A Chalmers University of Technology Bachelor's thesis.

Construction of an IoT-device transmitting data of the endangered Atlantic salmon via Sigfox-LPWAN

Bachelor's thesis in Electrical engineering

MATTIAS BLINGE

Department of Communication and Antenna Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Bachelor's thesis 2018:08

Construction of an IoT-device transmitting data of the endangered Atlantic salmon via Sigfox-LPWAN

MATTIAS BLINGE

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2019

Construction of an IoT-device transmitting data of the endangered Atlantic salmon via Sigfox-LPWAN
Mattias Blinge

IoT-Consult: Mikael Faklvidd, IoT-Sweden
Supervisor: Arni Alfredsson, Communication and Antenna Systems
Examiner: Tommy Svensson, Communication and Antenna Systems

Bachalor's Thesis 2019:02
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46(0)31 772 10 00

Gothenburg, Sweden 2019

Abstract

This thesis aims to present the construction of an Internet of Things (IoT) device utilizing Sigfox-Low-Power Wide-Area Network (LPWAN), in order to transmit data received from a tracking unit regarding Atlantic salmon in Göta river. This project was performed in collaboration with the County administrative board of Gothenburg, involved in the project "The Missing Salmon Project" and IoT-Sweden, responsible for the establishment of Sigfox-LPWAN in Sweden. The Missing Salmon project aims to provide evidence of why the Atlantic Salmon population has rapidly decreased.
The tracking unit utilizes Recommended Standard 485 (RS485) communication when transmitting the data by wire to a microprocessor programmed to process the data. The information is then wirelessly transmitted by a Sigfox modem to Sigfox-backend where the data is stored. The device was tested at three locations along the stream with different characteristics of coverage on the Sigfox service map. Furthermore, a minor comparative study of Sigfox LPWAN and LoRaWAN, (Long-Range Wide-Area Network), was made.

An issue was encountered when larger amounts of data was to be transmitted by wire from the tracking unit. This problem made the device unable to access sleep mode between the transmission which forced the device digital construction to be adjusted from what was originally thought. The main consequence of this deviation was that the power consumption greatly increased.
This difficulty was encountered at a critical time of the project and not enough time was available to produce a solution. Still the device was able to process and transmit data packages via Sigfox-LPWAN received from the tracking unit.

Three locations were tested to investigate how well the device perform due to different characteristics on the service map. The highest success rate was measured to 80% at the outlet of Göta River where the device was placed beneath a wharf in order to avoid attention to the public. This placement of the device might have affected the success rate negatively.

For this type of project where sensory data is transmitted, the limitations of Sigfox LWPAN do not interfere with the results. However, today only major cities including the surrounding area of Sweden has enough coverage to present Sigfox as an alternative network to GSM.

Keywords: LPWAN, IOT, Sigfox, LoRa, RS485, Serial Communication.

Acknowledgements

This BSc thesis was prepared at Chalmers University of Technolog in collaboration with IoT-Sweden and County administrative board of Gothenburg in fulfillment of the requirements for acquiring a BSc degree in Electrical Engineering. I am grateful to my supervisor Arni Alfredsson at Chalmers University of Technology for providing feedback and my examiner Tommy Svensson for the motivational meetings.
I would also like to thank Mikael Falkvidd at IoT-Sweden for the great support and expressing great interest in the project.
In addition I would like to express gratitude to Mikael Cremle and Mikael Ljung at County administrative board helping out with logistics and making me aware of the situation concerning the Atlantic salmon and many other species.

Mattias Blinge, Gothenburg, February 2019

Contents

1

Glossaries

- *CSS* - Chirp Spread Spectrum.
- *D-BPSK* - Differential Binary Phase-Shift Keying.
- *Downlink Message* - a message from an Application to a Device.
- *GSM* - Global System for Mobile communication.
- *IoT* - Internet of Things.
- *ISM band* - Industrial, Scientific and Medical radio bands.
- *LoRaWAN* - Communication protocol based on the LoRa technology.
- *RF* - Radio Frequency.
- *RSSI* - Receiver Signal Strength Indicator.
- *Sigfox* - Developers of Sigfox LPWAN.
- *UNB* - Ultra Narrow Band.
- *SNR* - Signal-To-Noise Ratio.
- *Uplink Message* - a message from a Device to an Application.
- *WiFi* - Wireless local area network.

2
Introduction

One of the most central topics of the world today is the growth of environmental disasters as a result of the rapid progression of the human species. The extensive use of rare earth metals and chemicals results in devastating effects on nature and biodiversity. Comparing massive amount of data is key for researchers struggling to locate the origin of why species that have been around for millions of years suddenly become endangered. By merging IoT and sustainability, massive collection of data will help the research to further decrease threats against biodiversity.
This report describes the construction and testing of an IoT prototype able to provide real time data feed of the information gathered by a tracking device used to monitor Atlantic salmon.
The Atlantic salmon has decreased by 70% over the last 25 years and this is a species that has existed for more than 60 million years [19]. The cause of the decline is unknown and as an attempt to prevent further decrease "The Missing Salmon Project" was launched with the aim to monitor the salmons movement from the river to the sea [19]. This will provide data of where the fish is lost and hopefully pin-point the causes of why the decline is occurring.
The tracking devices used when monitoring the fish movement are placed along a stream where the fish will make their way to the sea. However, today when extracting the information about the fish movement during the actual time of emigration the tracking devices need to be accessed manually. By knowing the exact time the salmon starts its journey towards the sea it will be easier to put constructions along the stream on hold that might have negative consequences on salmon population and thereby decrease potential disturbances.
This project is executed in collaboration with University of Gothenburg, Länsstyrelsen and IoT Sweden. Göteborgs Universitet is responsible for Swedens participation in "The Missing salmon project" with Prof. Johan Höjesjö as project manager and coordinator. Länsstyrelsen provides the equipment and performs installation of the tracking devices along Göta Älv. IoT Sweden is a Swedish company managing the implementation of Sigfox LPWAN in Sweden [7].

2.1 Delimitation

The project was limited to the construction of an IoT prototype utilizing Sigfox LPWAN. Therefore, no practical testing concerning LoRaWAN was conducted.
The purpose of the project was to construct a working prototype by utilizing a Sigfox modem enabling Sigfox LPWAN. Moreover, the project focused on describing the communication standards used in the construction of the device, thus to provide a comparative study of the two most widely established LPWAN which are Sigfox LPWAN and LoRaWAN.

3
Communication Standards

This chapter describes information gathered before executing the design and construction of the prototype.

3.1 LPWAN and IoT

IoT describes a network between endpoint devices, gathering mass scale physical data. In terms of digital revolution, IoT has been headlining with its marketing on smarter solutions and providing for a more sustainable future. Many opportunities arose with the term LPWAN in 2013 but it wasn't until recently that the size of the technical phenomena surfaced. In aspect of technical solutions to the current environmental problem, LPWANs popularity is increasing exponentially due to its low cost and energy efficiency compared to other communication technologies. Some battery powered devices transmitting via LPWAN are able to function for 10 years [8]. By 2020 it's estimated that there will be over 50 billion IoT-devices interlinked and transferring data to each other which is the reason of many companies competing of becoming the leading provider of the communication network [11].
LPWAN are radio based communication networks constructed to supply connectivity for mass scale IoT devices transferring information with low bit rate. Typical areas of usage include security, agriculture, parking, animal tracking, etc [11]. Bit rate and range of coverage depend on the specific technology used when designing the LPWAN. For example, alarm systems work better if the system is fast with a reliable transmission in relation to a system that is more energy efficient but has a higher failure rate. Therefore many different LPWAN types have been designed to cover the different needs.
The frequency of which LPWAN operates at is either licensed or unlicensed Industrial, Scientific and Medical (ISM),-band. Licensed frequency bands are purchased and give the buyer full control over the specific frequency spectrum. Unlicensed ISM bands are publicly free for use to whom it may be of interest. In Europe, LPWAN systems using the unlicensed ISM bands operate at the frequency spectrum 863 MHz -to 870 MHz whilst in the USA LPWAN utilizes the frequency span 902-928 MHz [14]. However, using the unlicensed spectrum comes with limitations of the packet size and the amount of times per hour a device is able to send. The limitations are a vital part in keeping consistent order and maintaining a functioning frequency band with low interference. The laws concerning the unlicensed ISM-band in Europe restrict the transmission to a duty cycle of 1% [6].

Due to the rapid expansion in interest of IoT devices many companies develop LPWAN with different modulation and characteristics to compete for the role as main supplier of network for these devices. Among these competitors the most developed are Sigfox and LoRaWAN which will be compared in this report to find the best alternative to this project.

3.1.1 SigFox LPWAN

Sigfox provides their own patented radio technology to an LPWAN providing an end-to-end IoT solution. Sigfox business model focuses on providing an LPWAN which is affordable and easy to implement. Sigfox constructs its LPWAN using radio base stations providing endpoint devices with coverage. The network allows bidirectional communication with the capacity to transmit uplink message of a maximum 12 bytes and downlink message of 8 bytes. However, only 4 downlink messages are allowed to be sent each day compared to 140 messages uplink. This is due to that Sigfox utilizes the unlicensed ISM-bands which comes with limitations. The amount of uplink messages transmitted per day is restricted to 140 per day because of the 1% duty cycle regulation by the European union [6]. 1% of one hour is equal to 36 seconds which is the amount of time a Sigfox modem legally is able transmit data. One message takes 6 seconds to transmit, therefore the modem is able to transmit a total of 6 messages per hour. This results in 144 messages per day, however Sigfox utilizes 4 of these for downlink, limiting the total amount of uplink messages to 140 per day [15]. By implementing the LPWAN using the unlicensed spectrum, large fees for private frequencies are avoided and therefore subscription plans for a reasonable amount is achieved.
As Sigfox prioritize long range communication and energy efficiency towards the endpoint devices, Sigfox utilizes the radio technology ultra narrow band, (UNB), in combination with differential binary phase-shift key, (D-BPSK), modulation which enables the coverage from one base station to reach 50km in rural areas [11]. This technology only utilizes 100 Hz of the operation band when transmitting a message and is therefore able to transmit long distances with low interference [17], [18].

3.1.2 LoRaWAN and LoRa

Like Sigfox Long Range (LoRa), is categorized as LPWAN and is designed to connect low power, battery operated devices to the internet. LoRa is the modulation technique used to generate the physical layer connecting the end-devices to the gateways whilst LoRaWAN is the data communication protocol developed by Semtech to enable low cost and battery efficient devices [2]. Key IoT specifications such as two-way communication, mobility and tracking services are important factors for the developers making it interesting and attractive to potential users [1]. LoRaWAN operates in the unlicensed ISM-spectrum and is therefore affected by the international laws regulating the activity on the frequency band explained in section 3.1. The network is deployed by integration of gateways interfacing the LoRaWAN protocol to the internet. Data sent from the end-point devices are received by the gateways where the data is transmitted to the "things" network such as the internet. Three

different classes of the end-point devices are available depending on intended application. The classes provide different characteristics w.r.t range, power consumption and latency [1].
The modulation technique LoRa utilizes is called chirp spread spectrum (*CSS*), which provides energy efficient, long range transmission. However, the long range requires a decrease in bit rate [25].
The bandwidth utilized by LoRaWAN is dependent on the operation region and frequency plan and may differ between 125, 250 and 500 kHz as seen in table 3.1 [4]. CSS modulation uses its entire bandwidth to transmit a message which makes is vigorous against interference. The broad bandwidth enables uplink messages up to 243 bytes to be transmitted [11]. Each message is transmitted to all base stations in range which increase the chance of successfully transmitting messages.

3.1.3 Technical differences

Table 3.1 contains the technical differences between the two LPWAN technologies Sigfox and LoRa.

Technical differences, Europe		
	Sigfox	**LoRa**
Modulation	D-BPSK	CSS
Licence	Unlicensed ISM Band	Unlicensed ISM Band
Frequency *(MHz)*	Europe 868	Europe 868
Bit rate *(kbps)*	0.1	0.3-50
Bandwidth *(kHz)*	0.1	125, 250 or 500
Range *(km)*	Rural 40 Urban 10	Rural 20 Urban 5
Interference resistance	High	High
Private networking	No	Yes

Table 3.1: The table presents specifications compared between the LPWAN technologies Sigfox and LoRa.

3.2 RS232

RS232 is an asynchronous wire data transmission method where the shift in signal level provides a binary number series. The bits are received sequentially at a defined rate. An example of this is presented in figure 3.1.
For Transistor–transistor logic (TTL),-level signals the binary value is perceived as a ”0” when the amplitude of the signal is between 0 and $0.8V$ and as a ”1” when the amplitude of the signal is stationed in the range $2V$ and $5V$.

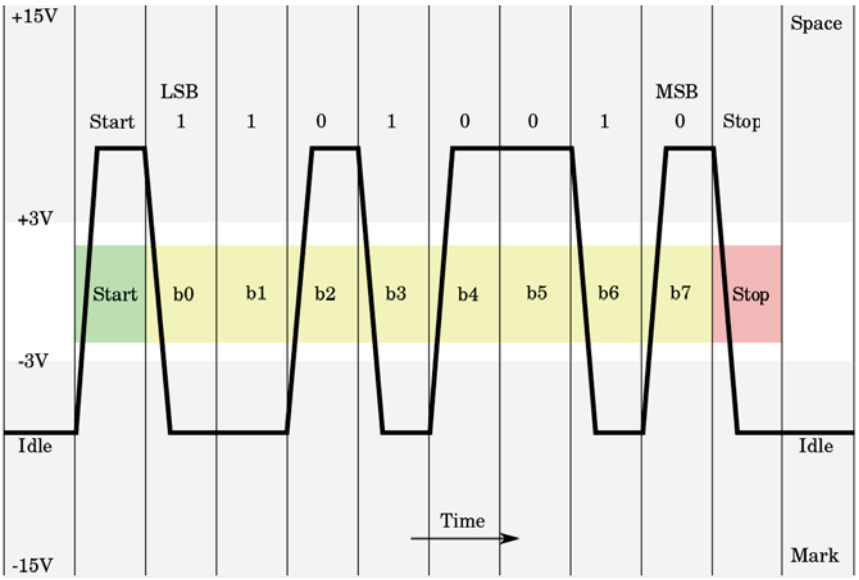

Figure 3.1: The picture illustrates how an analog signal is interpreted as a sequence of data that is being transmitted using serial communication [13].

To mark the beginning and end of each data package a start and stop bit is placed between the data bits. The start-bit synchronizes the package by dropping from ”1” to ”0” on an idle signal line. Indication that the stop bit is reached is presented that the signal is placed in an idle ”1”.
Baud rate is a term which determines the speed of the data transmitted or received by a system and is defined as the amount of symbols transmitted or received per seconds. Standard baud rates range from 1200 to 115200 symbols/s. Since serial communication operates asynchronously it is important that the same rate is set for both systems communicating with each other. If the baud rate differs the bits will not be interpreted correctly and bit errors will occur making the data package obsolete.

3.3 RS485

RS485 is a multipoint communication standard designed for devices to communicate in industrial and noisy environments. It utilizes a master/slave model where one device functions as master and is the main orchestrator of how the other slave-devices are to behave. In the communication system a maximum of 32 devices may be used and the signal may be transmitted in lengths of up to 1200m. The system is able to send data at the rate of 10 Mbits per second but the speed drastically depends on the distance between the node and the master. To reach the speed of 10 Mbits per second only a few meters of wire may be used [22].
The data is sent differentially via two signals which are the inverse of each other. An example of the signal is presented in figure 3.2. The advantage of differential signaling is due to its characteristic of higher tolerance to noise. Rather than the difference from signal to ground the signal is received as the difference from both signal wires. This results in that the transmitted signals both absorb equal amount of interference and therefore the difference between the signals are stable to the desired amplitude.

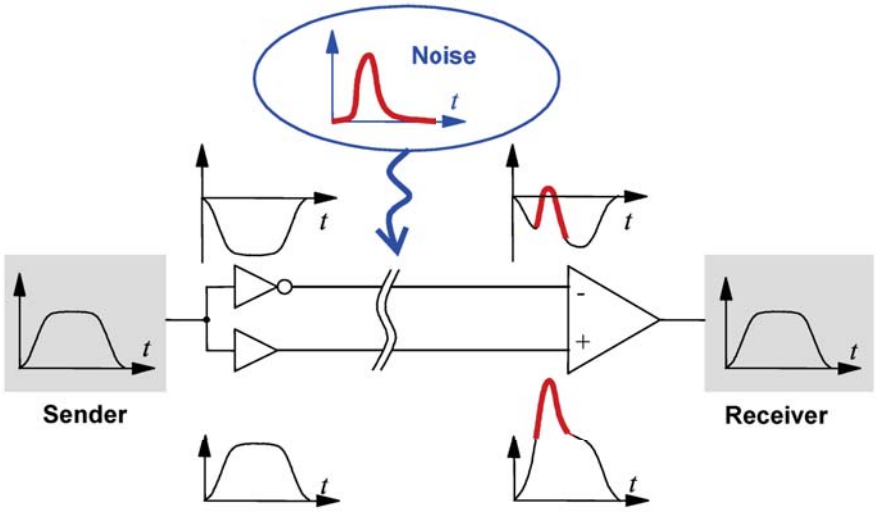

Figure 3.2: The picture illustrates the appearance of differential signals and how noise affects them [5].

A commonly encountered problem in signal wires that stretch for long distances are voltage reflections which may induce interference in the data. To prevent voltage reflection a resistor is placed between the signal wires, commonly the size of 120 Ω. To further prevent interference, the wires are commonly twisted which improves the electromagnetic compatibility [22].

3.4 Signal Quality

In telecommunication measurements are used to calculate signal reception quality forms of receiver signal strength indicator (RSSI), and signal-to-noise ratio (SNR). RSSI is an indicator of how well a device is able to interpret a signal from an accesspoint. By knowing the value of a signals RSSI makes it easier knowing if the signalqualifies for wireless data transmission. SNR states the relation between the signal power and power of the noise floor. If the received signal is to close to the noise floor data corruption might occur. Weak connectors, bad cables, nearby electrical equipment or perhaps that the device is near the distance limit may all be causes of decreased SNR.

Signal to interference and noise ratio, (SINR), demonstrates the strength of the desired signal compared to noise and interference in dB. If the interference on a signal equals zero the SINR would translate to SNR. The interference is extrinsic noise which originate from signal unwanted sources.
A study published on interference in the European ISM band with focus on LoRa and Sigfox states that "measurements show that there is a 22-33 % probability of interfering signals above -105 dBm within the mandatory LoRa and SigFox 868.0-868.6 MHz band in a shopping area and a business park in downtown Aalborg" which demonstrates vulnerability to LPWAN in more electrically dense areas[9].

4

Components and Intergrated circuits

This chapter provides technical explanation of the core elements used in the design of the device with the purpose to provide a clearer perspective of the integrated circuits and their function as well as the design structure.

4.1 Arduino Uno

Arduino is an open source micro-controller board supporting "Arduino programming language" which features different dialects from the common programming language C and C++ [20]. There are several different Arduino boards available but the most commercially known is the Arduino Uno which was used for this project. The Arduino is programmed to specifically orchestrate signals to different output pins connected to a circuit. The Arduino UNO has the following hardware specifications:

- ATmega328/P Microcontroller with 32KB programmable ISP flash memory.
- 14 Digital output/input pins.
- 6 analog inputs.
- 16 MHz quartz crystal.
- USB connection.
- Power supply jack.

The design and apperance of the microprocessor is seen in figure 4.1.

Figure 4.1: Illustration of an Arduino UNO board [24]

4.2 SIGFOX Breakout board BRKWS01

The Sigfox module possesses the ability to integrate a device to transmit small data via its attachable antenna using Sigfox-LPWAN. The supply voltage may vary between 1.8V to 3.6V but is typically set to 3.3V. Its low power consumption and advantageous size of 23.3mm x 21.3mm makes it easy to use in project designs [16]. The design and appearance of the module is seen in figure 4.2.

Figure 4.2: Illustration of a Sigfox breakout board BRKWS01 [16].

The board uses serial communication and data is transmitted and received via two pins, TX and RX. The serial communication operates at the rate of 9600 baud and utilizes 8 data bits and 1 stop-bit. It is controlled using serial AT commands which is a standardized way of communicating with a modem [3]. The command used when transmitting data is "$AT\$I = XXXXXXXXXXX$", where each X represents the data in hexadecimal value. Sensory data transmitted in the same data package are divided and processed to its correct form at the Sigfox backend.

4.3 MAXRS485

$MAXRS485$ is an integrated circuit which translates RS485 signals to UART-TTL level signals for serial communication and vice versa. The units supply voltage is $5V$ and as seen in figure 5.4, the module utilizes eight I/O pins.

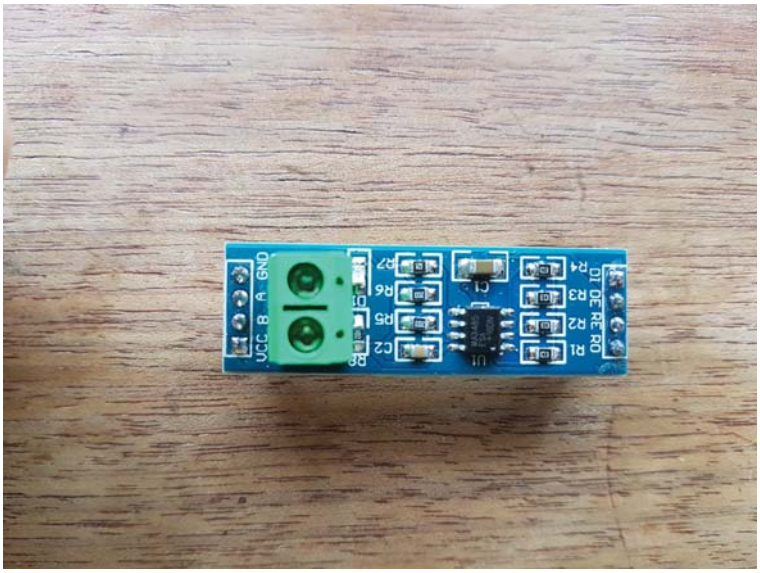

Figure 4.3: Illustration of a MAX RS485 board.

The I/O pins seen in figure 4.3 are specified in in table 4.1

Table 4.1: Pin configurations of $MaxRS485$.

PinName	-	*Function*
RO	Receiver Output-	Data Receive
RE	Receiver Enable-	Control pin
DE	Driver Enable -	Control pin
DI	Driver Input-	Data Transmit
VCC	Supply voltage	
B	RS485-	Inverted signal
A	RS485+	Signal
GND	Ground	

4.4 TBR 700 RT

The "TBR 700 RT" is a device designed to provide real time data feed for underwater animal and fish tracking and is presented in figure 4.4. The TBR device registers acoustic vibrations that originates from the associated tag placed on top or inside the fish. The frequency most commonly used for these devices is $69kHz$ but is dependent on the tags used [21]. Each tag carries an unique ID which is registered

when the tagged fish/animal passes the location of the TBR.

Figure 4.4: Illustration of a TBR 700 RT with associative RS485-to USB-cable for real time data transfer [21].

The associative RS485 to USB-cable presented in figure 4.4 may be connected to a computer or tablet in order to provide real time data [21]. The device utilizes RS485 communication protocol to transmit and receive the information at the rate of 115200 baud. There are two types of data packages transmitted from the TBR, sensory readings and tag detections. Each TBR is configurable to register sensory data in suitable intervals. The TBR sensory reading data package contains eight types of data separated by commas and is presented as follows: $\$000004, 0946709700, TBRSensor, 309, 10, 15, 69, 2330$". What the data represents is presented in table 4.2.

Table 4.2: Types of data presented in data package: TBR sensor reading.

\$000004	TBR serial number
0946709086	UTC UNIX Timestamp
$TBRSensor$	receiver sensor readings
309	Temperature(data-50)/10 -> °C)
10	Average noise level
15	Peak noise level
69	TBR listening frequency in kHz
2330	code running entry number in flash memory

The second type of data is when a Tag detection is made and is presented as following: $\$000004, 0946709086, 287, S256, 1, 0, 8, 2329$. The representation of this package is presented in table 4.3.

Table 4.3: Types of data presented in data package: Tag detection.

$000004	TBR serial number
0946709086	UTC UNIX Timestamp
287	millisecond timestamp
*S*256	code type
1	tagID
0	data
8	SNR
2329	code running entry number in flash memory

When connected to a computer the unit can be placed in listening mode where certain commands will extract the desired information about the fish.

4.5 Power supply

For power supply a power bank of model *PB-A303* was used which has a capacity of 20000 mAh. The output of the battery is 5V/1A.

5

Construction of the prototype

This section presents the construction and designing of a device, able to send data provided from a tracking unit via Sigfox-LPWAN. This project was divided into two main sections: "Establishing connectivity from the Arduino to the Sigfox modem" and "Establishing communication with the tracking device using RS485 interface". The Conties administrative board provided one of the tracking units which was necessary to perform tests and ensure that the prototype successfully was able to interpret and transmit data before performing the field test in Göta river.

5.0.1 Planning the construction

Details of how the TBR functions were obtained and a plan of how the device was going to operate was sketched. The main idea was to have the device becoming active once every ten minutes and ask the TBR for the relevant information. The Arduino would then process this data and send the package to Sigfox-backend. The device was to operate outdoor and therefore the container needed to be able to withstand rough weather.

5.1 Connectivity with Sigfox-LPWAN

When establishing connectivity with Sigfox-LPWAN the vital elements needed were a Sigfox modem with an attachable antenna and a programmable microprocessor providing the ability to communicate with the modem.
An Arduino UNO was used as programmable processor since it had all necessary requirements for this project. The wiring between the Arduino and modem was connected according to the schematic in figure 5.1.

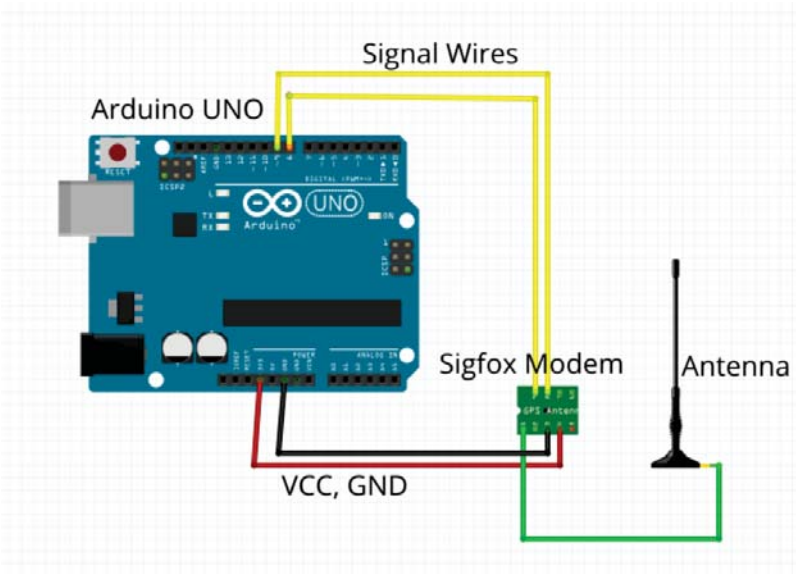

Figure 5.1: Illustration of the wiring between the Arduino and Sigfox modem.

The code presented in appendix A.0.1 was implemented to be able to verify the functionality of the circuit and modem. AT-commands were then sent from the Arduino serial monitor to ensure that the data was successfully transmitted to Sigfox-backend.

5.2 Communication from the Arduino to tracking device

This section explains the steps and decisions made when implementing the tracking unit to the system.

5.2.1 Verifying functionality of TBR 700

The functionality of the TBR and its serial communication was verified by connecting the "RS485 to USB" cable between the devices. The Arduino serial monitor was then used to transmit commands and receive data. A list of the TBR commands is provided in the TBR communication protocol [12].

5.2.2 Connection between TBR and MAX RS485

The Arduino receives and transmits data using $UART - TTL$ level signals whilst the TBR uses $RS485$ communication. These alternatives are not able to directly communicate with each other. To make communication possible between the devices a $MAXRS485$ unit which provides translation between UART and RS485 was used. In order to establish connection between the TBR and $MAXRS485$ an alternative socket to connect with the ending of the TBR-cable needed to be constructed.

Figure 5.2: The figure illustrated the cable end from the TBR. The socket pins are defined in table 5.1 TBR [12].

Figure 5.2 presents the end of the cable socket from the TBR which are to be connected to the $MaxRS485$.

Table 5.1: Pin definition of TBR cable presented in figure 5.2.

$PinNumber$	$Definition$
1	VCC(5 to 12VDC), external power input, may be left unconnected
2	Not used
3	Not used
4	RS485+
5	RS485-
6	Not used
7	Ground

The socket definition presented in table 5.1 show that the Max RS485 requires pin 4, 5 and 7 to connect between the devices. As previously stated in section 3.3, to ensure that the signal passing through the cable does not get exposed to interference some requirements of the cable need to be met. Since the device was not to be used in electrically noisy environments a five meter long twisted pair cable was used. The cable socket towards the TBR was constructed by soldering three connectors to the cable end to fit the pins 4, 5 and 7 presented in figure 5.2. On the other end of the cable female connectors was attached to the wires to ensure stability when connected to the male I/Os on the MAX RS485.

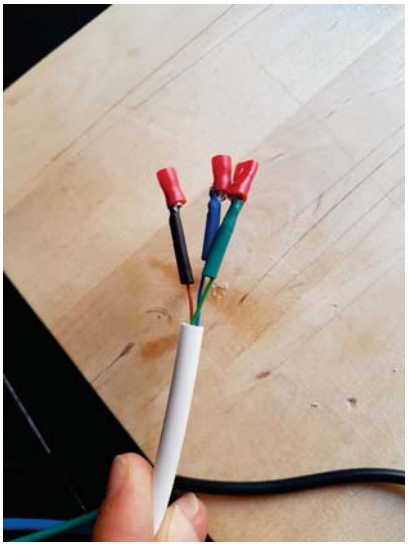

Figure 5.3: The figure shows the signal wires and cable connected between the TBR and MAXRS485.

To verify functionality of the modified cable presented in figure 5.3, a multimeter was used to ensure that the cable ends were connected.

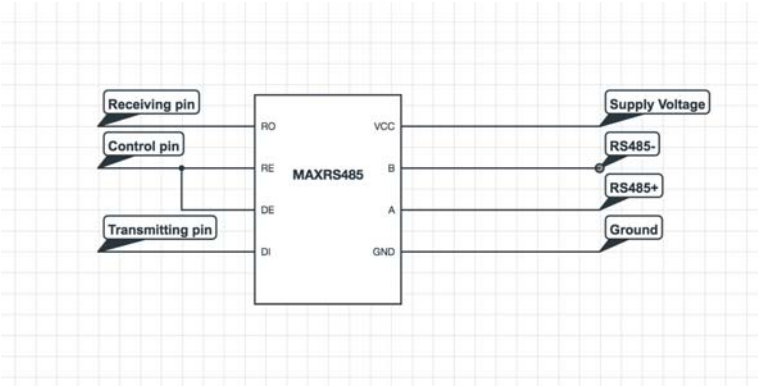

Figure 5.4: Electrical schematic of $MAXRS485$ [10].

Furthermore, the differential signals from the TBR were connected to "A" and "B" illustrated in figure 5.4. The two control pins, "RE" and "DE" were connected to each other with the purpose of merging the two pins into one control pin. When the control pin is set to "$HIGH$" it enables "DI" and when it is "LOW" it enables "RO". An active "RO" enables the $MAXRS485$ to receive data from the Arduino and pass it on towards the TBR and vice verse when "DI" is active. The control pin is set from the Arduino. Figure 5.4 presents the schematic of how the $MAXRS485$s I/Os was connected.

5.2.3 Connection TBR to Arduino

The Arduino and TBR were connected according to the schematic in figure 5.5. Tests to verify the functionality was conducted by sending commands to the TBR

and analyzing the results on the Arduino serial monitor. Connection was established but interference with the signal was experienced when larger data packages was sent from the TBR.

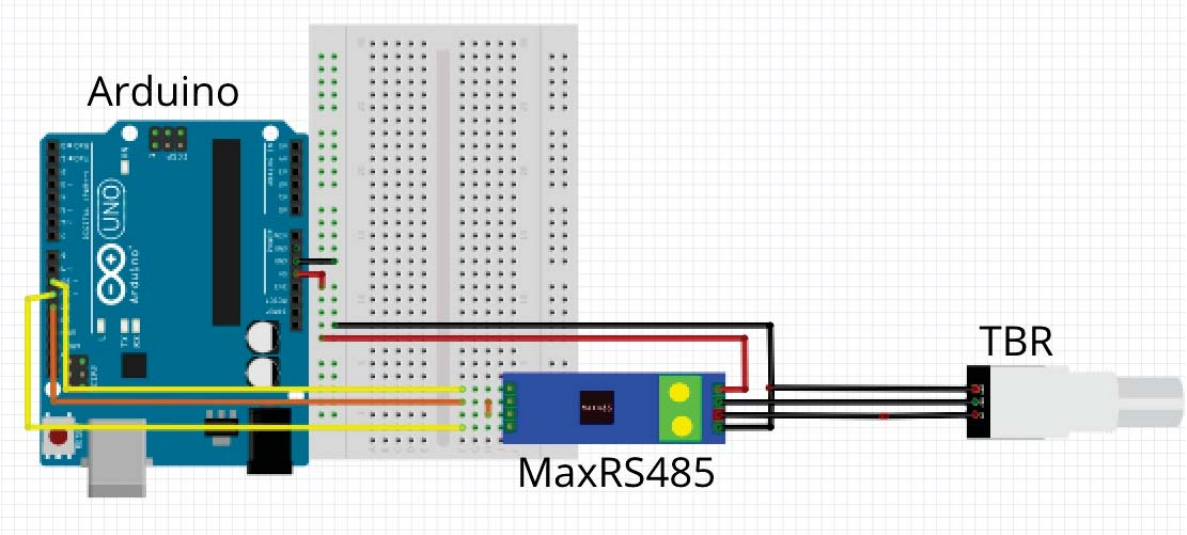

Figure 5.5: Schematic of the wiring when testing the connection between the Arduino and TBR.

As the problem was interfering with how the device was to be functioning, tests were conducted in order to find the location of the problem.

Two other cables were constructed and tested to ensure that the cable was working properly.

By measuring from point "A" to ground and point "B" to ground on the MaxRS485 missing the voltage reflection resistor was excluded.
Due to time limitations a decision was made to adjust to the situation and move forward with the programming of the Arduino.

5.3 Complete system

The Arduino to TBR and Arduino to Sigfox-circuit was merged and connected according to the circuit in figure 5.6. The digital output pin on the Arduino which produce the message signal has an amplitude of 5V when "High" and 0V when "Low". Since the Sigfox modem operate with 3.3V input a level-shifter was implemented to the circuit with the purpose of avoiding damage to the modem.

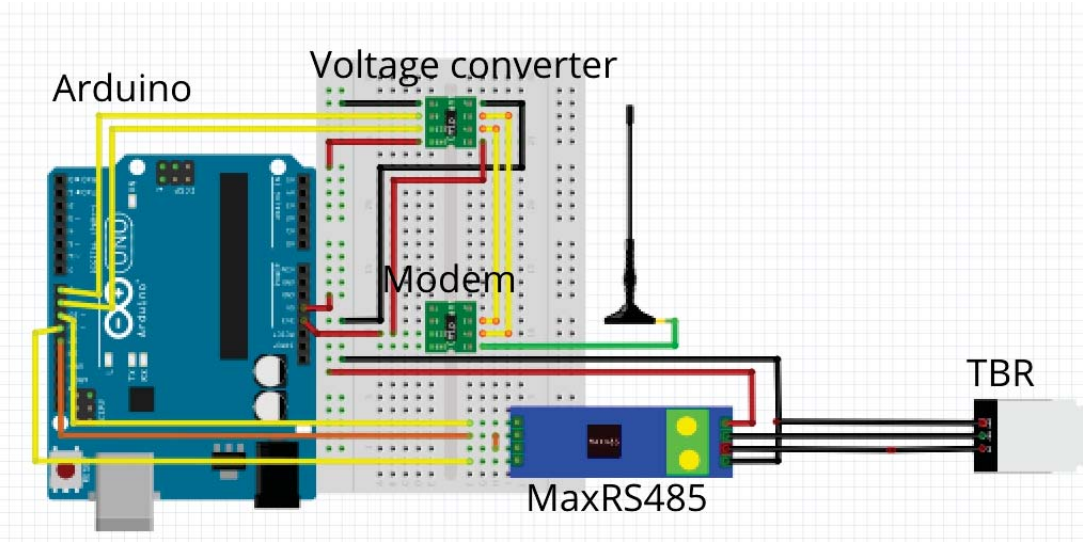

Figure 5.6: Schematic of the complete circuit.

5.4 Programming of the Arduino

By examining the TBR communication protocol and consulting with County administrative board the data package to be sent was to contain the following types of data: the total amount of fish tags detected, water temperature, and the ID of the last four tag detections. Since the package only may contain 12 bytes due to the limitations of sigfox LPWAN, two bytes were assigned to each type of data ensuring they held enough capacity. Equation 5.1 calculates the decimal value of two bytes or 16 bits.

$$65535 = 2^{16} - 1 \tag{5.1}$$

The equation ensure that enough capacity are allocated to suit its purpose. The received data is then processed in order to isolate the relevant data. Since Sigfox utilizes hexadecimal numbers the data needed to be converted from decimal to hexadecimal. Furthermore, for the message to be interpret correctly at Sigfox-backend, total amount of digits in the hexadecimal figure always needed to be four since the binary value of ”1111111111111111” is equal to ”$ffff$” hexadecimal. Meaning, if the hexadecimal figure didn’t make use of the most significant digit a "0" needed to be added to its location.

The finished code is presented in A.0.2 where it can be viewed with comments for every function. Figure 5.7 represent the flowchart of the finished code implemented on the Arduino.

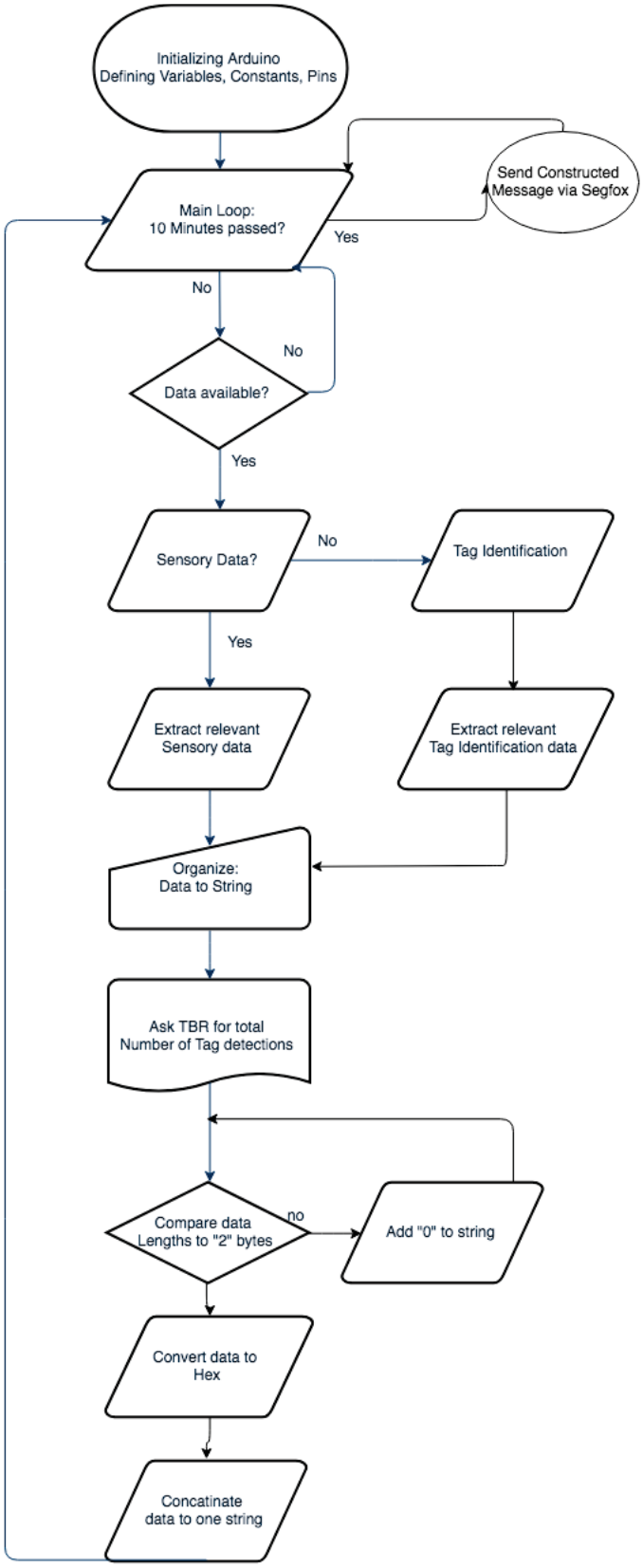

Figure 5.7: Flowchart representing the work flow and functions of which the process is built and implemented on the Arduino.

5.5 Modification of the device box

Once the system and code was running properly the box of which the electronics were placed in was modified to suit the needs of outdoor weather conditions. A transparent plastic box was used as base. Two holes were drilled in the box to situate the antenna on the top of the box and the signal cable from the TBR to be connected to the MaxRS485 from the bottom of the box. The battery was placed in the bottom of the box to provide stability and adhesive was used to attach the electronics to the side of the box above the battery. Silicon was placed around the edges of the box to provide protection against rain. Silica gel bags were attached on the sides to absorb condensation and humidity.

5.6 Planning of field test

Three different locations along Göta älv were chosen for testing depending on the characteristic of Sigfox service map. By conducting these tests, results of how efficient the network operates at different signal strengths were provided.
The first test was performed during the release of cultivated Atlantic salmon at location: 58°03'22.7"N 12°08'40.4"E, a few miles down the river from where the fish was released. According to the County administrative board, the fish need to acclimate to its new environment for a few days before they start moving towards the ocean. This location was selected because of the County administrative board interest in knowing at what time the salmon started to move from the place they were released.

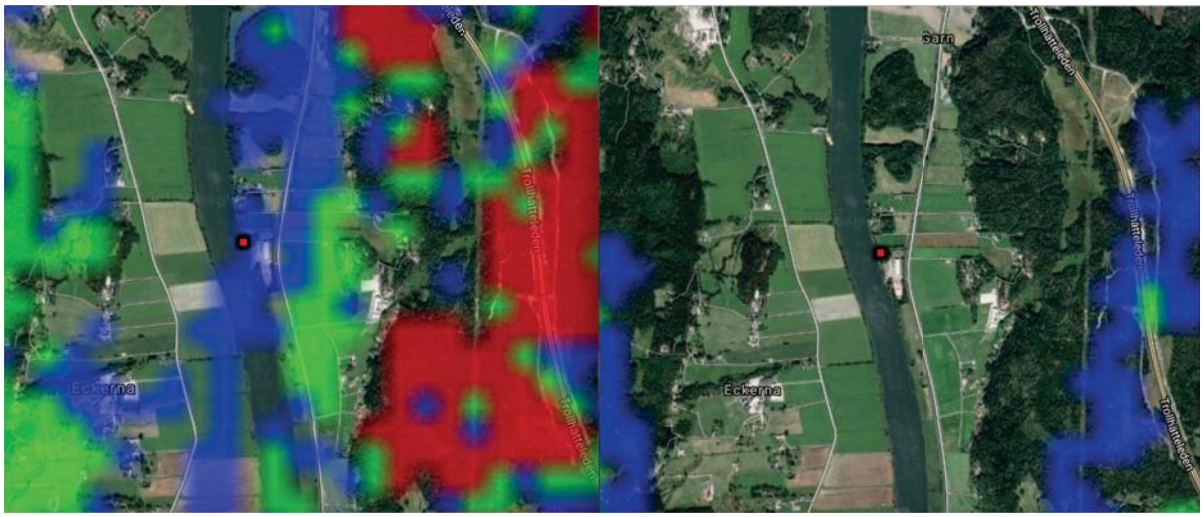

Figure 5.8: The figure presents the location of testing and the service map of the first test site. The left side of the picture demonstrate coverage without the the 20dB filter and the right side with the filter.

In figure 5.8 the Sigfox service map of the first test location is presented. The coloring represents how many base stations that cover a specific location, blue representing one base station, green representing two base stations and red representing

three or more base stations. A filter of 20dB may be applied to the service map to illustrate if the signal is strong enough to transmit data from inside a building. The left part of the picture does not have that filter implemented while on the right part it is implemented. From figure 5.8 the signal strength appears to be very low with only one coverage from one base station.

The second test was performed outside of Alvhem at the location 58°00'08.4"N 12°07'48.5"E presented in figure 5.9.

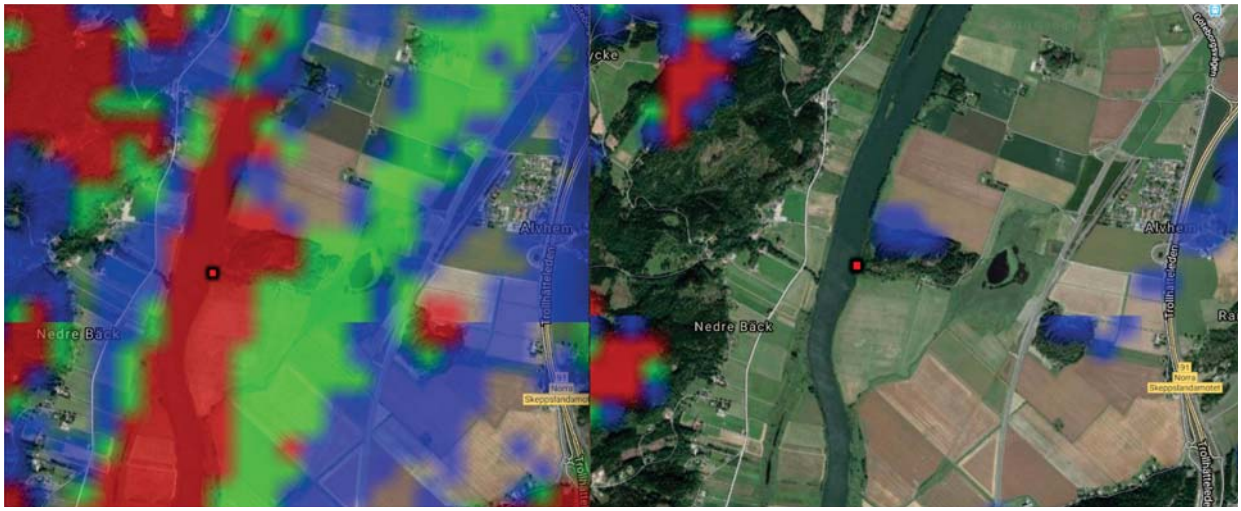

Figure 5.9: The figure presents the location of testing and the service map of the second test site with the 20dB filter to the right and without the filter to the left.

According to the service map in figure 5.9 there are three base stations covering the location. When the filter is applied, no coverage in the area is seen which implies a week signal.

The third and last location to be tested was performed at the outlet of Göta älv in Gothenburg at the location 57°41'12.0"N 11°53'12.4"E presented in figure 5.10.

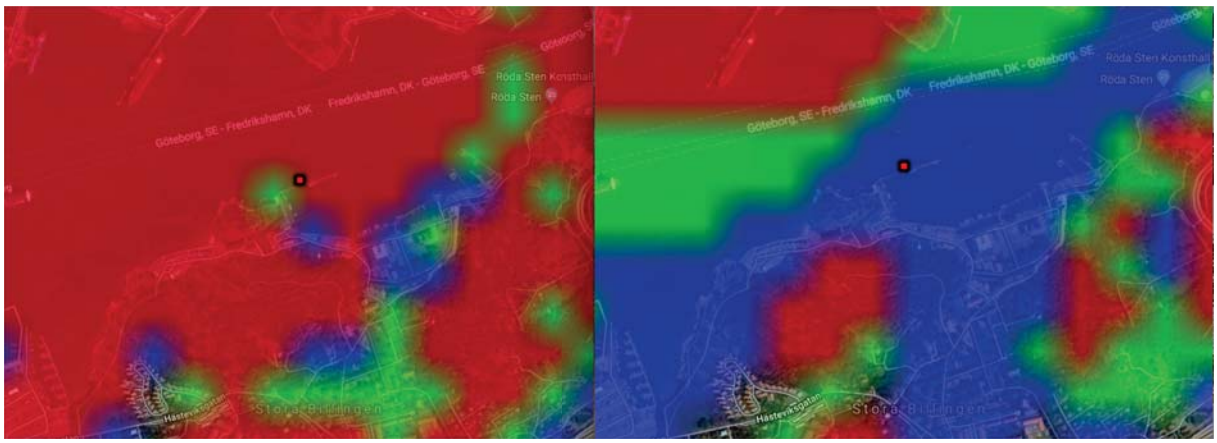

Figure 5.10: The figure presents the location of testing and the service map of the third test site with the 20dB filter to the right and without the filter to the left.

The service map in figure 5.10 implies very good coverage both with and without the filter.

6
Results

The original idea of how the IoT-device was supposed to function was not achieved which resulted in a decrease in the assumed battery life. Despite the interference issue the device was still able to receive, process and transmit data using Sigfox-LPWAN. The interference encountered when receiving large amounts of data was the main cause in not being able to follow the original idea. Other results of how well Sigfox-LPWAN performs with consideration to the service map are presented in this chapter.

6.1 Signal testing

The RSSI and SNR of the testing was obtained from the data bank collected of the specific Sigfox modem on Sigfox backend.
The first test took place between 2018-04-26 12:30 and 2018-04-30 09:00 which covers 93.5 hours. The device placement and surroundings from the first location are seen in figure 6.1. If the device would have had coverage 100% of the time, 561 messages would have been sent due to the fact that it is programmed to transmit every 10 minutes. The actual amount of messages transmitted was counted to 281. Therefore, the percentage of successfully transmitting a message was calculated to 50% at the first test site.

Figure 6.1: The figure presents device placement of first test location.

The second period was conducted between 2018-05-01 15:30 and 2018-05-04 09:00. The device placement and surroundings from the second location are seen in figure 6.2. This period covers 65.5 hours which should result in 393 messages if the device would have had coverage 100% of the testing time. However, only 40 messages were received during this period which generates the success rate to 10.2%.

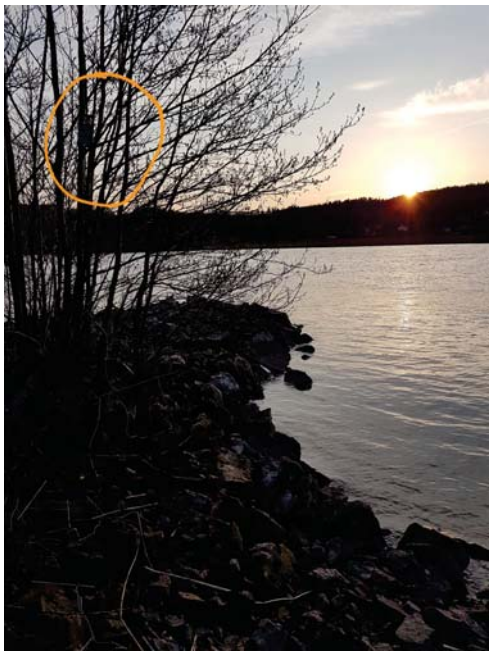

Figure 6.2: The figure presents device placement of second test location.

The RSSI and Average SNR during the first and second testing period are presented in figure 6.3 and figure 6.4. A compilation of the key values from these measurements may be observed in table 6.1 and 6.2.

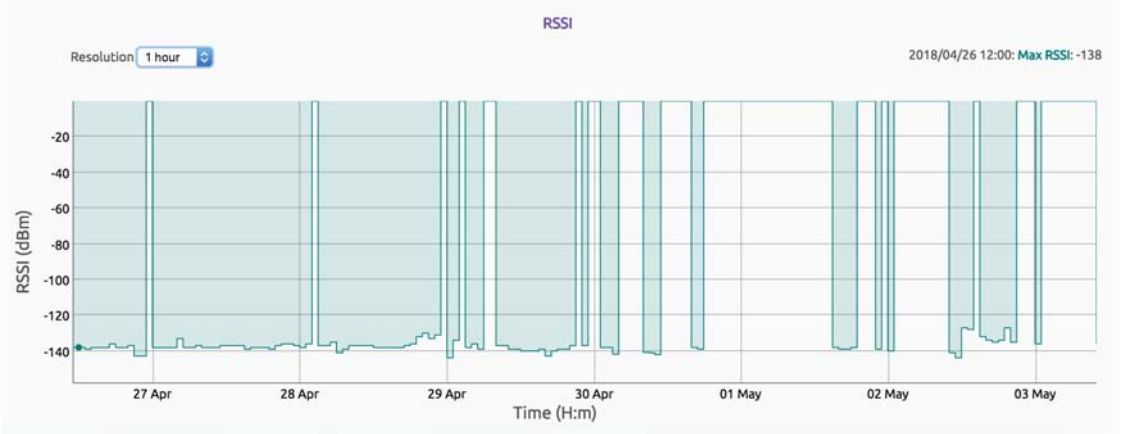

Figure 6.3: RSSI from the 26th of April til 3rd of May.

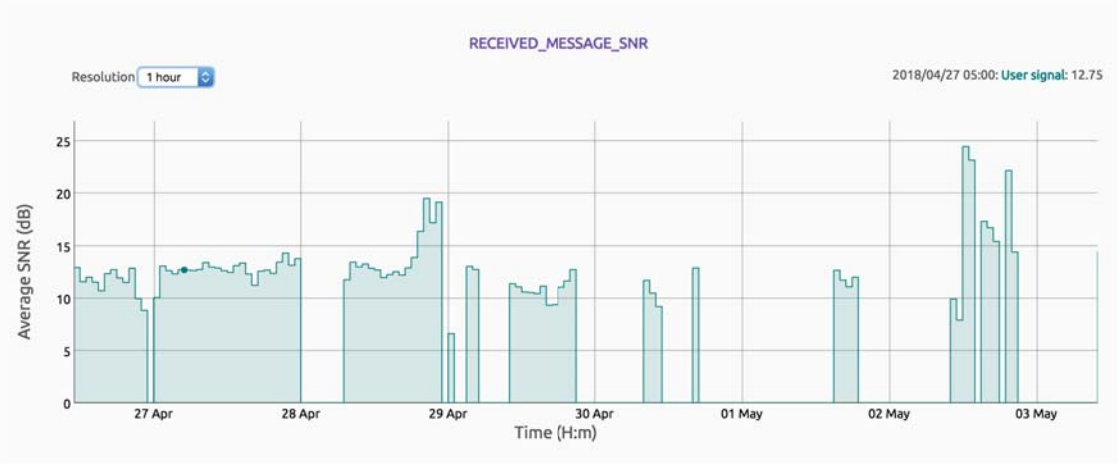

Figure 6.4: SNR from the 26th of April til 3rd of May

Table 6.1: RSSI and SNR on first test site.

First site	Mean value	MAX	MIN
RSSI	-138	-131	-142
SNR	13,7	19,4	7,2

Table 6.2: RSSI and SNR on second test site.

Second site	Mean value	MAX	MIN
RSSI	-139	-133	-142
SNR	17,2	24,1	8,0

Comparing the RSSI of the first and second test location no deviation was seen. However, the SNR level was varied and was higher at the second location.

The third test was conducted between 2018-05-07 13:00 and 2018-05-12 21:00 which gave a testing period of 128 hours. The total amount of messages supposed to be transmitted was 768. The actual amount of messages transmitted was 619 which gives a success rate of 80%.

Figure 6.5: The figure presents the device placement at third test location.

Figures 6.7 and 6.6 demonstrated the SNR and RSSI of the third and final test site. How the device was placed at the test site is presented in figure 6.5.

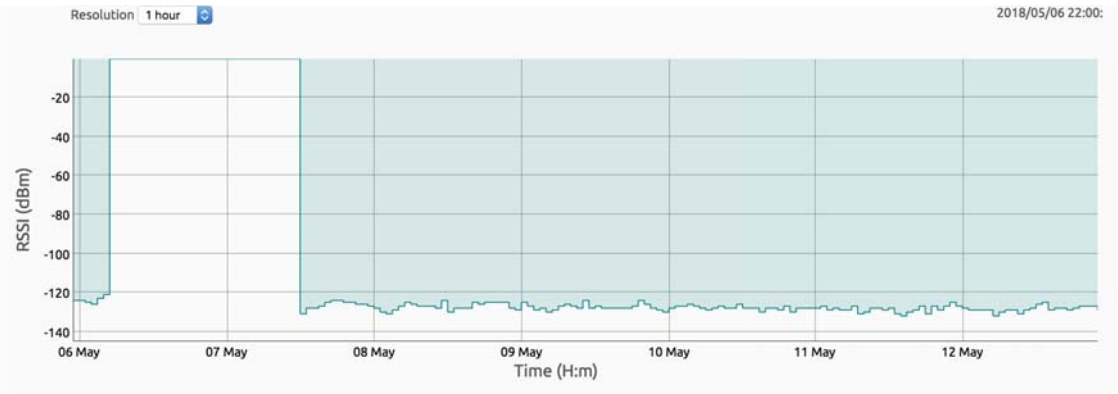

Figure 6.6: RSSI of the third testing period.

Figure 6.7: SNR of the third testing period.

Key values from figure 6.6 and 6.7 and are presented in table 6.3.

Table 6.3: RSSI and SNR on third test site.

Third site	Mean value	MAX	MIN
RSSI	-128	-123	-131
SNR	21,3	26,6	14,5

Comparing the RSSI of the third site to the first and second site, its value has greatly increased as well as the SNR level. Due to the variation of the success rate between the testing periods this difference of RSSI and SNR is not unexpected.
An example of how messages are presented at Sigfox-backend is shown in figure 6.8.

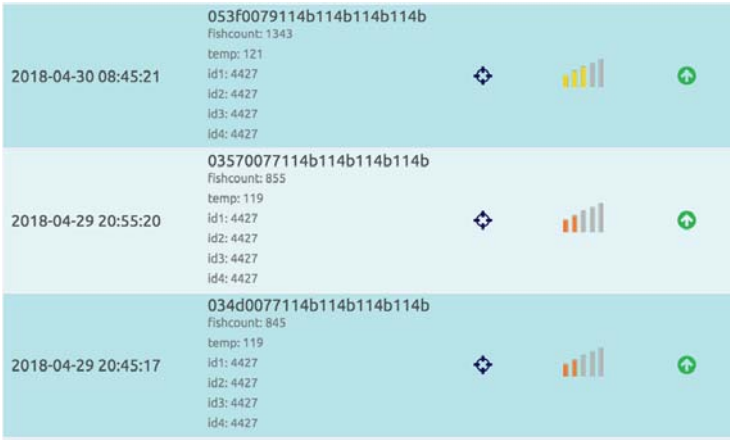

Figure 6.8: The figure illustrates three messages on Sigfox-backend received during the first testing period.

The string seen on the upper part of the message presented in figure 6.8 is decoded into the information seen below the string. The period from 20:55 to 08:45 the next day seen on the figure the device lost coverage to the network and was unable to transmit data. Also note the change from 855 to 1343 in "fishcount" during the same

period and that the same ID is presented in all four id slots. Perhaps fish 4427 was eaten by a pike stationed in the area.

6.2 Power efficiency

The longest period the battery was able to power the device for was 128 hours. Therefore, the power consumption of the device was calculated to 156 mAh.

7
Discussion

To conclude the project, implementation of IoT devices which provide continuous data feed every ten minutes concerning the Atlantic salmon in Göta Älv helped the county administrative board staff in ways of letting them know when a location of interest was passed by the fish. Mikael cremle, county administrative board, says *"A much needed device which I am surprised has not been presented to us earlier."* Since the device was unable to use power saving mode the power consumption was drastically increased and a much larger battery would be needed in order to provide a more suitable battery life for the application.
Reflecting on ethics regarding this project the methods used by the counties administrative board when implanting trackers into anesthetized fish by making incisions may be questioned. However, the aim of the missing salmon project is to research why the decrease occurs. By understanding the true nature of the project I find the this thesis both ethically and sustainably dependable. Researching why a decline like this occurs will provide important information and make way for similar projects in the future.
The data gathered by the prototype at the three different test locations have given information of the signal strength and quality compared to the coverage map with and without the filter applied. The success ratio varied significantly depending not only on coverage provided on the service map but perhaps due to other factors as weather, obstacles blocking the line towards the base station or fog [23]. The fact that the river is the lowest point in the surrounding does not make the position ideal for transmission of RF waves as well as the fact that Sigfox-LPWAN is still under construction and not extensively established outside the larger cites in Sweden. I therefore believe that the deviations experienced from the first and second test site are affected by these variables since better coverage was expected at the second test site according to the service map.
With good coverage on the service map and suitable placement the success rate of transmitting a message would surely have been greater than what was achieved in the third testing period where the placement was beneath a wharf in order to avoid attention to the public. The maximum send interval of Sigfox messages are every 10 minutes. The LPWAN is therefore suitable for projects that are not of the time-critical character. However, the easy implementation of a Sigfox modem together with its limitations makes it in my opinion suitable for applications collecting sensory data.
Since Sigfox LPWAN is not extensively deployed LoRaWAN, which support private networking, is a solution to stakeholders where Sigfox does not have coverage. Implementation of a private LoRaWAN requires gateways which significantly increase the

deployment costs compared to Sigfox LPWAN where the company is the deployer of the base stations providing coverage [11].
Monitoring crucial parameters of rivers, lakes and soil using Sigfox LPWAN at rural places outside the power grid would require autonomous solutions to Sigfox base stations. The topic was investigated by Emil Eskång and Gustaf Dahl in their Bachalor's thesis *"Researching possibilities for autonomous operation of a Sigfox radio base station north of the Arctic circle"* at Chalmers University. A solution similar to this would enable easier implementation of battery powered units gathering data of the wellness of the earth.

7.1 Further works

This section presents room for improvements if the opportunity was to be presented. One possible scenario of why the interference occur in the data transmission could be that the clock pulse dividing the bits is not 100% accurate to the speed of the TBR. If the pulse would fall out of sync with the bits sent from the TBR the receiver would misinterpret the data and display wrong symbols. This problem might be the issue since it's more likely to occur when larger data packages are sent. Improvements of the code to prevent repeats of the same fish ID are possible to implement.

Bibliography

[1] *About LoRaWAN™ | LoRa Alliance™*. URL: https://lora-alliance.org/about-lorawan.

[2] Quick Answer. *Difference Between LoRa and LoRaWAN*. URL: https://enablingsupport.zendesk.com/hc/en-us/articles/205089362-What-is-the-difference-between-LoRa-and-LoRaWAN.

[3] *AT-Command Structure*. URL: http://processors.wiki.ti.com/index.php/File:AT_Command_Structure.png.

[4] Romain Cambier. *LoRaWAN & The Things Network*. 2017. URL: https://www.thethingsnetwork.org/wiki/LoRaWAN/Home.

[5] *Differential signaling*. URL: https://en.wikipedia.org/wiki/Differential_signaling#/media/File:DiffSignaling.png.

[6] EUR-lex. *Journal of the European Commission*. URL: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:12008E102.

[7] IoT-Sweden. *Om oss _ IoT Sweden*. Stockholm. URL: https://www.iotsweden.net/sv/om-oss/.

[8] Cliff Microwave Journal and Dedham Vol. "Cellular , LPWAN Challenge Short-Range Wireless Solutions in IoT Markets". In: 60.Jul (2017), pp. 1–3. URL: https://search.proquest.com/docview/1919916392/fulltext/99F0BA3385A14599PQ/1?accountid=10041.

[9] M. Lauridsen et al. *Wireless Communications and Networking Conference (WCNC)*. Mar. 2017. DOI: 10.1109/WCNC.2017.7925650.

[10] Inc. Maxim Integrated Products. "RS-485 / RS-422 Transceivers ___ Next Generation Device Features CURRENT SHUTDOWN RECEIVERS ON RS-485 / RS-422 Transceivers". In: *Current* (2014), pp. 1–17. URL: https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf.

[11] Kais Mekki et al. "A comparative study of LPWAN technologies for large-scale IoT deployment". In: *ICT Express* (2018). ISSN: 24059595. DOI: 10.1016/j.icte.2017.12.005. URL: http://linkinghub.elsevier.com/retrieve/pii/S2405959517302953.

[12] T B R Sensor. "TBR 700 Communication Protocol". In: 0946709086 (2017).

[13] *Serial communication oscilloscope trace*. URL: https://commons.wikimedia.org/wiki/File:Rs232_oscilloscope_trace.svg.

[14] *Short-range device - Wikipedia*. URL: https://en.wikipedia.org/wiki/Short-range_device#SRD860.

[15] Sigfox. *Sigfox*. URL: https://build.sigfox.com/steps/sigfox.

[16] *SIGFOX BREAKOUT BOARD BRKWS01*. URL: https://partners.sigfox.com/products/sigfox-breakout-board-brkws01.

[17] Sigfox-Sigfox Techlology Overview. *Sigfox Technology Overview | Sigfox*. URL: https://www.sigfox.com/en/sigfox-iot-technology-overview.

[18] Tom Simonite. *SigFox, a specialist in ultra-narrow band technology for M2M and IoT - E-media, the Econocom blog*. 2014. URL: https://blog.econocom.com/en/blog/sigfox-a-specialist-in-ultra-narrow-band-technology-for-m2m-and-iot/.

[19] *Smolt tracking project*. 2018. URL: https://thefishsite.com/articles/smolt-tracking-project-launched.

[20] Arduino Store. *Arduino UNO*. 2014. URL: https://store.arduino.cc/usa/arduino-uno-rev3%0A.

[21] *Thelma Biotel*. Trondheim, Norway. URL: http://www.thelmabiotel.com/.

[22] *Understanding RS485_ Wiring, Connection, Monitoring Software - Windmill Software*. Manchester. URL: http://www.windmill.co.uk/rs485.html.

[23] *Weather versus propagation*. URL: http://www.tpub.com/neets/book10/40j.htm.

[24] Commons Wikimedia. *Arduino-uno-perspective-transparent - File_Arduino-uno-perspective-transparent*. URL: https://commons.wikimedia.org/wiki/File:Arduino-uno-perspective-transparent.png#/media/File:Arduino-uno-perspective-transparent.png.

[25] Technical Marketing Workgroup. "What is it ? A technical overview of LoRaWAN". In: November (2015).

A

Appendix 1

This chapter presents the code used at the two stages of the project.

A.0.1 Testing Sigfox module

Connectivity with the Sigfox modem was established using the prior code implemented on the Arduino.

```
//Author: Mattias Blinge
//Date:   Mars 2018

#include <AltSoftSerial.h>
// Library which enables transmition and receving through pin 8 and 9

#define SEND_INTERVAL 600000

AltSoftSerial altSerial;
// Set up the serial port connected to the Sigfox Modem


void setup() {
  Serial.begin(9600);
  //Defining baud rate to computer
  while (!Serial) ;
  // wait for Arduino Serial Monitor to open
  Serial.println("AltSoftSerial Test Begin");
  // Print on Serial monitor
  altSerial.begin(9600);
  // Defining baud rate to Sigfox Modem
  altSerial.println("AT\$I=10");
  // Command to test communication with Sigfox modem\cite{}

}


void loop() {
  char c;
```

```
  if (Serial.available()) {
    // If data is received from comuter, proceed
    c = Serial.read();
    // Place data in "c"
    altSerial.print(c);
    // Transmit "c" to Sigfox modem
  }
  if (altSerial.available()) {
    // If data is received from sigfox modem, proceed
    c = altSerial.read();
    // Place data in "c"
    Serial.print(c);
    // Transmit "c" to computer
  }

}
```

A.0.2 Finished code

The finished code is presented below with comments.

```
//Author: Mattias Blinge
//Date: April 2018


/*-----( Import needed libraries )-----*/
#include <AltSoftSerial.h>
// Library which enables transmission and receiving through pin 8 and 9
#include <SoftwareSerial.h>
// Library which enables transmission and receiving through pin 10 and 11
/*-----( Declare Constants and Pin Numbers )-----*/
#define SSerialRX        10
//Serial Receive pin TO RS485
#define SSerialTX        11
//Serial Transmit pin TO RS485
#define SigfoxRX         8
//Send message via Sigfox
#define SigfoxTX         9
//Send message via Sigfox

#define SSerialTxControl 12
//RS485 Direction control
#define RS485Transmit    HIGH
#define RS485Receive     LOW
#define NORMAL_DELAY 500
// In milliseconds
#define SEND_INTERVAL 600000
// Defining 10 minutes in milliseconds
#define ComputerSerial Serial

AltSoftSerial altSerial;
// Set up the serial port connected to the Sigfox Modem


SoftwareSerial RS485Serial(SSerialRX, SSerialTX);
// Set up the serial port connected to TBR


/*-----( Declare Global Variables )-----*/

String sendtoTBR;
int FishInt;
unsigned long lastSendTime;
```

```
/*--------------------(Initiating  Startup Code)------------------------- */

void setup() {
  altSerial.begin(9600);
  // Defining baud rate from Arduino to Sigfox Modem
  ComputerSerial.begin(9600);
  // Defining baud rate to computer
  RS485Serial.begin(115200);
  pinMode(SSerialTxControl, OUTPUT);
  // Defining

  digitalWrite(SSerialTxControl, RS485Receive);
  // Initiate Transceiver
  exitTBR();
  // Making sure TBR is in listening mode
  ComputerSerial.println("Booting");
  // Testing that communication with computer is established
  delay(10000);
  altSerial.println("AT\$SF=FFFFFF");
  // Test for coverage at start up
  lastSendTime = millis();
  // Reference time for While-loop
}

/*------------------(Loop process)---------------------*/

void loop() {


/*-----( Declare Local Variables )-----*/
  int numTempInt;
  String numTemp;
  String data_received;
  String SigFoxmess;
  int IDarray0;
  int IDarray1;
  int IDarray2;
  int IDarray3;

  while ((millis() - lastSendTime) < SEND_INTERVAL) {
  // Run program inside the loop for 10 min


    if (RS485Serial.available() > 0)
    // If data is received from the TBR proceed with the processing of the data
```

```
{
  data_received = RS485Serial.readStringUntil('\n');
  // Data is placed in a string
  String SensorCheck = getValue(data_received, ',', 6);
  if (SensorCheck == "69") {
  // Find out whether the data is a "Tag detection" och "Sensory data"
    numTemp = getValue(data_received, ',', 3);
    // Place the temperature value in String
    numTempInt = numTemp.toInt();
    // Convert numTemp from "string" to Integer.

  }
  else {
  // If the data received is a "tag detection"
    String IDtag1 = getValue(data_received, ',', 4);
    IDarray3 = IDarray2;
    IDarray2 = IDarray1;
    IDarray1 = IDarray0;
    IDarray0 = IDtag1.toInt();
    // Convert IDtag1 from "string" to Integer

  }
  getNumFishes();
  // Enter "getNumFishes" to fetch total number of tag detections from TBR
  rs485Read(500);
  /*--------------(For testing purposes)----------------*/

  //ComputerSerial.println(numTempInt);
  // Display Temperature on terminal as int
  //ComputerSerial.println(IDarray[0]);
  //ComputerSerial.println(IDarray[1]);
  //ComputerSerial.println(IDarray[2]);
  //ComputerSerial.println(IDarray[3]);
  //SigFoxmess = FishInt + numTempInt + IDarray[0]+
  IDarray[1] + IDarray[2] + IDarray[3];
  //ComputerSerial.println(SigFoxmess);




  //-----------(Making Sure StringFish is 16 bit)-------------
  String StringFish =  String(FishInt,  HEX);
  // Convert value to Hexadecimal

  while (StringFish.length() < 4) {
  // Add "0"s from the left until length equal to 4.
```

```
  StringFish = 0 + StringFish;
}

//-----------(Making Sure StringTemp is 16 bit)-------------

String StringTemp = String(numTempInt, HEX);
// Convert value to Hexadecimal

while (StringTemp.length() < 4) {
// Add "0"s from the left until length equal to 4.
  StringTemp = 0 + StringTemp;
}

//-----------(Making Sure StringIDtag0 is 16 bit)-------------

String StringIDtag0 = String(IDarray0, HEX);
// Convert value to Hexadecimal

while (StringIDtag0.length() < 4) {
// Add "0"s from the left until length equal to 4.
  StringIDtag0 = 0 + StringIDtag0;
}

//-----------(Making Sure StringIStag1 is 16 bit)-------------

String StringIDtag1 = String(IDarray1, HEX);
// Convert value to Hexadecimal

while (StringIDtag1.length() < 4) {
// Add "0"s from the left until length equal to 4.
  StringIDtag1 = 0 + StringIDtag1;
}

//-----------(Making Sure StringIStag2 is 16 bit)-------------

String StringIDtag2 = String(IDarray2, HEX);
// Convert value to Hexadecimal

while (StringIDtag2.length() < 4) {
// Add "0"s from the left until length equal to 4.
  StringIDtag2 = 0 + StringIDtag2;
}

//-----------(Making Sure StringIStag3 is 16 bit)-------------

String StringIDtag3 = String(IDarray3, HEX);
```

```
      // Convert value to Hexadecimal

      while (StringIDtag3.length() < 4) {
      // Add "0"s from the left until length equal to 4.
        StringIDtag3 = 0 + StringIDtag3;
      }

      //-----------(Concatinating the Sigfox message)-----------

      SigFoxmess = String("AT\$SF=" + StringFish + StringTemp + StringIDtag0 +
      StringIDtag1 +  StringIDtag2 + StringIDtag3);

      /*----------(For Testing Purposes)---------------*/
      //ComputerSerial.println(StringFish);
      //Print Hex value of number of Fish passed
      //ComputerSerial.println(StringTemp);
      //Print Hex value of last temperature read
      //ComputerSerial.println(StringIDtag0);
      //Print Hex value of last Fish tag scanned
      //ComputerSerial.println(StringIDtag1);
      //Print Hex value of 2:nd last Fish tag scanned
      //ComputerSerial.println(StringIDtag2);
      //Print Hex value of 3:rd last Fish tag scanned
      //ComputerSerial.println(StringIDtag3);
      //Print Hex value of 4:th last Fish tag scanned
      //ComputerSerial.println(SigFoxmess);
      // Print whole message


    }
  }
  altSerial.println(SigFoxmess);
  // Send message from Arduino to Sigfox modem.
  lastSendTime = millis();
}


//---------------FÅ FISKSTRÄNG-------------------
String getNumFishes() {
// Fetches the amount of tag detections made by the TBR
  sendTBR();
  // Place TBR in listening mode
  rs485Send("TD?");
  // Send "TD?" to TBR
  String response = rs485Read(500);
  // Read the from TBR for 500ms
```

```
  exitTBR();
  // Exit TBR listening mode
  String numFishes = getValue(response, '=', 1);
  // Process data received in "getValue" to fetch number of tag detections.
  FishInt = numFishes.toInt();
  // Convert numFishes from "string" to Integer.
  //ComputerSerial.println(FishInt);
  // Prints amount of tag detections made on computer monitor
  return numFishes;

}

/*
  //---------------Get TEMPERATUR-----(Not used)------------
  String getNumTemp() {
  sendTBR();
  rs485Send("PS?0001101");
  String response = rs485Read(1000);
  exitTBR();
  String numTemp = getValue(response, ',', 6);
  //ComputerSerial.println(numTemp);
  ComputerSerial.println(response);
  return numTemp;

  }
*/
//----------------------STRING SEPARATION-----------------------

String getValue(String data, char separator, int index)
// Splitting a string into pieces based on separation character
// and returns the item inbetween.
{
  int found = 0;
  int strIndex[] = { 0, -1 };
  // Contains the location of the beginning and end of the desired string
  int maxIndex = data.length() - 1;
  // Define the length of the string

  for (int i = 0; i <= maxIndex && found <= index; i++) {
  // If the end of the data string is not reached and the
  // index separator is not found, proceed
    if (data.charAt(i) == separator || i == maxIndex) {
    // If the character at position i is equal to separator character
      found++;
      // Increment "found"
      strIndex[0] = strIndex[1] + 1;
```

```
      strIndex[1] = (i == maxIndex) ? i + 1 : i;
    }
  }
  return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
  // Returns the data between strIndex[0] and strIndex[1]
}

//---------------TBR COMMANDS-------------------------

void exitTBR() {
  rs485Send("EX!");
  // Command to exit Listening mode of the TBR
  rs485Read(500);
  // Flush the "EX!" response
}

void sendTBR() {
// Placing TBR in listening mode
  digitalWrite(SSerialTxControl, HIGH);
  // Setting TX pin to "1"
  delay(10);
  RS485Serial.print("T");
  delay(2);
  // Transmitting the sequence "TBR" to the TBR places it in listening mode
  RS485Serial.print("B");
  // For the TBR to interpret the sequence delays
  delay(2);
  // must be placed between the letters.
  RS485Serial.print("R");
  RS485Serial.flush();
  digitalWrite(SSerialTxControl, LOW);
}


void rs485Send(String sendtoTBR) {
// Sub function of which writes transmitted data to the TBR
  digitalWrite(SSerialTxControl, RS485Transmit);
  // Place MAXRS485 in transmit mode
  delay(10);
  RS485Serial.print(sendtoTBR);
  // Transmit "s" to TBR
  RS485Serial.flush();
  // Wait until the all has been printed before proceeding
  digitalWrite(SSerialTxControl, RS485Receive);
  // Place MAXRS485 in receiving mode
}
```

```
//------------------READ FROM SERALPORT-------------------------------------


String rs485Read(unsigned long timeOutMS) {
  RS485Serial.setTimeout(timeOutMS);
  // Wait before proceeding
  digitalWrite(SSerialTxControl, RS485Receive);
  // Enableing receiving mode from TBR
  String readTBR = RS485Serial.readStringUntil('\n');
  // Place data form TBR in "s"
  sendtoTBR.trim();
  // Deletes white-spaces from the beginning and end of the string
  if (sendtoTBR.length() == 0) return "";
  // If nothing is read, return nothing.
  return readTBR;
}

//  ------------------------------------------------------------
```