Norwegian University of
Science and Technology

Communication system for an autonomous passenger ferry

Andreas Kaldestad Fjellhaugen

Master of Science in Electronics
Submission date: June 2018
Supervisor: Egil Eide, IES

Norwegian University of Science and Technology
Department of Electronic Systems

Abstract

NTNU is currently in the development process of an autonomous passenger ferry which will carry pedestrians and bicyclists across the channel between Ravnkloa and Vestre Kanalhavn in Trondheim harbour. To reduce the risk of potential collisions between the ferry and other vessels trafficking the channel, a wireless communication system should be developed. This master's thesis is about the development of such a wireless communication system. The system is going to broadcast navigational information gathered from the ferry such as position, course, and speed, at rapid intervals of one second with the use of two VHF radios, more specifically two microcontroller units of the type *EFR32™ Flex Gecko 169 MHz* from Silicon Labs. The system gathers navigational information from an on-board computer on the ferry by a USB data interface. This was resolved by using the built-in serial communication protocol UART in the microcontroller unit. When the navigational information is correctly acquired from the computer and made accessible on the VHF transmitter, the information is packed into a suitable message format with a similar structure to messages found in the NMEA 0183 communication protocol. The information is transmitted according to frequency regulation §8.7 found in the *regulation of general permissions for use of frequencies (The free use regulation)*, on 169.4 MHz VHF band. The transmitted signal is GFSK modulated with a 14 kHz deviation and a 36.8 kbps bitrate. When the transmitted message is received at the VHF receiver, it is processed and reformatted on to an appropriate NMEA 0183 message format. Received information is then transmitted over a new data interface which also uses the built-in serial communication protocol UART, to a chartplotter.

Samandrag

(Abstract in Norwegian)

NTNU er i gong med utvikling av ein prototype for ei autonom passasjerferje som skal frakte fotgjengarar og syklistar over kanalen mellom Ravnkloa og Vestre Kanalhavn i Trondheim hamn. For å redusera risiko for eventuelle kollisjonar mellom ferja og andre fartøy som trafikkerer kanalen skal eit trådlaust kommunikasjonssystem utviklast. Denne masteroppgåva handlar om utviklinga av dette trådlause kommunikasjonssystemet. Systemet skal kringkaste navigasjonsinformasjon til ferja som posisjon, kurs, og fart ved hyppige intervall på eit sekund ved å bruke to VHF radioar, nærmare bestemt to mikrokontrollarar av typen *EFR32^{TM} Flex Gecko 169 MHz* frå Silicon Labs. Systemet samlar navigasjonsinformasjon frå ein datamaskin om bord på ferja ved hjelp av eit USB-data grensesnitt. Dette vart løyst ved bruk av den innebygde serielle kommunikasjonsprotokollen i mikrokontrollaren, UART. Når navigasjonsinformasjonen er henta riktig inn frå datamaskinen og gjort tilgjengeleg på VHF-sendaren, blir informasjonen pakket inn i eit passande meldingsformat med ein liknande struktur til meldingar som finnes i NMEA 0183. kommunikasjonsprotokollen. Informasjonen vert overført i samsvar med frekvensregulering §8.7 som finnes i reguleringa av generelle tillatingar av frekvensar (fribruksforskrifta), på 169.4 MHz VHF-band. Det overførte signalet brukar ein GFSK-modulasjon med ein 14 kHz avvik og ein bitrate på 36,8 kbps. Når den overførte meldinga er motteken av VHF-mottakaren, blir den behandla og formatert på eit passande NMEA 0183-meldingsformat. Den mottatte informasjonen vert deretter overført over eit nytt datagrensesnitt som også brukar den innebygde serielle kommunikasjonsprotokollen UART, til ein kartplottar.

Preface

This master's thesis in "Signal Processing and Communication" with the course code TTT4920, carried out in the period between January and June 2018 concludes my study in the master program Electronics (program code MIEL). The master's thesis is a continuation of a specialisation project with the same name which was conducted during the autumn of 2017. This master's thesis is part of the research project Autonomous passenger and bicycle ferry between Ravnkloa and Vestre Kanalhavn. The research project looks at the possibility of developing and building an autonomous ferry that can carry passengers and cyclists across the canal between Ravnkloa and Vestre Kanalhavn. Being involved in the research project with the work done in this master's thesis has been exciting, challenging and educational. The master's thesis has been supervised by Associate Professor Egil Eide from the Department of Electronics and Telecommunications and executed and documented by the student Andreas Kaldestad Fjellhaugen.

I hope you will enjoy reading this thesis.

Trondheim, June 2018

Andreas Kaldestad Fjellhaugen

Andreas Kaldestad Fjellhaugen

Acknowledgement

I would first like to thank my supervisor Egil Eide for all the discussion, guidance and help he has provided me within this thesis. I would also like to thank fellow students to give me support or help me when I had questions, received input or help from them.

Contents

Glossary

American Standard Code for Information Interchange American Standard Code for Information Interchange (ASCII) is a character enconding standard used in electronic communication. 30

bit error rate The number of bit errors per unit time. 12

C A fast powerful general-purpose programming language. 39

endianness What sequential order bytes are arranged into larger numerical values when either stored in memory or transmitted over data links. 19

I²C Inter-Integrated Circuit (I^2C) is a multi-master, multi-slave synchronous serial computer buss, widely used in microcontrollers . 40

intersymbol interference A form of distortion in signals when symbols interferes with subsequent symbols. 14, 15

IrDA The Infrared Data Association (IrDA) is an interest group that provides communication protocols for wireless infrared communication. 40

latitude Geographic coordinate that specifies the north-south position of a point on Earth's surface. Ranges from 0° to 90° (North and South) [1]. 45, 46, 49

longitude Geographic coordinate that specifies the west-east position of a point on Earth's surface. Ranges from 0° to +180° eastward, and -180° westward [1]. 45, 46, 49

MATLAB MATLAB (**mat**rix **lab**oratory) is a powerful powerful numerical computing environment. 44

most significant bit The bit position in a binary number with the greatest value. 42

NMEA 0183 Communication protocol for maritime instruments [1]. i, iii, ix, 2, 30, 31, 34, 35, 42, 43, 55, 78

protocol A set of rules that determine how to communicate between one or more devices. 5, 6, 7

Python A high-level object-oriented, general-purpose interpreted, interactive, programming language.. 61, 62, 94

RS-485 RS-485, also known as TIA/EIA-485 is a serial multipoint communication standard with high transmission speed and over long distances. 40

RS-232 Recommended Standard 232 (RS-232) is a serial point-to-point communication standard commonly used in computer serial ports. 40

Acronyms

$\frac{Eb}{N0}$ energy per bit to noise power spectral density ratio. 11, 21, 52, 82

$\frac{Pr}{N0}$ received signal power to noise power spectral density. 21

I^2C Inter-Integrated Circuit. 40

4-FSK 4-Frequency Shift Keying. 10, 11, 12, 13, 37, 65, 77, 82

AGC Automatic Gain Control. 42

AIS Automatic Identification System. ix, 1, 28, 29, 30, 42, 47

ASCII American Standard Code for Information Interchange. 19, 30, 31, 45, 64

ASK Amplitude Shift Keying. 10, 12, 37

AWGN Additive white Gaussian noise. 52

BER Bit error rate. 11, 12, 21, 52, 77, 82

BFSK Binary Frequency Shift Keying. 10, 11, 12, 36, 46

bps bits per second. 18, 47, 51

CR carriage return. 31, 43

CRC Cyclic redundancy check. 42, 43, 46, 50

EHF Extra high frequency. 20

EIRP effective isotropic radiated power. 22, 23, 27, 28, 70, 71

ERP effective radiated power. 23, 27, 28, 51, 68, 70

ETSI European Telecommunications Standards Institute. 86

Free use regulation Regulation of general permissions for use of frequencies. 26, 85

FSK Frequency Shift Keying. 11, 12, 13, 46, 65, 77

FSL Free Space Loss. 23, 54, 55, 75

FSPL Free Space Path Loss. 23, 24, 47, 52

GFSK Gaussian Filtered Frequency Shift Keying. i, 36, 39, 68, 77, 82, 83

GMSK Gaussian Filtered Minimum Shift Keying. 29

GPS Global Positioning System. 57

GSM Global System for Mobile Communications. 20

HF High frequency. 20

HTTP Hypertext Transfer Protocol. 6

IDE integrated development environment. 39

IEEE Institute of Electrical and Electronics Engineers. 21

IMO The International Maritime Organization. 1, 28, 29

ISI Intersymbol interference. 14

ISO International Standards Organization. 6

LCD Liquid crystal display. 72

LF Low frequency. 20

LF line feed. 31, 43

LoS line-of-sight. 23

MCU microcontroller unit. 18, 34, 35, 39, 40, 42, 52, 55, 56, 58, 60, 61, 62, 66, 67, 70, 71, 72, 73, 74, 76, 78, 79, 82, 92, 94, 95

MF Medium frequency. 20

MFSK M-ary Frequency Shift Keying. 12

MMSI Maritime Mobile Service Identity. 1, 45, 46, 50

MPRP Maximum permitted radiated power. 27, 47, 51, 68, 70, 71

MSB Most significant bit. 42, 44

MSK Minimum Shift Keying. 11, 12, 13, 37, 46, 65, 67, 68, 77, 82

NKOM Nasjonal kommunikasjonsmyndighet. 70

NTNU Norwegian University of Science and Technology. 1

OBC On board computer. x, 35, 61

OBW Maximum occupied bandwidth. 27, 47, 51, 65, 68, 71, 77, 82

OOK On-Off Keying. 10, 11, 12, 13, 37, 65, 77, 82

OQPSK Offset Quadrature Phase Shift Keying. 11, 12, 13, 37, 65, 67, 77, 82

OSI model ISO OSI (Open Systems Interconnect) Reference Model. 6

PEL Plane Earth Loss. 24, 54, 55, 74, 75

PER Packet error rate. 2, 66, 67, 68, 72, 77

QPSK Quadrature Phase Shift Keying. 11, 12

RAM Random access memory. 36

RF radio frequency. 2, 7, 9, 56, 63, 67, 77, 82

RMC Recommended Minimum Navigation Information. 57, 78, 79, 80, 83

ROS Robot Operating System. 83

RSSI Received Signal Strength Indicaton. x, 2, 21, 63, 67, 71, 72, 73, 74, 75, 76, 78, 90

SHF Super high frequency. 20

SNR Signal-to-noise ratio. 11, 25, 53

SoC System On Chip. 34, 35, 39, 40

SOLAS Safety of Life At Sea. 1, 28

UART Universal asynchronous receiver transmitter. i, iii, 40, 41, 42, 55, 56, 58, 60, 61, 62, 64, 78, 79, 82, 83, 92

UHF Ultra high frequency. 20

USART Universal synchronous and asynchronous receiver transmitter. 36, 40

USB Universal Serial Bus. 2, 34, 35, 38, 40, 60, 62

VHF Very high frequency. i, iii, x, 1, 2, 20, 29, 34, 35, 37, 39, 41, 42, 50, 51, 55, 56, 60, 61, 62, 63, 78, 79, 80, 82, 83

VLF Very low frequency. 20

VSWR Voltage standing wave ratio. 38

Nomenclature

Physics Constants

c	Speed of light in a vacuum inertial system	$299,792,458\,m/s$
r	Earth's radius	$6371000\,m$
f	Frequency	Hz
K	Kelvin	$K =^\circ C + 273.15$
B	Bandwidth	Hz
λ	Wavelength	m
k	Boltzmann's constant	$1.38 \cdot 10^{-23}\,J \cdot K^{-1}$

Radio Communication

P_T	Transmit power	dBm
P_R	Received power	dBm
P_{TI}	Effective isotropic transmit power	dBm
P_{RI}	Effective isotropic received power	dBm
G_T	Transmitter antenna gain	dBi
G_R	Receiver antenna gain	dBi
L	Maximum acceptable path loss	dB
L_{FSL}	Free space path loss	dB
L_{PEL}	Plane earth path loss	dB
F	Noise figure	dB

P_N	Noise power	$J \cdot K \cdot Hz$
G/T	Receiver figure of merit	dB/K
T_0	Ambient room temperature	K
T_a	Antenna noise temperature	K
T_e	Equivalent noise temperature	K
d_f	Far-field distance	m
d'	Distance direct path	m
d''	Distance ground reflected path	m
h_t	Transmitter height	m
h_r	Receiver height	m

Modulation

T_s	Symbol period	m/s
α	Roll of factor	ranges between 0 - 1
$h_{eff}(t)$	Nyquist criterion impulse response	
$H_{RC}(f)$	Raised Cosine Filter transfer function	
$h_{RC}(t)$	Raised Cosine Filter impulse response	
$h_G(f)$	Gaussian Lowpass Filter transfer function	
$h_G(t)$	Gaussian Lowpass Filter impulse response	

Other Symbols

kn	Knot, a unit of speed	1 kn = 0.514444 m/s

Chapter 1

Introduction

Norwegian University of Science and Technology (NTNU) is currently in development of a prototype for an autonomous passenger ferry which will operate across the channels in Trondheim harbour. The harbour in Trondheim is a narrow congested area with heavy traffic, and without any crew on-board, the ferry passengers safety is a vital issue.

The ferry will transport pedestrians and bicyclists across the harbour following an on-demand concept, which will reduce the waiting time and create more frequent departures. The on-demand concept can be achieved by either using a mobile application or a call button on either side of the channel if the ferry is underway without any passengers. The ferry will have an electrical propulsion system with battery packs for running on-board systems and driving the propulsion. The recharging of the battery packs will be done while the ferry is by the dock while either idling or awaiting passengers to board.

1.1 Background

To further reduce the risk of a collision with nearby vessels the ferry will broadcast its navigational information to nearby vessels. Information such as position, heading, speed, time, status and Maritime Mobile Service Identity (MMSI)-number will be sent at an interval less or equal to 1 second with the use of a Very high frequency (VHF) radio or similar. Traditionally and also legislated by the Safety of Life At Sea (SOLAS) convention held by The International Maritime Organization (IMO), this is done by using the Automatic Identification System (AIS) which broadcasts any vessels navigational information to surrounding vessels to avoid and reduce the risk of collision.

However because of the requirement on transmit interval the AIS system is not sufficient enough since the shortest transmit interval AIS can provide is 2 second.

1.2 Problem Description

For achieving the desired broadcasting interval the licence-free 169.4 MHz frequency will be used to broadcast short messages from the ferry with navigational information. The VHF link has been successfully tested and verified in a specialisation project [2]. This master's thesis will, therefore, contain the following tasks:

1.3 Objectives

- Implement a Universal Serial Bus (USB) data interface from the computer on-board the ferry to the VHF transmitter.
- Design a compact message format that allows data to be transferred according to the frequency regulations for the 169.4 VHF band.
- Design and implement a data interface from the VHF receiver to a chartplotter using NMEA 0183 format.

1.4 Approach

This master's thesis is part a larger research project developing an autonomous passenger and bicycle ferry between Ravnkloa and Vestre Kanalhavn. The approach of this master's thesis has been as a development project, with solutions proposed based on theory and analysis of existing work. The solutions are implemented and a conclusion is drawn regarding whether each of the objectives listed previous section have been met.

1.5 Contributions

In the measurements of the Packet error rate (PER) and Received Signal Strength Indicaton (RSSI), a standalone test application provided by Silicon Labs named Range Test, is implemented. The application creates a RF link between two radio kits and sends predefined amounts of packets with configurable size and modulation from the transmitter to the receiver.

1.6 Limitations

Knowledge of the architecture of the ferry's computer system is limited. Testing of the implemented data interface between the on-board computer and VHF transmitter have therefore only used fixed data, oppose to actual navigational information gathered from various sources in the system. Measurements of RF link coverage is done with antennas supplied with the radio board, and due to lack of technical information about the supplied antennas the result may be misleading.

1.7 Thesis Outline

Chapter 2 - Theoretical Background provides the theoretical background used in this thesis. The main subjects are communication systems, serial communication, radio communication, and message formats.
Chapter 3 - Design and Implementation contains an analysis and a design proposal for each of the objectives, and also an analysis of the whole communication system. Hardware and tools used in the thesis also discussed.
Chapter 4 - Result and Discussion sets up different tests and measurement for each objective, results are presented and discussed for each of the thesis objectives.
Chapter 5 - Conclusion gives a conclusion of the thesis, and a recommendation for future work is presented.
Appendix A - Project Description contains project description in English and Norwegian.
Appendix B - Measurement data contains measurement data.
Appendix C - Serial Communication Appendix contains code and configuration used for the serial communication.

Chapter 2

Theoretical Background

This chapter intends to form a ground layer of useful theory and background that will be needed in the thesis later.

2.1 Communication system

2.1.1 Principle of Communication

Communication is essential for our lives, both analogue- and digital communication. People daily use a considerable amount of different communication systems to have a certain degree of coordination in their daily lives. Smartphones, computers, radio and television is just a handful of communication tools that utilise several different communication systems, all with the intent of sharing information. The word *communication* derives from the Latin word *commūnicāre*, meaning *to share*. The complexity of communication systems varies with the amount and complexity of tasks the system have to perform. A simplified explanation of how a communication system is organised can be done by dividing the system into five parts, as illustrated in Figure 2.1. This is the five-part classical architecture of a communication system that was originally described in *A Mathematical Theory of Communication*, by Claude Shannon while working at Bell Labs [3].

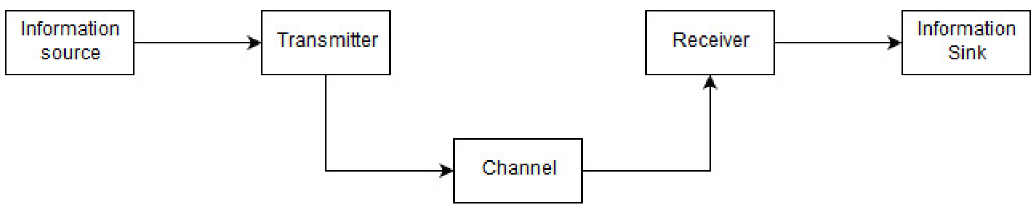


Figure 2.1: Elements of a communication system [4]

The first block in the block diagram is a *source* of information, the *source* could be a varity of different information such as music, pictures, data or video. When a source of information is selected, that is, what information the developer wish to send, the information must pass on in a *transmitter*. The transmitter is responsible for processing the information from the source in such way that it is ready for transmitting over the channel [4].

After the processing is done, the information is transmitted through a *channel* or transmission medium. The channel could be a cable, optical fibre or free space. Once the information flows through the transmission medium, a *receiver* is required for the information to make any sense. A receiver often has more tasks than just processing the received signal back to understandable information. Other tasks can be, for example, to synchronise with the transmitter or run error detection and correction. Once the receiver has finalised processing the received signal, the information is sent to the destination, which is called *sink* [4].

2.1.2 Communication Protocol

To reduce the risk of misunderstanding information exchanged between the transmitter and the receiver, a protocol for the communication is defined. A *protocol* is a set of rules for the communication defined prior to the actual start of the communication with the objective of getting both the transmitter and the receiver to "speak the same language". Without a predefined protocol for the communication, it would be impossible for the receiver to understand the received information from the transmitter. A protocol often consists of several different layers to reduce the design complexity. Names, numbers of layers, and each of the layers task differ from protocol to protocol. Each layer must be able to interact with the layer above and the layer below, as the layers communicate down in the layer stack at the transmitter and up in the layer stack at the receiver [5].

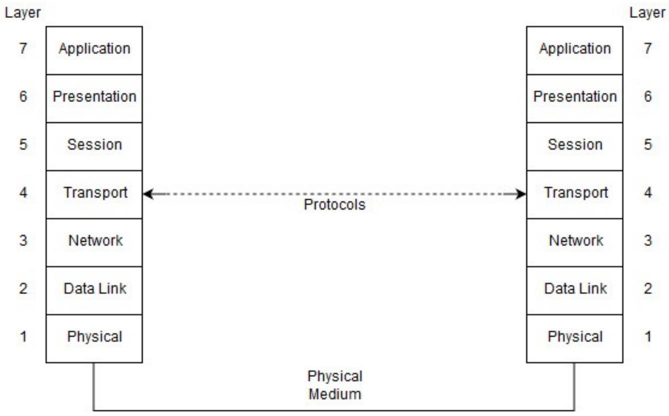


Figure 2.2: Seven layer OSI model for computer communication [4]

Communication protocols are often illustrated by the ISO OSI (Open Systems Interconnect) Reference Model (OSI model), which is a model based on a proposal made by the International Standards Organization (ISO) in a step towards making an international standardisation of protocols. This model has seven layers as illustrated in Figure 2.2, and each layer has different protocols inside [5].

The seven layers of the OSI model are a result of several factors. These factors and the goal of the OSI model can be summarised as [5]:

- Creating a layer when a different abstraction is needed.
- Having a well-defined function performed by each layer.
- Having a goal of defining internationally standardised protocols for the functions in each layer.
- Choosing the layer boundaries that minimise the information flow across the interfaces.
- Having the number of layers large enough to avoid having distinct functions in the same layer, and so small that the complexity of the protocol is manageable.

Some examples of different protocols that could be defined within each of the seven layers are found in Table 2.1, below.

Table 2.1: Layers and protocols for the OSI model

Layer	Protocols within this layer define:
Application (7)	How data is presented, etc Hypertext Transfer Protocol (HTTP).
Presentation (6)	Compression, decompression, data encryption, decryption.
Session (5)	How a session should be created, maintained and ended.
Transport (4)	Numbering of packets, sorting packets at receiver Request resending of missing packets.
Network (3)	External addressing, most effective routing.
Data Link (2)	How bits from the physical layer should form frames, local addressing, basic detection and correction.
Physical (1)	The physical link: such as plugs, contacts, cables. In short: how to transmit and receive raw bits successfully.

2.1.3 Wireless Communication

A wireless communication system follows the architecture that was presented in Chapter 2.1.1, but it requires some additional components inside some of the blocks illustrated in Figure 2.1. The basic components for the *transmitter* block that are necessary for a radio transmitter is illustrated in Figure 2.3.

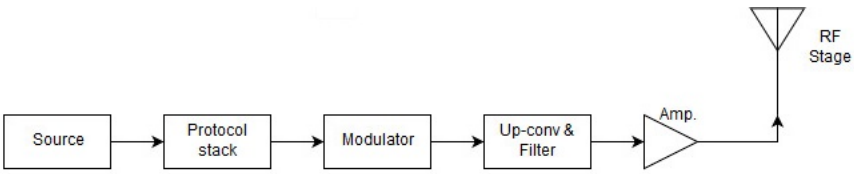


Figure 2.3: Basic radio transmitter [4]

At the start of the transmitter the *source* of information is sent into the *protocol stack* block. In this block, the information is packed according to how a protocol have defined this process in advance. A *protocol stack* might contain many layers and the number of layers is dependant on which protocol is being used. The protocols in the lowest layer always deals with the physical link, such as plugs, contacts, cables and voltages, that is in short how to transmit and receive raw bits successfully. Protocols must be able to interact with protocols in the layer above and the layer below the layer which the protocols is located at, as mentioned in Chapter 2.1.2. When the information is packed and ready it is modulated onto a fitting carrier wave, this is done in the *modulator* block [4].

After the signal have been modulated it is time for the *up-conversion stage*. Here, the signal is converted to a suitable radio frequency (RF) that is well fitted to transmit on. With newer technology, this block could be replaced by a modulator that does this work directly. The final stage before the *channel* is the *RF stage*, where the signal is amplified to a suitable power level and the signal is transmitted through an antenna. The power level of the signal depends on the systems desired transmission range, while the type of antenna depends on the frequency used, and the application [4].

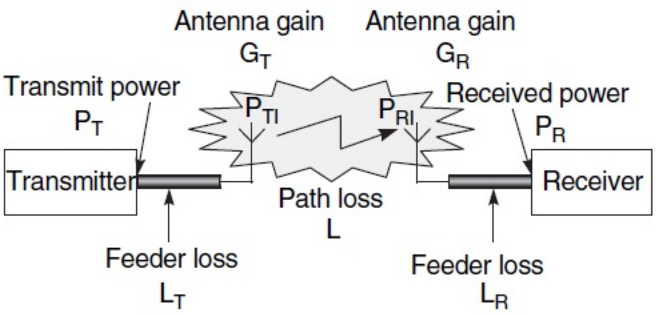


Figure 2.4: Elements of a wireless communication system [6]

For a wireless communication system one of the *channels*, or *transmission medium* is called *free space*. *Free space* transmission as illustrated in Figure 2.4 gives a number of properties that needs to be addressed, these properties are [4]:

- *Propagation loss*: The loss in signal strength as a ratio measurement when comparing the power of the transmitted signal to the power of the received signal between the antenna and the target. This is also called *path loss*.

- *Frequency*: How well different frequencies handles transmission compared to each other.

- *Time-varying*: Some channels have characteristics that vary with time, an example is Mobile radio channels. When either the receiver or the transmitter is moving the channel will experience changes in performance, caused by phenomena called *shadowing* and *fading*.

- *Non-linearity*: When trying to reduce the distortion that the transmitted signal has, the channel should ideally be made linear. But the channel might contain nonlinear components such as a repeater which could have an amplifier. This scenario happens on satellite channels, where the signal sent from earth is amplified by the satellite before being sent back to earth.

- *Interference*: Avoiding interference between users, since channels are often shared with many people that are using different multiplexing schemes to share the channel. An example of this is cellphone users sharing the same wireless channel in frequency and time.

- *Noise*: The unavoidable noise that reduces the performance of the system.

A more in-depth explanation of techniques that are used when designing a wireless communication system to take these properties into account follows in Chapter 2.3. In practice, these properties often occur in variation with each other in some way.

After the information is transferred wireless the *receiver* block receives the signal with the information. In the receiver block, most of the components are inverse functions of their counterparts that are found in the *transmitter*, but a few new components are needed as illustrated in Figure 2.5.

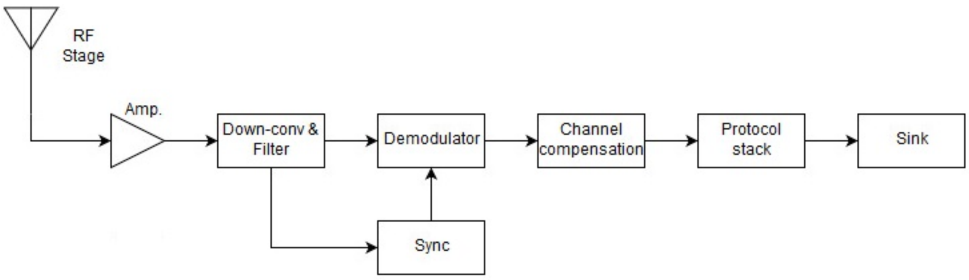


Figure 2.5: Basic radio receiver [4]

First up is the inverse *RF stage* which uses an antenna to collect RF energy at the right frequency. Then a *low-noise amplifier* is used to get the signal power to a level where the processing of the signal can be done without introducing to much noise. The second block, the *down-conversion* & *filter* is also an inverse which converts and filters the signal to a suitable frequency where demodulation of the message becomes easier. With new technology the signal can also go directly to *baseband*, this is called *direct-IQ-down-conversion*. When the signal is at a suitable frequency the demodulation and recovery of the transmitted message is done in the *demodulating* block. This is often a more complex part of the system in order to improve performance in modern receivers, compared to the classical receivers and their sequences of linear filters [4].

The blocks that didn't feature in the transmitter and that are only found in the receiver, are the *synchronisation (sync)* and *channel compensation blocks*. The synchronisation is required because of a difference in the *time* and *frequency clocks* used by the transmitter and receiver. The synchronisation technique depends on what type of modulation and multiplexing that is used. The *channel compensation* tries to reduce any of the channel properties that the signal was subjected for in the transmission. The last block of the receiver is the inverse *protocol stack*, which includes the instructions on how to assemble the message and often a verifying check of the received message [4].

2.1.4 Modulation

The process of encoding information from a source onto a carrier signal that is suitable for transmission is called *modulation*. A general technique is to translate a baseband signal, which is the source, to a bandpass signal that has a very high frequency compared to the baseband frequency. The bandpass signal is called the *modulated* signal while the baseband signal is called the *modulating* signal. A *demodulation* is this process reversed [7].

There are several different techniques when it comes to modulation, both in analogue- and digital modulation. The main difference between analogue and digital lies in the input- and the output signal. In analogue modulation, every value between minimum and maximum are considered legal, while in digital modulation only two values that are used to represent zero and one are considered legal.

Modulation can be done by varying the frequency, the amplitude, or the phase of the signal. The various techniques are called modulation schemes and each scheme comes with its own pros and cons in different situations. While there are lots of various modulation schemes, there are only five modulation schemes that can be used in this project due to the limitations of the hardware, as is further described in the hardware description in Chapter 3.1.2. These five modulations schemes are the following:

- Binary Frequency Shift Keying (BFSK)
- 4-Frequency Shift Keying (4-FSK)

- On-Off Keying (OOK) / Amplitude Shift Keying (ASK)
- Minimum Shift Keying (MSK)
- Offset Quadrature Phase Shift Keying (OQPSK)

When deciding which modulation to use several different factors play in, such as implementation cost, frequency regulation, the environment, and the Signal-to-noise ratio (SNR). Another important term in relation to the SNR in digital communication is the SNR per bit, also called energy per bit to noise power spectral density ratio ($\frac{Eb}{N0}$). When designing a communication system, one usually has a limitation on the number of acceptable bit errors the system can tolerate, called the Bit error rate (BER). Bit errors are bits received that have been altered due to various reasons such as noise, interference, and distortion. For achieving a certain BER the system requires a certain $\frac{Eb}{N0}$ for the various modulation schemes.

In Table 2.2 the bits per symbol and symbol rate for each of the five modulation schemes are listed. The symbol rate is also called *bitrate* or *baud rate* and it is the number of symbols transmitted per second. Bits per symbol is the number of bits used to represent each sent symbol.

Table 2.2: Bits per symbol and symbol rate for useful modulation schemes [7]

Modulation	Bits per symbol	Symbol rate
FSK/BFSK	1	$1 \cdot$ bit rate
OOK	1	$1 \cdot$ bit rate
MSK	1	$1 \cdot$ bit rate
4-FSK	2	$1/2 \cdot$ bit rate
QPSK/OQPSK	2	$1/2 \cdot$ bit rate

Frequency Shift Keying

Frequency Shift Keying (FSK) is a modulation scheme in which the frequency of the carrier signal varies with changes in the digital signal, while the amplitude and the phase of the signal remain the same. The simplest form of FSK is 2-FSK which is also known as Binary FSK (BFSK). In BFSK the carrier frequency is shifted either to a high or a low frequency for transmitting the corresponding binary 0 and binary 1.

M-ary Frequency Shift Keying

M-ary Frequency Shift Keying (M-ary FSK/MSFK) is Frequency Shift Keying with M transmitted signals that all have equal energy and duration. The signals are separated by 1/2 T_s Hz and situated orthogonal to each other. 4-FSK is an example of a M-ary digital modulation, in which there are 4 symbols ($M = 4$ *each with 2 bits of information* $(N = 2 = log_2 M)$), that are transmitting binary 00, 01, 10 and 11.

On-Off Keying

On-Off Keying (OOK) might be the simplest form of Amplitude Shift Keying (ASK) used. In ASK the amplitude of the carrier signal varies with changes in the digital signal, while frequency remains the same. With On-Off Keying one simply reduces the amplitude of the carrier signal to zero when transmitting a binary 0, and increases the amplitude when transmitting a binary 1.

Minimum Shift Keying

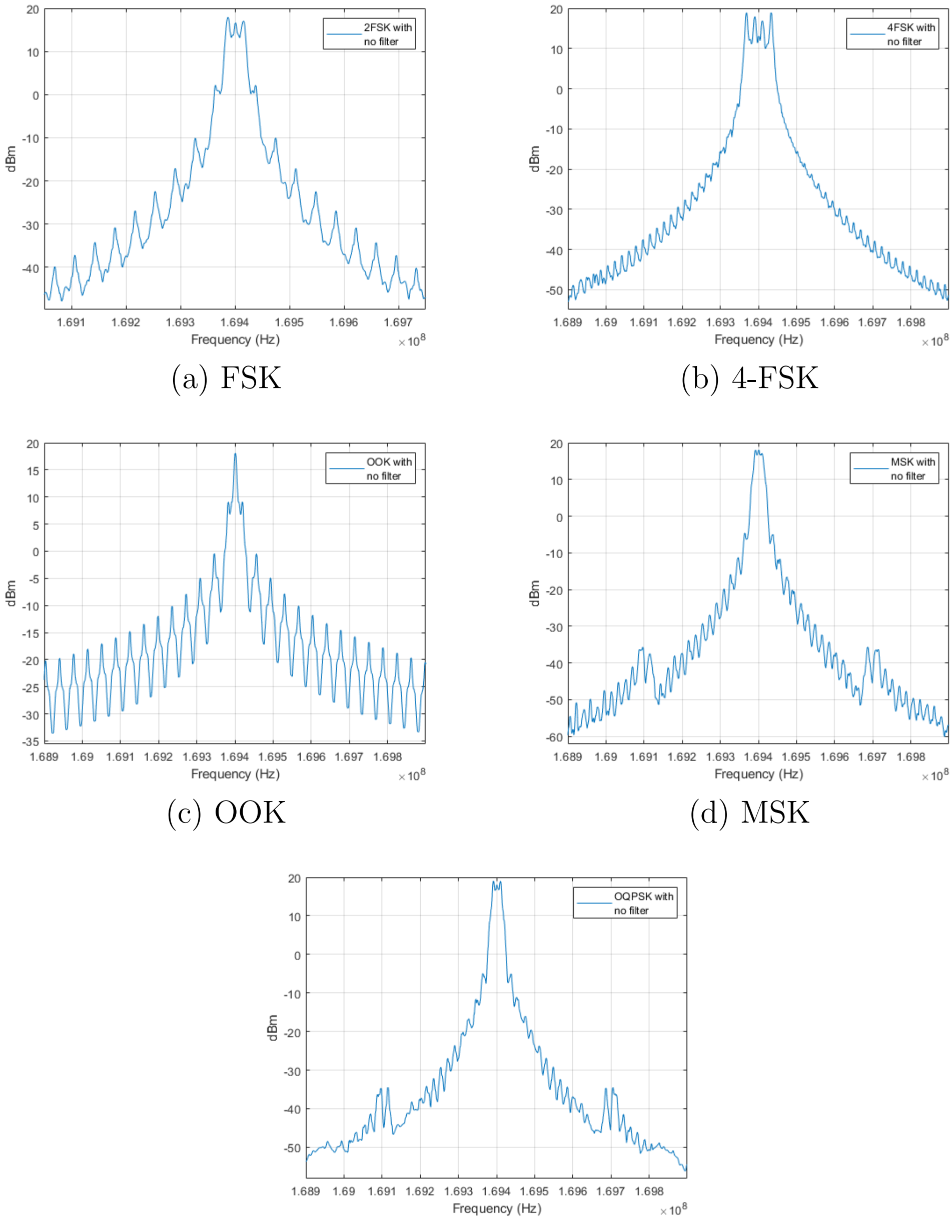
Minimum Shift Keying (MSK) is a special type of continuous-phase shift keying with that have a modulation index of 0.5 and a peak frequency deviation equal to 1/4 of the bit rate. Sometimes called *fast FSK* since it only uses half as much frequency spacing compared to conventional noncoherent FSK. The modulation index gives the lowest frequency spacing that still would give two coherently orthogonal FSK signals, hence the name minimum shift keying. Its a spectrally efficient modulation with good bit error rate (BER) performance, constant envelope and self-synchronising capability often used in mobile radio communication systems.

Quadrature Phase Shift Keying

Quadrature Phase Shift Keying (QPSK) is a modulation scheme in which phase of the carrier is divided into four equally spaced values, e.g. $0°$, $90°$, $180°$ and $270°$ or e.g. 0, $\pi/2$, π and $3\pi/2$. Each of the phase values holds 2 bits of information. Offset QPSK (OQPSK) is a modified version of QPSK that supports more efficient amplification and is less susceptible to side lobes and spectral widening. The OQPSK signal is similar to the QPSK signal, with the exception of the time alignment for the odd and even bits. QPSK also allows the signal to shift the phase by $180°$ each time which leads to undesired large amplitude fluctuations in the signal. OQPSK on the other hand only allows the signal to shift the phase by $90°$ each time.

Spectral Shape

Figure 2.6 shows how the spectrum of the various modulation schemes that have been presented in the previous sections look like when a randomly generated signal is modulated without the use of a pulse shaping filter. The bitrate is 36.8 kbps and the deviation is 14 kHz.



(a) FSK (b) 4-FSK

(c) OOK (d) MSK

(e) OQPSK

Figure 2.6: Spectrum to a random generated signal for various modulations

2.1.5 Pulse Shaping

Whenever a rectangular pulse passes through a band-limited channel the pulses spread in time. Pulses for one symbol might find themselves in succeeding time intervals interfering with the right pulses. This is called intersymbol interference (ISI) which leads to an increase in the probability that the receiver makes an error when detecting a symbol. There are different techniques that could be used to counter the intersymbol interference. One solution is to increase the bandwidth of the channel which reduces the intersymbol interference. However, it might not be desired to increase the bandwidth due to limitations on the bandwidth, e.g. the minimal bandwidth of mobile communication systems. Another solution would be to run the signal through a pulse-shaping filter in order to minimise the intersymbol interference [7]. Two such filters that are used to minimise the intersymbol interference are a *Raised Cosine Filter* and a *Gaussian Pulse-Shaping Filter*.

Raised Cosine Filter

The *Raised Cosine Filter* is the most popular pulse shaping filter used in mobile communications [7]. The filter is a *Nyquist Filter* because it satisfies the *Nyquist criterion*, which describes the conditions that results in no intersymbol interference. The Nyquist criterion impulse response $h_{eff}(t)$ can be mathematically described as:

$$h_{eff}(nT_s) = \begin{cases} K & \text{if } n = 0 \\ 0 & \text{if } n \neq 0 \end{cases} \tag{2.1}$$

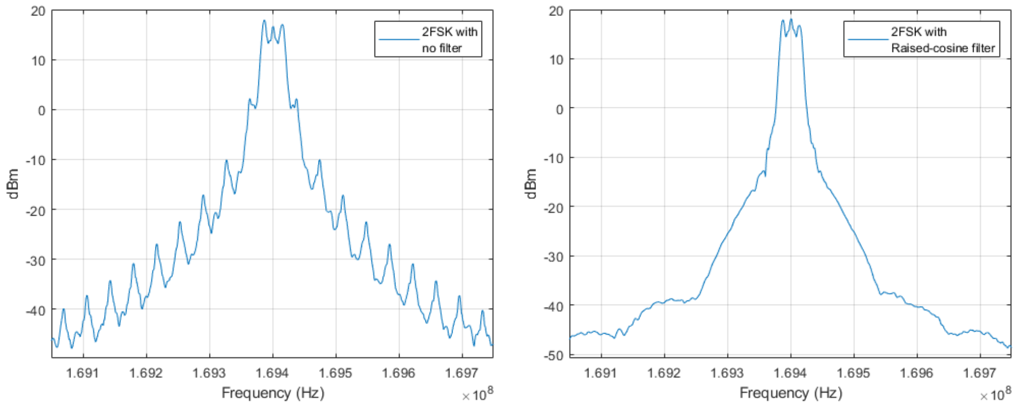
Where T_s is the symbol period, K is a non-zero constant and n is an integer [7].

For the raised cosine filter the transfer function is given by:

$$H_{RC}(f) = \begin{cases} 1 & \text{if } 0 \leq |f| \leq \frac{(1-\alpha)}{2T_s} \\ \frac{1}{2}\left[1 + cos\left[\frac{\pi(|f| \cdot 2T_s - 1 + \alpha}{2\alpha}\right]\right. & \text{if } \frac{(1-\alpha)}{2T_s} \leq |f| \leq \frac{(1+\alpha)}{2T_s} \\ 0 & \text{if } |f| > \frac{(1+\alpha)}{2T_s} \end{cases} \tag{2.2}$$

Where the roll-off factor α is between 0 and 1 and T_s is the symbol period [7]. The impulse response of the filter is found by taking the inverse Fourier transform, as shown in Equation 2.3.

$$h_{RC}(t) = \frac{sin\left(\frac{\pi t}{T_s}\right)}{\pi t} \cdot \frac{sin\left(\frac{\pi \alpha t}{T_s}\right)}{1 - \left(\frac{4\alpha t}{2T_s}\right)^2} \tag{2.3}$$



(a) Without a filter (b) With a raised-cosine filter

Figure 2.7: The spectrum of a 2FSK modulated signal

Gaussian Pulse-Shaping Filter

The *Gaussian Pulse-Shaping Filter* is another filtering technique that unlike the Raised Cosine Filter does not satisfy the Nyquist criterion. It is often used in Minimum Shift Keying due to its efficiency, but also in other modulation schemes that require power efficient nonlinear amplifiers [7]. The transfer function for the Gaussian lowpass filter is given by:

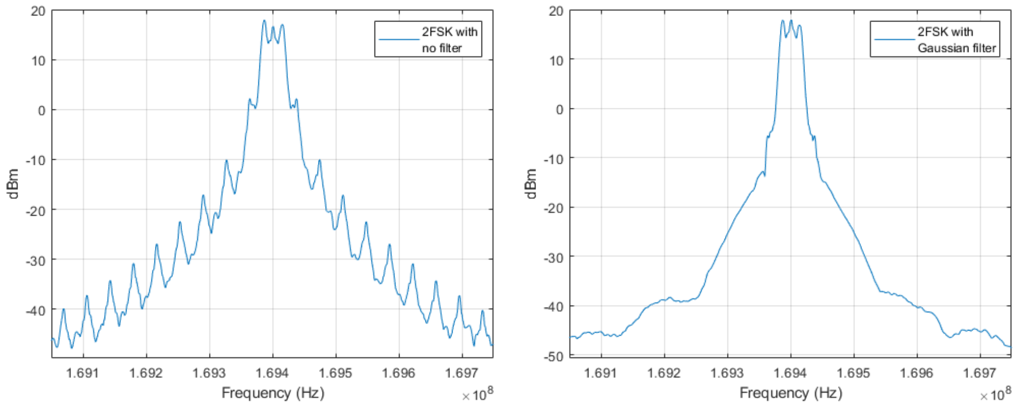
$$H_G(f) = exp(-\alpha^2 f^2) \tag{2.4}$$

Where α is related to the 3-dB bandwidth B as

$$\alpha = \frac{\sqrt{ln2}}{\sqrt{2}B} = \frac{0.5887}{B} \tag{2.5}$$

The impulse response is found by taking the inverse Fourier transform of Equation 2.4.

$$h_G(t) = \frac{\pi}{\alpha} exp\Big(- \frac{\pi^2}{\alpha^2} t^2\Big) \tag{2.6}$$



(a) Without a filter (b) With a Gaussian filter

Figure 2.8: The spectrum of a 2FSK modulated signal

Since the Gaussian-Pulse Shaping Filter does not satisfy the Nyquist criterion, reduction in the use of bandwidth will lead to a degradation in the performance of the system due to an increase in ISI [7].

2.2 Serial Communication

Serial communication is the process for transmitting data sequentially over a communication channel. In serial communication only a single bit is transferred at a time, while parallel communication transfers multiple bits at a time. Serial is therefore much cheaper to implement due to the fact that the amount of hardware and wiring are much smaller than the amount used in parallel. Serial communication is slower than parallel. This is because serial only transfers one bit at a time, while parallel can be built to transfer eight, sixteen, thirty-two at a time and so on. Figure 2.9 illustrates the speed and complexity for serial versus parallel, by visualising the transmission of the bits *01100011*.

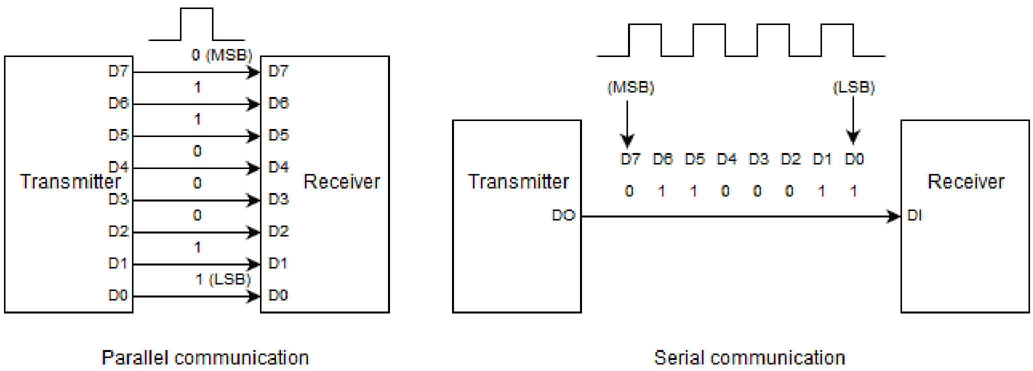


Figure 2.9: Serial vs parallel transmission

There are two operation modes in serial communication called *simplex* and *duplex*. Simplex only allows for transmission in one direction, while duplex allows transmission in both directions. The duplex operation mode can be further divided into *half-duplex* and *full-duplex*, where half-duplex don't allow the transmission to occur simultaneously in both directions, while the full-duplex allows both ends to transmit and receive simultaneously.

Serial communication is either done synchronously or asynchronously and these two groups of serial interfaces are called synchronous mode and asynchronous mode. In synchronous mode, the clock signal is paired with the data line so that all devices share a common clock. In the asynchronous mode, the data is transmitted without the use of a clock signal. Instead, the asynchronous mode transmits on a frame structure at a specified rate. Figure 2.10 shows how this frame structure is composed. Packets transmitted with this serial frame structure are configurable. The serial frame usually consists of data bits, parity bits, and synchronisation bits. The term synchronisation bits are used for the start and stop bits.

Start 1 bit	Data 5-9 bits	Parity 1 bit	Stop 1-2 bits

Figure 2.10: A configurable serial frame

How fast the serial frame is transmitted is specified by the baud rate, which defines how fast data is transmitted in a serial communication, it is measured in bits per second (bps). A commonly used baud rate is 9600 bps, but standard baud rates range from 110 bps to 256000 bps, with a baud rate of 115200 bps usually being the fastest microcontroller units (MCUs) uses. It is crucial that each side of the communication uses the same baud rate, different baud rate on either side will cause the transmitted data to be misinterpreted on the receiving end and render the data useless.

Starting with the data line in an *idle* state, the serial frame always starts with a start bit with value 0 which takes the data line out of idle. Then follows a *chunk* of data where the length is dependant on the character encoding. A popular encoding is ASCII, which have several different lengths. Both sides of the transmission need to have the same length for the data, and also agree to the same endianness of the data. Following the data is a parity bit, which is an optional bit used for error checking. Parity-bit checks whether the sum of the 1-bits or the 0-bits is either odd-parity or even-parity. This is done by keeping track of the number of 1-bits and 0-bits that are sent and depending on whether it looks for odd-parity or even-parity the parity bit results in the desired parity. If the result is the opposite parity then an error has occurred during the transmission. A flaw with this technique is that it can only guarantee detecting an odd number of errors. The serial frame ends with one or two stop bits that transitions the data line back to the idle state [8].

2.3 Radio Communication

The most common method for making a *channel* wireless as described in Chapter 2.1.3, is the use of radio waves. Radio waves are a type of electromagnetic radiation with longer wavelengths than infrared light. Radio waves uses frequencies from around 3 kHz to 300 GHz, and wavelengths from 1 mm to 100 km, as displayed in Figure 2.11.

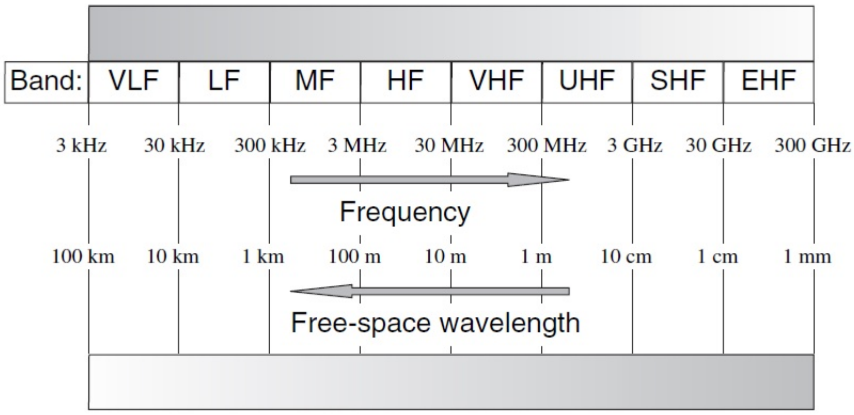


Figure 2.11: The electromagnetic spectrum for radio waves [6]

In recent times, the development of radio communication has sought to use ever higher frequencies. This is due to the tremendous bandwidths that are available at these high frequencies. With these higher frequencies the antennas that are needed for these systems can be really small. A example of this is mobile devices that use the Global System for Mobile Communications (GSM) cellular network which uses frequencies around 900 MHz have a small hidden antenna [9]. A downside with higher frequencies is that the system is more vulnerable to obstructing effects which reduces the range of the system. The impact on the systems range is a reduction in range as the size of obstructing objects increases relative to the wavelength of the system [6].

Table 2.3: Frequency bands [6]

Band name	Frequency range
Very low frequency (VLF)	3 − 30 kHz
Low frequency (LF)	30 − 300 kHz
Medium frequency (MF)	0.3 − 3.0 MHz
High frequency (HF)	3 − 30 MHz
Very high frequency (VHF)	30 − 300 MHz
Ultra high frequency (UHF)	0.3 − 3.0 GHz
Super high frequency (SHF)	3 − 30 GHz
Extra high frequency (EHF)	30 − 300 GHz

2.3.1 Link Budget

A useful tool for finding the required power for a receiver in a communication system is called a *link budget*. A link budget is a summation of all signal powers, losses, and gains that affects the performance of the system. As the calculation is only a summation it is fairly simple to do, but the result reveal very important information about the system performance given that the different elements used in the calculation, such as the different gains or the different losses have been accurately predicted or measured. The difference between the calculated link budget and the receiver's sensitivity is called the *link margin*, which is usually described in dB. A simple equation which describes the relationship between two antennas in free space is the *Friis transmission equation*.

$$\frac{P_R}{P_T} = G_T \cdot G_R \left(\frac{\lambda}{4\pi d} \right)^2 \tag{2.7}$$

Friis transmission equation describes the relationship between two antennas, A and B. Antenna A have gain G_T and transmit power P_T, antenna B have gain G_R and power received P_R, the distance between the two antennas is d and the wavelength of the transmitted signal is λ.

The link margin can also be found by having the required $\frac{Eb}{N0}$ for a certain BER subtracted from the calculated $\frac{Eb}{N0}$ for the system:

$$\text{Link margin } \Delta_{SNR} = \frac{E_b}{N_0} - \frac{E_b}{N_0} req \qquad (dB) \tag{2.8}$$

When the received signal power to noise power spectral density ($\frac{Pr}{N0}$) for the system is found in the link budget calculation, $\frac{Pr}{N0}$ can be related to $\frac{Eb}{N0}$ with the following equation:

$$\frac{P_r}{N_0} = \frac{E_b}{N_0} R_b \tag{2.9}$$

Where R_b is the bitrate. If the equation is rearranged it gives the systems calculated $\frac{Eb}{N0}$ for the link margin:

$$\frac{E_b}{N_0} = \frac{P_r}{N_0} - R_b \qquad (dB) \tag{2.10}$$

Received Signal Strength Indicaton

Received Signal Strength Indicaton (RSSI) is a figure of measurement of the received power at a receiver in telecommunication. Values, length of scale, and units of measurement used to describe the RSSI varies from manufacturer to manufacturer. The Institute of Electrical and Electronics Engineers (IEEE) defines RSSI as an optional parameter with a value of 0 to a RSSI max value. It is intended to be used in a relative manner, and there is no absolute accuracy specified for RSSI reading [10]. Although the unit of measurement varies from manufacturer to manufacturer, the most common unit used to describe RSSI values is in dBm.

2.3.2 Basic Propagation Models

Path Loss

Path Loss is the term used to describe loss due to radio wave propagation between the transmitter and receiver. There exists several propagation models that give a good approximation for path loss, which will be used when the link budget is calculated.

All the parameters that are needed in this section to define equations useful for calculating the path loss are illustrated in Figure 2.4 in Section 2.1.3. In the illustration, P_R is the power received, P_T the transmitted power, L_T is the feeder loss at the transmitter, L_R the feeder loss at the receiver, L is the path loss, G_R and G_T are the antenna gains expressed with the reference to an isotropic antenna. An isotropic antenna is an antenna which radiates power with unit gain uniformly in all directions. All gains G and losses L are expressed as power ratios and powers in watts.

The first equation is an equation for the received power P_R.

$$P_R = \frac{P_T G_T G_R}{L_T L_R L} \tag{2.11}$$

The effective isotropic radiated power (EIRP) is the total power that is radiated by a hypothetical isotropic antenna. This is also often used as a reference to antenna gains in wireless communication systems [7].

$$\text{EIRP} = \frac{P_T G_T}{L_T} = P_{TI} \tag{2.12}$$

Where P_{TI} is the effective isotropic transmitter power, the equation can be turned to find the effective isotropic received power P_{RI}:

$$P_{RI} = \frac{P_R L_R}{G_R} \tag{2.13}$$

Another term often used to denote maximum radiated power is the effective radiated power (ERP). ERP denotes the maximum radiated power compared to a half-wave dipole antenna, whereas EIRP compares to a isotropic antenna. The dipole antenna have a gain of 1.64 (2.15 dB) and the ERP is therefore 2.15 dB smaller than the EIRP of the same system. Equation 2.14 shows the conversion between ERP and EIRP when working with watt or dBm [7].

$$\begin{aligned} &\mathrm{EIRP}(W) = 1.64 \cdot \mathrm{ERP}(W) \\ &\mathrm{EIRP}(dBm) = \mathrm{ERP}(dBm) + 2.15dB \end{aligned} \tag{2.14}$$

The main objective of a propagation model is to predict the loss as accurately as possible so that the system can be optimised before installation. The maximum range of the system is when the received power drops below a- bar set for an acceptable communication quality, this is also called the *receiver sensitivity*. When the power in the system is expressed in terms of EIRP the path loss L can be expressed independently of the system parameters. The path loss is then a ratio between the received and transmitted EIRP, in other words, experienced loss if the system is ideal with zero feeder loss and isotropic antenna gains ($L_{R,T} = 1$ and $G_{R,T} = 1$). This is value for the path loss L for which the received power provides an acceptable communication quality, and it is called the *maximum acceptable path loss*:

$$\begin{aligned} L &= \frac{P_{TI}}{P_{RI}} = \frac{P_T G_T G_R}{P_R L_T L_R} \\ L_{dB} &= 10log\left(\frac{P_T}{P_R}\right) \end{aligned} \tag{2.15}$$

It should be noted that this definition is for an ideal system. Propagation loss may be different due to variations in the shape of the antennas radiating pattern, distance, and terrain between the transmitter and receiver or if the transmitter and receiver swap roles and so on.

Free Space Loss

The *Free Space Loss (FSL)*, also known as Free Space Path Loss (FSPL) is a basic propagation model used to find the attenuation between two antennas that are in line-of-sight (LoS). In *free space* there are no reflection, scattering or diffraction for the path between the transmitter and receiver. By following the definition of loss and rearranging the Friis transmission equation from Section 2.3.1, two equations can be set up for calculating the free space loss L_{FSL}:

$$L_{FSL} = \frac{P_T G_T G_R}{P_R} = \left(\frac{4\pi d}{\lambda}\right)^2 = \left(\frac{4\pi df}{c}\right)^2 \tag{2.16}$$

$$L_{FSL(dB)} = 32.4 + 20 \cdot log_{10}(d) + 20 \cdot log_{10}(f_{MHz}) \tag{2.17}$$

In the first equation, P_T and G_T is the transmit power and antenna gain for the transmitter, and P_R and G_R is the received power and antenna gain for the receiver, d is the distance between the two antennas in meters and λ is the wavelength of the transmitted signal. The second equations expresses the free space loss in decibels, here the distance d is given in kilometres and f is the frequency in megahertz.

The Free Space Path Loss propagation model is only valid for predicting P_R when the value of d are in the far-field of the transmitting antenna, also known as the *Fraunhofer region*. This region is defined by being beyond the far-field distance d_f, which is given by:

$$d_f = \frac{2D^2}{\lambda} \tag{2.18}$$

D is the largest physical linear dimension the antenna have and λ is the wavelength. The conditions $d_f \gg D$ and $d_f \gg \lambda$ have to be true for the distance d being in the far-field region.

Plane Earth Loss / Two-ray ground reflection model

The Plane Earth Loss (PEL), also known as the two-ray ground reflection model is a propagation model based on geometric optics. The model includes the path loss from both the direct path and the ground reflected path between transmitter and receiver. The model have decent accuracy for predicting the path loss for mobile radio systems over a few kilometres that uses tall towers, and also for small line-of-sight systems that operate in urban environments.

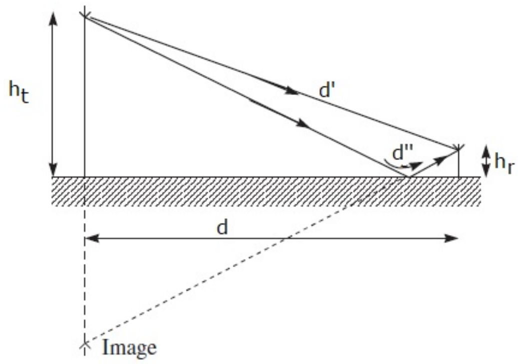


Figure 2.12: Two paths between transmitter and receiver

.

Figure 2.12 shows two paths between a transmitter and a receiver, the direct path d' and the ground reflected path d''. The flat earth distance between the transmitter and the receiver is d, while h_t and h_r are the transmitter and the receivers height over the flat earth [6]. The length of the two paths can easily be described with the use of simple geometry as:

$$\begin{aligned} d' &= \sqrt{(h_t + h_r)^2 + d^2} \\ d'' &= \sqrt{(h_t - h_r)^2 + d^2} \end{aligned} \tag{2.19}$$

The difference $\Delta = d'' - d'$ when $d \gg h_t + h_r$ can be approximated with the use of a Taylor series as

$$\Delta = d'' - d' \approx \frac{2h_t h_r}{d} \tag{2.20}$$

Equation 2.21 expresses the plane earth path loss in decibels without the antenna gains included, and Equation 2.22 expresses the loss in decibel with the antenna gain included.

$$L_{PEL(dB)} = 40log(d) - (20log(h_r) + 20log(h_t)) \tag{2.21}$$

$$L_{PEL(dB)} = 40log(d) - (10log(G_R) + 10log(G_T) + 20log(h_r) + 20log(h_t)) \tag{2.22}$$

Noise Figure

The ability to estimate a Signal-to-noise ratio (SNR) for a mobile communication system is important for determining the systems coverage and quality of service. SNR requirement is proportional to the link quality and the probability of error, and therefore important when deciding a suitable transmit power or receiver levels for various propagation models [7, p. 565]. *Noise figure* is a term used to express the noise at the output of a receiver to an equivalent level at the input of the receiver. It is defined as the ratio of the signal-to-noise ratio at the input of the amplifier to the signal-to-noise ratio at its output:

$$F = \frac{\text{Measured noise power out of device at room temperature}}{\text{Power out of device if device were noiseless}} \tag{2.23}$$

The noise power is found by multiplying Boltzmann's constant $k = 1.38 \cdot 10^{-23}\, J \cdot K^{-1}$ with the ambient room temperature T_0 and the equivalent bandwidth B of the measuring device:

$$P_N = kT_0B \tag{2.24}$$

Noise figure is also related to the *equivalent noise temperature*, T_e,

$$T_e = T_0 \cdot (F - 1) \tag{2.25}$$

where T_0 is the ambient room temperature, usually between 290K to 300K. Noise temperature is measured in Kelvin.

Complete systems can be characterised as a cascade of two-port elements. The equivalent noise temperature of the overall system can then be computed from the different noise temperatures and gains of the systems individual components:

$$T_{eq} = T_1 + \frac{T_2}{G_1} + \frac{T_3}{G_1G_2} + ... \tag{2.26}$$

The receiver Figure-of-merit G/T is an indication of how well the receiver antenna is working together with the receiving electronics to produce a useful signal, the larger G/T is, the better it is. G/T is given as a ratio by dividing the receiver gain G_R on the equivalent noise temperature of the overall system T_{eq}, or in decibel by subtracting:

$$G/T = G_R - T_{eq}(dB) \tag{2.27}$$

2.3.3 Frequency Regulations

The regulation of general permissions for use some frequencies in Norway is determined by *Post- og teletilsynet*. In Norwegian, the regulation is called

Forskrift om generelle tillatelser til bruk av frekvenser (fribruksforskriften), which in English is the *Regulation of general permissions for use of frequencies (Free use regulation).*

The first chapter of the regulation contains preliminary provisions, some that will have to be taken into consideration when deciding on whether a radio link is working within the legal boundaries. The preliminary provisions that are important are primary § 2.1 *Transmission time*, § 2.2 *Maximum occupied bandwidth*, § 2.4 *Equivalent isotropic radiated power (EIRP)* and § 3 *Disturbances*. The two important paragraphs of the first chapter are listed in the next subsections.

§ **2. Definitions[11]**

(1) *Transmission time*: the time a transmitter is active, expressed in per cent of one hour.
(2) *Maximum occupied bandwidth (OBW)*: the bandwidth which contains 99 per cent of the signals average power.
(3) *Channel separation*: distance between centre frequencies in two neighbouring channels.
(4) *Equivalent isotropic radiated power (EIRP)*: the product of the power transmitted to the antenna and the antenna gain in a given direction relative to an isotropic antenna (absolute or isotropic amplification).
(5) *Maximum allowed average EIRP spectral power density*: specified as EIRP per unit bandwidth; maximum average radiated power per unit bandwidth in the direction with the highest power under the conditions under which the measurement is performed.

§ **3. Disturbances[11]**

Frequency usage according to the regulations is not protected against interference from other lawful use of frequencies unless provided by this regulation. Frequency use according to the regulations shall be arranged so as not to reduce the quality of services operated on the basis of individual frequency licenses.

Short-range communications

The third chapter of the legislation, that is §§ 6−10, defines limitations for frequencies intended to be used by equipment for short-range communication.

These frequencies and their boundaries are listed in Table 2.4.

Table 2.4: Short distance communication frequencies [11]

§	Range (MHz)	MPRP	Transmission time	OBW
§7.1	169.400−169.475	500 mW ERP	10% <	50 kHz
§8.7	169.400−169.475	500 mW ERP	1,0%	50 kHz
§8.8	169.400−169.475	10 mW ERP	0.1% <	-
§8.11	433.050−434.790	10 mW ERP	10% <	-
§8.13	433.040−434.790	10 mW ERP	10% <	25 kHz
§8.18	869.400−869.650	500 mW ERP	10%	50 kHz
§17	2327−2390	2 W ERP	-	8 MHz
§21.4	869.650−869.700	25 mW ERP	10% <	25 kHz
§23.1	2400−2483.5	25 mW ERP	-	25 kHz

It should be noted that the legislation uses ERP instead of EIRP when defining power limitations, so Equation 2.14 must be used to convert the ERP values to EIRP values.

2.4 Message formats

Message formats are ways of arranging predetermined messages in a efficient structured so that the size of the message is reduced but not the content.

2.4.1 Automatic Identification System

The Automatic Identification System (AIS) is a maritime surveillance system introduced by the The International Maritime Organization (IMO) with the purpose of increasing the safety at sea for vessels and the environment and achieve an improvement in regulation of traffic and monitoring of traffic. The Safety of Life At Sea (SOLAS) convention held by the International Maritime Organization resulted in a requirement that every passenger vessel independent of size shall be fitted with AIS which must be operational at all times as long as there are no international agreements or rules that imply otherwise [12]. In principle, the requirement for having AIS can be summarised to apply [1]:

- All ships that are sailing on international voyages with size of 300 gross tonnage and upwards.

- All cargo ships that are not sailing on international voyages with size of 500 gross tonnage and upwards.
- All passenger ships, regardless of size.
- All fishing vessels longer than 45 m or of 300 gross tonnages and upwards.

The performance standards for AIS that were adopted in 1998 says that AIS shall be able to perform the following criteria [13]:

- AIS shall provide navigational information, such as the ships identity, type, position, speed, course, status and other relevant information automatically to other properly equipped ships, shore stations and aircraft.
- AIS shall similarly receive information from other properly equipped ships, shore stations and aircraft.
- Use this information to monitor and track ships, and exchange data with shore-based facilities.

Principle and Classes

There are defined five different AIS classes, each specified to allow communication amongst the other classes. These are the following classes [1]:

- *Class A*: for ships that comply with IMO requirements.
- *Class B*: for pleasure craft and other vessels that do not meet the IMO requirements.
- *Shore Stations*: shore-based station on land providing navigational information.
- *Aids-to-Navigation*: shore-based or mobile station providing location and status of an aid to navigation.
- *SAR aircraft*: Search and rescue aircraft.

AIS uses a VHF transmitter and broadcasts navigational messages on two channels located on the Marine VHF. The first channel, 87B have the frequency 161.975 MHz and is called AIS-1. The second channel, 88B have the frequency 162.025 MHz and is called AIS-2. AIS uses the modulation scheme Gaussian Filtered Minimum Shift Keying (GMSK) with a frequency modulation index of 0.5 and BT-product with maximum of 0.4 for the modulator and 0.5 for the demodulator. The transmission bit rate is 9600 bps ± 50 ppm and the bandwidth is 25 kHz [13].

The reporting interval for the different users of AIS depends on the ship class and the ship's dynamic conditions. The quickest interval is 2 seconds for some dynamic conditions for ship in Class A, while the slowest interval are 3 minutes. The nominal reporting interval tends to be higher if the ship are Class A and moving [13][1].

The content of an AIS message varies from message to message and there are currently 27 defined messages and 35 undefined that are reserved for future use. In Table 2.5 a default transmission packet including data fields and number of bits is listed.

Table 2.5: Default AIS Transmission Packet [13]

Ramp up	8 bits
Training sequence	24 bits
Start flag	8 bits
Data	168 bits
CRC	16 bits
End flag	8 bits
Buffer	24 bits
Total	256 bits

2.4.2 NMEA 0183

NMEA 0183 is a communication protocol for maritime instruments centred around the data-link layer developed by The National Marine Association, an ideal organisation for producers, distributors, retailers, educational institutions, and other people or organisations with an interest in maritime electronic. The communication protocol have a wide arrange of standardised messages that contain different information. Common for all the different variants is that they follow the message structure listed in Table 2.7, and usually have the serial configuration listed in Table 2.6 [1].

The structures of NMEA 0183 messages are a number of American Standard Code for Information Interchange (ASCII) characters which are set up as follows:

Table 2.6: NMEA 0183 standard serial configuration [1].

Baud rate	4800
Data bits	8
Stop bits	1 (or more)
Parity	None
Handshake	None

$TTSSS,Data_field1,Data_field2,...,*CS<CR><LF>

The first character of the message is a *"$"* if the message is a delimited conventional field message, and if the message is a special message the first character is a *"!"*. Following the start of the messages is two fields used to identify the talker, that is instrument type, and the type of message. The talker is identified by using two characters (TT), while the type of message is identified by using three characters (SSS). The next part of the structure is the data fields in the message, all of the data fields are separated by commas. Following the data fields is a checksum identifier * if the message contains a checksum. The checksum is represented by a two-digit hexadecimal number, and calculating the checksum is done by using a bitwise XOR for all the ASCII characters between the start of the message ($ or "!") and the checksum identifier (*). The bitwise XOR compares two and two bits and returns *"0"* if the two bits are the same and *"1"* they are different. The checksum is optional for some of the types of messages, and mandatory for others. The end of a NMEA 0183 message is the two control characters *<CR><LF>* carriage return (CR) and line feed (LF) [1].

Table 2.7: NMEA 0183 message structure [1].

ASCII character	Number of bits
$	8
TT	16
SSS	24
,	8
Data field	-
*	8
CS	16
$< CR >< LF >$	16
Sum min. required bits	96 bits

Chapter 3

Design and Implementation

This chapter is broken into several sections. The first section contains a detailed analysis of the communication system, hardware and tools intended to be used in the development of the communication system for the ferry. Then follows separate sections for each of the objectives for this master thesis, these were:

- Implement a USB data interface from the computer on-board the ferry to the VHF transmitter.
- Design a compact message format that allows data to be transferred according to the frequency regulations for the VHF band.
- Design and implement a data interface from the VHF receiver to a chartplotter using NMEA 0183 format.

Each of these sections contains an analysis of the objective, existing work, and a proposed design implementation with any improvement proposal. Results and discussion of the proposed design and implementation proposed in this chapter will follow in Chapter 4.

3.1 System Overview

The specialisation project that was conducted last autumn proposed a solution to the VHF link using two Silicon Labs proprietary wireless microcontroller units (MCU). The two same MCUs are intended to be used in this project, the EFR32$^{\mathrm{TM}}$ Flex Gecko Proprietary Wireless System On Chip (SoC). However, the specialisation project only verified that the VHF link was able to transmit and receive a constant predetermined packet with the same size as the proposed message format. The system also exceeded the limitations in the frequency regulation which were presented in Section 2.3.3.

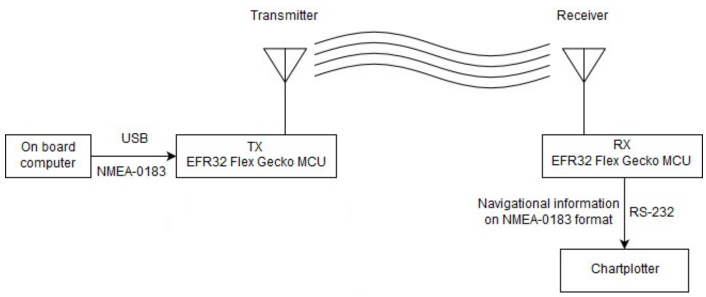


Figure 3.1: A simple block diagram of the communication system

To make the communication system operational for the autonomous passenger ferry, the VHF link have to send and receive packets containing real navigational information and avoid exceeding the frequency regulation. The navigational information will have to be gathered from the On board computer (OBC) and be made available for the transmitter MCU.

The navigational information has to be extracted from a source, in this case, an on-board computer, and transferred to the VHF transmitter via USB. The transmitter MCU then have to process the received information from the on-board computer and make it available for transmission. The navigational information has to be packed in a suitable message format so that the transmission is a success and follows the frequency regulations. This will be dependant on the result found in the analysis of the message format in Chapter 3.3. At the receiver, the received information will be processed and sent to a chartplotter using the NMEA 0183 format. As stated both the transmitter and receiver will be EFR32^{TM} Flex Gecko Proprietary Wireless SoC.

3.1.1 Hardware and Software

In any development process, the developer's available hardware and tools set limitations for the development process. When developing a communication system, the hardware for which the system will be built upon is a huge factor. It might be impossible to achieve the desired performance and configurations with a chosen hardware. Energy consumption, memory, operating temperature, and radio kit configurations are all factors that have to be weight up towards each other in order to find the right balance that works as intended for the desired system.

3.1.2 Hardware

EFR32^{TM} Flex Gecko 169 MHz 2.4 GHz and Sub-GHz Proprietary Wireless Kit

The EFR32^{TM} Flex Gecko is a wireless SoC developed by Silicon Labs. The Wireless Kit includes a 2400/169 MHz 19.5 dBm radio board that can be plugged directly into the motherboard, as visualised in Figure 3.2. A short list of key features for the microcontroller can be found in Table 3.1.

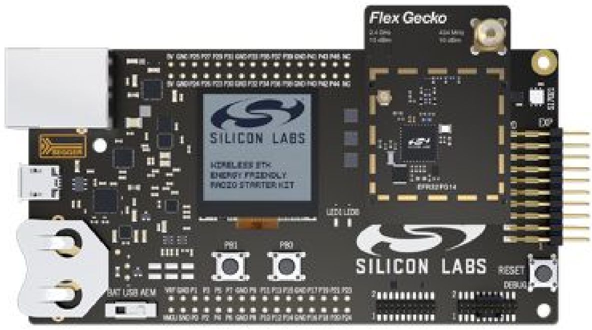


Figure 3.2: EFR32^{TM} Flex Gecko 169 MHz 2.4 GHz and Sub-GHz Proprietary Wireless Kit [14]

Table 3.1: Key features of the microcontroller [15]

256 kB flash program memory
32 kB RAM data memory
2.4 GHz and Sub-GHz radio operation
112.2 dBm sensitivity at 38.4 kbps, GFSK, 169 MHz
Low Energy Consumption
-8.4 mA RX current at 38.4 kbps, GFSK, 169 MHz
Wide Operating Range
1.8 V to 3.8 V single power supply
Temperature range from $-40°$ C to $+85°$ C
Supports USART and I^2C

The microcontroller supports several different modulation schemes combined with different pulse shaping filters. They are the following:

- Binary Frequency Shift Keying (BFSK)
- 4-Frequency Shift Keying (4-FSK)
- On-Off Keying (OOK) / Amplitude Shift Keying (ASK)
- Minimum Shift Keying (MSK)
- Offset Quadrature Phase Shift Keying (OQPSK)

All of the modulation schemes can be used with two different pulse shaping filters, the Gaussian Pulse-Shaping Filter and Raised Cosine Filter. Theory for the modulation formats is found in Chapter 2.1.4, while theory for the filters are found in Chapter 2.1.5.

Scan Antenna VHF73X

The antenna to be mounted on the ferry is named *VHF73X* and is produced by the Danish company SCAN Antenna A/S [16]. It is a 3 dB VHF antenna which is 1.44 m long which can also be customised to a customer specified frequency. It has a end-fed full 1/2 λ dipole antenna design, and a full omnidirectional radiation pattern, as shown in Figure 3.3. The antenna is made out of fibreglass, polytetrafluoroethylene (PTFE) (Teflon), polyethylene (PE) (plastic), copper, and chrome. The antenna also does not require a ground plane. Some of the antennas specifications are found in Table 3.2, while the radiation pattern of the antenna is plotted in Figure 3.3.

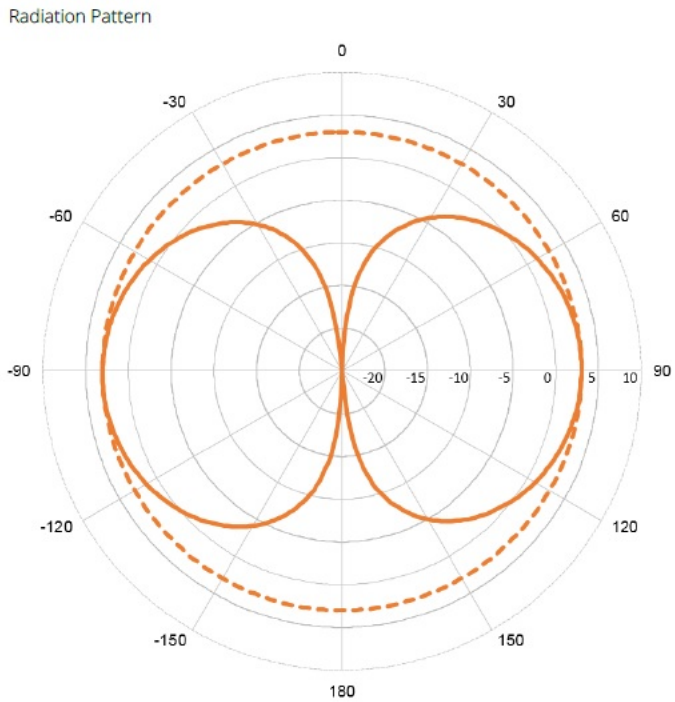


Figure 3.3: The radiation pattern of the SCAN Antenna VHF73X [16]

Table 3.2: SCAN Antenna VHF73X specifications [16].

Frequency	118 − 174 MHz (Specified by customer)
Bandwidth	Approx. 12 % of the specified center frequency
Impedance	50 ohm
Voltage standing wave ratio (VSWR)	<1.5
Polarisation	Vertical
Gain	2.1 dBi
Max. Input Power	100 W
Lightning Protection	Direct ground
Operating temperature	−55°C to +70°C
Survival Wind Speed	55 m/s (125 mph)

R&S®FSV Signal and Spectrum Analyser

The FSV Signal and Spectrum Analyser from Rohde & Schwarz is a useful tool when designing RF systems. The instrument has a frequency range up to 7 GHz. A number of different measurements can be done using this tool such as looking at the occupied bandwidth, transmission time, and the shape of the spectrum. Two female USB ports are located at the front panel, here a memory stick for extracting data can be attached and a mouse if one wishes to avoid using the touchscreen. The FSV Signal and Spectrum Analyser is the one used later in this report whenever it is stated that a signal and spectrum analyser are used to measure.

3.1.3 Software

Simplicity Studio™

Simplicity Studio™ is a integrated development environment (IDE) made by Silicon Labs as a development tool for Silicon Labs' EFM32, EFM8, 8051 MCUs, wireless MCUs, and ZigBee SoCs. The development language used in Simplicity Studio is C, a fast general-purpose programming language. Simplicity Studio contains a large number of software examples that demonstrate a variety of different applications, from simple transmits to range testing. All of the software examples comes with a built-in radio configuration, with the vast majority being configured to use 2-GFSK with different frequencies, bit rate, and deviation. The built-in radio configurations are also configurable for the user, both in the software examples and in projects that are created from scratch. Some of the receiver sensitives that are measured from these built-in radio configurations are listed in Table 3.3.

Table 3.3: Measured receiver sensitivity from built-in radio configurations [15]

Sensitivity	Reference signal
−124 dBm	2.4 kbps 2-GFSK , BT=0.5, $\Delta f = 1.2$ kHz, RX channel BW = 5.05 kHz, channel spacing = 12.5 kHz
−111.9 dBm	38.4 kbps 2-GFSK , BT=0.5, $\Delta f = 20$ kHz RX channel BW = 84.16 kHz, channel spacing = 100 kHz
−97.7 dBm	500 kbps 2-GFSK , BT=0.5, $\Delta f = 125$ kHz RX channel BW = 841.6 kHz

Among some of the extra tools available in Simplicity Studio is a tool for monitoring the energy consumption of the MCU, named *Energy Profiler*. The Energy Profiler and enables user monitoring and plotting of the energy consumption of the MCU in real-time. Another vital tool of development is the *Flash Programmer* which is a tool for flashing and erasing code running on a MCU. There is also a built-in serial console called *Device Console*.

3.2 On-board Computer to VHF Transmitter

3.2.1 Objective

The first objective of this thesis is gathering navigational information and other data from the on-board computer of the ferry and making it available to the VHF transmitter.

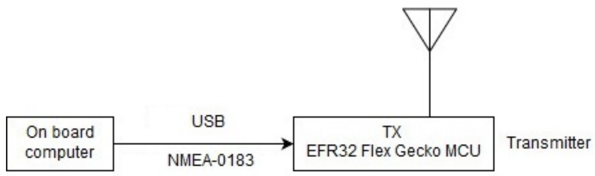


Figure 3.4: On-board computer to VHF transmitter block diagram

As visualised in Figure 3.4 the transmitting MCU and the on-board computer will be connected to each other through a USB cable.

Existing Work

The Flex Gecko MCU supports two different serial communication protocols, Universal synchronous and asynchronous receiver transmitter (USART) and Inter-Integrated Circuit (I^2C). The USART protocol also includes a Universal asynchronous receiver transmitter (UART) protocol. A big difference between UART and I^2C is the number of devices included in a communication link. UART is suitable for communication between two devices while the I^2C can have a large number of devices that are either working as masters or slaves. I^2C is a type of synchronous communication while UART is a asynchronous communication type, this means that there is no clock signal included in the UART communication compared to the I^2C communication. UART is not suitable for communication over long distances due to the voltage used. Since the distance between the on-board computer and the MCU is very short and there is only two devices communicating between each other, the UART communication is a suitable option to implement.

Proposed Design Implementation

A UART interface between the on-board computer and the transmitting MCU should be created. UART is well suited for this task since the interface is between two devices, the on-board computer and the MCU, and the cable distance between the two devices is also very short. The EFR32^{TM} Flex

Gecko wireless SoC built in USART interface also supports several more protocols then I^2C, as listed in Table 3.4.

Table 3.4: USART Asynchronous vs. Synchronous Mode [17]

Communication Mode	Supported Protocols (Standard)
Asynchronous (UART)	RS-232, RS-485 (w/external driver), IrDA and ISO 7816
Synchronous (USART)	SPI, MicroWire, 3-wire

Silicon Labs have a recommended UART configuration, as listed in Table 3.5. The on-board computer requires a script for gathering the information, and a Python script would be suitable. The Python script and the VHF transmitter have to run the same UART configuration from Table 3.5, as this is a necessity for serial communication for avoiding data corruption.

Table 3.5: Recommended UART configuration [18]

Baud rate	115 200
Data bits	8
Parity bits	none
Stop bits	1
Flow control	none

Once the Python script starts gathering and transmitting data to the VHF transmitter, the incoming data requires some sort of handling. The handling of incoming data from the on-board computer can be done by creating a *structure*. Structures in C are user-defined data types that can be used to combine data items of different kinds, such as *characters* or *integers*. The structure intended to handle the data flow from the on-board computer should contain five different data items. These are a data buffer with configurable buffer size, read index, write index, byte counter, and a Boolean overflow indicator. Table 3.6 lists the configuration of the structure *rxBuf*, which is created for handling incoming data.

Table 3.6: rxBuf structure configuration

Variable name	Data type	Function
data	uint8_t	Data buffer
rdI	uint32_t	Read index
wrI	uint32_t	Write index
pendingBytes	uint32_t	Count of bytes yet handled
overflow	bool	Buffer overflow indicator

Note that the baud rate specified for this UART configuration only applies for the interface between the on-board computer and the VHF transmitter.

3.3 Message Format

3.3.1 Objective

The second objective of this thesis is designing a compact message format that allows the gathered information from the on-board computer to be transmitted according to the frequency regulations for the VHF band described in Section 2.3.3. This objective has existing work from a specialisation project conducted last autumn, and there are also similar systems in existence. The specialisation project proposed a message format inspired by the NMEA 0183 [2], and one already existing system that sends out navigational information is AIS as described in Section 2.4.1.

3.3.2 Existing Work

Starting with the proposed message format from the specialisation project, the message format is only a data field in the packet structure transmitted by the VHF transmitter. The packet structure used by the Flex Gecko MCU is shown in Figure 3.5 and is entirely configurable.

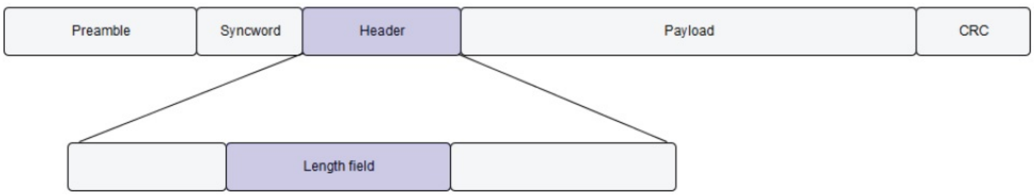


Figure 3.5: The radio packet structure [19]

The packet starts with a *preamble* field, which is a data field that contains alternating ones and zeros. The alternating pattern typically follows the structure of *"0101"* or *"1010"*, and it is transmitted with the most significant bit (MSB) first. The purpose of the preamble data field is to make the receiver aware that a message could arrive from the transmitter. The Automatic Gain Control (AGC) ensures that the receiver avoids saturation and optimises the performance by adjusting the receiver gain [15]. The length of the preamble pattern determines how aware the receiver is while waiting for a message. If the pattern is to short the receiver will not be aware enough and if it's too long the receiver will enter saturation. After the preamble data field comes a synchronisation word (*syncword*) that is used for indicating the start of the payload, which is also transmitted with MSB first. A good syncword have an autocorrelation which has one high distinct main lobe and small side lobes. Following the syncword is a *header* which is an optional field located before the *payload*. The *length* field in the header is required if the transferred payload has variable length, as in that case, it stores the length of each payload. The payload is the data field containing the information intended to be transmitted. The length of this field is dependant on the length of the message format and it can either be a fixed length or a variable length, with the latter one the header field is needed. The last field of the packet is a *Cyclic redundancy check (CRC)*, which also is an optional field. CRC codes are a type of error-detection codes. When a packet is sent a checksum follows each packet and when the receiver receives a packet it calculates a checksum and compares it to the received checksum. If the two checksums match each other then the received packet is valid, if they don't match each other then the received packet contains some type of error and the packet will be discarded. The structure and size of the packet components, especially the payload which contains the navigational information must be structured so that one of the frequency regulations described in Section 2.3.3 are met.

The complete message proposed in the specialisation project contained a total of 416 bits as shown in Table 3.7 and the packet structure is illustrated in Figure 3.6. The message format was inspired by NMEA 0183 with some of the constant elements such as commas (,), asterisk (*), line feed and carriage return (<CR><LF>) being removed to reduce the size of the payload. The other data fields used in the packet structure was the preamble field, a syncword, and the packet ended with a CRC-code, with the header field being excluded from the structure since the length of the payload was constant.

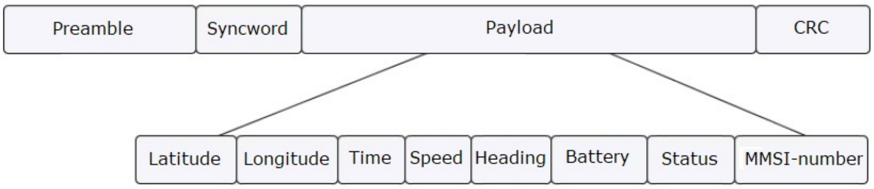


Figure 3.6: Proposed packet structure from the specialisation project [2]

Table 3.7: Proposed message format from the specialisation project [2]

Data fields	Number of bits
Preamble	40 bit
Syncword	16 bit
Package	344 bit
CRC	16 bit
Sum	416 bit

Preamble

The preamble data field had a length of 40 bits of alternating ones and zeros configured to transmit with MSB first, which means the pattern is [0101...01]. It is worth noting that the maximum length the preamble data field can have is $2^{21}-1 = 2097151$ [19].

Syncword

The synchronisation word used in the specialisation project was *F6 8D* which in binary is *11110110 10001101* and have a length of 16 bits. It is worth noting that the maximum length of the synchronisation word that can be used is 32 bits [19]. The autocorrelation calculated in MATLAB is illustrated in Figure 3.7 below.

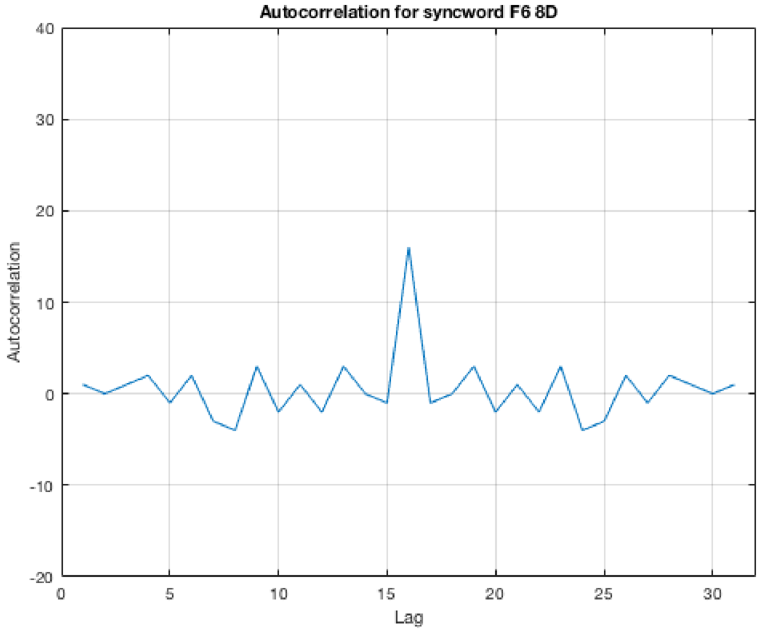


Figure 3.7: Autocorrelation for the syncword F6 8D

Payload

The proposed payload field contained the ferry's latitude, longitude, local time, the speed of the ferry, the heading of the ferry, the battery voltage of the battery pack on-board, the navigational status, and the MMSI-number associated with the ferry. Table 3.8 list each of the information fields structure, name, number of ASCII character used and number of bits used for each information field.

Table 3.8: Proposed payload from the specialisation project [2]

ASCII characters	Information	Number of ASCII characters	Number of bits
ll.lllll	Latitude	8	64
yy.yyyyy	Longitude	8	64
hhmmss	Time	6	48
x.x	Speed	3	24
www	Heading	3	24
vv.vv	Battery voltage	5	40
A	Navigational Status	1	8
xxxxxxxxx	MMSI-number	9	72
Sum		43 characters/bytes	344 bits

The information put together in the order from latitude to the MMSI-number gave the following payload string:

ll.lllllyy.yyyyyhhmmssx.xwwwvv.vvAxxxxxxxxx

The decision on the number of bytes required for each information field was justified for the following reasons:

The latitude and longitude information fields had five decimal places. These five decimal places were needed in order for the ferry to broadcast an acceptable precision on the ferry's position within the narrow operating area. Two decimal numbers and a decimal point were also included resulting in a total of eight characters. The time format which represents the time of transmission comes from the standard NS-ISO 8601 [20], although the colons separating hours from minutes and minutes from second were excluded. The ferry's speed varies between 1 kn and 9 kn and the heading varies between 0° and 360°. The battery voltage varies between 0V and 48.00V. The navigational status is represented by a single character and therefore only requires one byte. A MMSI-number have a 9-digit structure [21].

Cyclic redundancy check (CRC)

The Cyclic redundancy check field is one of the optional fields in the packet structure but was included in the message format from the specialisation project. The type of CRC chosen in the radio configuration for that message is named *CRC-16* [2].

Modulation

The modulation scheme proposed for the system in the specialisation project was Binary Frequency Shift Keying with a Gaussian filter and a bitrate of 4,8 kbps and 1,2 kHz deviation. The arguments made for this choice are that FSK is generally a good choice for power saving systems because non-linear amplifiers can be used, and that FSK is generally more easy to implement than for example MSK.

Link Budget

The specialisation project has a link budget calculated using FSPL as propagation model for a distance of 2000 m. This link budget did not include antenna gain or noise figures. The link margin was found to be 47.26 dB by comparing received power P_R to the second sensitivity listed in Table 3.3 in Section 3.1.3. A more thorough link budget should, therefore, be calculated.

Frequency Regulations

The performance of the system in the specialisation project is listed in Table 3.9. This system exceeds the maximum transmission time from regulation §8.7 in Table 2.4. The reason for this is that the packet transmitted had a size of 416 bits and a low bitrate of 4800 bps.

Table 3.9: System performance achieved in specialisation project [2]

Maximum permitted radiated power (MPRP)	69.18 mW
Maximum occupied bandwidth (OBW)	6.5 kHz
Maximum transmission time	8.67%

Automatic Identification System

Due to the transmission interval for the system, which is less or equal to one second, as described in Section 1.1, the Automatic Identification System is not a viable option. This is because the shortest transmission interval AIS can provide is two second, as described in Section 2.4.1.

3.3.3 Proposed Design Implementation

Taken the analysis of the proposed packet structure in the specialisation project, there are quite a few improvements that are needed in order to fulfil a chosen frequency regulation. For starters should some of the data fields be further improved. The proposed packet structure for this objective will have the same structure as the one proposed in the specialisation project as Figure 3.6 illustrates, but with improvements for the syncword and the payload. The modulation also has to be robust and operate within the chosen frequency regulation, which was not achieved in the specialisation project. A more precise link budget can also be calculated given the new information regarding the antennas, and noise figures, but also with the use of other propagation loss models.

Preamble

The preamble pattern of 40 bits used in the specialisation project proved efficient in making the receiver aware that a message could arrive, thus the same preamble will be used in this packet structure.

Syncword

The synchronisation word from the specialisation project will be changed to a longer word which has a better autocorrelation. The Silicon Labs support team suggest that the synchronisation word *2D D4* are used when a 16-bit long word is desired and recommends that the word *5A 0F BE 66* is used when a 32-bit long word is desired. The 32-bit long syncword have a binary pattern of *01011010 00001111 10111110 01100110* when it is transmitted over the air. The autocorrelation of the syncword is plotted in Figure 3.8.

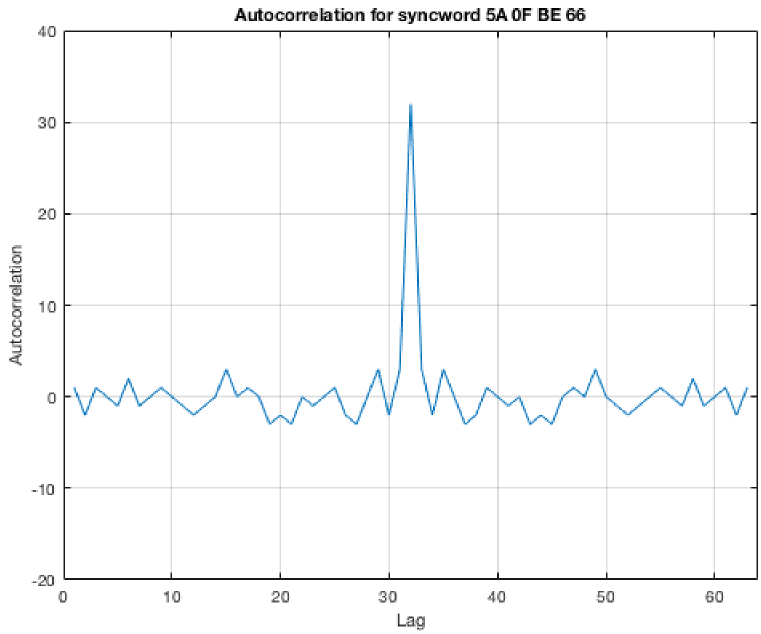


Figure 3.8: Autocorrelation for the syncword 5A 0F BE 66

Payload

The payload proposed in the specialisation project has nine bytes that can be removed to reduce the message size. The first two decimal places in the ferry's position will be constant and due to the ferry's small operating area, and they can, therefore, be removed from both the latitude and the longitude. The four period punctuation marks in the payload are also excess information not needed and can be removed. And a byte can be removed from the battery voltage because the accuracy is good enough for the first decimal place. Table 3.10 lists the new payload proposal.

Table 3.10: New payload proposal [2]

ASCII characters	Information	Number of ASCII characters	Number of bits
lllll	Latitude	5	40
yyyyy	Longitude	5	40
hhmmss	Time	6	48
oo	Speed	2	16
www	Heading	3	24
vvv	Battery voltage	3	24
A	Navigational Status	1	8
xxxxxxxxx	MMSI-number	9	72
Sum		34 characters/bytes	272 bits

The new structure of the payload on string format in order from latitude to MMSI-number would look like following:

lllllyyyyyhhmmssooxxxvvvAxxxxxxxxx

Cyclic redundancy check

The CRC-code used in the specialisation project proved efficient there and thus can be in this message as well, the CRC-code was a CRC-16 code.

New Message Format

The size and structure of the new message format are summarised in Table 3.11 below.

Table 3.11: New message format

Data fields	Number of bits bits
Preamble	40 bit
Syncword	32 bit
Package	272 bit
CRC	16 bit
Sum	360 bit

Modulation

There are five different modulation schemes available that can be combined with different pulse shaping filters, as mention in Section 3.1.2. The selection of modulation and configuration must take into account the *Free use regulation* and meet the requirements for the regulation which will be chosen in Section 3.3.2. Since it is rather quick work configuring the VHF radios, a thorough analysis of different modulations, pulse shaping filters and deviation should be done. A signal and spectrum analyser can be used to measure transmission time, radiated power and occupied bandwidth.

Frequency Regulation

From the problem description in Section 1.2 it is stated that the project intends to use the licence-free 169.4 MHz VHF. With that in mind only three of the regulations listed in Table 2.4 from Section 2.3.3 are useful, since the other regulations concerns higher frequencies beyond the licence-free 169.4 MHz VHF. The three regulations §7.1, §8.7 and §8.8 are listed in Table 3.12 below.

Table 3.12: Useful frequency regulations [11]

§	Range (MHz)	MPRP	Transmission time	OBW
§7.1	169.400−169.475	500 mW ERP	10% <	50 kHz
§8.7	169.400−169.475	500 mW ERP	1.0%	50 kHz
§8.8	169.400−169.475	10 mW ERP	0.1% <	-

The three regulations can be used for tracking, tracing, data acquisition and meter reading as described in the standard EN 300 220 [22]. The free regulation states that §7.1 shall be used for meter reading, which leaves §8.7 and §8.8 to be used for tracking, tracing and data acquisition. Looking at Table 3.12 it is clear that §8.7 have much larger boundaries than §8.8, and §8.7 should, therefore, be the system goals limitation. As stated in Section 3.3.2 the previous work failed to meet this requirement, but with an increase in bps and a reduction in the size of the message format this goal can be achieved. An equation to find the minimum bitrate x which fulfil regulation §8.7 can be expressed as:

$$\frac{360bit}{x} = 0.01 \Rightarrow x \geq 36000bps \tag{3.1}$$

From the Equation 3.1 it is clear that the minimum bitrate the system must have in order to fulfil regulation §8.7, is 36 kbps. The bitrate is therefore set to 36.8 kbps to avoid exceeding the maximum transmission time from regulation §8.7.

Link Budget

Two different basic propagation models were introduced in Chapter 2.3.2, the Free Space Path Loss Model and the Plane Earth Loss Model. The link budget calculated in the specialisation project only had basic components included, and the received power was compared with a given sensitivity from the MCUs datasheet resulting in an inaccurate link margin. A more thorough link budget can be calculated by introducing more information such as antenna gain, noise figure, BER, and $\frac{Eb}{N0}$ for the different modulations. However, there is no limit to the number of bit errors in the system specification. One error roughly every third hour should be an acceptable BER. The link budget below is calculated with BER $= 10^{-4}$, $\frac{Eb}{N0}$ values for this BER for each modulation are found in Figure 3.9. The distance is set to 2000m which is well beyond the actual distance over the channel. The antenna noise temperature is approximated value, and the cable loss between the MCUs and the antenna is estimated to be zero due to the short distance and lack of information surrounding a value. Transmitter power and antenna gain are taken from the datasheets of the respective components, and the transmitter and receiver heights are an approximate value.

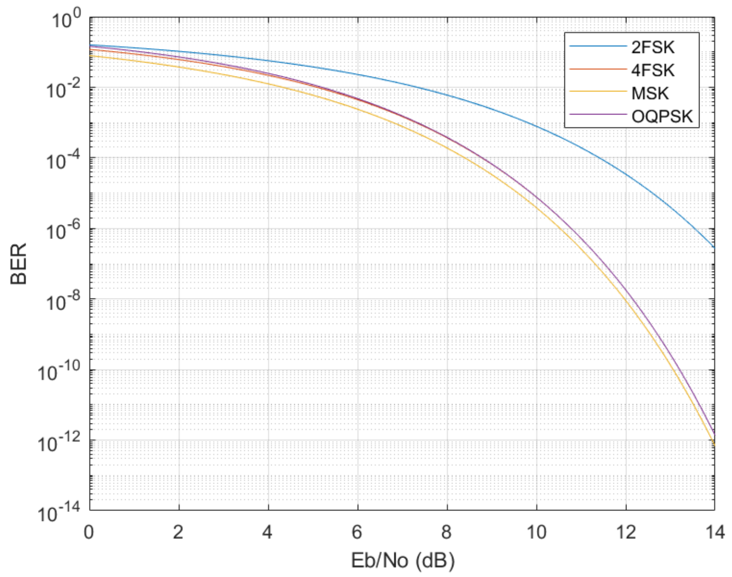


Figure 3.9: BER vs $\frac{Eb}{N0}$ for an Additive white Gaussian noise (AWGN) channel

Table 3.13: Link Budget for FSL and PEL respectively

Quantity	Formula or symbol	Result
Transmitter		
Transmitter Power	P_T	18.5 dBm
Antenna gain	G_T	2.1 dBi
EIRP	$P_T + G_T$	20.6 dBm
Channel		
Distance [m]	d	2000 m
Distance [km]	D	2 km
Carrier frequency	f_c	169.4 MHz
Transmitter height	h_t	3 m
Receiver height	h_r	3 m
Free space loss	$L_{FSL(dB)} = 32.4 + 20 \cdot log_{10}(D) + 20 \cdot log_{10}(f_{MHz})$	83.05 dB
Plane Earth Loss	$L_{PEL(dB)} = 40log(d) -(20log(h_r) + 20log(h_t))$	112.95 dB
Receiver		
Receiver noise figure	F	7 dB
Antenna gain	G_R	2.1 dBi
Antenna noise temp	T_a	290 K
System noise temp	$T_e = 290(10^{\frac{F}{10}} - 1)$	1163.4 K
Equ. receiver noise temp	$T_{eq} = 10log(T_{sys} + T_a)$	31.62 dBK
Boltzmann's constant	k	$1.38 \cdot 10^{-23}\ J \cdot K^{-1}$
Receiver G/T	$\frac{G}{T} = G_R - T_{eq}$	-29.52 dB/K
Signal-to-noise ratio for FSL	$\frac{P_R}{N_0}$=EIRP$-$ $L_{FSL}+\frac{G}{T} - k$	107.63 dB
Signal-to-noise ratio for PEL	$\frac{P_R}{N_0}$=EIRP$-$ $L_{PEL}+\frac{G}{T} - k$	77.72 dB

The SNR is a simple summation of the systems gain and loss.

Quantity	Formula or symbol	Result
Signal		
Data rate	R_b	36,8 kbps
Data rate [dB]	$R_b[dB] = 10log(R_b)$	45.65 dB
Energy per bit to noise ratio with FSL	$\frac{E_b}{N_0} = \frac{P_R}{N_0} - R_b[dB]$	61.97
Energy per bit to noise ratio with PEL	$\frac{E_b}{N_0} = \frac{P_R}{N_0} - R_b[dB]$	32.06
Required $\frac{E_b}{N_0}$ for 2FSK	$\frac{E_b}{N_0}req$	11.4 dB
Required $\frac{E_b}{N_0}$ for 4FSK	$\frac{E_b}{N_0}req$	8.8 dB
Required $\frac{E_b}{N_0}$ for MSK	$\frac{E_b}{N_0}req$	8.4 dB
Required $\frac{E_b}{N_0}$ for OQPSK	$\frac{E_b}{N_0}req$	8.8 dB
Link margin with FSL as propagation model		
Link margin 2FSK	$\Delta_{SNR} = \frac{E_b}{N_0} - \frac{E_b}{N_0}req$	50.58 dB
Link margin 4FSK	$\Delta_{SNR} = \frac{E_b}{N_0} - \frac{E_b}{N_0}req$	53.18 dB
Link margin MSK	$\Delta_{SNR} = \frac{E_b}{N_0} - \frac{E_b}{N_0}req$	53.58 dB
Link margin OQPSK	$\Delta_{SNR} = \frac{E_b}{N_0} - \frac{E_b}{N_0}req$	53.18 dB
Link margin with PEL as propagation model		
Link margin 2FSK	$\Delta_{SNR} = \frac{E_b}{N_0} - \frac{E_b}{N_0}req$	20.66 dB
Link margin 4FSK	$\Delta_{SNR} = \frac{E_b}{N_0} - \frac{E_b}{N_0}req$	23.26 dB
Link margin MSK	$\Delta_{SNR} = \frac{E_b}{N_0} - \frac{E_b}{N_0}req$	23.66 dB
Link margin OQPSK	$\Delta_{SNR} = \frac{E_b}{N_0} - \frac{E_b}{N_0}req$	23.26 dB

The link margin for the four available modulation schemes is in the region of 50 dB when the FSL model is used, which in this case could be an unrealistic model to use. When the PEL model is used the link margin is around 30 dB lower. Figure 3.10 shows how path loss increases for the two models over the distance of 2000 m in the link budget. Since the ferry will be operating in an area which could be described as an urban environment the PEL models link margin might yield the most accurate prediction.

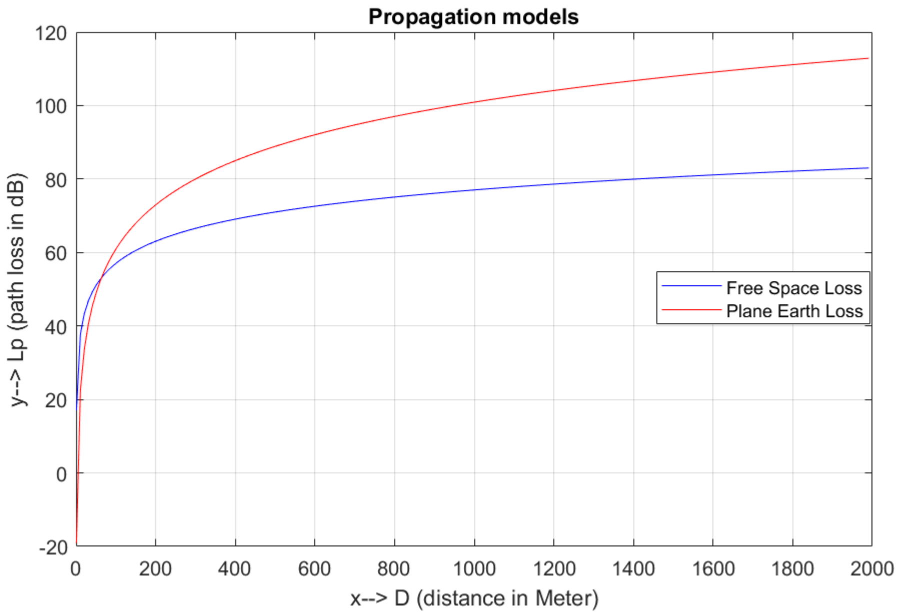


Figure 3.10: Path loss for FSL and PEL over a distance of 2000m

Source Code

Code that handles transmitting has to be implemented at the VHF transmitter, the same MCU which gathers navigational information. In the previous objective a structure for handling the incoming data from the on-board computer where created. A function for generating a payload is therefore created. This function has a pointer to the data buffer from the rxBuf which then becomes the payload. After a payload is successfully transmitted the structure is emptied and awaits new data from UART. The whole process is synchronised with the use of a timer, which assures that navigational information is transmitted each second according to the system specification from Section 1.1.

3.4 VHF Receiver to Chartplotter

3.4.1 Objective

The third and last objective of this thesis is the design and implementation of a data interface from the VHF receiver to a chartplotter using NMEA 0183 format.

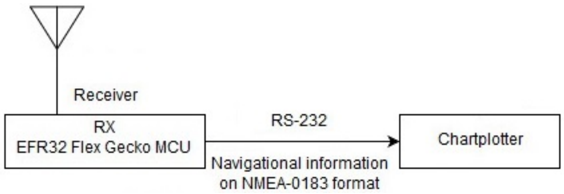


Figure 3.11: VHF receiver to chartplotter block diagram

The receiver MCU receives the payload proposed in Section 3.3 over RF. The payload then has to be processed and formatted on to a suitable NMEA 0183 format on the MCU before being transmitted over a data interface to a chartplotter.

3.4.2 Existing Work

This objective is similar to the first objective. Both objectives involve the implementation of a data interface between two devices where one of the devices are the Flex Gecko MCU. It is fair to say that the same factors apply for this interface as for the interface in the first objective, that is to say, that the distance between devices is short and only two devices are connected. UART communication is, therefore, a suitable option for this interface as well.

3.4.3 Proposed Design Implementation

Data Processing

The received navigational information has to be processed and formatted onto a suitable NMEA 0183 format, and two structures, as listed in Table 3.14 could be used in this process. The first structure should handle the received data from the transmitter and the second structure should handle the processed data awaiting transmission over the data interface.

Table 3.14: Configuration structure

Received data structure

Variable name	Data type	Function
data	uint8_t	Data buffer
rdI	uint32_t	Read index
wrI	uint32_t	Write index
pendingBytes	uint32_t	Count of bytes yet handled
overflow	bool	Buffer overflow indicator

Awaiting transmission structure

Variable name	Data type	Function
data	uint8_t	Data buffer
rdI	uint32_t	Read index
wrI	uint32_t	Write index
pendingBytes	uint32_t	Count of bytes yet handled
overflow	bool	Buffer overflow indicator

Recommended Minimum Navigation Information (RMC) is a NMEA 0183 sentence which contains a vessels position, speed, heading, date and time of message, and magnetic variation. It is a sentence commonly emitted by GPS units [23]. Table 3.15 lists the message fields of a RMC sentence.

Table 3.15: RMC message fields

Field	Description
1	UTC of position fix in hhmmss.ss
2	Status, V=Warning, A=Valid
3	Latitude
4	North or South (N or S)
5	Longitude
6	East or West (E or W)
7	Speed over ground in knots
8	Heading
9	Date, ddmmyy
10	Magnetic Variation, degrees
11	East or West (E or W)
12	Checksum

The payload proposed in Section 3.3.3 contains all this information except date and magnetic variation. The magnetic variation can be estimated to be 1° East in the area around Trondheim [24]. The received payload has to be formatted so that each data field contain right information. $\$GPRMC$ have to be included first in the sentence, and checksum calculation has to be performed of the information and added to data field 12. At the end control characters <CR><LF> must be added. The structure of the message in the awaiting transmission structure should then look like following:

$GPRMC,1,2,3,4,5,6,7,8,9,10,11,12,<CR><LF>

Data Interface

Table 3.16 lists a configuration proposal for the UART interface between the receiver MCU and a chartplotter.

Table 3.16: UART configuration chartplotter

Baud rate	4800
Data bits	8
Parity bits	none
Stop bits	1
Flow control	none

A thing to notice is that the configuration is similar to the one used between the on-board computer and the transmitter MCU, except for the baud rate. The baud rate is set to 4800 which is typically the baud rate used in the NMEA 0183 communication protocol, as mentioned in Section 2.4.2.

Chapter 4

Result and Discussion

This chapter contains the results and discussion of the proposed design implementations for the three objectives of this thesis.

4.1 On-board Computer to VHF Transmitter

The first objective of this thesis was to implement a Universal Serial Bus (USB) data interface from the on-board computer on the ferry to the VHF transmitter. The previous chapter presents a proposal for the design of the data interface, by using the built-in UART serial interface in the MCU. To gather the navigational information from the on-board computer a Python script should run continuously on the on-board computer. The Python script will have to be set up with the same serial configuration as the UART configuration used on the MCU. A structure for handling the data flow between the two devices is also proposed. To verify that the proposed solution for the data interface works, three different tests are conducted in the next subsections. Each of the tests will look at how the recommended UART configuration and the proposed data handling structure works in practice.

4.1.1 Test: Serial Communication between PC and VHF transmitter

The two first tests are run with one of the MCU acting as transmitter and the other MCU acting as receiver, both are connected to a PC with USB cable.

In the first test the serial console PuTTY is used to open a serial session on both MCUs using the proposed UART configuration from Table 3.5. PuTTY allows users to set up serial sessions that can either be used to print from or write to different devices. To test the data interface, the payload string from Section 3.3.3 is copy-pasted into the transmitters serial console. This should then trigger the transmit function to the MCU if the UART communication works correctly. To monitor the test, the first serial port is connected to the transmitter and the other serial port connected to the receiver. Each time the string was copy-pasted into the serial console for the transmitter, the structure for handling incoming data from UART worked as intended. The transmit function successfully triggered when a payload was ready for transmission. The payload was then transmitted and correctly printed at the receivers serial console before the data handling structure was reset. Therefore, the test is considered a success.

In the second test, a Python script is written which has an inclusion of the serial support library PySerial and the time library. The script uses the PySerial library to set up a serial interface with the UART configuration from Table 3.5. The transmitter MCU is connected to the COM-port specified in the Python script. A for loop is constructed which transmits the payload string from Section 3.3.3 continuously with time delays between each transmission. The time library is used to create these time delays, which is 1 second each. The delay between each transmission is added to simulate the transmission interval that is part of the system specifications from Section 1.1. A serial session is opened to the receiver MCU using PuTTY and set up with the same UART configuration as the Python script uses. The Python script is executed on the PC from the Command Prompt in Windows. When this happened the for loop in the script started transmitting the payload with a time delay between each transmission as intended. Each second the transmitter MCU received data from the PC, the data was correctly placed into the structure created for handling the data flow. The transmit function in the transmitter was once again successfully triggered when data was present in the structure. The payload was correctly transmitted by the transmitter MCU and received by the receiver MCU which printed the data correctly in the receiver serial console. The data handling structure was also reset as should be expected. This test is also considered a success.

4.1.2 Test: Serial Communication between OBC and VHF transmitter

The last test for this objective is conducted on the autonomous ferry. The transmitter MCU is connected to PORT 1 on the ferry's on-board computer with a USB cable, as visualised in Figure 4.1.

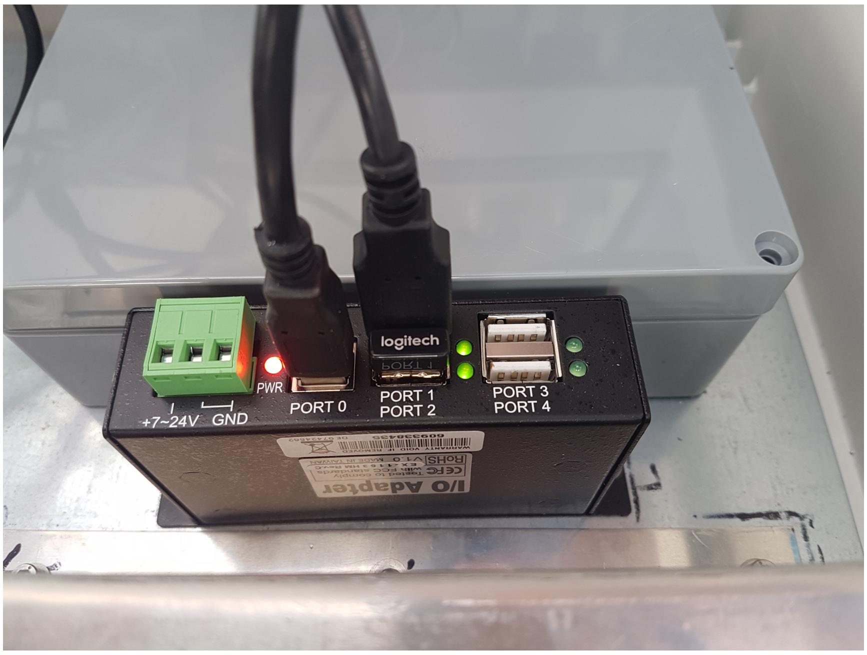


Figure 4.1: VHF transmitter connected to PORT 1 on the on-board computer

A similar Python script to the script used in the previous test is written on the on-board computer using the same structure and libraries, but earlier release code is used since the on-board computer uses Python 2.7. The receiver MCU is connected to a PC and a serial session is opened, ready to print the receiving information from the on-board computer. When the Python script was executed from the terminal the payload this time was successfully transmitted from the on-board computer to the transmitter MCU. As in the two previous tests, the transmit function triggered when the data handling structure contained data. The payload was transmitted, the data handling structure correctly reset and the receiver received the payload correctly and printed the data in the receiver serial console. The last test is also considered a success.

4.1.3 Summary of OBC to VHF Transmitter

Using UART has proven to be a good choice for the serial communication between the on-board computer and the VHF transmitter. All three tests done on the first objective have had successful results. All data transmitted with the use of the UART were successfully received at the MCU without any misinterpretation of the data. The number of devices and the low cable distance between the devices, as well as the possibility for an asynchronous mode, made this serial communication link fitting for UART, as stated in Section 3.2.1. However, as this objective is the implementation of a USB data interface from the on-board computer on the ferry to the VHF transmitter, gather navigational information on the on-board computer also have to be done. Two of the tests in this section have shown that a Python script can be used to transmit data with UART. If a Python script is written that is tasked with collecting actual navigational information from various processes that have stored the desired information, it would be easy to include an extension of code which utilises the pySerial library to transmit data to the VHF transmitter. This script should be written by the developer responsible for the layout of the ROS system ferry user, or someone with thorough knowledge of other ROS systems. The navigational information must be formatted as the payload string from the payload proposal in Section 3.3.3, that is a 34-byte long integer string.

4.2 Message Format

The second objective of this thesis was to design a compact message format that allows data to be transferred according to the frequency regulations for the 169.4 MHz VHF band. An improvement to the existing message format from the specialisation project conducted last autumn, which was based on NMEA-0183 is laid out in Section 3.2.1. The improvement proposed alterations in the syncword and payload data fields in the packet structure and that an analysis should be conducted for selecting a suitable modulation scheme since the frequency regulation is exceeded in the specialisation project. This section starts with the analysis for finding the suitable modulation scheme. The analysis is then followed by laboratory measurement and calculations verifying that the selected modulation scheme does meet the frequency regulation. At the end of the section are outdoor measurements of the RSSI for the radio link.

4.2.1 Modulation

Spectrum Analysis

Each different modulation and configuration's spectrum can be analysed using a signal analyser. The output of the plug-on radio board is connected to the RF input on the signal analyser, as Figure 4.2 visualises. The signal spectrum for the different modulations and their configuration should bear resemblance to their theoretical spectrum, if not the signal could prove difficult to detect. The modulations that have the desired spectral characteristics should also not occupy too much bandwidth, as the frequency regulation chosen in Section 3.3.3 states that the 99% occupied bandwidth cannot exceed 50 kHz.

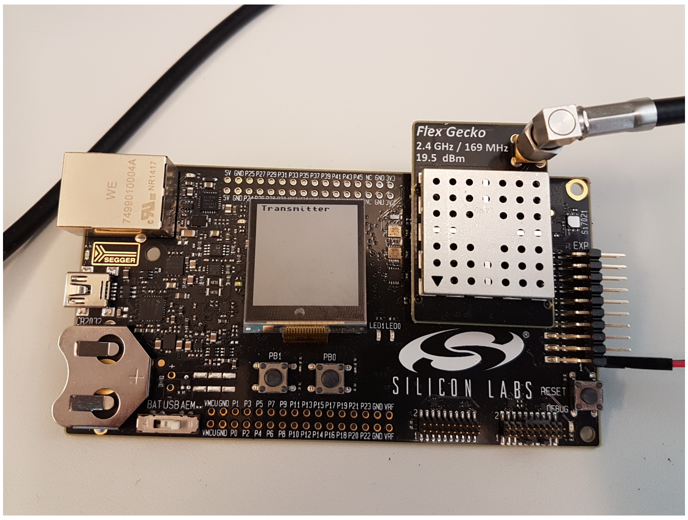


Figure 4.2: Spectrum analysis setup with a coaxial cable connected to the output found on the plug in radio board

Since the payload is a 34 digit integer string where each digit varies between zero and nine, a random digit generator is created for simulating the actual payload. The random generator creates a random string once each second and the transmit function transmits the signal to the signal analyser, this is done 50 times. Before each measurement starts the **Max Hold** instrument function is activated. This function assures that the spectrum of each signal fills up for each sweep until all of the signal components are detected [25]. When all 50 packets have successfully been transmitted, "Ready for measurement" is printed in the serial console using UART. The **OBW** measurement function is then used to calculate a selected percentage of the signals occupied bandwidth, in this case, the percentage is 99 %. For each measurement the data is exported on a ASCII file format for data storage [25]. All signals are transmitted using the proposed bitrate of 36.8 kbps from Section 3.3.3. The measured modulations and configurations are listed in Table 4.1.

Table 4.1: Measurement results of the different spectrums

	Modulation	Filter used	OBW	Deviation used	Peak dBm
1	2-FSK	No filter	74.96 kHz	14 kHz	17.9 dBm
2	2-FSK	Gaussian	47.61 kHz	14 kHz	17.9 dBm
3	2-FSK	Raised cosine	45.48 kHz	14 kHz	18.2 dBm
4	2-FSK	Raised cosine	47.75 kHz	15 kHz	18.1 dBm
5	2-FSK	Gaussian	101.32 kHz	36.8 kHz	19.84 dBm
6	4-FSK	No filter	92.61 kHz	11 kHz	18.9 dBm
7	4-FSK	Gaussian	78.14 kHz	11 kHz	18.8 dBm
8	4-FSK	Raised cosine	75.25 kHz	11 kHz	18.9 dBm
9	OOK	No filter	301.03 kHz	10 kHz	18.1 dBm
10	OOK	Gaussian	79.57 kHz	10 kHz	16.9 dBm
11	OOK	Raised cosine	75.25 kHz	10 kHz	17.7 dBm
12	MSK	No filter	47.75 kHz	10 kHz	18.1 dBm
13	MSK	Gaussian	41.96 kHz	10 kHz	18.5 dBm
14	MSK	Gaussian	41.96 kHz	15 kHz	18.5 dBm
15	MSK	Raised cosine	41.96 kHz	15 kHz	18.6 dBm
16	OQPSK	No filter	46.3 kHz	15 kHz	19 dBm
17	OQPSK	No filter	46.3 kHz	18 kHz	19 dBm
18	OQPSK	Gaussian	41.96kHz	18 kHz	19 dBm
19	OQPSK	Raised cosine	39.07 kHz	18 kHz	19 dBm

From the results in Table 4.1 it is clear that any configurations using OOK or 4-FSK can be discarded right away, since their 99% OBW is higher than 50 kHz. However 2-FSK, MSK, and OQPSK all have different configurations with 99% OBW lower than 50 kHz. The remainder of the modulation schemes spectrum were compared to the spectrum of their theoretical equivalents and the best result for each modulation scheme were picked out. Figure 4.3 shows the measured spectrum for these three modulation, which are number 2, 13, and 18 from Table 4.1. The MSK and 2-FSK modulation both have a Gaussian filter, while the OQPSK modulation have a Raised cosine filter.

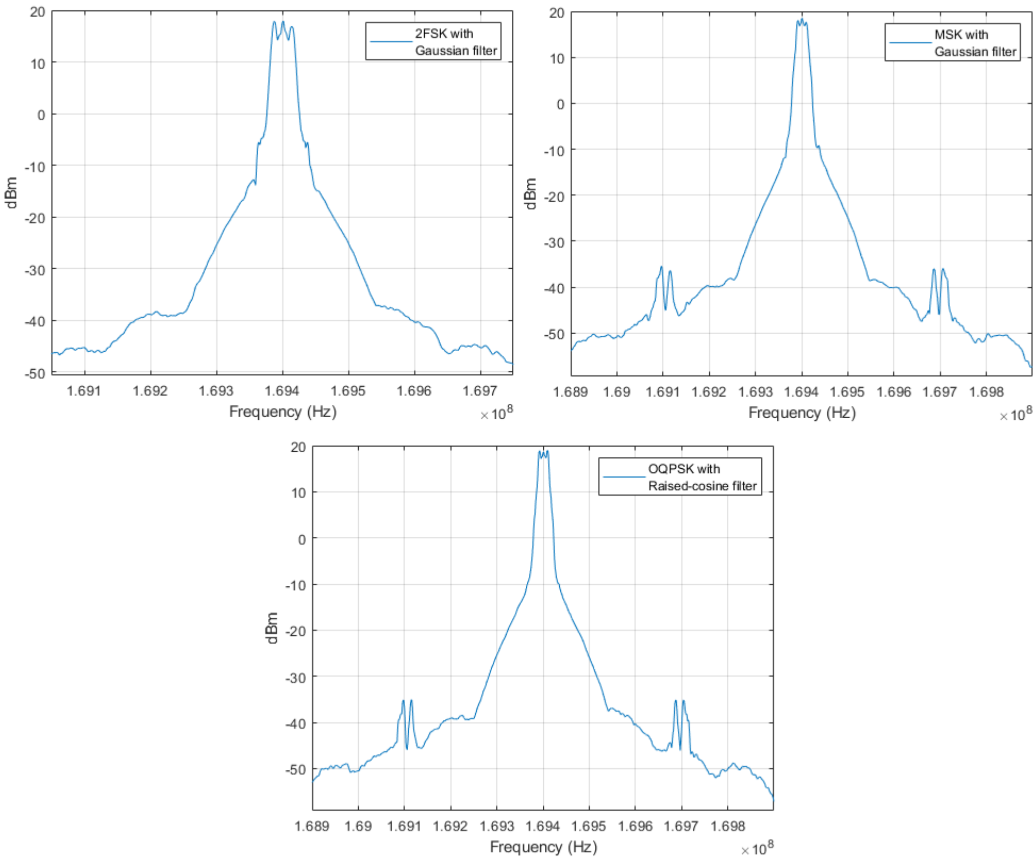


Figure 4.3: Best spectrum measured for FSK, MSK, and OQPSK

The modulation selected ultimately must have a low packet error rate to ensure that the navigational information is continuously received by surrounding vessels so that collisions are avoided. The packet error rate is therefore analysed for these three modulations in the next section.

Packet Error Rate

Packet error rate (PER) rate is the number for the percentage of incorrectly received data packets. It is calculated using Equation 4.1, where P_{Tx} is the number of packets transmitted and P_{Rx} is the number of packets received. A test application for calculating PER is one of the software examples found in Simplicity Studio. This application can be used to calculate the PER for the different modulations.

$$\text{Packet Error Rate (PER)}[\%] = \frac{(P_{Tx} - P_{Rx})}{P_{Tx}} \cdot 100 \tag{4.1}$$

The test application sets up one MCU as a transmitter and the other MCU as a receiver. The length of each packet, transmit power, and number of packets is configurable. Each packet transmitted contain the address of the transmitter and the number of packets sent, this number increments for each packet sent. At the receiver each packet and its address are checked and if their addresses match the packet number is stored. To simulate distance between the transmitter and the receiver, several attenuators are coupled between the transmitter and receiver. The power received over a distance of 2000 m in an ideal situation is calculated using Friis transmission from Section 2.3.1 in the equation below:

$$\begin{aligned} P_R(dB) &= P_T + G_T + G_R + 20 \cdot log_{10}\left(\frac{\lambda}{4\pi d}\right) \\ P_R(dB) &= 19.5dBm + 2.1dBi + 2.1dBi \\ &+ 20 \cdot log_{10}\left(\frac{\frac{299792458m/s}{169.4MHz}}{4 \cdot \pi \cdot 2000m}\right) = -59.35dBm \end{aligned} \tag{4.2}$$

After the attenuation, the signal is connected to a power divider which divides the signal in two. The first output is connected to a signal analyser and the second output is connected to the receiver MCU. The power divider used has a 3 dB *coupling loss* (L_C) and a 0.35 dB *insertion loss* (L_I), which needs to be included in the measurement when selecting the attenuators. The signal strength expected should then be $P_R - L_C - L_I = -62.7dBm$, and corresponding to a distance of 2000m between transmitter and receiver. This value includes the gain from the Scan Antenna VHF73X, coupling loss, insertion loss and transmit power. This desired P_R value of -62.7 dBm was measured when the attenuators in series had a value of 80 dB and the transmit power were reduced with 2.3 dBm.

In the PER measurement each of the three modulations sent 50 000 packets with the same length as the proposed message format in Table 3.11 in Section 3.3.3. The resulting percentage packet loss is listed for each modulation tested in Table 4.2 below.

Table 4.2: Packet error rate results

Modulation	Value
2GFSK modulation number 2	0%
GMSK modulation number 14	82.1%
OQPSK modulation number 18	65.9%

Looking at the results in Table 4.2 suggests that there either is a major problem at the receiver or the test bench setup. To see if there is a problem with the test bench, various levels of attenuation, increase and decrease in bitrate, is tested for both the MSK- and OQPSK-configuration, along with increasing or reducing bitrate for each test. However, there is no resulting impact on the amount of packet loss for either of the configurations for these tests. The manufacturer of the MCU was therefore contacted for support and the results of the tests, and the test bench setup was conveyed to the manufacturer. The test was recreated by the manufacturer who reported back similar PER measurements for the same modulation and configurations [26]. Another thing noticed during the packet error tests was that the RSSI values for the RF-link appeared to differ from the power received measured on the signal analyser. This suspicion is confirmed in the range test datasheet, where it is stated that a systematic offset will appear due to radio configuration and matching network [27]. This systematic offset has to be measured using the signal analyser with various levels of attenuation on the RF-link when the RSSI measurement is done in the ferry's operating area.

So although MSK combines really well with the Gaussian pulse shaping filter and creates a clear spectrum and also is a very popular modulation scheme for mobile radio communication systems, MSK along with OQPSK cannot be used due to the high packet loss at the receiver. The 2-GFSK modulation had a very good PER despite having an undesired spike in the centre of the spectrum. This undesired spike was also observed on the signal analyser on numerous different bitrates, deviations and frequencies during the test bench setup. The results of the PER test indicate that the spike does not appear to be harmful and of any significance for a successful detection of a message when a signal is received. It should be noted that the spike does affect the amount of occupied bandwidth, but the 99 % OBW for the signal is still below the limit of 50 kHz. Modulation number two from Table 4.1, that is the GFSK with a deviation of 14 kHz and a bitrate of 36.8 kbps is therefore chosen as modulation.

4.2.2 Frequency Regulation

Section 3.3.3 proposes that the system must meet the frequency regulation §8.7 as listed below in Table 4.3.

Table 4.3: Regulation §8.7 [11]

§	Range (MHz)	MPRP	Transmission time	OBW
§8.7	169.400−169.475	500 mW ERP	1.0%	50 kHz

Transmission Time

Table 4.4: Size of transmitted message

Data field	Number of bits
Preamble	40 bit
Syncword	32 bit
Payload	272 bit
CRC	16 bit
Sum	360 bit

Table 4.4 lists the proposed message format with the sizes of the different data fields. The system must according to the specifications in Section 1.1 transmit navigational information at an interval less or equal to 1 second. Regulation §8.7 states that the transmission time shall not exceed 1%. One per cent of the transmission interval of 1 second gives a maximum transmission time of 10ms. The actual transmission time D_T for the proposed message can be found by dividing N number of bits on the bitrate R, as shown in the equation below.

$$D_T = \frac{N}{R} = \frac{360bit}{36800bps} = 0.0097826s = 9.78ms \tag{4.3}$$

The transmission time can also be measured at the laboratory to further verify that Equation 4.3 is correct, by using a signal analyser. The measurement of the transmission time at the laboratory with the signal analyser is plotted below in Figure 4.4.

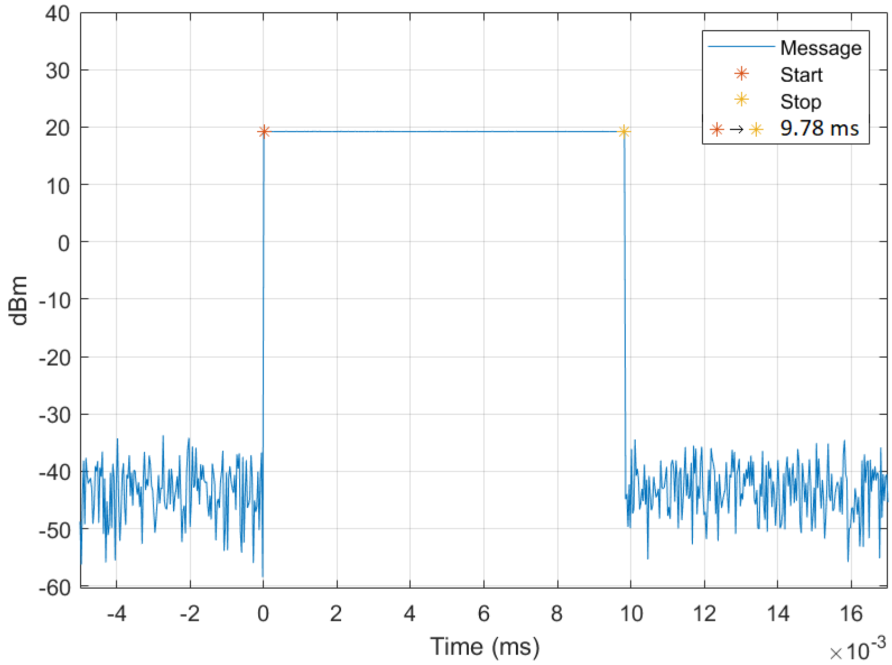


Figure 4.4: Transmission time for the transmitted message

Both the equation result and the plot shows that the transmission time of the proposed message at the proposed bitrate of 36.8 kbps does not exceed the maximum transmission time of 1%.

Maximum Permitted Radiated Power (MPRP)

The Maximum Permitted Radiated Power (MPRP) from regulation §8.7 in Table 4.3 is defined as Effective Radiated Power (ERP). To verify that this is actually the case NKOM was contacted. NKOM confirmed that the MPRP for regulation §8.7 is defined in ERP [28]. The signal analyser at the laboratory and the Reference Manual for the MCUs operates with the term Effective Isotropic Radiated Power (EIRP), and a conversion between ERP and EIRP is therefore needed. Equation 2.14 that is found in Section 2.3.2 lists two useful equations for converting between EIRP and ERP.

$$EIRP(W) = 1.64 \cdot ERP(W) = 1.64 \cdot 500mW = 820mW \qquad (4.4)$$

Using the first equation one finds that the MPRP for regulation §8.7 is equal to 820 mW when denoted in EIRP. The datasheet of the Flex Gecko lists the transmit power for Sub-GHz operations as *up to 20 dBm* [15]. A more accurate value can be measured in the laboratory with the use of a signal analyser. For the proposed modulation the transmit power was measured to be 17.9 dBm with the signal analyser. Since this measurement is in dBm it's necessary to use Equation 4.5 to convert the value from dBm to mW.

$$P(mW) = 1mW \cdot 10^{\frac{P(dBm)}{10}} = 1mW \cdot 10^{\frac{17.9}{10}} = 61.66mW \tag{4.5}$$

The MPRP from regulation §8.7 is 820 mW EIRP and the measured EIRP from the laboratory is 61.66 mW, which is well within the limit of the regulation.

Maximum Occupied Bandwidth OBW

From the definition in Section 2.3.3 the OBW is the bandwidth which contains 99 per cent of the signals average power. Regulation §8.7 states that the OBW shall not exceed 50 kHz. Using the built-in software tools in the signal and spectrum analyser, the measured OBW with a RBW and a VBW of 5kHz is 47.61 kHz for the chosen modulation. The signal measured in the laboratory is plotted in Figure 4.5 below.

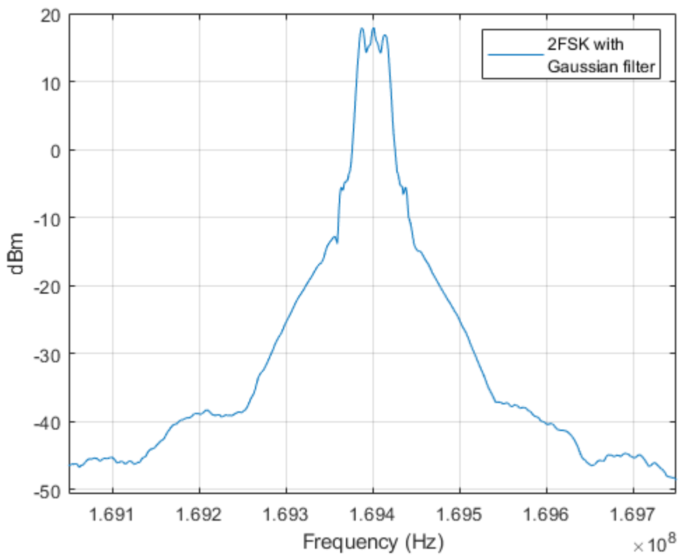


Figure 4.5: Spectrum for the chosen modulation, 2-GFSK with 36.8 kbps bitrate and 14 kHz deviation

4.2.3 Received Signal Strength Indicaton Measurement

During the packet error test in Section 4.2.1 an offset was found between the RSSI value provided by the MCU and the received power measured with the signal analyser. A calibration of the actual RSSI values for the chosen modulation is therefore done first, before performing measurements of the RSSI values in different zones within the ferry's operating area.

RSSI Calibration

For the calibration of the RSSI values, the same test bench setup used in the packet error rate test is used. The test application which calculated the PER in the packet error rate test is also used, as this applications displays the RSSI value measured on the MCUs Liquid crystal display (LCD) when the synchronisation word of a packet is received [27].

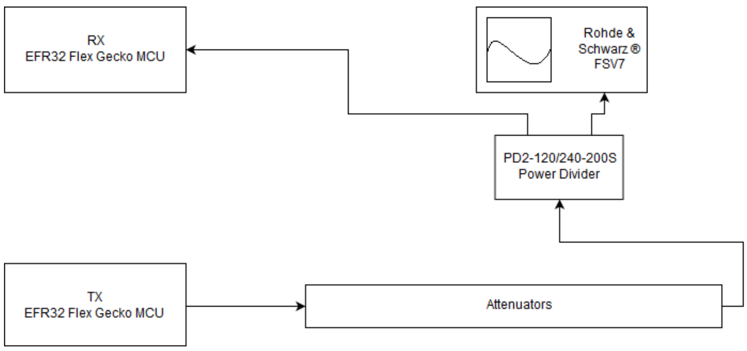


Figure 4.6: RSSI calibration test bench

The calibration starts with a 30 dB attenuation coupled between the transmitter MCU and the power divider. The attenuation used increases gradually for each measurement done, and the last attenuation value used was 113 dB. The calibration found an average difference between the RSSI value displayed on the LCD and the received power at the signal analyser of 14.5 dBm.

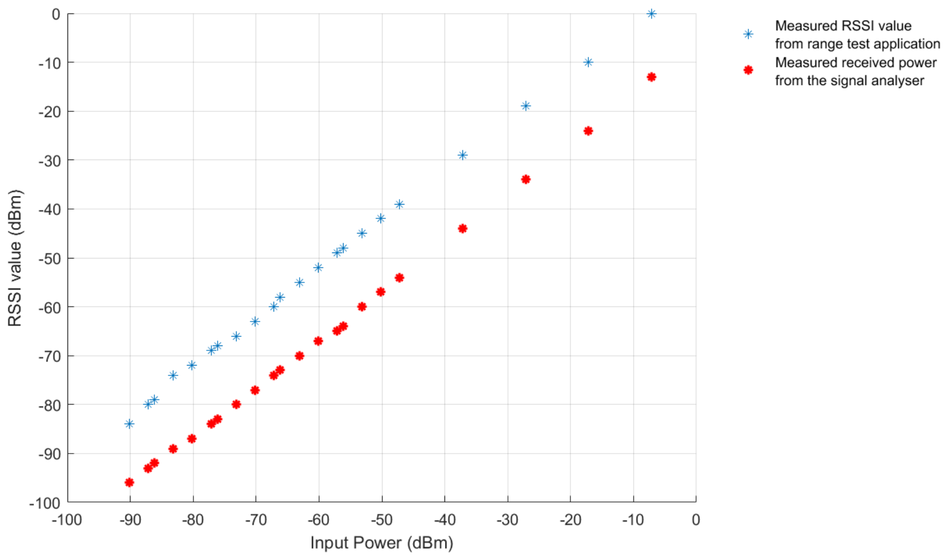


Figure 4.7: Difference in the RSSI values from the MCUs given value to the measured value from the signal analyser

Figure 4.7 shows there appear to be an almost linear difference between the RSSI values provided by the MCU compared to the measured values from the signal analyser. At the highest attenuations used in the test, rapid fluctuations in measure values occurred, and it was difficult to get an accurate reading on the measured values. This could have been a result of the high amount of coupling the highest attenuations required, as well as the resulting signal strength becoming very low. The results found in the calibration will have to be taken into account when the RSSI measurements in the operating area of the ferry are undertaken. The offset found in the calibration also confirms the information provided by one of the applications manuals which states that an offset exists for the various modulations [27].

RSSI Measurement Trondheim Harbour

With the results of the RSSI calibration acquired, an outdoor measurement down in Trondheim harbour in the ferry's operating area is performed. The test application that was used in the packet error test and the calibration of the RSSI is also used here to measure the RSSI. In this test, both MCUs are used, with one being a transmitter and one a receiver. The different locations visited during this test is marked with numbers, and red or green balloons in Figure 4.8. The receivers locations are marked with a green balloon, the letter R, and a measurement number. The transmitters locations are marked with a red balloon, the letter *T*, and a measurement number. The measurement numbers have matching pairs, which show how the transmitter and receiver were located in relation to each other during the test. In Figure 4.8 the left side of the figure shows an overview of the area where the measurements were conducted. The majority of the measurements were done in the area around the channel between Ravnkloa and Vestre Kanalhavn, but there is also a measurement data from the channel to Trondheim market square. The right side of the figure shows clusters of measurement numbers. The clusters are measurements where either the transmitter or the receiver remained stationary, while the opposite radio changed location. The clusters can be seen on both sides of the widest area of the channel.

Figure 4.8: Measurement location, overview picture on the left side and two magnified areas on the right side

Figure 4.9 shows the measured RSSI values over a distance in meters, and the expected signal strength over distance for both propagation models in the link budget. The RSSI values are corrected for the offset found during the calibration in the previous Section 4.2.3. Most of the RSSI values are in proximity of the expected received signal strength for the chosen propagation model, the PEL model. However, the result of this test can be considered slightly conservative since the measurement is done with the antenna supplied with the MCUs, and there is no data available on the supplied antennas gain.

Some of the lowest RSSI measurements dip below the lowest value the MCU can measure when they are corrected for the offset [15].

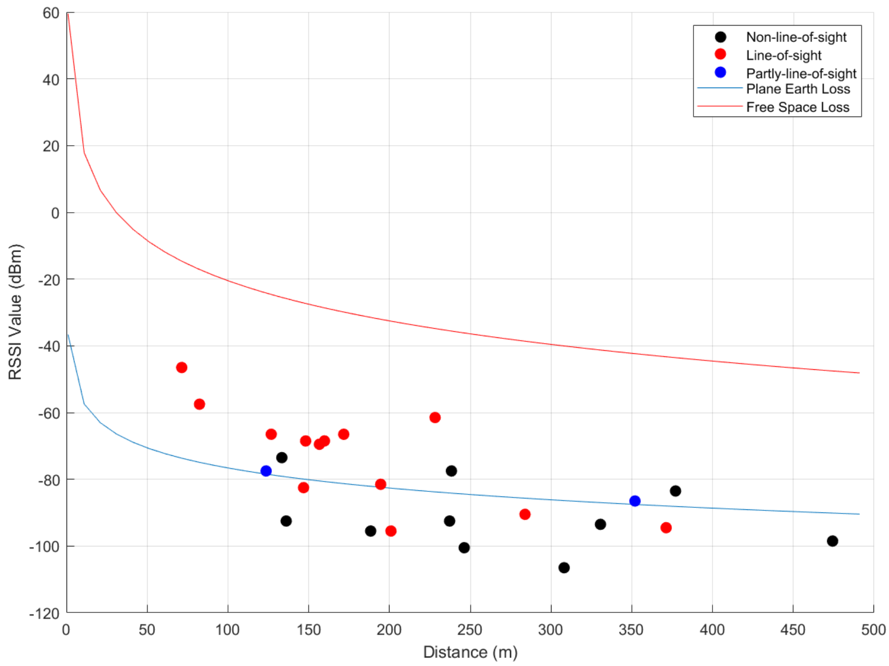


Figure 4.9: RSSI values measured and the expected signal strength for the FSL and PEL propagation model found in the link budget

The distance between transmitter and receiver during the measurements were found using the Haversine formula [29], which calculates the great-circle distance between two points on a sphere given the two points latitude and longitude, with high accuracy. The Haversine formula is given in Equation 4.6 and Equation 4.7 shows the Haversine function. The distance between the two points is d, the radius of the sphere is r, and $\varphi_{1,2}$ and $\lambda_{1,2}$ are latitude and longitude respectively for point one and two given in radians.

$$hav\frac{d}{r} = hav(\varphi_2 - \varphi_1) + cos(\varphi_1)cos(\varphi_2)hav(\lambda_2 - \lambda_1) \tag{4.6}$$

$$hav(\theta) = sin^2(\frac{\theta}{2}) \tag{4.7}$$

By rearranging Equation 4.6 the distance between two points can be calculated, using Equation 4.8:

$$d = 2 \times r \times arcsin\left(\sqrt{sin^2(\frac{\varphi_2 - \varphi_1}{2}) + cos(\varphi_1)cos(\varphi_2)sin^2(\frac{\lambda_2 - \lambda_1}{2})}\right) \quad (4.8)$$

Table 4.5 lists each measurement number in reference to Figure 34.8 The table lists RSSI value, PER, if there were free space between transmitter and receiver, and the distance between the transmitter and the receiver. All the coordinates used in the distance calculation is found in Table B.1 in Appendix B.1, the radius used for Earths sphere is 6371000 meters.

Table 4.5: Results of RSSI measurement

# [a]	RSSI [b]	PER[c]	FS [d]	d [1]	#	RSSI	PER	FS	d
1	-43	0	Yes	82.40	14	-92	45	No	308.13
2	-32	0	Yes	71.41	15	-84	3.2	No	474.29
3	-47	0	Yes	228.22	16	-69	0	No	377.08
4	-52	0	Yes	126.79	17	-79	0.2	No	330.60
5	-59	0	No	133.38	18	-63	0.1	No	238.37
6	-81	10	No	188.37	19	-54	0	Yes	159.76
7	-78	0.1	No	237.23	20	-63	0	Semi	123.64
8	-78	1	No	136.07	21	-68	0	Yes	146.83
9	-52	0	Yes	171.69	22	-67	0	Yes	194.56
10	-55	0	Yes	156.64	23	-76	0	Yes	283.90
11	-54	0	Yes	148.12	24	-72	0	Semi	351.98
12	-81	0.3	Semi	200.91	25	-80	0	Yes	371.28
13	-86	5	No	246.28					

[a]#: Measurement number
[b]RSSI (dBm)
[c]PER [%]
[d]FS: Free space between locations
[d]d: Distance between locations [m]

From looking at Figure 4.7 and Table 4.5 it is clear that for the most part, the RSSI value decreases when the distance between the transmitter and the receiver increases. This decrease in RSSI value is what one would expect. Three of the measurements do have a lower RSSI value after the offset has been corrected then the lower limit of −98 dBm which is the lowest RSSI value the MCU can measure according to specifications in the datasheet [15]. Measurement number three stick out with a good RSSI value for its distance compared to the other line-of-sight measurements. In measurement twelve the transmitter is located inside a tunnel beneath the road with the line-of-sight partly blocked due to a small stone wall. If comparing measurement twelve to measurement six, which is not in line-of-sight, one sees that their RSSI values are the same even though measurement twelve only is partly in line-of-sight. It could be assumed that this blocking and the location had an impact on the propagating waves and therefore the RSSI value.

4.2.4 Summary of Message Format

Section 3.3.3 proposed improving the message format from the specialisation project, by altering some of the data fields and finding a suitable modulation scheme by conducting an analysis. Altering the data fields reduced the message size from 416 bits to 360 bits. In the analysis for the modulation scheme, the first part investigated how several configurations for each modulation schemes spectrum behaved when transmitting a random generated payload. All configurations of OOK and 4-FSK are discarded from further analysis due to high OBW. The best configurations for FSK, MSK, and OQPSK are further analysed with a packet error test.

The packet error test calculates the percentage of incorrectly received data packets. For simulating the distance d which the link budget from Table 3.13 in Section 3.3.3 is calculated with, an 80 dB attenuation are connected between the transmitter and receiver and transmit power is reduced by 2.3 dBm. In the link budget BER $= 10^{-4}$ was used, this corresponds to one bit error per 10 000 bit transmitted. In each of the PER measurements, 50 000 packets were transmitted, each containing 360 bits giving a total number of 18 million transferred bits. OQPSK and MSK both have the most promising spectrums but their PER are alarmingly high ranging from 65.9% to 82.1%. Based on these findings, various configurations, bitrates, and attenuations are further thoroughly tested for OQPSK and MSK in collaboration with customer support from the manufacturer. These configurations did not have any impact on the PER. The manufacturer has stated that this problem may require a long investigation [26]. The FSK configuration, on the other hand, has no packet loss during the PER test, despite having an undesired spike located in the middle of the spectrum. This undesired spike also appears in spectrums measured from built-in radio configurations that use various bitrates, filters, and deviation. Since there is no packet loss for the FSK configuration the spike does not appear to be harmful. The spike could be a result of a carriage leakage.

Based on the findings made in the modulation analysis, modulation number two from Table 4.1, the GFSK with a 14 kHz deviation and a 36.8 kbps bitrate is chosen for this project. The Gaussian filter was preferred over the raised cosine filter due to the shape of the spike located in the right side of the spectrum. The spike appears almost flat compared to the spike on the left side of the spectrum when the raised cosine filter is applied. When the Gaussian filter is used, one can clearly see the spike located on the right side of the spectrum. During the packet error test, an offset was found in the RSSI value provided by the MCU compared to the received signal strength measured at the signal analyser. This offset requires a calibration of the RSSI values measured outdoor in Trondheim harbour. Since a modulation now is selected, the RF link have to operate within the proposed frequency regulation from Section 3.3.3, that is regulation §8.7 in Table 4.3. Calculations and measurements confirm that this is the case, and the selected modulation is operating within regulation §8.7.

In the calibration of RSSI values, the same test bench setup used in the packet error test is set up. By gradually increasing attenuation between the transmitter and receiver, a 14.5 dBm average offset is detected between RSSI values from the MCU and received signal strength from the signal analyser. The discovery of this offset confirms information provided from one of the accompanying application manuals for the radio board, which states that an offset exists [27]. The outdoor measurement is conducted in Trondheim harbour where the ferry will be in traffic. Twenty-five measurements are performed over different distances and visibility between the VHF transmitter and the VHF receiver. The antennas used in this measurement are not the Scan Antenna VHF73X which is intended to be used on the ferry. The result of the RSSI measurement is that the coverage in the vicinity around the channel between Ravnkloa and Vestre Kanalhavn is quite good.

4.3 VHF Receiver to Chartplotter

The third objective of this thesis is to design and implement a data interface from the VHF receiver to a chartplotter using NMEA 0183 format. Section 3.4 proposes that the interface between the VHF receiver and the chartplotter also should use UART, but with a configuration suitable for the specifications on the chartplotter.

A Simrad NSE12 chartplotter is used for testing the data interface from the VHF receiver. The NSE12 has one NMEA 0183 port which uses the

RS-422 protocol. This port can output one NMEA 0183 sentence for listening devices, and receive one NMEA 0183 sentence talking devices. The chartplotters datasheet contains details about the wiring of the NMEA 0183 communication protocol, a configuration for the serial port, and NMEA 0183 sentences supported by the chartplotter. One of the supported NMEA 0183 sentences are the RMC sentence that is proposed to use in Section 3.4.

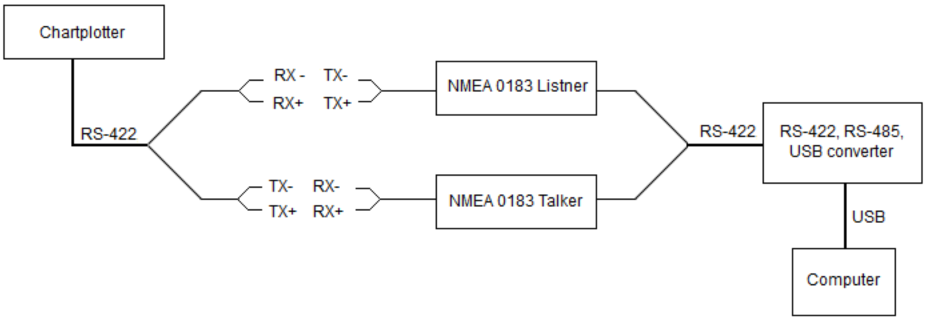


Figure 4.10: Block diagram of implemented data interface from the VHF receiver to Simrad NSE12

Table 4.6: NMEA 0183 wiring

Pin	Colour	Description
8	Orange	RS-422 RX-
7	Green	RS-422 RX+
6	Yellow	RS-422 TX+
5	Blue	RS-422 TX-

The wiring of interface is done accordingly to the datasheets of the Simrad NSE12 [30] and the Exsys Ex-1303 USB to RS-422/485 converter [31], as shown in Figure 4.10, with the pin numbering found in Table 4.6. A simulator named NemaStudio [32] were used for simulating NMEA 0183 sentences. NemaStudio used the same serial configuration proposed for the UART interface between the MCU and the chartplotter. To test that the connection and configuration for the data interface between the VHF receiver and the chartplotter works, NemaStudio is used to send simulated messages with positions from Trondheim harbour. When NemaStudio starts transmitting RMC sentences these then should be printed on the chartplotter. However, when the transmission started no RMC sentences was printed on the chartplotter, which was unexpected.

A further look at the interface wiring is done with the use of an oscilloscope. The *NMEA 0183 Talker* wire is measured when NemaStudio transmits RMC sentences and multiple clear pulse waves appear on the spectrum. The *NMEA 0183 Talker* wire is measured when the chartplotter transmits RMC sentences. A serial session is opened PuTTY for printing of the received RMC sentences from the chartplotter. The RMC sentences are correctly printed with the same structure as the sentences transmitted from NemaStudio. Multiple clear pulse waves appear on the oscilloscopes spectrum simultaneously as the sentences are printed in PuTTY. The pulse waves measured at both wires appears very similar, which they should since they transmit the same RMC sentence. However, the messages sent to the chartplotter do not appear on the screen.

Since the RMC sentence transmitted from the chartplotter is correctly printed and the wires connected to the chartplotter have distinct pulse waves measured, the data interface from the VHF receiver to the chartplotter works. However, the messages received by the chartplotter does not appear on the screen. There might be other settings in the chartplotter that requires configuration so that information from the RMC sentences appear on the screen.

Chapter 5

Conclusion

This thesis set out to develop a communication system for an autonomous passenger ferry in order to reduce the risk of collision. The communication system consists of a data interface from the on-board computer located on the ferry to a VHF receiver, the design of a compact message which transmits according to regulation §8.7, and a data interface from the VHF receiver to a chartplotter.

Communication between the on-board computer and the VHF transmitter was implemented with a UART interface, one of the serial communication protocols supported by the MCU. Several tests were run successfully on the interface with good results. A new message format is created and transmitted according to frequency regulation on 169.4 VHF band, using GFSK modulation with a 14 kHz deviation and a 36.8 kbps bitrate. This modulation was selected based on an analysis conducted on each of the available modulation schemes. The analysis discarded OOK and 4-FSK first due to their OBW exceeding the frequency regulation. Although OQPSK and MSK has the lowest OBW, these configurations also were discarded due to a high packet loss, which ranged from 65.9% to 82.1%. In talks with the manufacturer, the manufacturer recreated the same tests and received the same result in packet loss. The manufacturer stated that the problem with packet loss for OQPSK and MSK require a long investigation [26]. Both a Gaussian and raised cosine filter could have been used but the Gaussian filter was selected based on the spectrum it provided. The RF coverage in the ferry's area of operation is based on RSSI measurements that provided good results for the coverage in the vicinity of the channel between Ravnkloa and Vestre Kanalhavn. A link budget for the system which includes noise figure, antenna gains, and uses the plane earth path loss model is calculated. The resulting $\frac{Eb}{N0}$ value from the link budget is compared to the $\frac{Eb}{N0}$ value required for BER$= 10^{-4}$, and the link margin for the system is found to be 20.66 dB.

Similar to the interface between the on-board computer and VHF transmitter, the interface between VHF receiver and chartplotter is implemented with a UART interface. Testing and measurements performed can conclude that the interface works as intended. However, the messages transmitted to the chartplotter does not appear on the screen.

5.1 Further Work

As stated in the summary Section 4.1.3 of the first objective, a Python script which gathers the navigational information and combines the data to the proposed payload string needs to be running on the on-board computer. The script would best be written by either the developer responsible for the layout of the ROS system used by the ferry or developers with thorough knowledge on other ROS systems. The localisation of desired information is much easier for developers with a larger knowledge of a given system. For the second objective, further collaboration with the manufacturers support team, regarding the undesired spike found in the centre of the spectrum for the chosen GFSK modulation should be further work. Although the spike did not affect the receiving, it occupies unnecessary bandwidth. In the third objective, the received RMC sentences do not appear on the screen. It may be that some settings on the chartplotter are properly configured, and the chartplotters settings could be looked at further.

Bibliography

[1] Norvald Kjerstad. *Elektroniske og akustiske navigasjonssystemer*. Tapir Akademisk Forlag, 2008. ISBN 978-82-519-2288-3.

[2] Andreas Fjellhaugen. Kommunikasjonssystem for autonom ferge. 2017. Specialisation project, Department of Electronic Systems.

[3] Claude Shannon. A mathematical theory of communication. 1948. URL `http://math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf`.

[4] Simon Haykin and Michael Moher. *Communication Systems*. Wiley, 2010. ISBN 978-81-265-2151-7.

[5] Andrew S. Tanenbaum and David J. Wetherall. *Communications Standard Dictionary*. Pearson, 2011. ISBN 978-0-13-212695-3.

[6] Simon R. Saunders. *Antennas and Propagation for Wireless Communication Systems*. Wiley, 1999. ISBN 0-471-98609-7.

[7] Theodore S. Rappaport. *Wireless Communication Principles and Practice*. Pearson, 2002. ISBN 0-13-042232-0.

[8] Giorgio Rizzoni. *Principles and Applications of Electrical Engineering*. McGraw-Hill, 2004. ISBN 0-07-288771-0.

[9] David M. Pozar. *Microwave and RF Wireless Systems*. Wiley, 2001. ISBN 978-0-471-32282-5.

[10] IEEE, 2012. URL `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6178212`. IEEE 802.11-2012 standard.

[11] Nasjonal kommunikasjonsmyndighet, 2016. URL `https://lovdata.no/dokument/SF/forskrift/2012-01-19-77`. Forskrift om generelle tillatelser til bruk av frekvenser (fribruksforskriften), Regulation of general permissions for use of frequencies (Free use regulation).

[12] SOLAS, 2002. URL http://solasv.mcga.gov.uk/regulations/regulation19.htm. International Convention for the Safety of Life at Sea (SOLAS V).

[13] International Telecommunication Union (ITU). Recommendation itu-r m.1371-5 (02/2014), 2014. URL http://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1371-5-201402-I!!PDF-E.pdf. Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band.

[14] Silicon Labs, 2017. URL https://www.silabs.com/support/getting-started/proprietary-wireless/flex-gecko. Flex Gecko product overview.

[15] Silicon Labs, 2017. URL https://www.silabs.com/documents/public/data-sheets/efr32fg1-datasheet.pdf. Flex Gecko datasheet.

[16] SCAN Antenna A/S, 2017. URL http://www.scan-antenna.com/product/vhf73x. SCAN Antenna VHF73X Datasheet.

[17] Silicon Labs, 2017. URL https://www.silabs.com/documents/public/reference-manuals/EFR32xG1-ReferenceManual.pdf. Flex Gecko Reference Manual.

[18] Silicon Labs, 2013. URL https://www.silabs.com/documents/public/application-notes/AN0045.pdf. USART/UART - Asynchronous mode - AN0045 - Application Note.

[19] Silicon Labs, 2017. URL https://www.silabs.com/documents/public/application-notes/an971-efr32-radio-configurator-guide.pdf. Flex Gecko radio configuration.

[20] ISO, 2004. URL https://www.standard.no/nettbutikk/produktkatalogen/produktpresentasjon/?ProductID=158392. NS-ISO 8601:2004. Data elements and interchange formats - Information interchange - Representation of dates and times.

[21] ITU, 2015. URL https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.585-7-201503-I!!PDF-E.pdf. Recommendation ITU-R M.585-7 (03/2015), Assignment and use of identities in the maritime mobile service.

[22] European Telecommunications Standards Institute (ETSI). http://www.etsi.org/deliver/etsi_en/300200_300299/30022001/02.04.01_40/en_30022001v020401o.pdf, 2012. ETSI EN 300 220 standard.

[23] url=http://www.tronico.fi/OH6NT/docs/NMEA0183.pdf Klaus Betke, year=2011. The NMEA 0183 Protocol.

[24] Statens Kartverk Sjø. *Den Norske Los - Bind 1*. Gunnarshaug Trykkeri AS Stavanger, 2010. ISBN 978-82-90-65326-7.

[25] Rohde-Schwarz, 2017. URL https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_manuals/gb_1/f/fsv_1/FSVA_FSV_UserManual_en_10.pdf. Rohde-Schwarz FSVA/FSV Signal and Spectrum Analyzer - Operating Manual.

[26] Ferenc Plesznik, 2018. Silicon Labs employee (personal communications), contacted for support from Silicon Labs regarding the major packet loss found in Section 4.2.1.

[27] Silicon Labs, 2017. URL https://www.silabs.com/documents/public/user-guides/UG147-Flex-Gecko-Range-Test-Demo.pdf. Flex Gecko Range Application Manual.

[28] Tor-Inge Kvaksrud, 2018. Senior engineer at NKOM (personal communications), provided a confirmation on terms used in the Free use regulation in Section 2.3.3.

[29] C. C. Robusto. The cosine-haversine formula, 2017. URL https://www.jstor.org/stable/pdf/2309088.pdf. Mathematical Association of America, Last visited 07.12.17.

[30] Simrad, 2017. URL http://ww2.simrad-yachting.com/en-GB/Products/Discontinued-Products/NSE12-Chartplotter-en-gb.aspx. Simrad NSE12 Chartplotter Installation Manual.

[31] Exsys, 2018. URL https://www.exsys.ch/media/files_public/edrjucqmvme/ex-1303.pdf. EX-1303 USB 1.1 to Serial RS-422/485 cable datasheet.

[32] Sailsoft, 2016. URL http://www.sailsoft.nl/nemastudio/usermanual/nemastudio_userguide.pdf. NemaStudio User Manual.

Appendix A

Project description

A.1 English

Communication system for a autonomous passenger ferry

NTNU is currently developing a prototype autonomous passenger ferry which will be used to carry passengers and bicyclists across channels in Trondheim harbour. The ferry will navigate and operate without crew, and the main focus will be to demonstrate passenger safety when it is operated in a congested area with other boat traffic present.

To reduce the risk of collision to a minimum, the ferry should broadcast its position, heading and speed to other boats at a short time interval using VHF radio or similar. Traditionally this is done using the Automatic Identification System (AIS) which broadcast this kind of information over the maritime VHF band. However, the operation of the ferry requires update rates of typically 1 sec, which the AIS-system can not provide.

Therefore, we intend to use the licence-free 169.4 MHz VHF band and broadcast short messages from the ferry every second. The VHF link has been successfully tested and verified in a specialisation project. The master's project will therefore contain the following tasks:

- Implement a USB data interface from the computer on board the ferry to the VHF transmitter.
- Design a compact message format that allows data to be transferred according to the frequency regulations for the VHF band.
- Design and implement a data interface from the VHF receiver to a chartplotter using NMEA 0183 format.

A.2 Norsk

Kommunikasjonssystem for en autonom passasjerferge

NTNU er i gang med å utvikle en prototype for en autonom passasjerferge som skal brukes til å frakte fotgjengere og syklister over kanalen i Trondheim havn. Fergen vil navigere og være i operasjon uten mannskap, og hovedfokus vil være å demonstrere passasjerenes sikkerhet når fergen opererer i ett tett trafikkert område med andre fartøy tilstede.

For å redusere risiko for kollisjoner til et minimum, må fergen kringkaste sin posisjon, kurs og fart til andre fartøy i hyppige intervall ved bruk av en VHF radio eller lignende. Tradisjonelt sett er dette løst ved å bruke det Automatiske Identifikasjonssystemet (AIS) som sender slik informasjon over maritimt VHF bånd. Grunnet hyppige sendeintervall, typisk 1 sekund, vil ikke AIS systemet være tilstrekkelig.

Med bakgrunn av dette vil det være nødvendig å utvikle et kommunikasjonssystem som skal sende på 169.4 MHz VHF bånd, i henhold til fribruksforskriften, for å kunne sende korte meldinger hvert sekund fra fergen. VHF linken har allerede blitt testet og verifisert i ett fordypningsporsjekt. Master prosjektet vil derfor inneholde følgende oppgaver:

- Implementere USB data grensesnitt fra en datamaskin om bord fergen til VHF-sender.
- Design ett kompakt meldingsformat som gjør det mulig å sende data i henhold til fribruksforskriften.
- Design og implementer et datagrensesnitt fra VHF-mottakeren til en kartplotter med NMEA0183-format.

Appendix B

Measurement data

B.1 Positions of RSSI measurements

Table B.1: Coordinates for RSSI measurement

	Transmitter		Receiver	
#	Latitude [N]	Longitude [E]	Latitude [N]	Longitude [E]
1	63.435881	10.399436	63.43514	10.39946
2	63.435881	10.399436	63.435875	10.398000
3	63.435881	10.399436	63.434842	10.395478
4	63.435706	10.397142	63.434842	10.395478
5	63.435843	10.396956	63.434842	10.395478
6	63.435843	10.396956	63.434356	10.395141
7	63.435843	10.396956	63.433986	10.393535
8	63.435385	10.395016	63.433986	10.393535
9	63.435385	10.393289	63.433842	10.393162
10	63.435250	10.393265	63.433842	10.393162
11	63.435163	10.392778	63.433842	10.393162
12	63.43563	10.39258	63.433842	10.393162
13	63.436028	10.392365	63.433842	10.393162
14	63.436576	10.394173	63.433842	10.393162
15	63.435184	10.392830	63.430981	10.394456
16	63.435184	10.392830	63.431835	10.394023
17	63.435184	10.392830	63.432280	10.394256
18	63.435184	10.392830	63.433099	10.393945
19	63.435184	10.392830	63.433782	10.393533
20	63.435184	10.392830	63.434755	10.395124
21	63.435184	10.392830	63.434911	10.395719
22	63.435184	10.392830	63.434992	10.396719
23	63.435184	10.392830	63.435086	10.398535
24	63.435184	10.392830	63.435159	10.399908
25	63.435184	10.392830	63.435645	10.400225

Appendix C

Serial Communication Appendix

C.1 UART Settings

Three functions are written for the UART serial communication. The code listed under in Code C.1 is used to implement the UART serial communication in both MCUs.

```
#include "uartdrv.h"
#include "em_usart.h"

#define BUFFERSIZE          128

// Local variables
static UARTDRV_HandleData_t uartHandle0; /* UART driver handle
    */
static UARTDRV_Handle_t  testHandle0 = &uartHandle0;
static uint8_t rxByte;

// Structures
volatile struct circularBuffer {
  uint8_t data[BUFFERSIZE]; /* data buffer */
  uint32_t rdI; /* read index */
  uint32_t wrI; /* write index */
  uint32_t pendingBytes; /* count of how many bytes are not yet
    handled */
  bool overflow; /* buffer overflow indicator */
}
rxBuf, txBuf = { { 10 }, 0, 0, 10, false };

// Function prototypes
void UART_Init(void);
```

```
static void UART_tx_callback(UARTDRV_Handle_t handle, Ecode_t transferStatus, uint8_t *data,
                             UARTDRV_Count_t transferCount);
static void UART_rx_callback(UARTDRV_Handle_t handle, Ecode_t transferStatus, uint8_t *data,
                             UARTDRV_Count_t transferCount);

/*****************************************************
* UART functions
*****************************************************/
/*****************************************************
* This function called when the UART TX operation completed
*****************************************************/
static void UART_tx_callback(UARTDRV_Handle_t handle, Ecode_t transferStatus, uint8_t *data,
                             UARTDRV_Count_t transferCount)
{
  static uint8_t txCnt = 0;

  if (transferStatus == ECODE_EMDRV_UARTDRV_OK)
  {
    txCnt++;
  }
}


/*****************************************************
* This function called when the UART RX operation completed
*****************************************************/
static void UART_rx_callback(UARTDRV_Handle_t handle, Ecode_t transferStatus, uint8_t *data,
                             UARTDRV_Count_t transferCount)
{

  if(transferStatus == ECODE_EMDRV_UARTDRV_OK)
  {
    uint8_t rxData = *data;
    rxBuf.data[rxBuf.wrI] = rxData;
    rxBuf.wrI = (rxBuf.wrI + 1) % BUFFERSIZE;
    rxBuf.pendingBytes++;

    /* Flag Rx overflow */
    if (rxBuf.pendingBytes > BUFFERSIZE) {
      rxBuf.overflow = true;
    }
  }
  /* RX the next byte */
  UARTDRV_Receive(testHandle0, &rxByte, 1, UART_rx_callback);

```

```c
}

/*****************************************************
* UART Initialization
*****************************************************/
void UART_Init(void)
{
  /* Enable VCOM */
  GPIO_PinModeSet(BSP_BCC_ENABLE_PORT, BSP_BCC_ENABLE_PIN, gpioModePushPull, 1);
  GPIO_PinOutSet(BSP_BCC_ENABLE_PORT, BSP_BCC_ENABLE_PIN);

  /* uart init */
  UARTDRV_Init_t initData;

  DEFINE_BUF_QUEUE(EMDRV_UARTDRV_MAX_CONCURRENT_RX_BUFS, rxBufferQueueI0);
  DEFINE_BUF_QUEUE(EMDRV_UARTDRV_MAX_CONCURRENT_TX_BUFS, txBufferQueueI0);


  /* UART init */
  initData.port                 = USART0;
  initData.baudRate             = 115200;
  initData.portLocationTx       = USART_ROUTELOC0_RXLOC_LOC0;
  initData.portLocationRx       = USART_ROUTELOC0_RXLOC_LOC0;
  initData.stopBits             = (USART_Stopbits_TypeDef) USART_FRAME_STOPBITS_ONE;
  initData.parity               = (USART_Parity_TypeDef) USART_FRAME_PARITY_NONE;
  initData.oversampling         = (USART_OVS_TypeDef) USART_CTRL_OVS_X16;
  initData.mvdis                = false;
  initData.fcType               = uartdrvFlowControlNone;
  initData.rxQueue              = (UARTDRV_Buffer_FifoQueue_t *) &rxBufferQueueI0;
  initData.txQueue              = (UARTDRV_Buffer_FifoQueue_t *) &txBufferQueueI0;

  UARTDRV_Init(testHandle0, &initData);
  UARTDRV_Receive(testHandle0, &rxByte, 1, UART_rx_callback);
}
```

Code C.1: UART configuration

C.2 Python script for serial communication

The Python script used for testing the serial interface between a PC and the MCU in Section 4.1.1. Uses the time and PySerial library and is written in Python 3.4.

```
import time
import serial
s = serial.Serial(
    port = "COM4",
    baudrate = 115200,
    parity = serial.PARITY_NONE,
    bytesize = serial.EIGHTBITS,
    stopbits = serial.STOPBITS_ONE
) # Configured as the proposal in Table \ref{tab:uart_cfg}
  suggested

while True:
    time.sleep(0.5)
    # s.write transmits the hard coded payload
    s.write(b'llllyyyyyhhmmssooxxxvvvAxxxxxxxxx')
    time.sleep(0.5)
```

Code C.2: Python script for testing serial communication

C.3 PuTTY configuration for Windows

1. Download and install the PuTTY terminal emulator
2. Open PuTTY and select Serial under the Connection category
   (a) Configure the serial line with the same values as Figure C.1
   (b) Enter the COM# of the MCU, such as COM3
   (c) Click Open

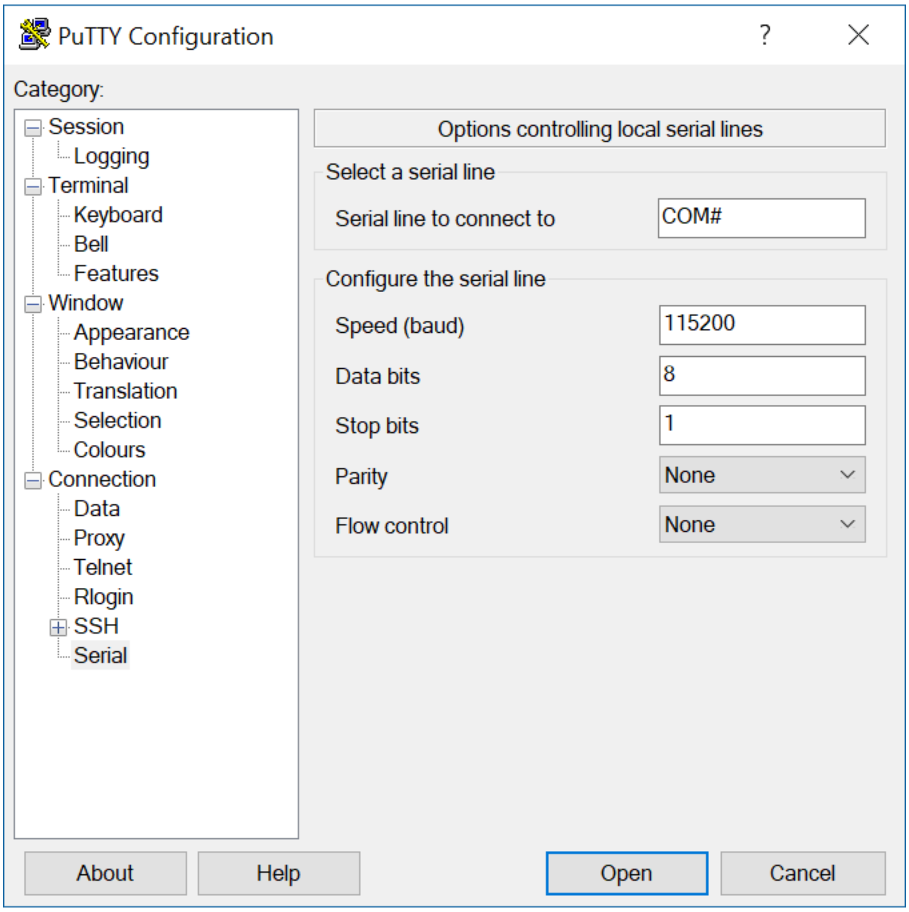


Figure C.1: PuTTY configuration