is the Wi-fi technology since this is a newer and higher bit-rate technology. This means that the least expensive technologies are the two XBee modules.

The bit rate proved to be irrelevant to the project for all technologies except the 868 MHz XBee. This is due to the very small amount of data which needed to be transferred. Still, even the 868 MHz Xbee may have barely reached the necessary bit rate for this project, but with little room for expansion. If the system were to be expanded or the refresh rate improved in the future, the 868 MHz Xbee would not suffice.

Latency is one of the most determining factors for the system since it varies more with technology than the previously mentioned parameters. Highest performing in this category was SmartMesh IP which had a relatively steady latency of around 20 ms. This was then followed by 2.4 GHz XBee which stayed at around 26 ms. Slightly slower was Wi-fi which had an average latency of 33 ms. Second to last were the 868 MHz XBee modules, which showed an average latency of 78 ms. Dead last was the SmartMesh WirelessHART at around 110 ms. In this regard, the 868 MHz module is on the limit of being too slow, while SmartMesh WirelessHART is too slow for this application.

Finally we have join time, this is the determining parameter for this project. The system must be able to rejoin if it loses connection, and it must be done relatively fast. Worst in this category was SmartMesh WirelessHART, which had an average join time of above 30 s. The other SmartMesh protocol was much better at this with an average join time of around 5.4 seconds on the best settings. However, both of these values are too high for this application since the wireless network should re-establish a connection in less time than it takes to reach the wired controls. For Wi-Fi the join time was either the same or lower than that of SmartMesh IP, but it was still too long at its lowest average of 3.4 s. The 868 MHz XBee also yielded poor results in these tests, with a join time of around 7.5 s. This means that the best and only technology to show promising results in this regard was the 2.4 GHz XBee which had a join time of below 250 ms. It still sometimes reached 2.5 s but in most cases the join time was really fast.

5.4.1 Security & reliability

The security of the network is measured in its ability to withstand external attack as well as avoiding internal failures. Since privacy is not really a concern one does not need to care if someone is able to intercept and read messages. Although it is really important to be able to authenticate all messages to make sure they come from the correct source and not from some malicious hacker spoofing the joystick or another boat. This can be achieved in multiple ways with multiple levels of security. But to make it as secure as possible it is recommended to use a HMAC algorithm to encrypt and sign each packet.

Adding an extra layer of HMAC encryption could lead to increased latency. However, since HMAC is basically a combination of a hashing and encryption algorithm the lag is very implementation specific. The algorithm can be optimized to have negligible latency.

The second thing to consider would be the systems ability to recover from internal failures. These types of failure includes everything that can cause the joystick to lose communication with the access point. If two transceivers are used on two different frequencies it will be really hard to jam both of them. Also the fact that they are on 2 different networks means that it would require that both networks fail at the same time.

6

Conclusion

The optimal solution would be a system which can have low join time, low latency, good penetration, high enough bit rate, be durable, and resistant to jamming. None of the mentioned technologies can fulfill all of these roles at the same time. But a combination of them can. The best combination is to use two XBee systems, one operating on the 2.4 GHz band and one on the 868 MHz band. The 2.4 GHz XBee have a fast join time, low latency and high bit rate, while the 868 MHz XBee has good penetration and durability. By using this combination it would be difficult to jam and the system would be less susceptible to other wireless devices due to its use of two frequency bands.

It is recommended to also consider Bluetooth 5 when it is available on the market, as it has promising specifications.

7
Alternative Solutions

After reaching the conclusion about the optimal solution, it may be worth to describe alternative solutions and why they are not considered optimal.

7.1 SmartMesh IP & 2.4 GHz XBee

Due to its low latency, it would make sense to use SmartMesh IP in combination with the 2.4 GHz XBee. The first downside to this solution is the interference between the devices and the low resistance to jamming. During normal operation, this solution would have better performance while running on the backup system. In the end this improvement for the back-up system would not be worth the extra cost and the higher interference susceptibility when compared to using for example the 868 MHz XBee.

7.2 SmartMesh IP & 868 MHz XBee

This solution would be an alternative as long as it does not loose connection. Due to the long join time of SmartMesh IP, the system would be running on the slower 868 MHz XBee for longer, which increases the odds of both systems failing at the same time. The solution would still be viable and have the benefits of a dual-band system but would not be as good as the final one. It would also be more difficult to develop due to the lack of available documentation concerning SmartMesh IP.

7.3 Other technologies than SmartMesh IP & XBee

Smartmesh IP outshined SmartMesh WirelessHART in most aspects, meaning it does not make sense to use SmartMesh WirelessHART for this application when the two technologies are so similar. Wi-Fi proved to be inadequate for this project, since its only advantages over the other technologies was its high bit rate and good encryption. For latency, price and join time Wi-fi was worse than XBee while security was better than XBee and almost as good as SmartMesh IP. XBee is however deemed safe enough, especially with the existence of hardware encryption chips. Many Wi-Fi modules are also known to have high power consumption.

8

Bibliography

[1] Gordon Kelly, "802.11ac vs 802.11n - What's the difference between the Wi-Fi standards?" 2015, accessed: 2017-01-30. [Online]. Available: http://www.trustedreviews.com/opinions/802-11ac-vs-802-11n-what-s-the-difference

[2] Q. Incorporated, "Ieee802.11ac: The next evolution of wi-fi standards," San Diego, USA, 2012, accessed: 2017-05-23. [Online]. Available: https://www.google.se/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwj4lPq044XUAhVLkSwKHYMkDV4QFggqMAA&url=https%3A%2F%2Fwww.qualcomm.com%2Fmedia%2Fdocuments%2Ffiles%2Fieee802-11ac-the-next-evolution-of-wi-fi.pdf&usg=AFQjCNEV1yg8HNx2uyIm3r74QNXXr7bjcg&sig2=W04wwY_fLHAIy_tdAXifMQ

[3] Thinktube, "Why Wi-Fi Direct can not replace Ad-hoc mode," accessed: 2017-01-30. [Online]. Available: http://www.thinktube.com/tech/android/wifi-direct

[4] *ZigBee Specification*, ZigBee Standards Organization, San Ramon, California, January 2008, accessed: 2017-04-05. [Online]. Available: https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2011/kjb79_ajm232/pmeter/ZigBee%20Specification.pdf

[5] *XBee/XBee-PRO RF Modules*, Digi International Inc., Minnetonka, Minnesota, USA, September 2009, accessed: 2017-04-05. [Online]. Available: https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf

[6] *XBee-PRO 868 RF Modules*, Digi International Inc., Minnetonka, Minnesota, USA, February 2011, accessed: 2017-04-05. [Online]. Available: https://elmicro.com/files/digi/xbee-pro-868-manual.pdf

[7] D. M. Pozar, *Microwave and RF wireless systems.* New York: Wiley, 2001.

[8] D. Chen, M. Nixon, A. K. Mok, and S. (e-book collection), *WirelessHART: real-time mesh network for industrial automation*, 1st ed. New York, London:

Springer, 2010.

[9] M.Nixon, "A comparison of wirelesshart and isa100.11a," Emerson Process Management, Texas, USA, Tech. Rep., 2012, accessed: 2017-01-21. [Online]. Available: http://www2.emersonprocess.com/siteadmincenter/PM%20Central%20Web%20Documents/wirelesshart-vs-isa-WP.pdf

[10] W. T., D. L., S. J., and P. K., "Technical overview of smartmesh ip," in *Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2013, accessed: 2017-04-05. [Online]. Available: http://ieeexplore.ieee.org/document/6603731/

[11] T. Watteyne, L. Doherty, J. Simon, and K. Pister, "Technical overview of smartmesh ip," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, 2013, pp. 547–551, accessed: 2017-01-30. [Online]. Available: http://ieeexplore.ieee.org.proxy.lib.chalmers.se/stamp/stamp.jsp?arnumber=6603731

[12] P. Yi, A. Iwayemi, and C. Zhou, "Frequency agility in a zigbee network for smart grid application," in *Innovative Smart Grid Technologies (ISGT)*. IEEE, 2010, pp. 1–6.

[13] G. Litovsky, "A look into bluetooth v4.2 for low energy products," May 2015, accessed: 2017-05-22. [Online]. Available: https://www.google.se/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwir4djmxoXUAhVJ1ywKHXLhALkQFgguMAE&url=http%3A%2F%2Fwww.edn.com%2FPdf%2FViewPdf%3FcontentItemId%3D4439356&usg=AFQjCNESHnJBoaPmsIzHxWs0fZMQBYTLog&cad=rja

[14] M. Zanchi, "Bluetooth low energy," Sunnyvale, USA, 2016, accessed: 2017-01-26. [Online]. Available: http://www.pmeasure.com/en/home/assets/bluetooth-low-energy_whitepaper.pdf

[15] B. S. Inc, "Bluetooth core specification5.0 faq," Kirkland, USA, 2016, accessed: 2017-01-26. [Online]. Available: https://www.bluetooth.com/specifications/adopted-specifications

[16] L. Labs, "Symphony link a revolutionary wireless system for wide-area iot networks," accessed: 2017-05-23. [Online]. Available: http://www.link-labs.com/symphony

[17] T. Parker, "Wi-fi preps for 900 mhz with 802.11ah," September 2013, accessed: 2017-05-23. [Online]. Available: http://www.fiercewireless.com/tech/wi-fi-preps-for-900-mhz-802-11ah

[18] F. Rezha and S. Shin, "Performance analysis of isa100.11a under interference

form an ieee 802.11b wireless network," *IEEE Transactions on industrial informatics*, vol. 10, no. 2, pp. 919–927, 2014, accessed: 2017-02-15. [Online]. Available: http://ieeexplore.ieee.org/document/6746084/

[19] decision-making confidence, "How to use the pugh matrix," accessed: 2017-05-23. [Online]. Available: http://www.decision-making-confidence.com/pugh-matrix.html

[20] L. T. Corp, "Smartmesh ip application notes," accessed: 2017-02-15. [Online]. Available: http://cds.linear.com/docs/en/application-note/SmartMesh_IP_Application_Notes.pdf

[21] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos, "Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization," in *2011 International Conference on Localization and GNSS, ICL-GNSS 2011*, 2011.

[22] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," pp. 1–15, 1996.

[23] H. Krawczyk, R. Canetti, and M. Bellare, "Hmac: Keyed-hashing for message authentication," 1997, accessed: 2017-04-26. [Online]. Available: https://www.ietf.org/rfc/rfc2104.txt

[24] M. Bellare, R. Canetti, and H. Krawczyk, "Message authentication using hash functions: The hmac construction," *RSA Laboratories' CryptoBytes*, vol. 2, no. 1, pp. 12–15, 1996.

[25] N. Hunn, B. (e-book collection), and I. Books24x7, *Essentials of short-range wireless*, 1st ed. Cambridge;New York;: Cambridge University Press, 2010.

A

Source code: Tests

These are the functions used to perform the tests mentioned in Section 3. The initial setup and the way the functions are called differ depending on what technology is being used.

A.1 Join time

To measure the join time the time between power up and the first message has to be measured. This function in called in the Arduinos' loop function and will run 1000 times before printing the results in a comma separated values (CSV) format, then finally blocking with an endless loop.

The wireless module is powered from pin 23 and is continuously toggled by the code while checking for messages.

```
String csv = ""; // The results string
int i = 0;       // counter
unsigned long startTime; // The timer
void JoinTime()
{

  // While we've less than 1000 measurements
  if (i < 1000 && i >= 0)
  {
    startTime = millis(); // Start timer
    digitalWrite(23, HIGH); // Power on


    while (!Serial1.available());    // Wait for first message


    res = millis() - startTime;      // Stop timer.

    // Put comma after each value except last one
    if (i == 999)
      csv += String(res);
    else
```

```
      csv += String(res) + ",";

    Serial.print('#');
    Serial.print(i++);
    Serial.print(" ");
    Serial.println(res);

  } else if (i >= 1000) {
    // Print results
    Serial.println(csv);
    i = -1;
  } else if (i == -1) {
    // Turn off module, flush serial input buffer, then block
    Serial.println("Done!");
    digitalWrite(23, LOW);
    while (Serial1.available())
      Serial1.read();
    while (true);
  }

  // After each measurement; power off, wait until discharged, flush input buffer.
  digitalWrite(23, LOW);
  delay(5000);
  flushSerial1();
}
```

Listing A.1: "Join time test function"

A.2 Latency

To test the latency, 100 packets of 1 byte is sent from the joystick, then the access point answers with 1 byte. The time between sent and received package is measures. Then the results are averaged over the amount of successful measurements. Finally the average round trip time is divided by 2 manually to get the latency of the network.

```


void rttTest()
{
  int results[100];
  int i;
  unsigned long sum = 0;
  int drops = 0;
  memset(results, 0, sizeof(results));

  Serial.println("Starting rtt test...");

  Serial1.setTimeout(5000);
  int maxVal = 0;
  for (i = 0; i < 100; i++)
  {
    Serial.print("Sending packet #");
    Serial.println(i);

    results[i] = rtt();
    if (results[i] == -1) {
      drops++;
      Serial.println("Packet dropped");
    } else {
      sum += results[i];
      maxVal = max(results[i], maxVal);
      Serial.print("RTT: ");
      Serial.println(results[i]);
    }
  }


  Serial.print("Average rtt is ");
  Serial.print(sum / (100 - drops));
  Serial.println("ms");
  Serial.print("Max RTT: ");
  Serial.println(maxVal);
  Serial.print(drops);
  Serial.println(" packets were dropped.");

  while (true);
}





unsigned long tim1; // timer
int rtt()
{
  while(!Serial1.availableForWrite());

  tim1 = millis();
  int res = 0;

  Serial1.write('s');

  if (Serial1.find('r'))
  {
    res = (millis() - tim1);
  } else {
    res = -1;
  }

  return res;

}
```

Listing A.2: "Latency test function"

B

FMEA

The FMEA, seen in table B.2, is a results of a safety analysis made on the system. It shows what types of errors can occur, how severe the error is, how often it occurs and how easy the fault is to detect. The severity, occurance rate, and detectability is rated with a number between 1 and 5. The ratings are explained in table B.1.

Table B.1: The severity, occurrence and detection ratings explained.

Rating	Severity (Sev)	Occurrence (Occ)	Detection (Det)
1	No injuries may be caused, but general safety is affected by this failure	Failure occurrence is very unlikely	Certain detection of the failure
2	Light injuries may be caused by this failure	Relatively few failures occur	High chance of detecting this failure
3	Medium injuries may be caused by this failure	Occasional failure occurrence	Medium chance of detecting this failure
4	Severe injuries may be caused by this failure	Frequent failure occurrence	Low chance of detecting this failure
5	Fatal injuries may be caused by this failure	Persistent failure occurrence	Failure cannot be detected

Table B.2: Table showing the FMEA.

FMEA No.:	Item	Function	Failure Mode	Failure Cause	Failure Effect		Sev	Severity Reasoning	Occ	Occ Reasoning	Failure Detection	Det	Det Reasoning	Risk	Failure Handling	
					Local	Global									Vehicle	Crew
1	Joystick	Send joystick state to access point	Sends same value	Software bug	The microcontroller won't sample the joystick and keep sending the same value	The vehicle will receive bogus joystick values and act upon them.	4	The wireless control system is meant for docking only and therefore even if it crashed it will not be fatal.	1	Preventive methods in place on the access point. Watchdog timer placed on joystick	A continuous counter value is sent with each message and if 5 messages in a row display a faulty value the boat will go into neutral.	1	If the counter is not constantly increasing the vehicle will know something is wrong and assume connection has been lost.	4	If the vehicle is equipped with a Digital Positioning System (DPS) it will be activated, otherwise the vehicle will go into neutral and sound an alarm. The watchdog timer will reset the joystick if it freezes.	The crew will have to move to a wired control station until wireless signal is regained.
2	Joystick	Send joystick state to access point	Loses connection	Interference	N/A	The vehicle will receive no joystick values. Driver will have no control over the vehicle with the wireless controller	4	The wireless control system is meant for docking only and therefore even if it crashed it will not be fatal.	2	Preventive methods in place on the access point. By using acknowledgements one can determine packet loss rate and have an indicator for that on the joystick	A high RSSI with 100% packet loss usually means too much interference	2	If no messages arrive for more than 2 seconds but the RSSI value still is high one can assume that messages didn't arrive because of interference.	16	If the vehicle is equipped with a DPS it will be activated, otherwise the vehicle will go into neutral and sound an alarm.	The crew will have to move to a wired control station until wireless signal is regained.

Table B.3: Continuation of the table showing the FMEA.

FMEA No.:	Item	Function	Failure Mode	Failure Cause	Failure Effect		Sev	Severity Reasoning	Occ	Occ Reasoning	Failure Detection	Det	Det Reasoning	Risk	Failure Handling	
					Local	Global									Vehicle	Crew
3	Joystick	Send joystick state to access point	Loses connection	Loses power source	Joystick will not function	The vehicle will receive no joystick values. Driver will have no control over the vehicle with the wireless controller	4	The wireless control system is meant for docking only and therefore even if it crashed it will not be fatal.	2	Preventive methods in place on the access point. Battery indicator placed on joystick	A low RSSI value and no packets usually means the joystick is dead. The battery level will be reported to the vehicle periodically.	1	The battery level will be reported periodically. If the joystick loses it's power source for an unknown reason one can detect a low RSSI value and no packets.	8	If the vehicle is equipped with a DPS it will be activated, otherwise the vehicle will go into neutral and sound an alarm.	The crew will have to move to a wired control station until wireless signal is regained.
4	Access Point	Act as a bridge between wireless and CAN interface	Receives wrong value	Interference, software bug, network security compromised	The AP will assume the values are legitimate and continue sending them to the HCU (Helm Control Unit)	The vehicle will receive bogus joystick values and act upon them.	5	Could potentially set the vehicle in a harmful state and crash.	1	For this to happen multiple faults has to happen at once. First of all the self healing capabillities of the XBee protocol has to fail. Then someone must either send some interfering signal on both the 868MHz and the 2.4 GHz frequencies, or one could inject packets if the hacker gets a hold of the encryption key.	All data is sent over both 868MHz and 2.4 GHz frequencies.	1	If the data differs when it arrives at the access point one can assume the network is compromised.	5	If the vehicle is equipped with a DPS it will be activated, otherwise the vehicle will go into neutral and sound an alarm.	The crew will have to move to a wired control station until wireless signal is regained.