

## REPORT FROM 6<sup>TH</sup> SEMESTER PROJECT SPRING 2016

PRH612 Bachelor thesis

IA6-5-16

### Read, control and communication unit for manholes



**Address: Kjølnes ring 56, N-3918 Porsgrunn, Tel: +47 31 00 80 00, [www.usn.no](http://www.usn.no)**

Bachelor's programmes – Master's programmes – Ph.D. programmes

## REPORT FROM 6<sup>TH</sup> SEMESTER PROJECT SPRING 2016

**Course:** *PRH612 Bachelor thesis*

**Title:** *Read, control and communication unit for manholes*

This report is a part of the evaluation result for the course

**Project group:** *IA6-5-16*

**Availability:** *Open*

### Group participants:

*Courage Asemota*

*Stine Marie Bakke*

*Amar Muminagic*

*Quynh Thao Nguyen*

*Oriol Pausas*

**Supervisor:** *Morten Pedersen*

**External examiner:** *Jan Petter Hestvik*

**Examiner:** *Hans Petter Halvorsen*

**Project partner:** *Willy Durholt*

**Approved:** \_\_\_\_\_

### Summary:

The background to this report was the bachelor thesis in the 6<sup>th</sup> semester of Computer Science and Industrial Automation at University College of Southeast Norway.

The goal of the project was to improve an existing system for intelligent manholes. The new system will be exposed for hazardous and wet environment and needs to be battery driven, which means that all components need to be water and gas proof with low power consumption, power supply from 3.3V – 5.0V. In addition, the system should be cheaper than the current system.

The system will be monitored every hour. All measurements will be stored in the microcontroller's memory in an array and send to the database 3 times per day. Alarms will be activated whenever the cover is opened or the water level, gas content or temperature values are over critical values. GSM with 2G will be used for data transfer

The group took advantage of previous project experience and programming skills to improve the existing system.

At the end of the project, the system was considered adequate but not optimal. Further improvements will be needed.

The group is satisfied with the result and experience the group got from the project.

## PREFACE

This bachelor thesis was completed by a group of five students of engineering. Four of the group members were students of Computer Science and Industrial Automation at University College of Southeast Norway. The last member was an exchange student of Polytechnic University of Catalonia in Spain, studying Industrial Electronic and Automation Engineering. The project was carried out in the 6<sup>th</sup> semester, spring 2016.

Data tools that were used: Energia, CA Erwin Modeler, Microsoft Project 2016, Microsoft Word 2013, Microsoft Excel 2013, Microsoft Visio 2013, Microsoft Visual Studio, Microsoft SQL Server 2012, Fritzing, Google Drive, Kicad and Arduino

Everyone can read the report; however, there is an advantage with basic knowledge and understanding with automation and control system.

The picture in the front page was taken by Oriol Pausas

Porsgrunn, date:

\_\_\_\_\_

Courage Asemota

\_\_\_\_\_

Stine Marie Bakke

\_\_\_\_\_

Amar Muminagic

\_\_\_\_\_

Quynh Thao Nguyen

\_\_\_\_\_

Oriol Pausas

## NOMENCLATURE

ANSI- American National Standard Institute  
CE- Communaute Europeenne  
CISC- Complex Instruction Set Computer  
CMD 1- Command  
EDGE- Enhanced Data rates for GSM Evolution  
EEPROM- Electrically Erasable Programmable Read-Only Memory  
GB- Gigabyte  
GND- Ground  
GPRS- Global Packet Radio Service  
GSM- Global System for Mobile Communication  
GUI- Graphical User Interface  
IDE- Integrated Development Environment  
IP- Ingress Protection rating  
ISO- International Organization for Standardization  
LSD- Low Self Discharge  
mA- milliAmpere  
MOSFET- metal–oxide–semiconductor field-effect transistor  
ms- millisecond  
NEMA- National Electrical Manufacturers Association  
NOR-  
OTP ROM- One-time programmable Read-only Memory  
RAM- Random Access Memory  
RF- Radio Frequency  
RISC- Reduced Instruction Set Computer  
RoHS- Restriction of Hazardous Substances  
RTD- Resistance Temperature Detectors  
SCADA- Supervisory Control And Data Acquisition  
SD card- Secure Digital Memory card  
SDHC cards- Secure Digital High Capacity  
SPI- Serial Peripheral Interface Bus  
SQL- Structured Query Language  
VOC- Volatile organic compounds

# TABLE OF CONTENT

Preface.....	3
Nomenclature .....	4
Table of content.....	5
<b>1 Introduction .....</b>	<b>7</b>
1.1 Reader guide.....	7
<b>2 Research .....</b>	<b>9</b>
2.1 Visits.....	9
2.1.1 <i>Ulefos Jernværk</i> .....	9
2.1.2 <i>Skien municipality</i> .....	10
2.1.3 <i>HSN Vestfold</i> .....	10
2.2 Microcontroller .....	11
2.2.1 <i>Which one is better for this application?</i> .....	13
2.2.2 <i>General properties of MSP430 family</i> .....	13
2.2.3 <i>Which device type is better to choose?</i> .....	14
2.2.4 <i>Ultra-low-power:</i> .....	14
2.2.5 <i>FRAM-based MCUs</i> .....	14
2.2.6 <i>MSP430FR5969 Launchpad</i> .....	15
2.3 Communication .....	16
2.3.1 <i>GSM/Antenna</i> .....	16
2.3.2 <i>SD-card</i> .....	23
2.3.3 <i>Bluetooth</i> .....	25
2.4 Sensors .....	26
2.4.1 <i>Water level sensor</i> .....	26
2.4.2 <i>Cover sensor</i> .....	27
2.4.3 <i>Gas sensor</i> .....	28
2.4.4 <i>Temperature sensor</i> .....	29
2.4.5 <i>Gas proof enclosure and connector</i> .....	30
2.5 Software research.....	30
2.6 Battery.....	31
2.6.1 <i>Battery calculations</i> .....	33
2.7 Ideas implemented .....	35
2.8 Cost estimation .....	36
<b>3 Overview .....</b>	<b>38</b>
3.1 System overview .....	38
3.2 Unit overview.....	39
3.3 Software overview .....	40
<b>4 Schematic of the Unit .....</b>	<b>41</b>
4.1 Unit battery.....	42
4.2 MOSFET.....	42
<b>5 Software .....</b>	<b>43</b>
5.1 Software Module .....	43
5.2 Software Application .....	43
5.2.1 <i>Software application flow chart</i> .....	43
5.2.2 <i>Class diagram</i> .....	45
5.2.3 <i>Use Case Diagram</i> .....	45
5.3 Database server .....	46
5.3.1 <i>ER Diagram</i> .....	46
5.3.2 <i>Table script</i> .....	47
<b>6 Unit Programming .....</b>	<b>48</b>

<b>6.1 Microcontroller .....</b>	<b>48</b>
6.1.1 <i>What is Energia?</i> .....	48
<b>6.2 Specification: Unit block diagram. ....</b>	<b>48</b>
<b>6.3 Design: Code flow diagram. ....</b>	<b>49</b>
<b>6.4 Implementation .....</b>	<b>51</b>
6.4.1 <i>Controlling the time using RTC_B.</i> .....	51
6.4.2 <i>Controlling the low power mode &amp; GPIOs</i> .....	52
6.4.3 <i>Reactivating the pins</i> .....	53
6.4.4 <i>Creating the array</i> .....	54
6.4.5 <i>Convert and send the data</i> .....	56
6.4.6 <i>Converting the array into a string</i> .....	56
<b>7 Testing .....</b>	<b>58</b>
7.1 Unit test.....	58
7.1.1 <i>Input</i> .....	58
7.1.2 <i>Expected output</i> .....	59
7.1.3 <i>Actual output</i> .....	60
7.1.4 <i>Testing sensors</i> .....	62
7.1.5 <i>Testing antenna/GSM and SD card</i> .....	62
7.1.6 <i>Testing power consumption</i> .....	62
7.2 Software test .....	62
<b>8 Discussion.....</b>	<b>64</b>
8.1 Components not finished .....	64
8.1.1 <i>SD-card</i> .....	64
8.1.2 <i>GSM and database connection (CRITICAL)</i> .....	64
8.1.3 <i>Gas proof box and connectors</i> .....	64
8.1.4 <i>Battery misunderstanding</i> .....	65
8.1.5 <i>Sealing the box with polyurethane</i> .....	65
8.2 Implementations in the future.....	65
8.2.1 <i>GSM and database connection</i> .....	65
8.2.2 <i>Use of GSM technology</i> .....	65
8.2.3 <i>Implementing a SD-card</i> .....	66
8.2.4 <i>Implementing a camera inside the manhole</i> .....	66
8.2.5 <i>Bluetooth</i> .....	66
8.2.6 <i>Gas proof box and connectors</i> .....	66
8.2.7 <i>Considering separation or unification of batteries with unit</i> .....	67
8.2.8 <i>Safety measures</i> .....	67
8.2.9 <i>Gas sensor readings</i> .....	67
8.2.10 <i>Smaller system</i> .....	67
8.2.11 <i>Switch sensor for water (CRITICAL)</i> .....	67
8.3 Comparison of Xepto unit vs. new unit .....	68
<b>9 Summary .....</b>	<b>70</b>
<b>References .....</b>	<b>71</b>
<b>Attachments.....</b>	<b>74</b>

# 1 INTRODUCTION

The project was based on developing and improving an existing intelligent manhole system for Ulefos Jernværk.

The assignment background was that Ulefos Jernværk had cooperated with Xepto earlier and they had developed a system to monitor water level in manholes. The assignment is to improve the existing system. The new system should contain two parts: Communication unit, and control unit with a battery package. It should be easy to service, easy to install and have a low cost. Furthermore, that data acquisition and status of alarms easily can be used by other systems. The collected data can be analyzed later.

The goal with the project is to assess the current solution for intelligent manhole covers, look at the cost-efficiency and the functionality of the solution. The solution should be an improvement of the current solution. Meaning a cheaper product than today has longer lifetime, and a greater functionality. The current solution sends an alarm, only when the water flows over, and the solution can only send information, not receive. An improvement would be to read water level when wanted and have an option for the unit to receive a command to read status. With the current solution, the whole manhole cover with seat has to be changed. This is for the alignment of the cover-sensor. The cover sensor is a mechanical switch implemented in the cover, with the unit. This again gives the customer a higher cost (ca. 1000% more expensive). A better solution is to have a unit that can be added on the side of the manhole, or wherever the customer wants, with the possibility to add features later, and without the need to change the cover and seat, meaning the cover switch/sensor is implemented in the unit or on the side of the manhole, not in the cover, as it is in the current solution.

Methods to reach the goal were to use C++ knowledge to program the microcontroller from Texas Instruments with Energia® software, SQL Server to create the database and Visual studio to set up communication between server and database.

It was decided to restrict the project to measure water level, gas content, temperature in the unit (box) and control the present of the cover. The status of the manholes will be stored in the database and showed in the GUI.

The assignment, goal formulation, WBS and progress plan are all added as attachments for further reading.

- Attachment A: Assignment
- Attachment B: Goal formulation.
- Attachment C: WBS
- Attachment D: Progress plan

## 1.1 Reader guide

In the report, the word “system” represents the whole system “Read, control and communication unit for manholes”. “Unit” represents microcontroller, sensors, hardware and code for hardware. “Software” represents GUI, database and installation.

- Chapter 1 is about Introduction
- Chapter 2 is about Research

- Chapter 3 is about System Overview
- Chapter 4 is about Schematic of the unit
- Chapter 5 is about Software
- Chapter 6 is about Unit programming
- Chapter 7 is about Testing
- Chapter 8 is about Discussion
- Chapter 9 is about Summary



## 2 RESEARCH

This chapter is about the research that was performed. It was important to find components that met the requirements for the system. The components will be exposed to wet and hazardous environment like moisture and gas. Moreover, the system has to remain alive during a long time, meaning years, therefore, the power consumption has to be low.

It has been important to look for components that have/are :

- Low power consumption
- Power supply from 3.3V-5.0V
- Waterproof
- Gas proof
- Withstands temperature between -40 to +70 degrees Celsius

### 2.1 Visits

In the course of the research period, the project group visited different sources. The first visit was Ulefos Jernværk at the beginning of the research period. It was important to know what Ulefos expected this project group to do and what they wanted to improve in the existing system.

The next visit was Skien municipality. Skien is one of Ulefos's customers. They have tested the existing system and the group wanted to know the most important measurements that they needed to monitor and how the system should be improved.

The last visit was at University College of Southeast Norway in Vestfold. Since the visit was conducted late in the process, the purpose of the visit was to see if they had any solution that can be used in the future system.

#### 2.1.1 Ulefos Jernværk

Some of the most important ideas of what Ulefos wants are described below.

About the system:

- Appealing and robust design
- Easy to integrate
- Applies smartphones and PC to facilitate the work of maintenance personnel and planners
- High life endurance and good performance
- Full control of the manhole covers
- Design the components box that ice growing could not affect the box. The box can be placed under the cover

About the control and monitoring:

- Monitoring water level
- Monitoring if the manhole cover is positioned correctly
- Monitor the need for the collection of sludge
- Monitor whether the manhole cover is open
- Possible to find a cover in bad conditions.
- Camera can take situation picture of manhole
- Status of water flowing in the pipes

- Status and control of valves and pumps.
- Measurement of temperature. Important if the manhole has equipment that is critical temperature
- Counting of replications

### 2.1.2 Skien municipality

The important ideas the project group obtained from visiting Skien municipality is described below.

- There are many different manholes, with different sizes and different locations.
- Skien has many pump stations, Wastewater in an area comes to a pump station, and every pump station has its own control system. The pressure sensor detects the blocked water line.
- Manual valves are used to close the water pipes so that workers can go down in the manholes.
- Intelligent system from Xepto was placed under the manhole cover, they used it and changed battery package once. The next time the workers checked out the system because of malfunction, they found out that the battery was ok, intelligent box looked fine externally, but all component inside the box were corroded. They thought the problem was H<sub>2</sub>S gas and moist. Skien sent the system back to Xepto.
- They replaced the nonworking system with a new system placed inside the manhole on wall. Skien municipality think it is better.
- They do not have any problem with ice growing as Ulefos mentioned (both under cover and beside the wall).
- Under the cover, there are two main points, the presence of moist and the H<sub>2</sub>S gas flowing up to the manhole.
- They currently only have a pressure sensor that can detect if the water rise up more than 3 meters.
- Current antenna is not good; it has poor signal strength. There is no connectivity at all.
- Each worker has a gas sensor placed on their leg when they go down in the manhole. The gas sensor may not detect gas before the worker steps down in the manhole, then, they already are exposed to the gas.
- They expect the group to make a system that is waterproof, gas proof, with better antenna and a better battery package.

### 2.1.3 HSN Vestfold

The project group visited the University College of Southeast Norway in Vestfold. The group was received by the head of the laboratory of the «Institute for micro- and nano system technology», Mr. Ole Henrik Gusland. He presented and explained some of projects that they have done. These projects are based on developing pressure sensors for various purposes like: airbag for cars, altitude measurement in planes, measure pressure of veins in the heart, pressure controller for tires and blood pressure.

Later on, he took us on a tour around the research department. The group saw some of the machines and equipment they use. Figure 2.1 is picture taken from the tour.

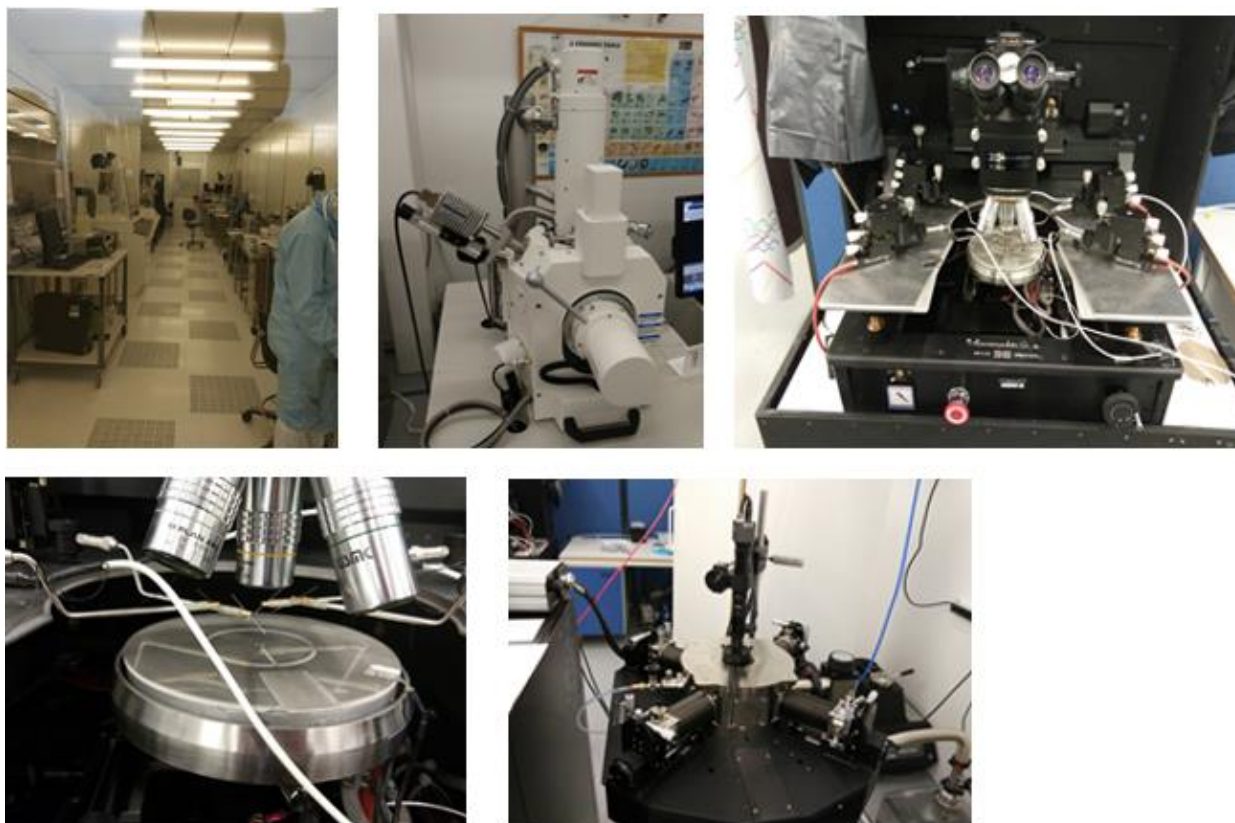


Figure 2.1: Picture from the visit in Vestfold

After the tour, the group explained the project goals, he gave a suggestion on level measurement using pressure sensor. However, this idea did not meet the criteria of the project. Therefore, he might thought it would be good to visit another company related to the group project, Aiwell in Tønsberg.

Aiwell is a small company, and work with water flow systems. They were very interested in the group project and they wanted to cooperate. They said they could provide every sensor that the group needs. However, they had only one sensor that did not meet the criteria of the group. Furthermore, there was a misunderstanding, they thought the group were students of water and building engineering.

This was a very interesting and informative visit, however, it was not relevant for the group project solutions.

As a conclusion for this chapter, it had been a misunderstanding, they thought the group project needed pressure solutions.

## 2.2 Microcontroller

A microcontroller is a small computer (SoC) on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of Ferroelectric RAM, NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general-purpose applications consisting of various discrete chips.

Microcontrollers are used in automatically controlled products and devices, such as automobile

engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. [1]

Figure 2.2 gives a short overview of the specifics to different microcontrollers, bits, memory devices, instruction set and architecture.

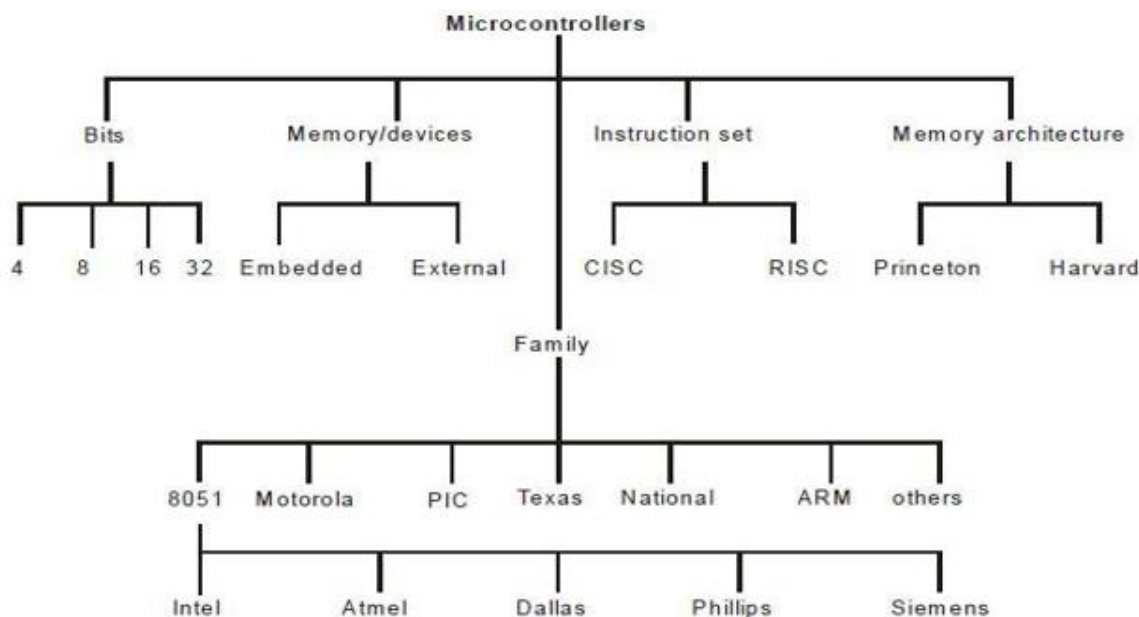


Figure 2.2: Microcontrollers [2]

By choosing the needed specifics for the task in hand, it is possible to find the most suitable microcontroller.

The bits defines where the arithmetic and logic operations are done. 8-bit microcontrollers can use 8 bits, meaning a range of 0x00 - 0xFF (0 - 255), 16-bit microcontrollers can use 16 bits 0x0000 - 0xFFFF (0 - 65535), and so on. Higher bit also increases precision and performance.

Memory devices defines if a microcontroller has embedded memory or external memory.

Instruction set is divided in two sets: CISC and RISC. CISC is a Complex Instruction Set Computer. It allows the programmer to use one instruction in place of many simpler instructions. RISC: Stands for Reduced Instruction set Computer; this type of instruction sets reduces the design of microprocessor for industry standards. It allows each instruction to operate on any register or use any addressing mode and simultaneous access of program and data. [2]

Architectures are split in two types: Harvard and Von Neumann/Princeton. Harvard architecture is defined by that the instruction memory and the data memory are separate, controlled by different buses, and sometimes have different sizes. For microcontrollers, the instructions are usually stored in "read only" memory, and data is in RAM or registers. Princeton is defined by data and instructions share memory space, so one could do things like dynamic compilation to generate instructions in RAM and then execute them.

Some different microcontroller families are ARM, 8051, AVR, PIC, MSP430. The biggest differences and reasons for choosing one over the other is price and power consumption. For the read, control and communication unit, the power consumption was the most important part. [4]

Table 2.1 gives a brief overview of the different microcontroller families.

Table 2.1: Different microcontrollers

	ARM	8051	AVR	PIC	MSP430
Bus Width	32-bit mostly also available in 64-bit	8-bit for standard core	8/32-bit	8/16/32-bit	16-bit
Communication Protocols	UART,USART, LIN,12C,SPI,CAN, USB, Ethernet, 12S, DSP, SAI(seial audio interface), IrDA	UART, USART, SPI, 12C	UART, USART, SPI, 12C ( special purpose AVR support CAN, USB, Ethernet)	PCI, UART,USART, LIN, CAN, Ethernet, SPI, 12S	UART,USART, LIN, 12C, SPI, 12S, IrDA
Speed	1 clock/ instruction cycle	12 clock/ instruction cycle	1 clock/ instruction cycle	4 clock/ instruction cycle	6 clock/ instruction cycle
Memory	Flash, SDRAM, EEPROM	Rom,SRAM, FLASH	Flash, SRAM, EEPROM	SRAM, FLASH	SRAM, FLASH
ISA	RISC	CISC	RISC	Aome feature of risc	Some feature of RISC
Memory Architecture	Modifie Havard architecture	Von Neumann architecture	Modified Harvard	Harvard architecture	Von Neumann architecture
Power consumption	Low	Average	Low	Low	Ultra low
Families	ARMv4,5,6,7 and Cortex series	8051 variants	Tiny, Atmega, Xmega, special purpsor AVR	PIC16, PIC17, PIC18, PIC24, PIC32	MSP430X, MSP430FR57xx, MSP430lxxx to x6xx series
Community	Vast	Vast	Very good	Very good	Average
Manufacturer	Apple, Nvidia, Qualcomm, Samsung, Electronics, and TI, etc.	NXP, Atmel, Silicon Labs, Dallas, Cyprus, infineon, ect	Etmel	Microchip	TI
Cost	Low	Vary low	Average	Average	Average
Other Feature	High speed operation	Known for its Standard	Cheap, effective	Cheap	Known for Ultra low power operation
Popular Microcontrollers	LPC2148 ARM CORTEX-mo TO arm Cortex-M7, ect	AT89C51, P89v51, ect.	Atmega8, 16, 32, Arduino Community	PIC18Fxx8, PIC16f88X, PIC32MXXX	MSP430G2552, MSP430 launchpad

As seen in Table 2.1, for power consumption the MSP430 family is the family with least power consumption. [4]

### 2.2.1 Which one is better for this application?

There is not the perfect microcontroller for each application, even for this project. There are plenty of families with many different microcontrollers with many properties; however, the one that can be adapted better to this project has to be chosen.

It has been evaluating the characteristics of each  $\mu$ C, and, according to low power consumption property, the MSP430 is the family with more possibilities for this project.

### 2.2.2 General properties of MSP430 family

Several features make the MSP430 suitable for low-power and portable applications:

- The CPU is small and efficient, with a large number of registers.

- It is extremely easy to put the device into a low-power mode. No needs of special instruction: bits in the status register control the mode. An interrupt awakes the MSP430, and then, returns automatically to its low-power mode after handling the interrupt.
- There are several low-power modes, depending on how much of the device should remain active and how quickly it should return to full-speed operation.
- There is a wide choice of clocks. Typically, a low-frequency watch crystal runs continuously at 32 KHz and it is used to wake the device periodically. An internal, digitally controlled oscillator (DCO), which restarts in less than 1  $\mu$ s in the latest devices clocks the CPU. Therefore, the MSP430 can wake from a standby mode rapidly, perform its tasks, and return to a low-power mode.
- A wide range of peripherals are available, many of which can run autonomously without the CPU for most of the time.

### 2.2.3 Which device type is better to choose?

According to search the lowest power consumption microcontroller, the best one for this application is the MSP430FRxx with FRAM memory incorporated. They are the lowest power MCUs as shown in the picture Figure 2.3.

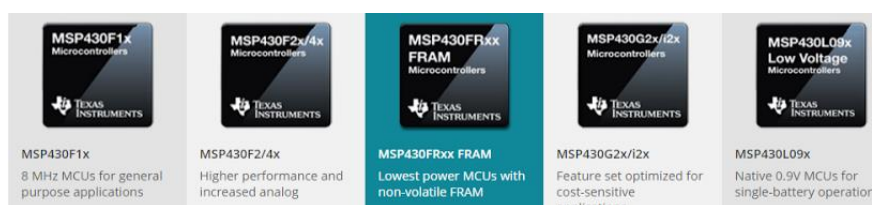


Figure 2.3: Possible devices available [6]

### 2.2.4 Ultra-low-power:

MSP ultra-low-power microcontrollers are ideal in applications where the majority of the microcontroller life is spent in standby. The latest addition to this family is the series of MSP430FRx FRAM-based microcontrollers, which offer high write speeds and nearly infinite endurance for enhanced data-acquisition and logging from sensors as well as security capabilities and new features that include IR modulation logic and dual analog-front-end is for rotation detection.

### 2.2.5 FRAM-based MCUs

The MSP430 ultra-low-power microcontroller (MCU) family features a series of FRAM-based devices. The MSP430FRxx FRAM series offers a full portfolio, see Figure 2.4, of devices ranging from 4 KB to 128 KB of non-volatile memory.

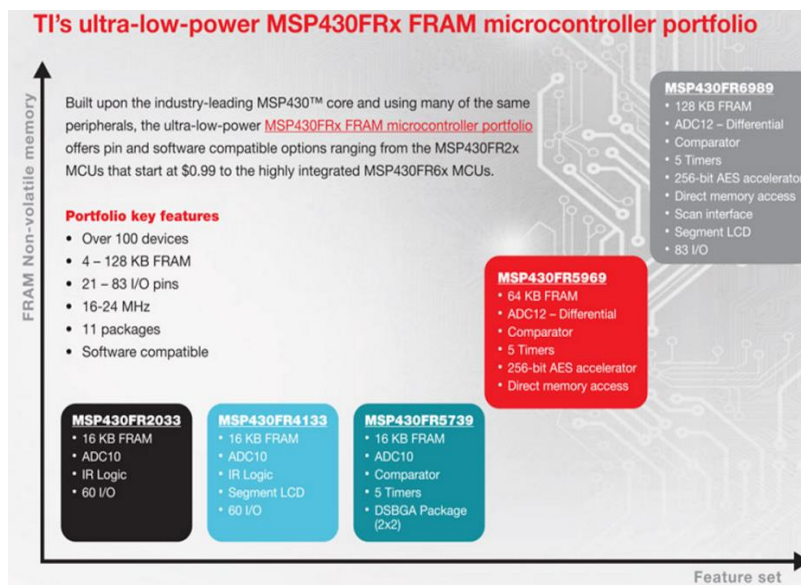


Figure 2.4: Portfolio of FRAM devices [7]

After seeing all the types of microcontrollers of MSP430FRxx it has to be chosen the best one for this application. [8]

Firstly, according to the Figure 2.4, it is possible to remove the MSP430FR4133 & MSP430FR6989 from the list because they both have an LCD screen built-in, which is useless. After it, MSP430FR2033, MSP430FR5739, and MSP430FR5969 can be chosen.

Continuing looking at the idea of the lowest power consumption, TI says its lowest power consumption microcontroller is MSP430FR5969.

Finally, the best option is the MSP430FR5969, which is the world’s lowest power MCU series (codename: Wolverine) with a medium-sized memory footprint (up to 64 KB FRAM).

### 2.2.6 MSP430FR5969 Launchpad



Figure 2.5: MSP430FR5969 Launchpad [9]

The MSP-family was chosen because of a key feature, which is not available in other microcontrollers. The feature is called LPM (Low power mode). This shuts down certain areas of the microcontroller to save power. Interrupts are available, that can wake up the microcontroller from any low power mode, and decide whether to continue sleep or return to active mode. [10][11]

Further explanation of each letter in “MSP430FR5969” can be found in Attachment E: MSP430FR5969 Letter explanation.

Table 2.8 gives a short overview of what the LPM’s can do.

Table 2.8: Overview of LPM’s

	CPU	MCLK	DCO	SMCLK	ACLK
LPM0	Inactive	Inactive	Available	Available	Available
LPM1	Inactive	Inactive	Inactive	Available	Available
LPM2	Inactive	Inactive	Inactive	Inactive	Available
LPM3	Inactive	Inactive	Inactive	Inactive	Available
LPM4	Inactive	Inactive	Inactive	Inactive	Inactive

Other features for the MSP430FR5969 are:

- Wide supply voltage range (1.8V - 3.6V)
- Ultra-low power modes: From 100  $\mu$ A/MHz active mode to 0.4 $\mu$ A in LPM3 (Low Power Mode 3)
- Ultra-low power FRAM up to 64 kb and unified memory with Program + Data + Storage in One Single Space
- Real-Time Clock (RTC) With Calendar and Alarm Functions, available in low power mode.
- Edge-Selectable Wake From LPM on All Ports
- Programmable Pullup and Pulldown on All Ports
- Enhanced Serial Communication
- UART With Automatic Baud-Rate Detection
- Flexible Clock System
- Fixed-Frequency DCO With 10 Selectable Factory-Trimmed Frequencies
- Low-Power Low-Frequency Internal Clock Source (VLO)
- EnergyTrace++ Technology measures MCU current consumption. Uses a DC-DC solution to measure the time density of charge pulses. The EnergyTrace++ window allows users to view power data and compare power consumption.

## 2.3 Communication

One of the main reasons of building this unit is for the ability to communicate, to the microcontroller, to the world without wires. Therefore, all the kind of communication chosen is wireless.

For being able to send all the data from the unit, it is going to be explained which type of GSM module was chosen. Moreover, for future abilities of the system, it had been added a Bluetooth, which is also explained later.

In addition to this chapter, it is been described the SD card and its properties, the communication between the microcontroller and SD card for storing information.

### 2.3.1 GSM/Antenna

This chapter will explain the GSM-technology and other relevant communication technologies.



### 2.3.1.1 GSM/GPRS

GSM/GPRS is used to establish communication between a computer and a GSM-GPRS system. Global System for Mobile communication (GSM) is an architecture used for mobile communication in most countries. Global Packet Radio Service (GPRS) is an extension of GSM that enables higher data transmission rate. GSM/GPRS module consists of a GSM/GPRS modem assembled together with power supply circuit and communication interfaces (like RS-232, USB, etc) for computer. The modem is the soul of such modules; see Figure 2.6.

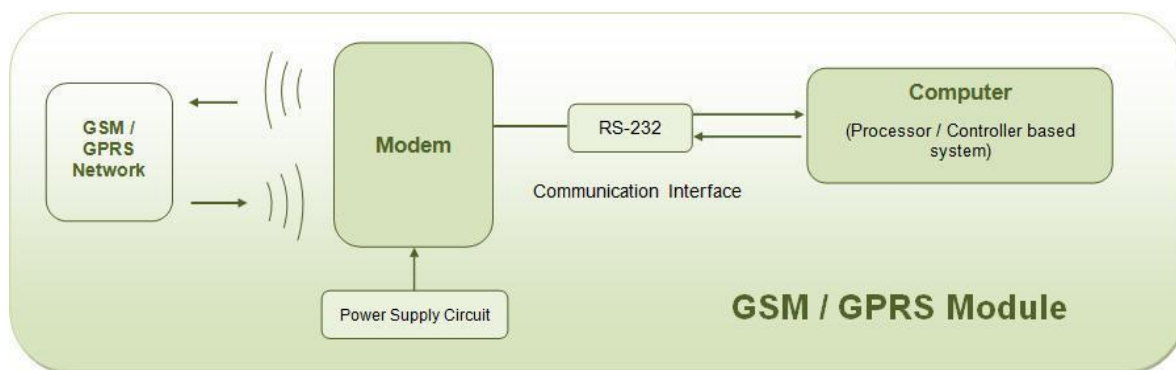


Figure 2.6: GSM-GPRS system [11]

### 2.3.1.2 Wireless modems

Wireless modems are the modem devices that generate, transmit or decode data from a cellular network, for establishing communication between the cellular network and the computer. These are manufactured for specific cellular network (GSM/UMTS/CDMA) or specific cellular data standard (GSM/UMTS/GPRS/EDGE/HSDPA) or technology (GPS/SIM). Wireless modems like other modem devices use serial communication to interface, and need AT commands for communication, with the computer (any microprocessor or microcontroller system).

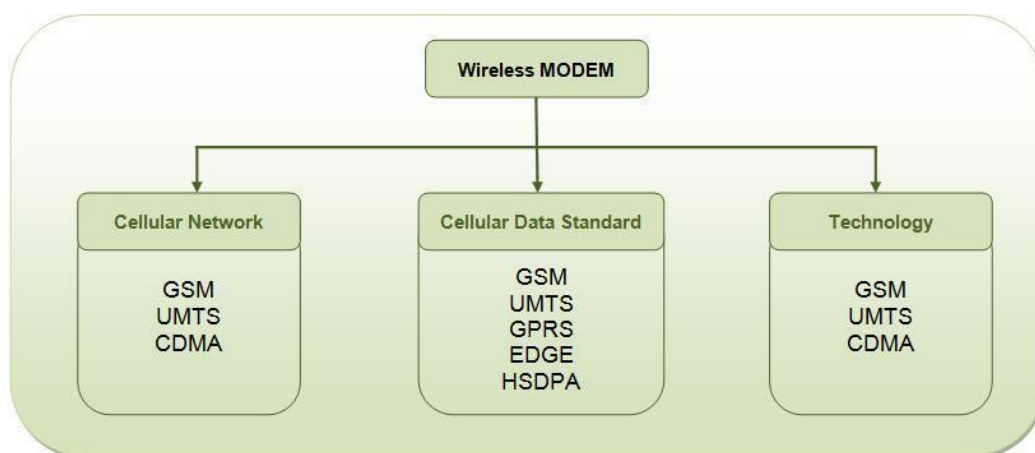


Figure 2.7: Wireless Modem [11]

### 2.3.1.3 GSM/GPRS Module

A GSM/GPRS module assembles a GSM/GPRS modem with standard communication interfaces like RS-232 (Serial Port), USB etc., so that it can be easily interfaced with a computer or a microprocessor / microcontroller based system. The power supply circuit is also built in the module that can be activated by using a suitable adaptor.

The GSM/GPRS module always needs a computer or external processor/controller to receive AT commands from. GSM/GPRS module itself does not provide any interface between the user and the network, but the computer to which module is connected, is the interface between user and network.

An advantage that GSM/GPRS modules offer is that they support concatenated SMS, which may not be supported in some GSM mobile handsets

### 2.3.1.4 Difference between GSM mobile and GSM module

A GSM mobile is a complete system in itself with embedded processors that are dedicated to provide an interface between the user and the mobile network. The AT commands are served between the processors of the mobile termination and the terminal equipment. The mobile handset can also be equipped with a USB interface to connect with a computer, but it may or may not support AT commands from the computer or an external processor/controller.

### 2.3.1.5 Current speed transmission and technology

Wireless technology is still in development, with different technologies to choose from for different units. Each successor to the mobile network technology has one numeric number followed by the alphabet 'G'. This is of the form 'xG'. Therefore, 2G means 2<sup>nd</sup> generation, 3G means 3<sup>rd</sup> generation, 4G means 4<sup>th</sup> generation and so on. Each generation is more advanced than the previous standard.

Currently, 4G is the latest standard, but many big organizations have already started their work with a technology in between 4G and 5G called 4G plus.

In Table 2.2, there is a short view of the current technology a speed available in the market. [13][14]

Table 2.2 GSM Technology available

Technology		Speed
GSM (Global System for Mobile Communications)	2G	9.6 Kbps
GPRS (General <u>packet</u> radio services)	2.5G	56 Kbps
EDGE (Enhanced Data rates for GSM Evolution)	2.75G	170 Kbps
<u>UMTS</u> (Universal Mobile Telecommunications System)	3G	384 Kbps
<u>HSPA</u> (software upgrade to <u>UMTS</u> )	3.5G	42 Mbps
<u>WiMAX/LTE</u>	4G	50 Mbps

Telenor© has a coverage map for the public to see their actual coverage in Norway, when choosing telecommunication operator. This is a pinpoint for choosing the wanted technology, with manholes being placed in all parts of Norway. The map is in Norwegian with the coverage separated in two categories:

- “Basis” - Basic connection quality.
- “Meget god” - excellent connection quality.

The blue dot represents University College of South East Norway, department Porsgrunn. The colors for comparing the technologies are: Yellow/orange for 2G, light blue for 3G, dark blue for 4G and purple for 4G plus. The Figure 2.8, Figure 2.9, Figure 2.10 and Figure 2.11 gives an overview of the different generations and their basic quality, for Norway. [15]

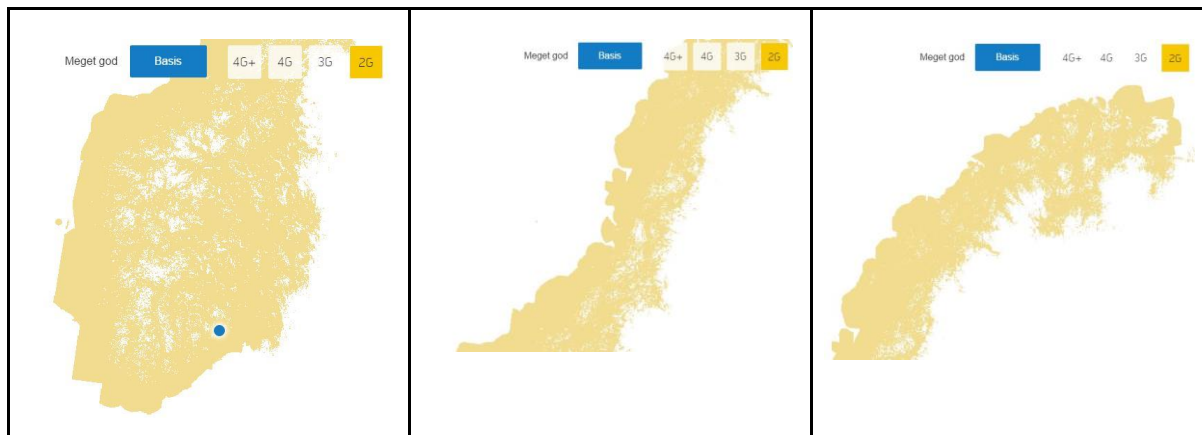


Figure 2.8: 2G connection quality Norway “basic” (south-, mid- and northern Norway)

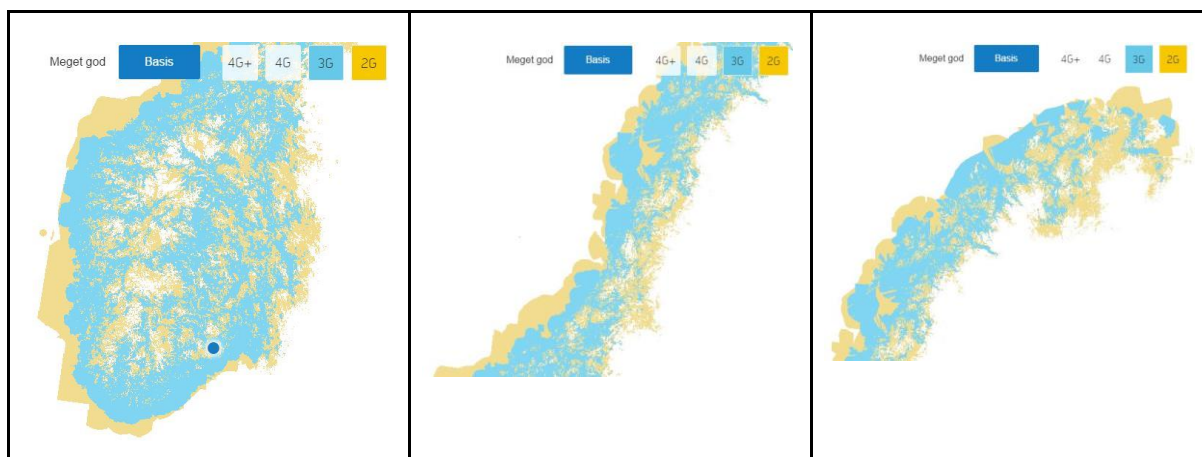


Figure 2.9: 2G and 3G connection quality Norway “basic” (south-, mid- and northern Norway)

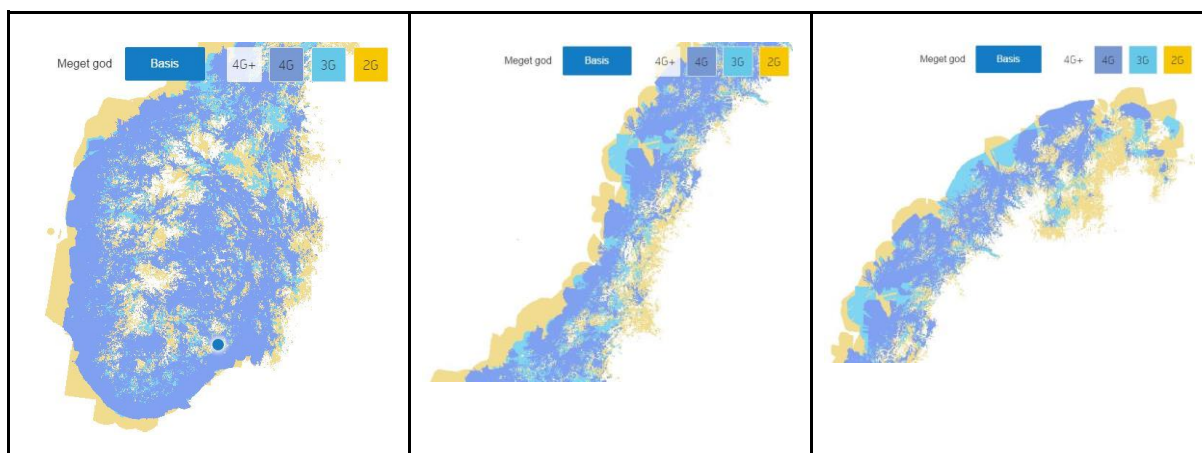


Figure 2.10: 2G, 3G and 4G connection quality Norway “basic” (south-, mid- and northern Norway)

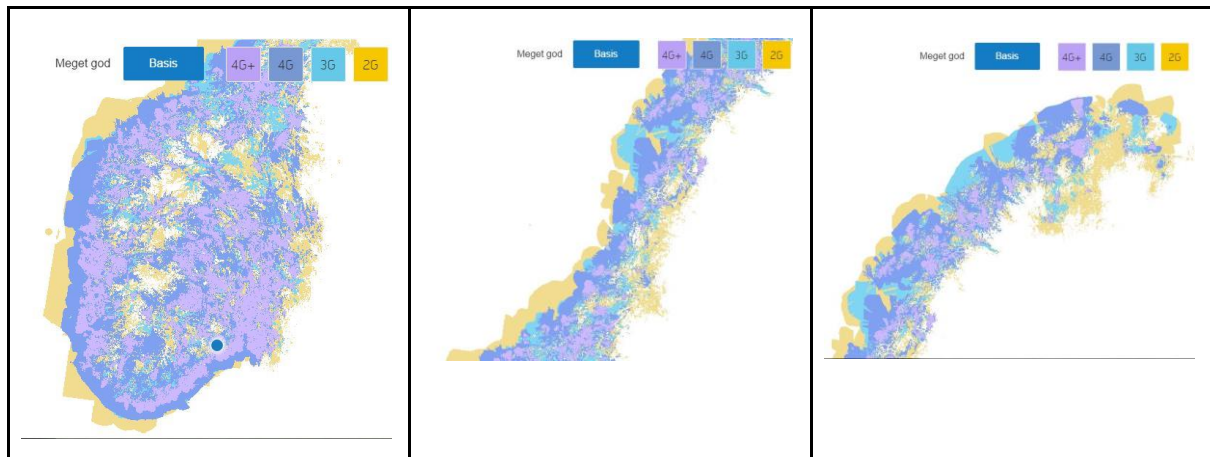


Figure 2.11: 2G, 3G, 4G and 4G plus connection quality Norway “basic” (south-, mid- and northern Norway)

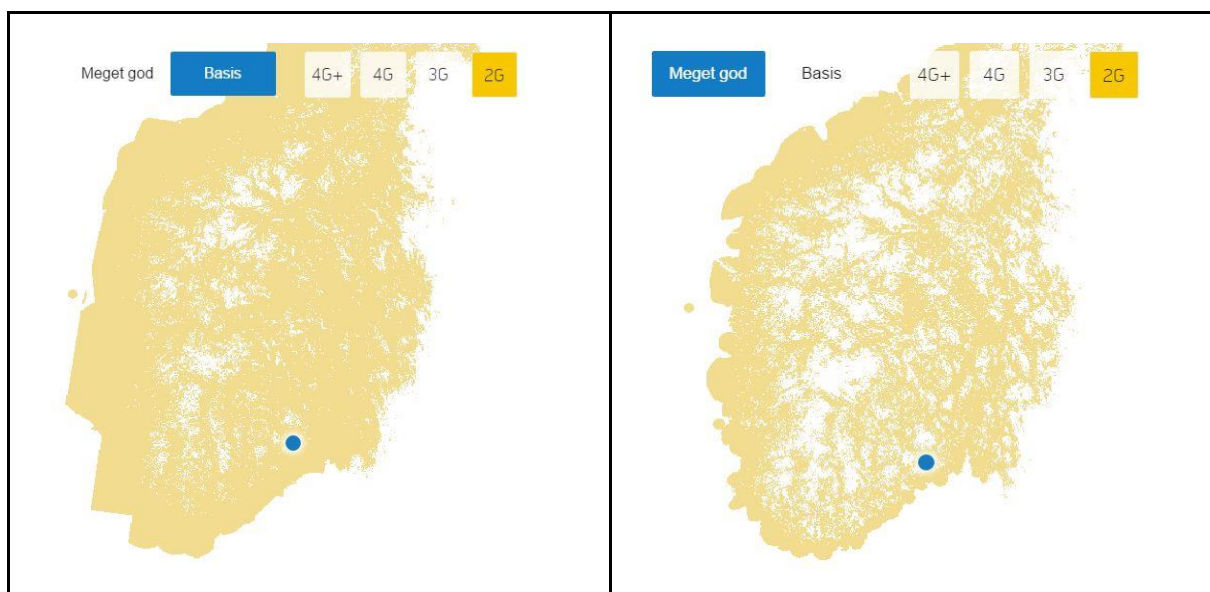


Figure 2.12: Basic 2G coverage compared to 2G excellent quality for South Norway

A good quality is limited by location, compared to a basic quality. Figure 2.12 gives an overview of the difference between basic 2G coverage, compared to excellent 2G quality. This difference is even greater when comparing a newer standard like 4G plus.

Basic 4G plus coverage compared to excellent 4G quality South Norway is shown in Figure 2.13.

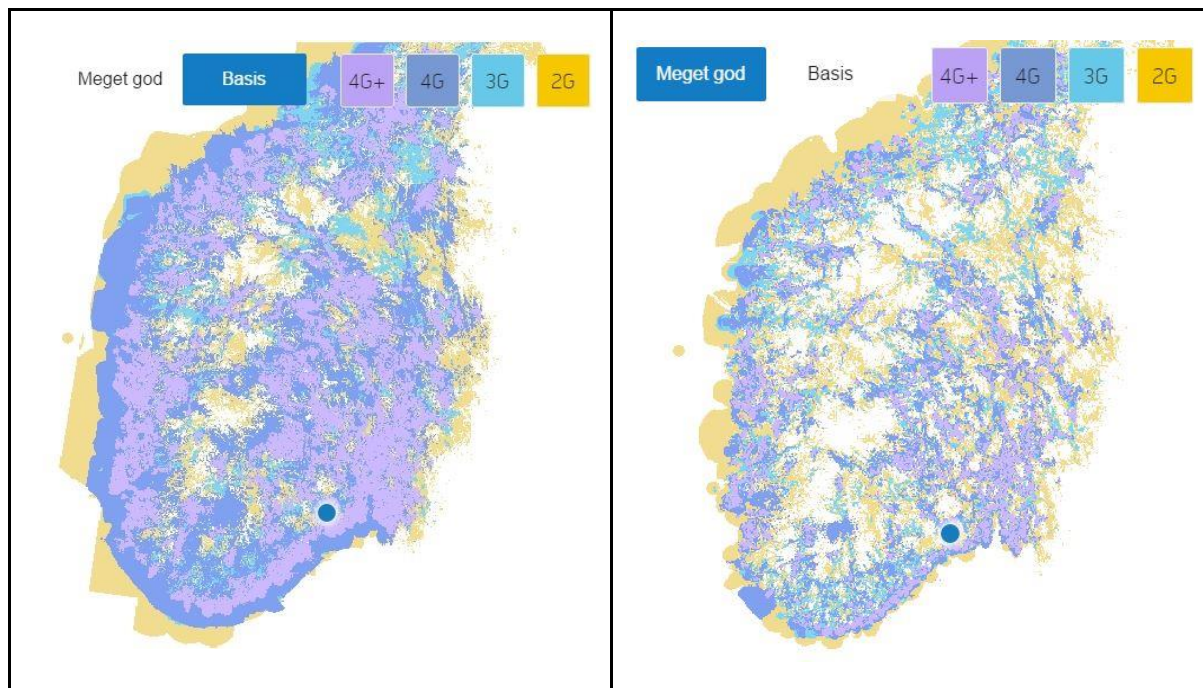


Figure 2.13: Basic 4G plus coverage compared to excellent 4G quality South Norway

The coverage and data speed is important when choosing the right technology for the read, write and communication unit. The data transmission is limited to sizes that are smaller than 1 MB, meaning the need for fast transmission is not an important factor. The more advanced technology, the more complex the code is for the programmer. In addition, as stated in the tables of connectivity, the unit should be connectable in all of Norway; therefore the 2G that is supported throughout Norway is the most suitable solution.

### 2.3.1.6 Choosing the right SIM module

The different modules of SIMCom are shown in the Figure 2.14, with comparisons. Also stated in the picture are the temperatures in which the modules can work.

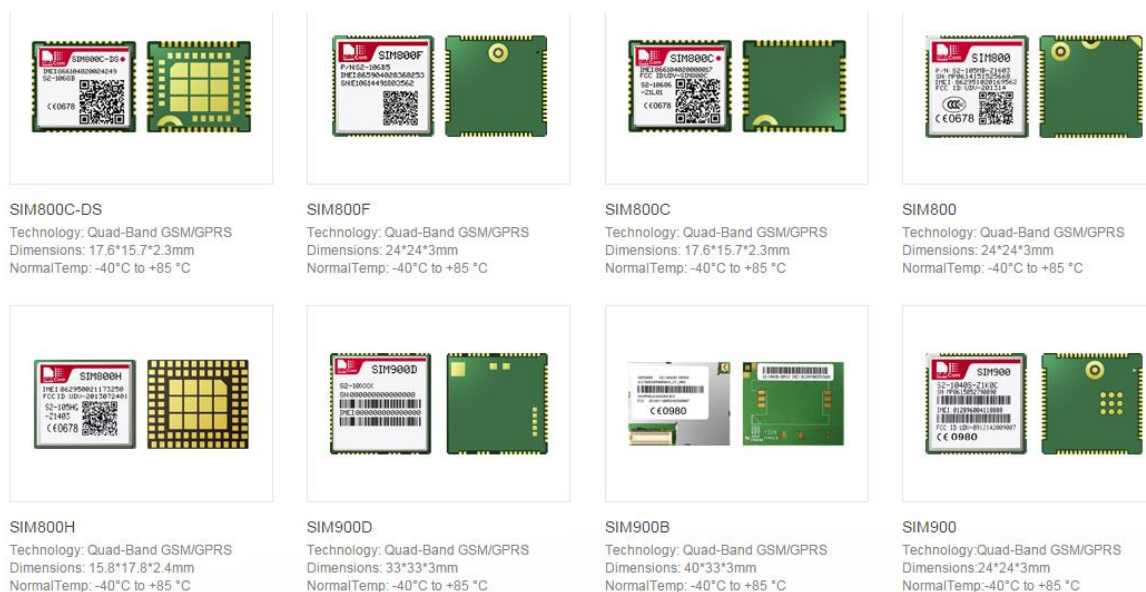


Figure 2.14: SIM chips available [16]

For choosing the most suitable GSM module, popularity and community support was the biggest key features. The chosen GSM module is SIMCom SIM800.

These are some of the SIM800 properties and key-features.

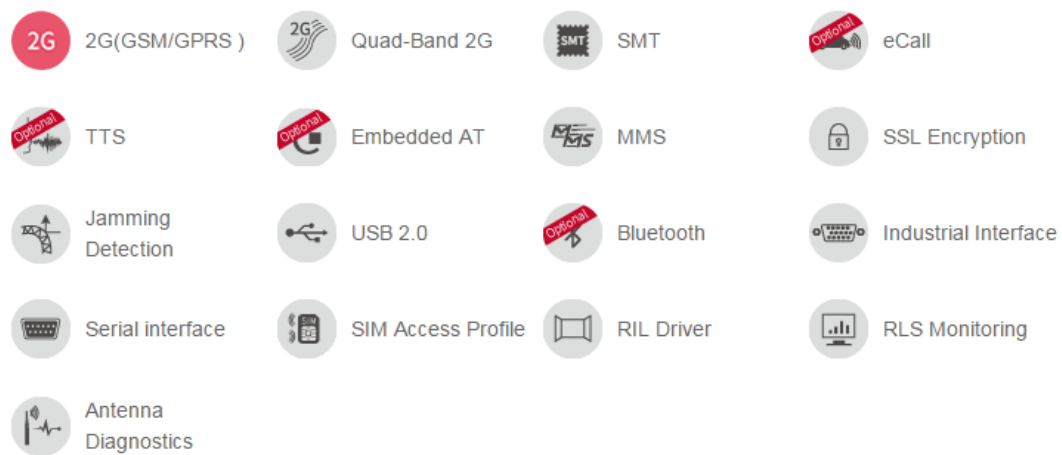


Figure 2.15: SIM800 Properties [17]

SIM800 is a complete Quad-band GSM/GPRS solution in a SMT type, which can be embedded in the customer applications.

SIM800 support Quad-band 850/900/1800/1900MHz, it can transmit Voice, SMS and data information with low power consumption. It can fit into slim and compact demands of customer design.

Datasheet: [17]

### 2.3.1.7 Final module chosen, SIM800L

SIM800L is a quad-band GSM / GPRS module, LGA package. Its stable performance, the appearance of small, cost-effective and can meet various needs of customers.

SIM800L operating frequency of GSM / GPRS 900 / 1800MHz, with low power consumption transmission of voice, SMS and data information.

The power supply of the module has to be connected to an external battery. That is because during the transmission, the GSM module has to be fed with 2 Amps as a peak, which is too much for the microcontroller, thus it can be burned.

The Figure 2.16 shown the SIM800L board ready to connect.



Figure 2.16: SIM800L Chosen

### 2.3.2 SD-card

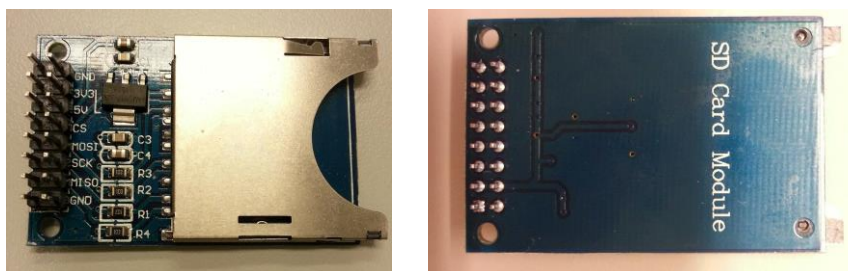


Figure 2.17: Socket for SD card (front and back views)



Figure 2.18: SD card used (front and back views)

(It was initially planned to use SD-card for storing information, although at the end it was not implemented. See chapter 2.3.2.1 Why SD card is removed from the microcontroller for more information)

The SD-card will be used to store local information. Every unit will have a SD-card. In Figure 2.17 and Figure 2.18 above, the SD-card and SD-card module are presented. SD-cards with more memory than 2GB are SDHC-cards and SPI for SDHC-cards is not supported. The unit will only store text, so there is no need for a larger SD card. SDHC is a newer standard, which means for Windows 7 computers and older, the user will need to have specific software to access the SDHC-cards. The lifetime of a SD-card is based on number of times it can be written to. The average lifetime as of today is 100,000 writing cycles.

Datasheet: [19][20]

There are different storage solutions to use with a microcontroller, narrowed down to EEPROM, SD and flash memory. SD has greater accessibility, with EEPROM needing more equipment to read. EEPROM has a max memory of 1MB, while SD-cards, that are not SDHC, have a max memory of 2GB. In addition, EEPROM is limited because the whole chip has to be erased in its entirety. Flash memory is an alternative to the two above, but with little community support. Another point is that bits do not limit the SD-card.

The type of brand of SD-card is chosen for the prototype of the unit. Current available SD-cards are Transcend 2GB, Integral 2GB, Canon 8MB, Canon 32MB and PNY Technologies 256MB (more SD-cards are available for ordering online) The SD-cards under 2GB are older cards that were acquired for testing, and to assess the necessity of size compared with read/write speed. A major point is the availability on the market for the different SD-cards. 2GB SD-cards are found commonly, because of their storage space.

Today SD-cards are often found in cameras, with high resolution meaning more pixels and size of the pictures exceeding 5MB per picture, meaning the cameras need greater storage. This is reflected by SD-cards only increasing in size (GB), but not all the technologies are compatible with microcontrollers (i.e. SDHC). SD-cards with storage space under 2GB are not common.

Furthermore choosing a SD-card with more space increases life expectancy. “The thinking behind this is simple: with a limit on the number of times data can be written to SD cards, and the fact that data written to the device should be spread out into untouched areas before going back to the beginning, there is less chance of writing to the same area of the card. Choosing 16GB, over 8GB, will cut by half the number of rewrites. In theory this will double the life expectancy of your storage.”

Table 2.3 below shows the different SD-cards size from different brands:

Table 2.3: SD-cards size from different brands

Brand	Size
Canon	8MB
PNY Technologies	256MB
Integral	2GB
Transcend	2GB

SD-cards are operating on 3.3V, which is the voltage of the chosen microcontroller. This means seamless connectivity with no adapters or converters. The power consumption of an idle SD-card is too high compared to the microcontroller and battery-life. This can be resolved with coding. The SD-card can be turned completely off, which means that the power consumption is 0A, when the card is not in use. When writing to a SD-card the power consumption is 75 mA, while reading from a SD-card is 60 mA. [21][22]

Table 2.4: Operating time SD-card

Timing	Typical	Maximum
Block Read Access Time	0.5 ms	100 ms
Block Write Access Time	0.5 ms	250 ms
CMD1 to Ready after Power-up	50 ms	500 ms
Sleep to Ready	1 ms	2 ms

### 2.3.2.1 Why SD card was removed from the microcontroller

In this chapter the equations and mathematics behind the reason for excluding the SD-card will be presented.

How many times can 64 kB (microcontroller memory) be written inside 2 GB (SD-card) of memory?

64 kB of FRAM from the microcontroller ads up to 2GB - 32768 times:

$$\frac{2.1 * 10^6 kB}{64kB} = 32768 \quad (2.1)$$



Meaning for one rewrite in a SD-card, there is a need for 32768 rewrites in the microcontroller. 32768 rewrites in a microcontroller per SD-card rewrite, times 100 000 (number of times a SD-card can be rewritten) is  $3.2768 \cdot 10^9$ :

$$32768 * 100000 = 3.2768 * 10^9 \quad (2.2)$$

As stated earlier, the microcontroller can be rewritten  $10^{15}$  times, meaning there is a deficit of approximately  $10^4$  for the number of rewrites in the SD-card.

$$10^{15} - 10^9 = 10^4 \quad (2.3)$$

What is the maximum life expectancy of the SD card? First, it is needed to establish number of writes per day, and how many writes the memory can handle. Number of writes per day is 24, and total writes that the memory can handle are 2700.

$$\frac{2700}{24} = 112.5 \quad (2.4)$$

The number of days the microcontroller can be written to is 112.5, until the information needs to be deleted, meaning a rewrite. The number of days until a rewrite, multiplied with total number of rewrites, equals to the theoretical lifetime of the microcontroller.

$$112.5 * 10^{15} = 1.125 * 10^{17} \quad (2.5)$$

Table 2.5: Lifetime of FRAM memory

Lifetime in days	$1.125 * 10^{17}$
Lifetime in years	$3.1 * 10^{14}$

This theoretical lifetime was only based on rewrite-cycle, with other factors also determining the lifetime.

### 2.3.3 Bluetooth

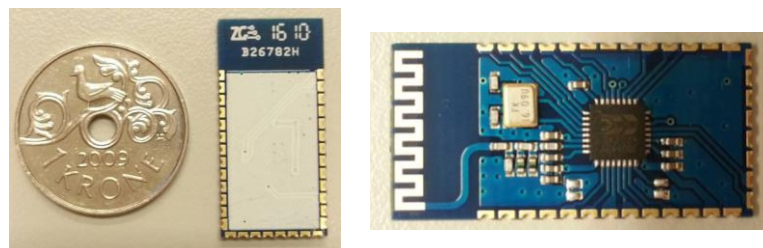


Figure 2.19: Bluetooth component

(It was initially planned to use Bluetooth for connectivity, although at the end it was not implemented. See 8.2.5 for more information).

The Bluetooth in Figure 2.19 is installed for future features. Customers can add camera and other components to the manhole, and connect to the Bluetooth not to break the gas-seal of the microcontroller. The Bluetooth is restricted to limited range and it cannot penetrate thick metals (manhole-cover) and asphalt. This is why the Bluetooth solution is only limited for connecting to devices inside the manhole. When the cover is open, the Bluetooth range increases.

This can be used if the control-center loses contact with the unit, meaning the GSM is not connected, or the unit is broken. This gives the opportunity for a worker to go to the broken unit and connect with Bluetooth to update the system, or repair any problems with the GSM.

The brands of Bluetooth offered are narrowed down to shell-Bluetooth devices and Bluetooth boards. The shell devices are too big for the chosen microcontroller. The Bluetooth boards are connected either to a breadboard or directly to the microcontroller.

The Bluetooth boards are commonly designed for hobbyists. This means few brands with industrial data-sheets, but big hobby communities for help.

The features of the Bluetooth are seen in Table 2.6

Table 2.6: Bluetooth features

Active mode current	45 mA
Sleep mode current	0.78 $\mu$ A
Power supply voltage	3.3V
Operating time	100 ms

## 2.4 Sensors

In this chapter it is explained the kind of sensors chosen to detect all the variables needed and all the logic statements. It is extremely important to understand how all of them work, knowing all the voltage ratings and the current drains, therefore is possible to design a schematic without errors.

### 2.4.1 Water level sensor



Figure 2.20: Ultrasonic sensor

Ultrasonic sensor, shown in Figure 2.20, will be used to measure water level in the manhole. If the water level rises to a critical level, an alarm will be sent to the database and presented in the GUI.

There are many ways to measure water level: Magnetic level gauge, RF Transmitter, Radar, Ultrasonic, Magnetic Switch, Float switch, RF switch, Vibrating fork and thermal dispersion, amongst others.

Ultrasonic sensor was chosen because it is a non-contact sensor. Ultrasonic level sensors measure the distance between the transducer and the surface, using the time required an ultrasound pulse to travel from a transducer to the fluid surface and back. The MB7052 has low power consumption; therefore, it is excellent for battery-based systems. The sensor is acoustic and electrical noise resistance. It is robust and withstands immersion up to half an hour without any harmful effect.

The MB7052 has been on the market since July of 2009, and is unlikely to be removed from the market in the near future. The sensor is CE marked and RoHS. Table 2.7 shows some of sensor features:

Table 2.7: Water level Sensor features

Power supply	3- 5.5V
Operational current	3.4mA
Operational temperature	-40 to 70 degrees
Detection range	20- 765 cm
Sealing	IP67

Datasheet: [25][26][27][29]

## 2.4.2 Cover sensor



Figure 2.21: Cover sensor

Cover sensor, shown in the Figure 2.21, is used to detect if the manhole cover is opened, then an alarm will be sent to the database and presented in the GUI.

There are many types of sensors for detecting the status of the cover: Hall Effect sensor switch, motion sensor, and mechanical sensor. The mechanical switch was chosen because of its simplicity and the easiest way to install and program, and as it is a switch, it does not consume power.

SSCEB31C is one of Honeywell micro switch medium-duty limit switch series, which has been designed for reliability in the harshest of environments. It has the possibility of being submerged

into water or oil. Nema 6 is temporarily submersible, 6P withstands occasional prolonged submersion.

Its robust design, corrosion limiting features, flexible PUR cable and super sealing allows it to operate fully submerged in a range of fluids. Its small size allows operation in confined spaces. A range of different actuators offers flexibility in application. The sensor is CE-marked.

Following in Table 2.8 is some of the sensor features.

Table 2.8: Features specification

Max. Current rate	5A
Supply voltage	DC
Housing Material	Zinc Die- Cast
Sealing	IP68, NEMA 1, 3, 6, 6P, 13
Operating temperature	-40 to 90 degrees

Datasheet: [24]

### 2.4.3 Gas sensor



Figure 2.22: TGS 2602 gas sensor

The gas sensor, shown in the Figure 2.22, will measure hydrogen sulfide gas ( $H_2S$ ) in the manhole. It will be placed outside the enclosure to be able to check the gas content. When the gas reaches a set level, the microcontroller will send an alarm, the data will be sent and stored in the database and presented in the GUI.

There are three different gas sensors, which can detect the  $H_2S$  gas: TGS2602, MQ136 and TGS 825. TGS 825 is the best sensor based upon Ceramic materials and is therefore resistant to severe environmental conditions, however it was impossible to find a store that provides TGS 825, and thus it had to look up at other alternatives. The sensitivity of the MQ136 will be reduced or the unit damaged when in contact with water, therefore the TGS2602 was chosen.

TGS 2602, has high sensitivity to VOCs and odorous gases. It has low power consumption and the external circuit that it needs to function is simple.

The sensing element is comprised of a metal oxide semiconductor layer formed on the alumina substrate of a sensing chip together with an integrated heater. In the presence of detectable gas, sensor conductivity increases depending on gas concentration in the air. The electrical circuit can convert the change in conductivity to an output signal, which corresponds to the gas concentration.

The TGS 2602 has high sensitivity to low concentrations of odorous gases such as ammonia and H<sub>2</sub>S generated from waste materials in office and home environments.

The sensor requires two voltage inputs: heater voltage and circuit voltage. The heater voltage is applied to the integrated heater in order to maintain the sensing element at a specific temperature optimal for sensing. Circuit voltage is applied to allow measurement of voltage across a load resistor, which is connected in series with the sensor.

The sensor has been on the market at least since 2005 and it is not possible to know how long it will be on the market. It is only finished products that incorporating CE Directives, therefore this sensor is not CE-marked. Table 2.9 shows some of the sensor specifications:

Table 2.9: Specification

Detection range	1-30ppm
Heater voltage	5.0V AC/DC
Circuit voltage	5.0V DC
Heater current	56±5mA
Average consumption	61mA
Response time	30s

Datasheet: [30]

#### 2.4.4 Temperature sensor



Figure 2.23: DS18B20 Dallas temperature sensor

The temperature sensor, shown in Figure 2.23, will be used for measuring the temperature inside the enclosure. Therefore, it is not necessary to have a sensor that is waterproof or gas proof. The sensor has a range of temperatures within -40 °C to +70 °C and has a low power consumption.

There are many different types of sensors: RTD, Thermocouple and Thermistor. It is possible to choose between analog and digital sensors. Since digital sensors are easier to work with than an analogue sensor, it is decided to choose a digital sensor.

There are many different digital temperature sensor to choose from, and many different brands, which offers some interesting options. It was chosen a sensor with three pins (table 2) technically called RTD.

The sensor that has been chosen, DS18B20 from Dallas Semiconductor, measure temperature in a range within -55°C to +125°C with an accuracy at ±0.5°C in a range of -10°C - +85°C which is more than adequate for the application, as specified in Table 2.10. This kind of sensors are called 1-wire bus protocol, it means that it requires only one data line for communication (transmit the temperature) with the microcontroller, and obviously, it needs the power and GND wire. It converts temperature to 12-Bit Digital Word in max 750ms. When the sensor reaches a set level,

the microcontroller will send an alarm, the data will be sent and stored in the database and presented in the GUI.

The sensor has been on the Market at least since 2007 and it is not possible to know how long it will be on the market. It is only finished products that are incorporating CE Directives, therefore this sensor is not CE-marked. Table 2.10 below shows the sensor specifications:

Table 2.10: Specification

Temperature range	-55°C to +125°C
Supply Voltage	3V-5.5V
Accuracy (-10°C - +85°C)	±0.5°C (-10°C - +85°C)
Accuracy (-85°C - +125°C)	±2°C
Standby Current	0.001mA
Active Current	1.5mA
Time Slot	60-120µs
Recovery time	minimum 1µs
Read Data Valid	15µs

Datasheet: [32]

### 2.4.5 Gas proof enclosure and connector

Gas proof-connector and enclosure is needed to implement to the system to protect all electronic component from moist and hydrogen sulfide gas. Two challenges are needed to be solved. Some gas proof connector and enclosure vendor have been contacted. The gas proof connectors are expensive. After a discussion with Mr. Willy Durholt from Ulefos Jernvæk, it was decided not to use the gas proof connector for this test, although it can be implemented in the future. Further information about the connector-offer and prices see Attachment F: Gas proof connector offer.

Following are the price for gas proof protectors from to difference vendors.

- Marechal Electric Technor vendor. Three connector with socket cost 220,50€  
For more detailed info., visit the website [32]
- Nema7 vendor. The enclosure with size 6 x8 x4 inch cost 625 \$.  
For more detailed info., visit the website [33]

## 2.5 Software research

The objective of the software is to display important information from the database and be easy for users to use. The program is expected to operate in Windows 10. The appropriate developmental tools have to be chosen to achieve these goals. Though the group is familiar to some certain developmental tools used in the University, it has to be determined if they are most appropriate for the task.

Visual studio is the most familiar developmental tool to the group, moreover it was chosen because it is a Microsoft product, which is the best to develop program for Windows.

Furthermore, it is the most suitable for the project because it has other built-in tools like form designer for building user's friendly GUI application, class designer and database schematics.

Same as Visual Studio, SQL Server is most popular for database development in the University, but was chosen because it is most appropriate to use SQL Server to develop database for programs built in Visual Studio. Moreover, SQL Server is consistent across new versions of SQL Server. Furthermore, SQL Server can be used to retrieve large amount of records from a database quickly and efficiently. Additionally, it uses a long-established standard, which is being adopted by ANSI and ISO. Non SQL database do not adhere to any clear standard. [34][35]

## 2.6 Battery

Every electronic component need a specific voltage and current to operate efficiently. Therefore, choosing the right battery that meets the voltage requirement of a device and at the same time live up to its expected level of performance is very important. Furthermore, with the intention to determine which battery is most appropriate to power the unit, analysis of three main type of batteries was carried out, these included Alkaline, Nickel Metal Hybrid (NiMH) and Lithium-ion batteries. [36] [37]

The following factors were considered very important in choosing the right battery for the unit because of the environmental conditions and circumstances of the places where the manhole would be used:

- Ability to withstand extreme temperature.
- Low rate of discharge under load
- Ability to be recharge
- Economic viability
- Environmental effect
- Low self-discharge in all temperatures
- Possibility to withstand hydrogen sulphide gas. the

Result of the batteries analyses is presented in Table 2.11.

Table 2.11: Batteries analyses

Properties	Non - rechargeable	Rechargeable		
	Alkaline	Alkaline	NiHM (LCD)	Lithium-ion
Self-discharge rate	Low	Low	Low	High
	Retain 80% if not used	Retain 95% if not used	Retain 75% if not used	Retain 98% if not in use
Performance at low drain rate	Very good	Good	Good	poor
Performance at high drain rate	Poor	Poor	Good	Excellent
Long shelf-life	Up to 5 years	Up to 7 years	5 to 7 years	4 - 6 years
Temperature	from -20° to 50°	from -18° to 55°	from -20° to 60°	from -40° to 60°
Energy density	Moderate	Low	High	High
Percentage of capacity avail. when used in cold temperature (0 degree)	31% – 75% (more capacity loss at higher drain rate)	35% – 75% (more capacity loss at higher drain)	91 %	-
Voltage output	1,5 V	1,5 V	1,2 v	3,7 V
Freezing temp. effect	Voltage drops rapidly	Voltage drops rapidly	Voltage drops but better performance than alkaline.	good performance, voltage dropping at the end of the cycle.
Over charge tolerance	Not application	low	moderate	Very low
Minimum discharge level	Can completely be discharged	Can completely be discharged	May completely be discharged	Cannot completely discharge. Minimum level is usually 1,2 V
Environmental issue (recycling)	Available for some alkaline	Available for some alkaline	Available	Available

From information in Table 2.11, it is seen that alkaline battery is best in self-discharging and in low-drain devices, but it is poor performance in low temperature and problem with recharging makes it less preferable.

Lithium-ion battery perform best in extreme temperature and has longest shelf life, but its poor performance in low-drain devices and problem with complete discharging of its capacity render it less preferable as well.

Nickel-metal-hybrid (LSD – Low self-discharge) battery is most preferred for the project because of its ability to perform well in low-drain devices and its rechargeable properties. Its ability to discharge completely and its low voltage drop during freezing temperature makes it stand out. [38][39]



## 2.6.1 Battery calculations



Figure 2.24: “Enesense” Battery Chosen [18]

The batteries chosen are from a brand called “Enesense”. They are Ni-MH with 1.2V and 8500 mAh capacity. The batteries are sold in packs of two batteries, and then it has been ordered two packs, so four batteries in total.

Before continuing, it is needed to know some useful concepts about batteries.

- Connecting batteries in series adds the voltage of the two batteries, but keeps the same amperage rating (also known as Amp Hours).
- Connecting batteries in parallel connections will increase your current rating, but the voltage will stay the same.

The GSM module works at a minimum voltage of 3.4V, meaning that, putting 3 batteries in series, the voltage is increased until 3.6V, which is enough for the GSM.

However, the capacity remains at 8500 mAh, which gives the unit a short period of lifetime. Therefore, a fourth battery is used in parallel to the other three, to be able to increase the capacity up to 17000 mAh, see Table 2.12. Then, the disposition should be like Figure 2.25: Final battery disposition Figure 2.25.

Table 2.12: Final battery properties

	V	mAh
<b>One battery capacity</b>	1,2	8500
<b>3x battery in series</b>	3,6	8500
<b>3x series &amp; 1x parallel</b>	3,6	17000
<b>Final Battery Capacity</b>	3,6	17000

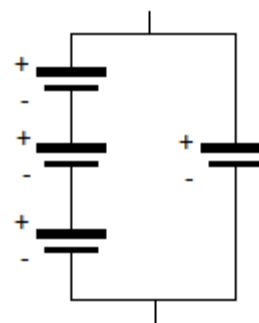


Figure 2.25: Final battery disposition

The next step was to calculate the use of energy that the unit will have during a year. For being able to figure out the total consumption, the power consumption is split in three scenarios, each one with its own consumption and time active.

Therefore, at the end of the table there is the theoretic lifetime of the system, at optimal conditions. The theoretic lifetime is obtained from dividing the total capacity of the batteries with the consumption per year calculated, see Table 2.13.

Table 2.13: Calculus of theoretic lifetime

		Scenario 1	Scenario 2	Scenario 3
<b>Mode</b>		<b>Sleep</b>	<b>Active</b>	<b>Transmitting</b>
<b>Current consumption</b>	(mA)	0,0004	30	60
<b>Time required</b>	(s)		1	5
<b>Times per day</b>	(u)		24	3
<b>Total time needed per day</b>	(s)	86361	24	15
<b>Consumed charge</b>	(mAh)	0,00960	0,2	0,25
<b>Consumed charge year</b>	(mAyear)	84,1	1752	2190
<b>Total</b>	(mAyear)	4026,06		
<b>Theoretic lifetime</b>	(years)	4,22		

However, this previous estimated lifetime is not accurate, because the self-discharge was not calculated.

The battery company specifies that their batteries can hold up to 70% of the total capacity after 5 years. Then, for doing an approximation of the self-discharge, a linear discharge was assumed, due to lack of information from the company about the batteries, see Table 2.14.

With that information, it is possible to plot a graph and compare it with the theoretic lifetime, see Figure 2.26, and then, make an accurate approximation of the real lifetime.

Table 2.14: Battery specification

Self discharge				
<b>Initial capacity</b>	(%)	100	<b>Initial time for capacity retained</b>	(years) 0
<b>Capacity retained</b>	(%)	70	<b>Time for capacity retained</b>	(years) 5

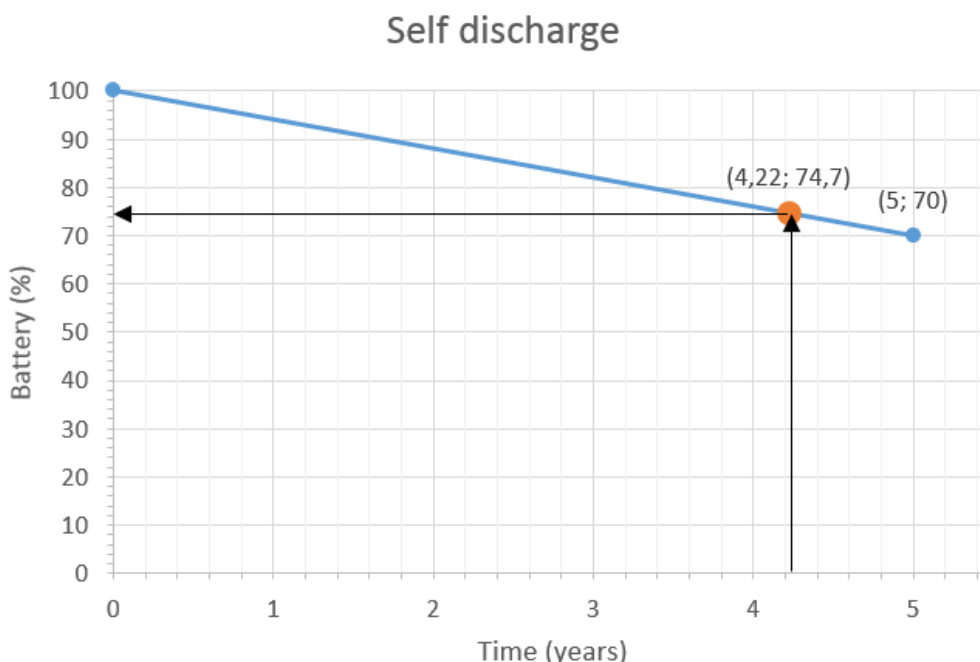


Figure 2.26: Plot of battery specification and theoretic lifetime point

Table 2.15: Real lifetime calculus

Current self discharge		
Time for actual capacity retained	(years)	4,22
Actual capacity retained	(%)	74,7
Real time active calculus	0,747 * 4,22	
Real lifetime (years)	3,2	

Therefore, after the self-discharge deduction, it is concluded that the real lifetime of the batteries for this unit should be 3.2 years in optimal conditions; it is calculated in the Table 2.15.

It is needed to say that the whole unit is going to be under many different environmental conditions, thus the battery lifetime is just an approximation.

## 2.7 Ideas implemented

In the assignment there was stated different ideas for the project. The group was free to choose from the ideas, although some were more important than others. The ideas and if they were implemented, or not, are presented.

About the system, see Table 2.16.

Table 2.16: Checklist about the system

Idea implemented	Work
YES	Appealing and robust design
YES	Easy to integrate
SOME	Applies smartphones and PC to facilitate the work of maintenance personnel and planners
YES	High life endurance and good performance
SOME	Full control of the manhole covers
YES	Design the components box that ice growing could not affect the box. The box can be placed

About the control and monitoring, see Table 2.17.

Table 2.17: Checklist about control and monitoring

Idea implemented	Work
YES	Monitoring water level
YES	Monitoring if the manhole cover is positioned correctly
NO	Monitor the need for collection of sludge
YES	Monitor whether the manhole cover is open
NO	Possible to find a cover in bad conditions.
NO	Camera can take situation picture of manhole
NO	Status of water flowing in the pipes <sup>1</sup>
NO	Status and control of valves and pumps. <sup>2</sup>
YES	Measurement of temperature. Important if the manhole has equipment that is critical temperature
NO	Counting of replications
YES	Measurement of H <sub>2</sub> S gas. <sup>3</sup>

## 2.8 Cost estimation

All the components are bought from reputable companies with the purpose:

- Components are available for several years
- Provide large amount of the different components
- Spare parts are available
- Cost-effective

Table 2.18 shows overview of total cost without shipping.

<sup>1</sup> Pipes are monitored in pump-stations.

<sup>2</sup> Valves controlled manually, not possible with electronics.

<sup>3</sup> Measurement of gases was not expected from Ulefoss, but researched and added.

Table 2.18: Cost overview

Unit Parts Cost		Dollar	8,56			
		Euro	9,39			
Component	Model	Cost	Units	Quantity	Price in Kr	Company Seller
Microcontroller	MSP430FR5969	15,99	\$	1	136,9	RS Component AS
Gsm	SIM800L	10,23	€	1	96,1	Electrodragon
Antenna	2PCs Quad-band	2,19	€	1	20,6	Electrodragon
Antenna adapter	2PCs IPEX(IPX) to SMA	1,54	€	1	14,5	Electrodragon
SD-Card	Integral SD MLC 2Gb	10	Kr	1	10	-
SD-Card socket	SD card Module	17	Kr	1	17	Electrodragon
Bluetooth	H-06	52	Kr	1	52	Electrodragon
Ultrasonic Sensor	MB7052	1984	Kr	1	1984	Maxbotix
Switch	SSCEB31C	65,45	€	1	614,5755	Honeywell
Temperature Sensor	DS18B20	3,95	\$	1	33,8	RS Component AS
Gas sensor	TGS2602	127,65	Kr	1	127,7	RS Component AS
Box	Koblingsboks	149	kr	1	149	Clas ohlson
Batteries	Enesense	456	Kr	2	456	Batterionline
Regulators	Step-up module	138,9	Kr	1	138,9	Kjell & Company
Total			Kr		3712,0	

In addition, an approximation of the unit assembly cost is done in Table 2.19, which shows how much it would cost to assemble all the parts shown in the table above.

Table 2.19: Unit assembling cost

Unit Assembling Cost		
Time needed for weld	h	0,5
Component	Unit	Price
New PCB	Kr	5
Piece of Lead	Kr	5
Worker	Kr/h	150
Total	Kr	85,0

A new PCB means a board where all the components are mounted and welded. It is assumed to buy PCBs in big packs. That is why the cost is so cheap.

As well, it is assumed the price per hour of a common worker is 150 kr/h. Obviously, it depends on the company that mounts the unit.

It is also important to consider, once the unit is ready, how long it would take to install it in the manhole-wall. It is assumed that to screw the wall, put the mounting structure and finally install the unit requires at least 1.5 hours of work.

Table 2.20: Unit installing cost

Unit Installing Cost		
	Unit	Price
Time needed installing	h	1,5
Worker	Kr/h	150
Total	Kr	225,0

Finally, when all the costs are taken into account and the component expenses, the final estimation will be **4022 kr**.

The price in hours for developing the system are not taken in account.

### 3 OVERVIEW

The following chapter will contain a short overview of the system, unit and software. As seen in the research part, the components that were chosen for the unit are: a microcontroller, GSM-module, water level-, gas-, temperature-, cover sensor, and a battery as a power source.

The software consists of a database and a GUI, with installation program.

The unit and the software combined makes up the system.

#### 3.1 System overview

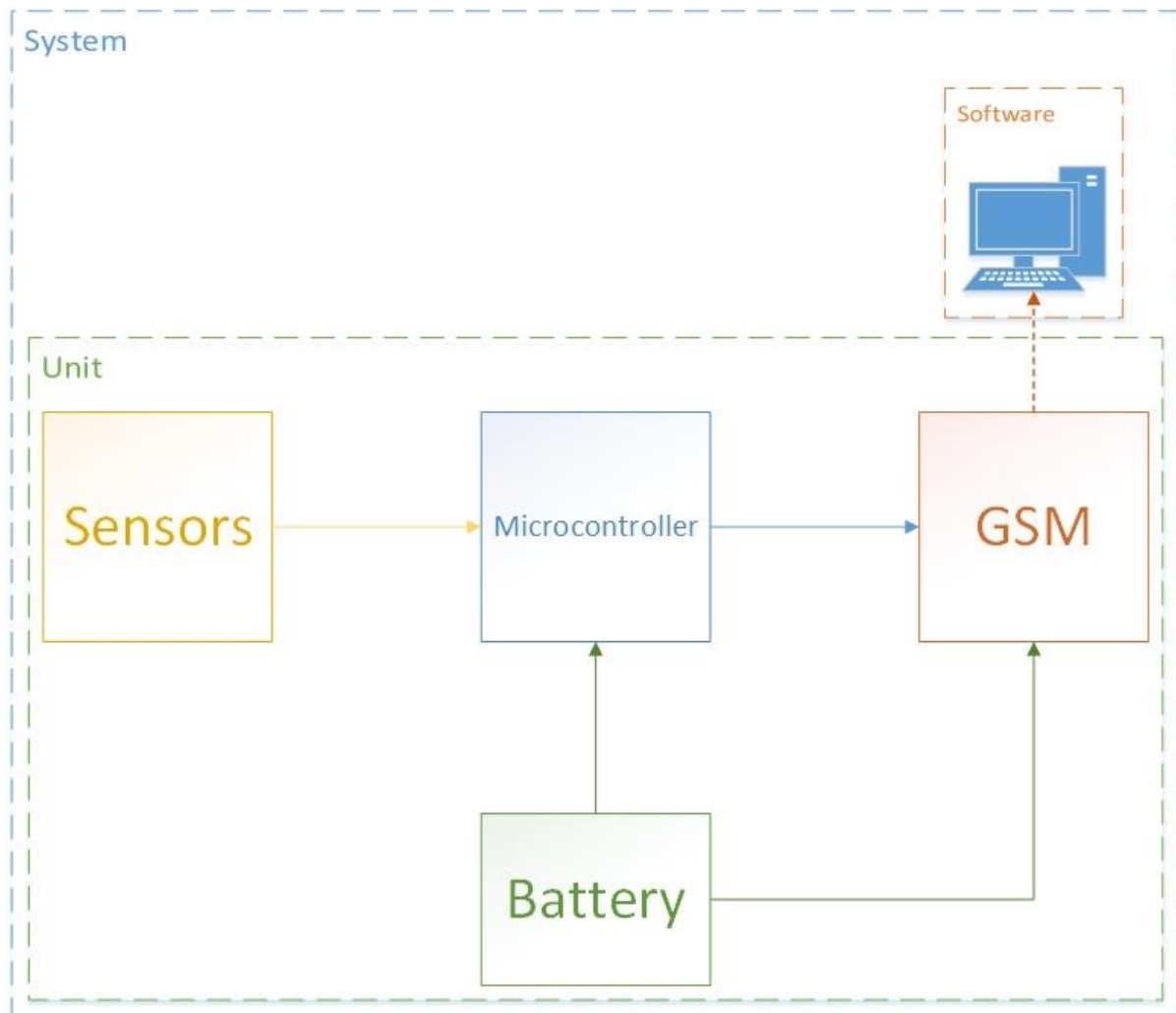


Figure 3.1: System overview

As seen in the Figure 3.1, the sensors provide information to the microcontroller; the microcontroller processes this information and sends it to the GSM. The GSM uses wireless technology to transmit the information to the database. The battery feeds the microcontroller (and sensors from there) and the GSM-module. More details about the components and how they work will be presented in the specified chapters.

## 3.2 Unit overview

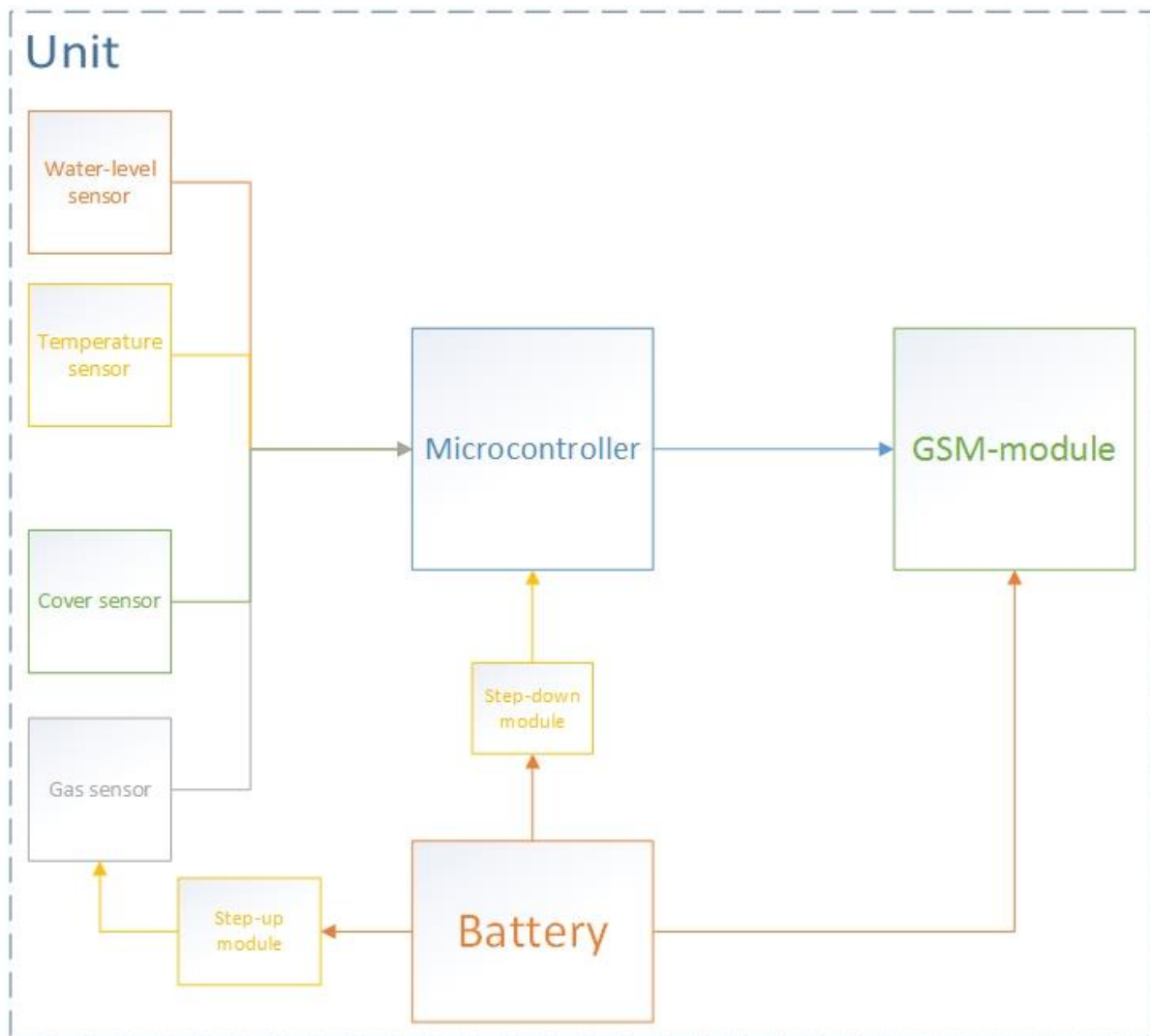


Figure 3.2: Unit overview

As seen in Figure 3.2, this is a more specific overview of the unit, but with the same characteristics as the system for the different components. The voltage needs to be stepped down for the microcontroller, with the battery voltage being too high, and opposite for the gas sensor, with the voltage being stepped up to reach the wanted level. This will be explained further in the specific chapters.

### 3.3 Software overview

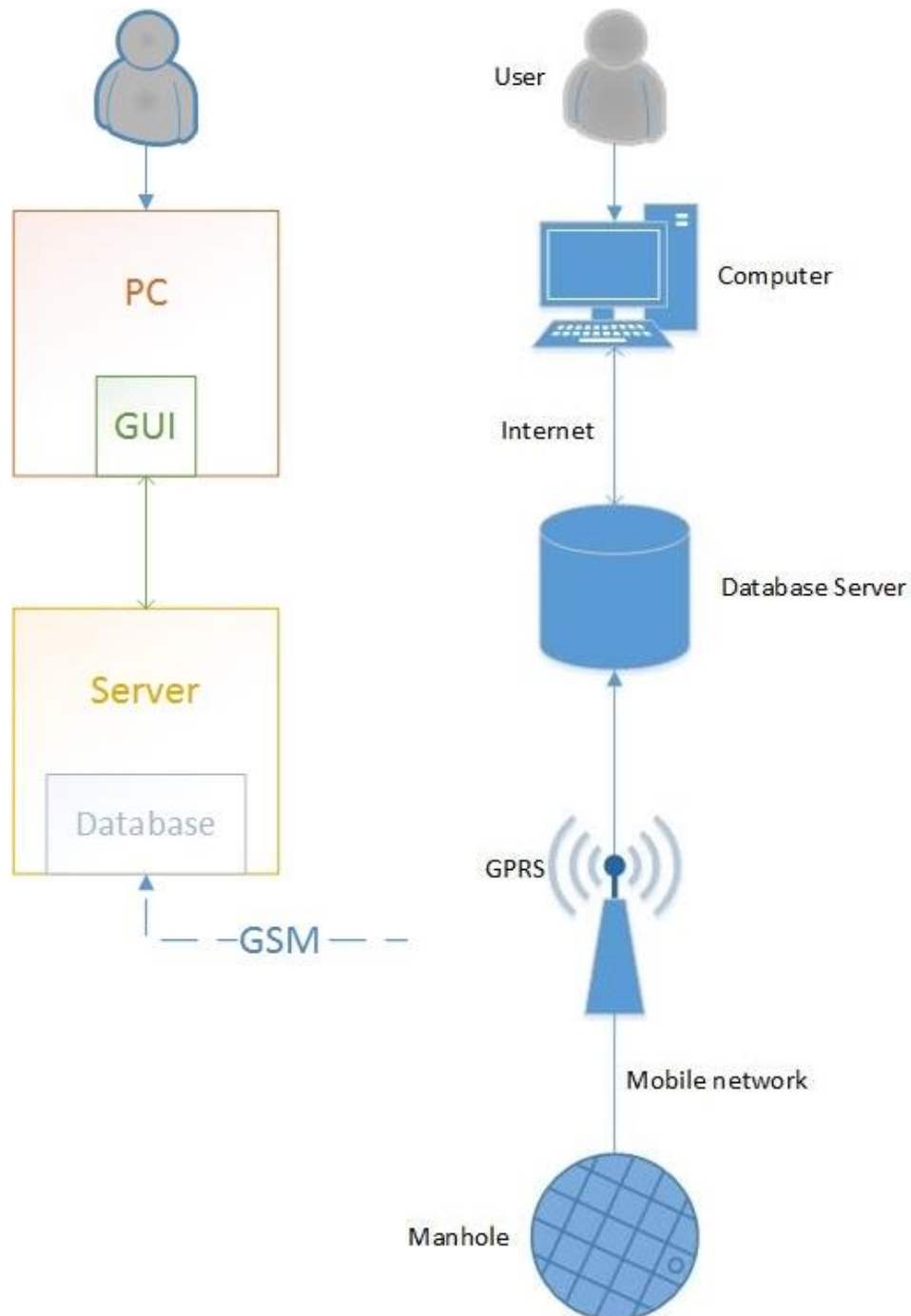
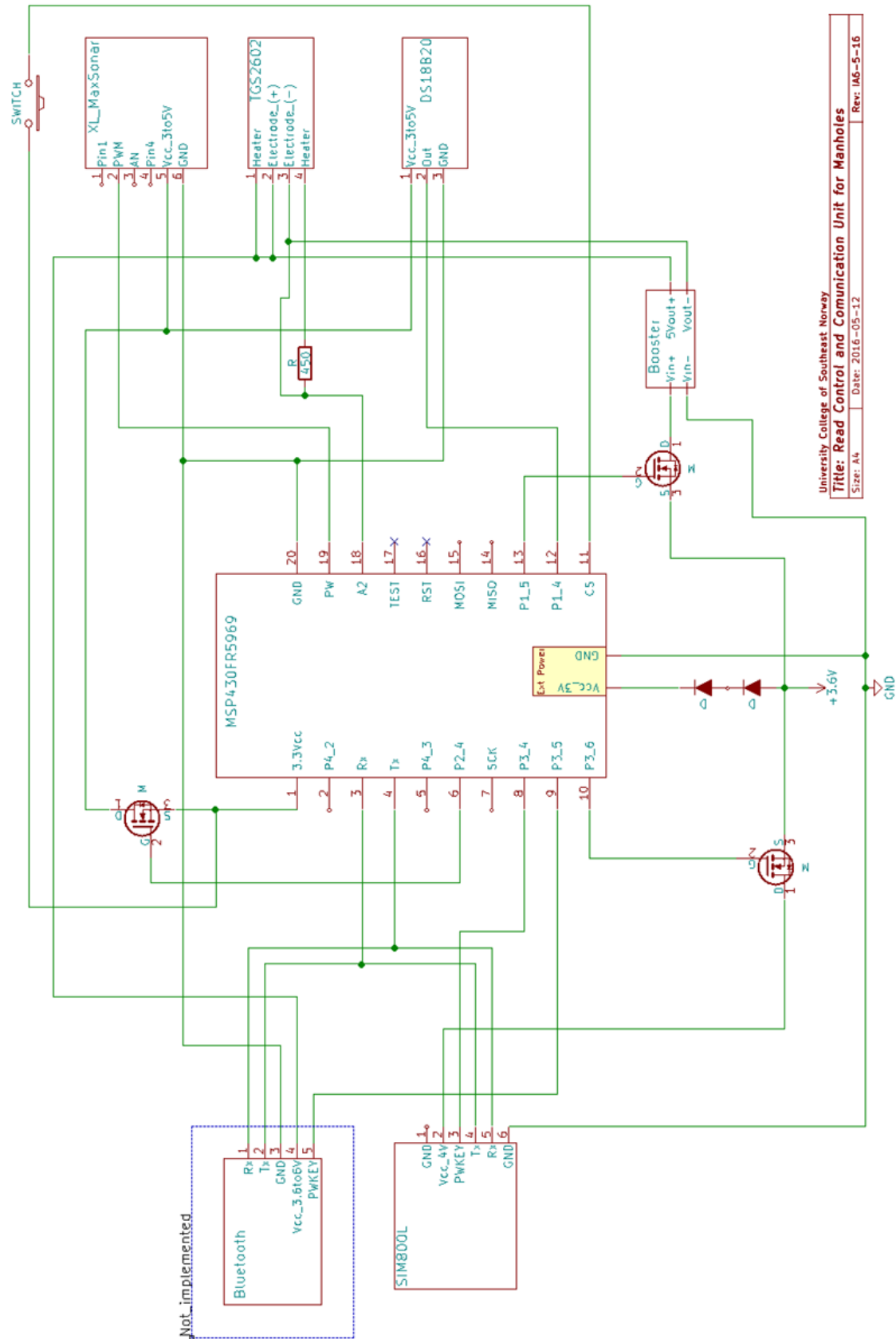


Figure 3.3: General and specific overview of software

The picture Figure 3.3 gives a more specific overview of the software. The server receives information from the GSM, and either saves to the database, or shows the information in the GUI.



# 4 SCHEMATIC OF THE UNIT



University College of Southeast Norway  
 Title: Read Control and Communication Unit for Manholes  
 Size: A4 Date: 2016-05-12 Rev: IA6-5-16

In the schematic of the unit shown above, it can be seen the disposition of all the components that make the entire unit, and, how they are connected to the microcontroller.

It is extremely important, before to constitute the whole unit, to test each component individually to see its behavior and not to commit mistakes, which they can cause a huge failure in the microcontroller or other component.

## 4.1 Unit battery

The battery that is feeding the whole unit gives a voltage of 3.6 V. It is thought in that way to feed directly the GSM module, which the minimum voltage it can afford is 3.4 V. Besides, it is inside the range it can supply the 2A peak that it needs.

However, the microcontroller needs to be fed using a voltage of 3V. As the battery supplies 3.6V, it is used a simple solution to adapt it to the microcontroller. Using two diodes made of Germanium which they have each one a dropping voltage of 0.3V, it is possible to create a total dropping voltage of 0.6V. Then, it becomes inside the range of microcontroller feeding.

Note that a diode does not affect the performance of the microcontroller because its physical composition of non-resistive component.

For supplying energy to the gas sensor, or the Bluetooth, which is not used, it is needed to step up the voltage to 5V. It possible using a voltage booster step up regulator plugged from the battery as input and feeding the gas sensor with 5V as output.

## 4.2 MOSFET

There was a problem when the unit is going to sleep; all the components are still connected to the battery, meaning a constant current consumption. However, there is a smart simple solution, by using a MOSFET transistor, the microcontroller is able to deactivate all the components, or in other words, all the power supply wires. Therefore, the battery is just feeding the microcontroller, and it is this one, which decides what to do with the power supply of the components. Then, when it awakes, it activates the MOSFET to drain current to the components, otherwise, when go to sleep, it deactivates the MOSFET.

Note that the cover sensor cannot be under a MOSFET effect because, as it is a switch (ON – OFF), if it is disconnected from the power supply, it will never be able to create an interruption.

It is important to note that the microcontroller always has to be connected to the battery without interruptions of power supply; otherwise, the chip (MSP430FR5969) of the Launchpad board could be damaged.

## 5 SOFTWARE

This chapter describes how users can interact with data from the Read, Control and Communication unit. It ensures that data from the manholes are stored in tables for future references. The data is used to save vital information about software administrator, manholes and the company. Further information about the software requirements and design can be found in Attachment G: SRS & SDD

### 5.1 Software Module

The software has two main modules, Microsoft Visual Studio application and the Database Server. Visual Studio application host the application's Graphical Users Interface (GUI) and the programming codes in C# language, while the Database Server hosts tables that stores data from the manholes and administrators.

As illustrated in Figure 5.1, data from the manhole is sent from the system unit via GSM/GPRS to the Database. These data is displayed on the GUI. Other data can also be sent from the GUI to the database by the administrator. Administrator send data to the database when new manholes are registered or when an editing task is performed. In other words, both GUI and the Database work hand-in-hand to execute the software task with the help of the programming codes.

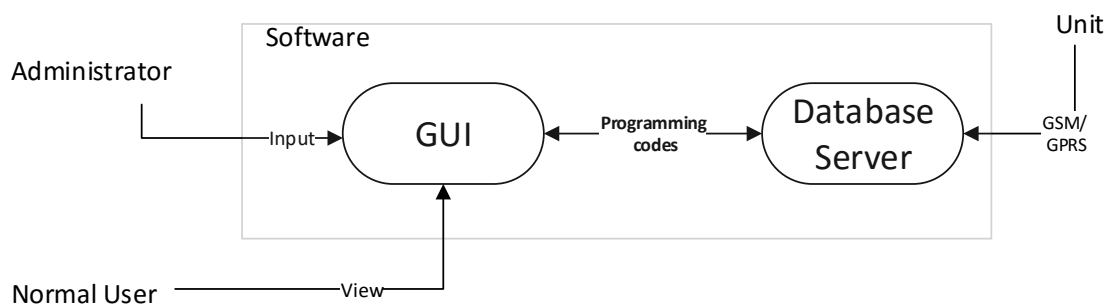


Figure 5.1: Software Module

### 5.2 Software Application

This is the process involved in developing the codes and GUI needed to satisfy end users requirements. It deals with the design and implementation the application.

#### 5.2.1 Software application flow chart

It is the beginning of the design of the program and represents the logical sequence of events in the software application of the system. Figure 5.2 illustrates how the users navigates the GUI template to interact with the systems unit through the Database Server. Further details and full code can be found in Attachment H: Software code

Flow chart for software Read, Control and Communication

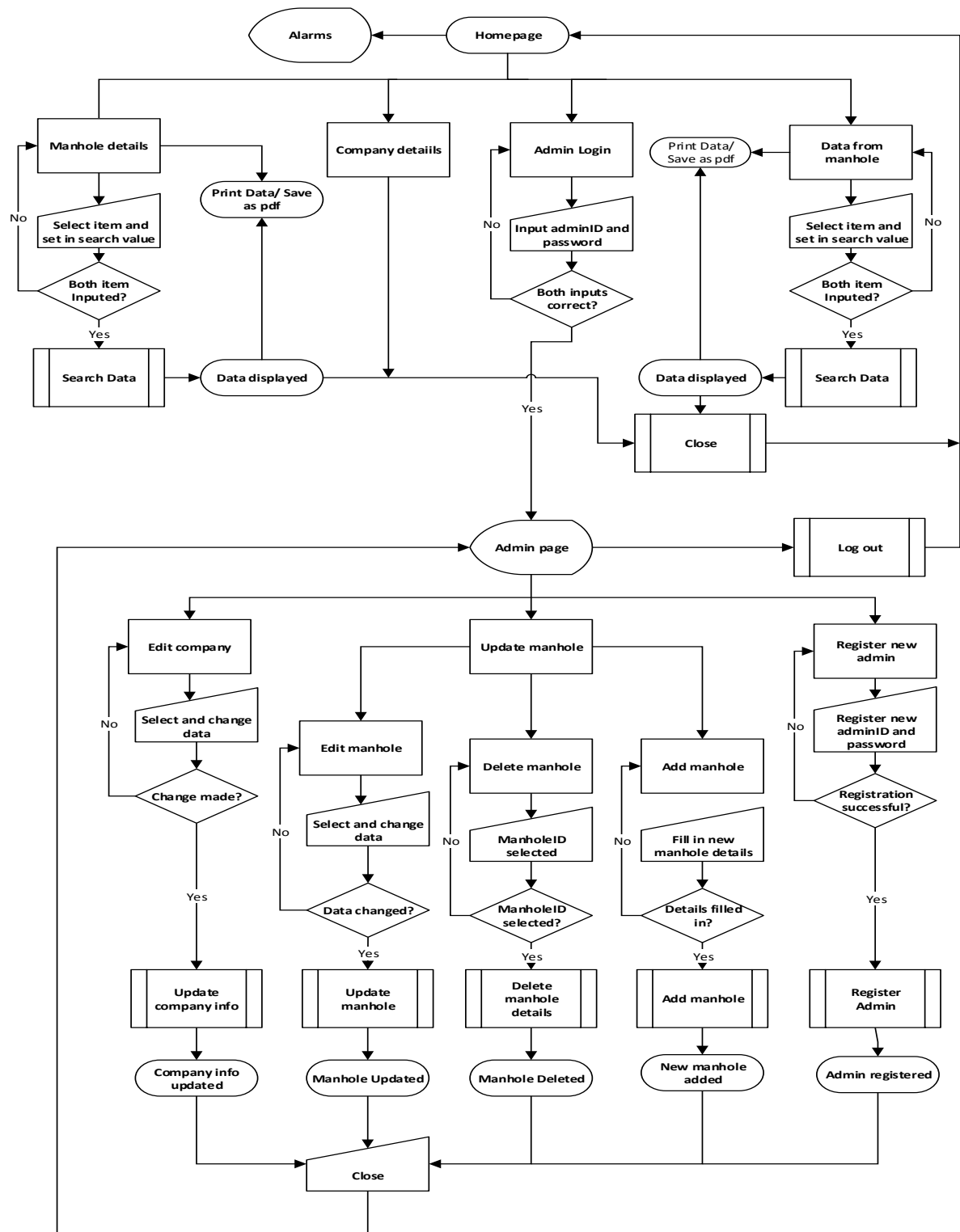


Figure 5.2: Software flow chart

### 5.2.2 Class diagram

It was used both for the general conceptual modelling of the systematics of the software application. It is the building block of the program code and it is part of the design of the Software application. It represents both the main elements, interactions in the application, and the classes that were programmed. Figure 5.3 represent the classes that were programmed to develop the software application.

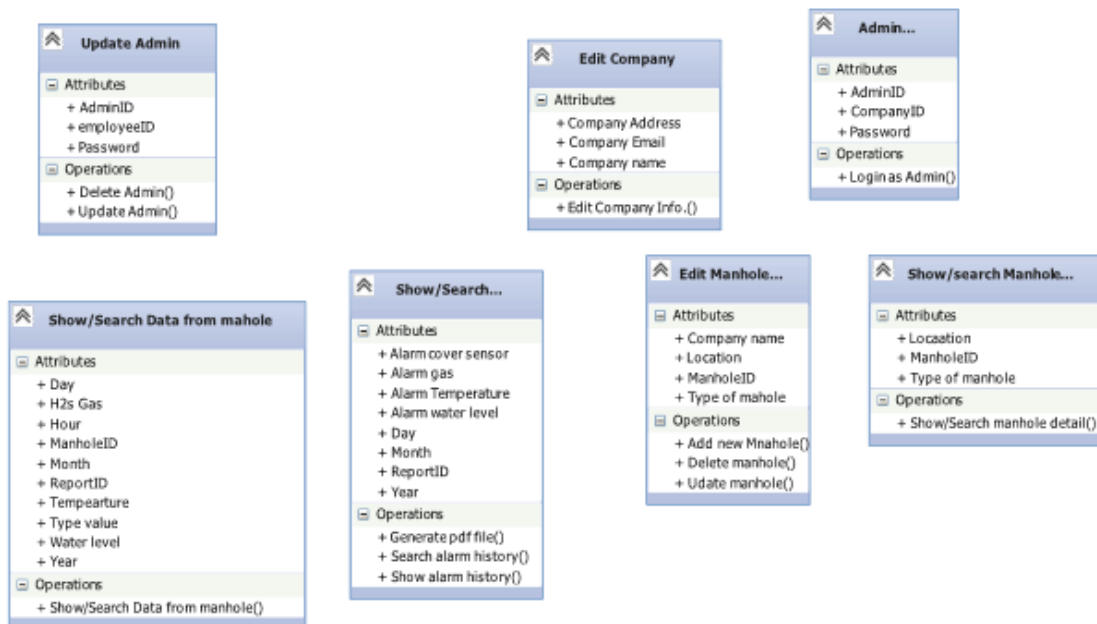


Figure 5.3: Class diagram

### 5.2.3 Use Case Diagram

This shows the relationship between the user and the different use cases the user is involved in. In other word, it illustrates the user interaction with the system. As shown in Figure 5.4, the use case diagram indicates how the various actors interacts with the system. The actors in this case are the regular user, administrator, database and the printer/pdf file.

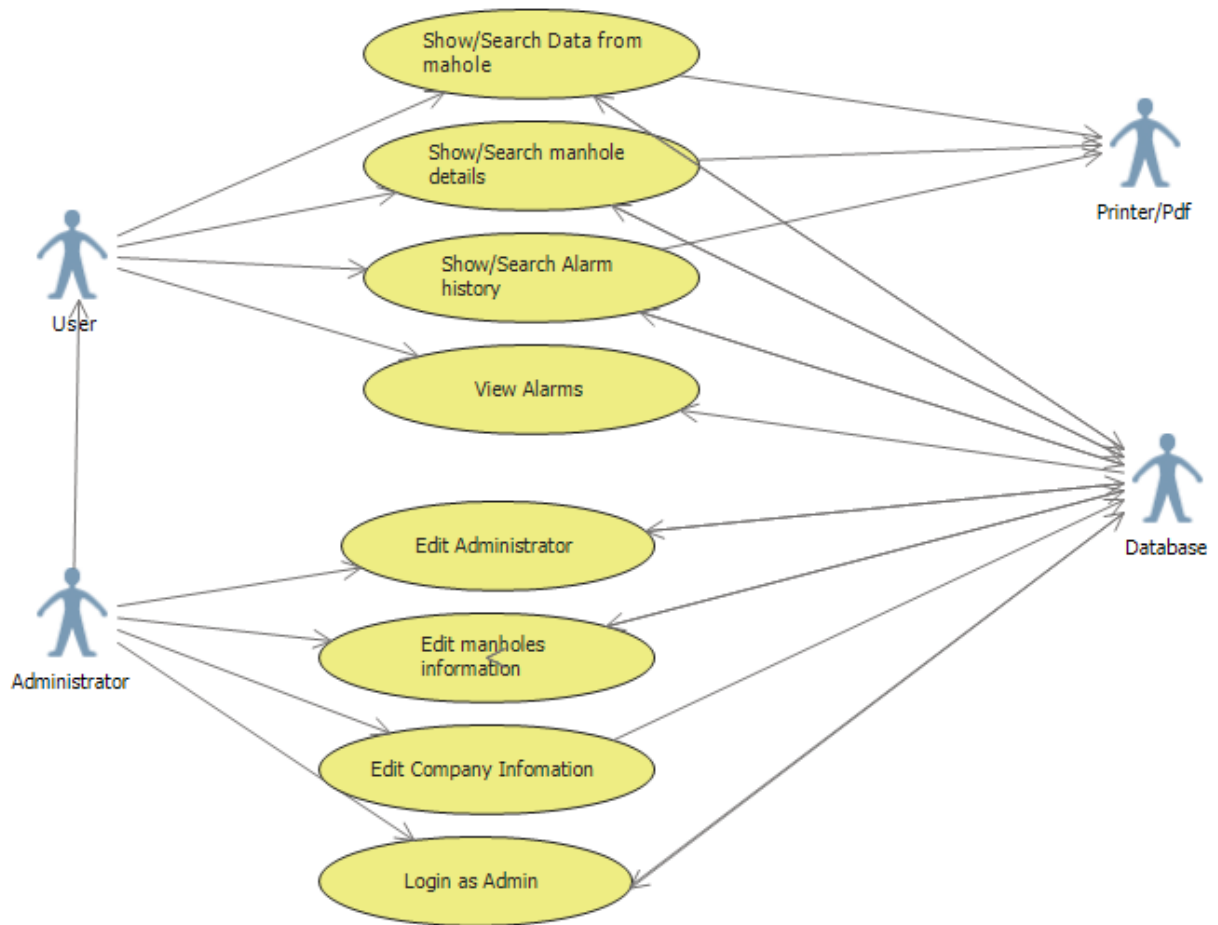


Figure 5.4: Use case diagram

### 5.3 Database server

This section involved in designing, modeling and developing the Database.

#### 5.3.1 ER Diagram

The table script was used for the general conceptual modelling of the tables and columns in the Database. It is the development source of the Database and represents the tables and their relationships. Figure 5.5 shows both the various tables and columns in the Database and its relationship.

The “table script” was developed and generated in “CA Erwin data modeler”. “CA Erwin data modeler” is a program use for the modelling and designing of table and its relationship for database development.

It is important to note that the Database for the software was created in the SQL Server, but the tables and columns were generated through the “Tables script”.

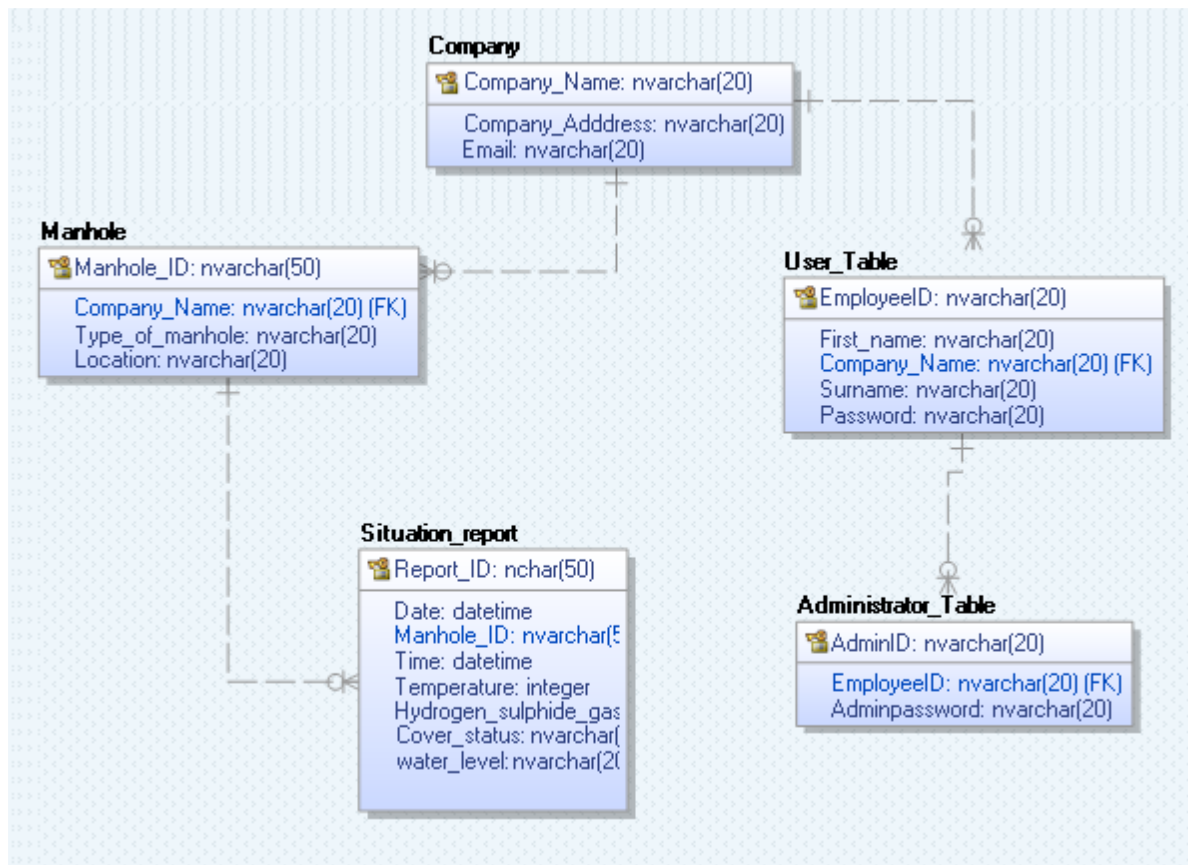


Figure 5.5: Erwin diagram

### 5.3.2 Table script

This is the SQL Query script generated from the ER Diagram to develop tables and columns in the Database. It is the source of building exact tables and columns used in developing the Software application. For details of the Table script, refer to the Table Script Document.

## 6 UNIT PROGRAMMING

This chapter will explain the programming of the unit and its components.

### 6.1 Microcontroller

To be started in the microcontroller MSP430FR5969 Launchpad, or called Wolverine, it is going to be used Energia 17 Software from TI® as a program to create the final code for the unit. Final code with comments can be found in Attachment I: Unit code and, information about the libraries and variables used in Attachment J: Tables.

#### 6.1.1 What is Energia?

Energia is an open-source electronics prototyping platform started by Robert Wessels in January of 2012 with the goal to bring the Wiring and Arduino framework to the Texas Instruments MSP430 based LaunchPad. Energia includes an integrated development environment (IDE) based on Processing.

For more information visit the Energia webpage, where is possible to download the software. [39]

### 6.2 Specification: Unit block diagram.

The Figure 6.1 shows what the main idea of the software is.

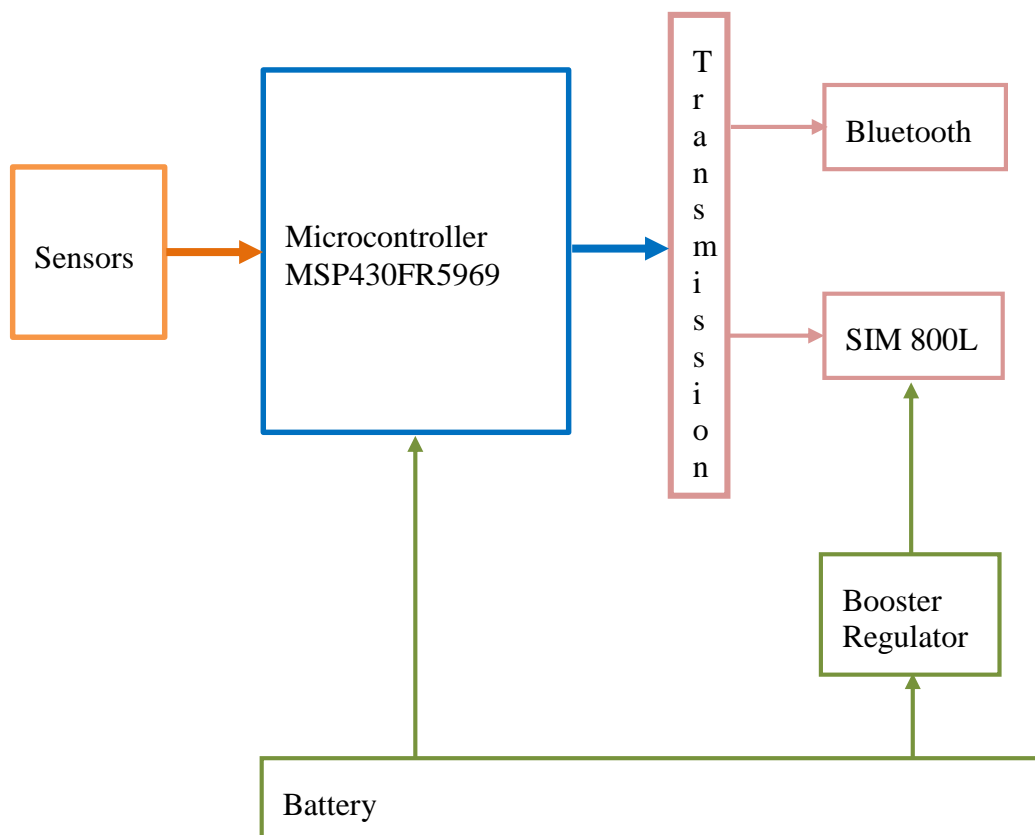


Figure 6.1: Unit block diagram



As shown in the block diagram above, the sensors send measurements to the microcontroller when required. By the same way, the microcontroller stores these measurements into an array.

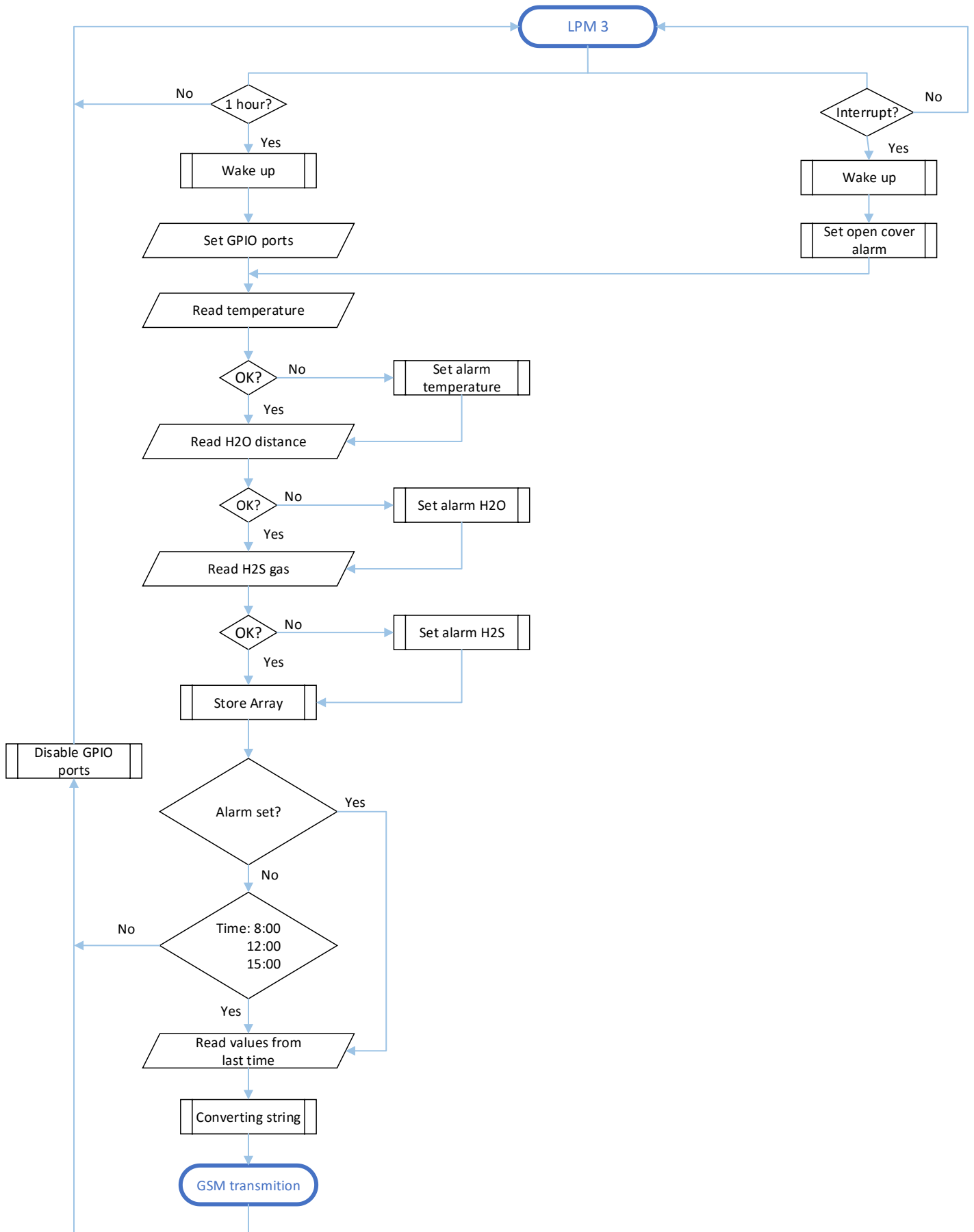
If the microcontroller commands to send the information, this array is converted into a string of characters from the last line sent and measured, until the new line of measurements. This string is created because it is sent just a word data type with all the numbers inside. It is an easy way to transmit all the data in one shot.

Then, the string is transmitted to the GSM module SIM800L, and if it is needed, to the Bluetooth.

Obviously, the battery is feeding the microcontroller; however, the SIM800L needs to be fed directly from the battery also. This is because the GSM module generates peaks of current of 2A during microseconds when doing the transmission of data, so if the microcontroller feeds the GSM, the microcontroller can be burnt.

### **6.3 Design: Code flow diagram.**

This is a specific flow diagram to show how the unit will run the code and which commands are done every moment that it is active. It is possible to see the main idea and the full functionality of the microcontroller.



## 6.4 Implementation

Now, there is going to be described some of the most important parts of the code for the better understanding. In addition, there are some parts, for instance, the sensors, that they are better described into the test part, this is because they are studied and explained individually.

### 6.4.1 Controlling the time using RTC\_B.

```
//-----library for time-----
#include <RTC_B.h> //library to set the RTC_B clock mode

byte firstAlarm = 8; //8:00
byte secondAlarm = 12; //12:00
byte thirdAlarm = 15; //15:00
int timetochek = 3600; //check every 1 hour

byte hourIncrement;
volatile uint16_t _lhr_ticker = 0;
volatile boolean dostuff_every_lhr = false;
```

Figure 6.2: RTC\_B library and alarms

In this first part of the code corresponding to the Figure 6.2, see the initialization of the RTC\_B library. This library enables the configuration of the internal RTC of the microcontroller with simple commands.

This RTC supports Calendar mode, with a single Alarm configurable to latch on Day of Week, Day of Month, Hour or Minute. Any combination of these criteria may be switched on to provide a flexible alarm system.

It can be seen also the initialization of the alarms, which are going to be explained in the next step.

```
//time inicalization
rtc.begin(); // Start RTC calendar mode
rtc.begin( MONDAY, 4, 10, 2016, 8, 29, 0);
rtc.attachPeriodicInterrupt(1, flagOneHourTick); //alarm every one hour
```

Figure 6.3: Time initialization

Figure 6.3 shows the way of initialize the RTC calendar mode inside the Setup(). Afterwards, it is needed to declare the current time; the microcontroller has to know from the beginning which time it is to start counting.

Then, when it is able to count, attach a periodic interrupt to a function, in this case, called “flagOneHour”. This attachment will generate an interrupt every second in the RTC calendar. This library supports *1/second* down to *1/16384-second* ticks using the attachPeriodicInterrupt() feature. That is why it is placed a “1” in the attachPeriodicInterrupt(), meaning one second.

```

//-----Alarms-----
void flagOneHourTick() { //interrupt created every second

    _lhr_ticker++; //variable to count the seconds
    if (_lhr_ticker >= (timetocheck)) { //if is reached the time wanted
        _lhr_ticker = 0;
        dostuff_every_lhr = true; //run the main program
        hourIncrement++; //variable to know the current hou:
        wakeup();
    }
}

```

Figure 6.4: Alarm every hour

In the Figure 6.4 is seen what is done inside the interruption. It counts the seconds (“\_lhr\_ticker”), and when the counter reaches 3600 seconds, equivalent to one hour, it wakes up the CPU of the microcontroller.

“hourIncrement” is a variable that store each time the microcontroller wakes up every 3600 seconds, or in other words, each hour. It was created for knowing when the microcontroller has to send the string that contains all the measurements. As it is shown in the first part of code, now it sends the string when “hourIncrement” match with first, second or third alarm variables, three times a day.

## 6.4.2 Controlling the low power mode & GPIOs

```

GPIOoff(); //Restore Port settings
sleepSeconds{10000000}; //go to LPM3;clock will be updated; consumption registered: 1uA

```

Figure 6.5: Low Power mode 3

The Figure 6.5 , which is placed at the end of the “void loop”, shows the command “sleepSeconds” is written. Using this, the microcontroller must go into the sleep mode, concretely, into the low power mode 3 or LPM3. Its written 10 millions of seconds to be sure it never wakes up, until an interruption wakes it up.

However, just writing the command for going to sleep does not mean to get the lowest power consumption specified into the datasheet. Actually, the current consumption was around 600 uA remaining into the LPM3.

The problem was that all the I/O pins were active, which that means current flowing through them. To solve it, before each time the microcontroller must go to sleep, it must to deactivate all the I/O pins; otherwise, they are all consuming power.

To reach it, is as easy as write into the registers to set all of the ports available at zero. Moreover, fill up the registers of the directories at the maximum value. Then, the microcontroller can go to the sleep mode, concretely the LPM3, and going to sleep consuming less than 1 uA.

```

void GPIOoff(){
  WDTCTL=WDTPW+WDTHOLD;           // Stop watchdog timer

  // Port Configuration
  // Disable the GPIO power-on default high-impedance mode to save power
  P1OUT = 0;
  P1DIR = 0xFF;

  P2OUT = 0;
  P2DIR = 0xFF;

  P3OUT = 0;
  P3DIR = 0xFF;

  P4OUT = 0;
  P4DIR = 0xFF;

  PJOUT = 0;
  PJDIR = 0xFFFF;
}

```

Figure 6.6: Disabling ports Code and WDT

The watchdog timer (WDT) default behavior is to reset the microcontroller unless the firmware periodically lets it know everything is running fine. If the firmware is stuck in a loop or otherwise stops operating as expected, the watchdog is not fed and will reset the chip. Therefore the WDT always needs to be active, in other words, consuming power. In addition, if it is active, the WDT does not allow going to LPM3 because it needs the CPU and clocks to be active for counting time.

Then, as the WDT is useless for the unit/application, it is deactivated.

### 6.4.3 Reactivating the pins

When the code has to restart because there is an interrupt, the pins are still deactivated, they are not set as a default by the microcontroller itself. Therefore, when they are not set, the unit cannot take the measurements. The answer from the sensors cannot be detected because the respective pins are disabled.

For that reason, in the Figure 6.7, it is seen how to declare the pins required as inputs for the sensors, and as outputs for the MOSFET, setting them to high.

```

//set sensors and mosfet pins
//mosfets
pinMode(P2_4, HIGH); //feeds the sensors
pinMode(P1_5, HIGH); //feeds the regulator and gas sensor
pinMode(P3_6, HIGH); //feeds de GSM module

//sensor pins
pinMode(gasSensorPin, INPUT); //Set P3.0 SEL as Input- gas
pinMode(tempSensorPin, INPUT); //Set P3.0 SEL as Input- temp
pinMode(ultrasonicPin, INPUT); //Set P1.2 SEL as Input- ultrasonic

```

Figure 6.7: Set sensor pins and MosFet pins

## 6.4.4 Creating the array

### 6.4.4.1 Initialize the array

To initialize the array is necessary the following commands shown in the Figure 6.8.

```
//creating the array in the FRAM position
#define PERSIST __attribute__((section(".text")))
uint8_t myArray[2700][12] PERSIST;
```

Figure 6.8. Define the Array into FRAM memory

The MSP430FR5969 just has 2 Kb of RAM, which is nothing to operate and store all the array. However, as this board has 64 Kb of FRAM, which is more than enough for the application, this large array was placed into the FRAM registers. Therefore, it gets more space into the RAM place. Then, the string is going to have more capacity.

For this reason, it is used the definition of “PERSIST”, which allows the array to go into FRAM position. Global variables are declared as being part of the ".text" linker section, so they live in FRAM and are not copied to/from RAM on boot up.

In addition, when defining the array, it has to end with the word “PERSIST” to let the microcontroller know where it has to be placed. Now the array will live in FRAM and will not hog up any of the RAM. It will co-exist with program code as shown in the Table 6.1.

Table 6.1: Microcontroller memory disposition

64 Kb FRAM	2 Kb RAM
<ul style="list-style-type: none"> <li>- Sketch</li> <li>- myArray [2700][11]</li> </ul>	<ul style="list-style-type: none"> <li>- myString</li> </ul>

The table above is the disposition of the microcontroller’s memory.

### 6.4.4.2 Storing data

The idea is to store all the measurements that are read from the sensors into a bi-dimensional array of 2700 lines per 12 columns as a maximum, no more due the FRAM reaches the maximum capacity.

Storing 2700 lines and dividing them per 24 checks per day, corresponding to the 24 hours of a full day, then it is get the time until the array gets full of data. Therefore, it means it is time to wipe out the array and start a new one.

$$Days = \frac{2700 \text{ lines}}{24 \text{ lines/day}} = 112 \text{ days}$$

This array has to have a sort of all the values, which is shown in the Table 6.2.

Table 6.2: Sort of values

0	1	2	3	4	5	6	7	8	9	10	11
Day	Moht	Year	Hour	Minute	Temperature value	Distance value	Gas value	Cover Alarm	Temp. Alarm	Dist. Alarm	Gas Alarm

All the information is going to be send in this sort of values. For reaching it, is as simple as when storing, doing it in the right sequence. The code in Figure 6.9 shows how it works.

```

// read the sensors
TEMP = read_temperature();
DIST = read_dist() ;
GAS = read_gas();
// storing the values in an array
j=0;
//get the time and store in thearray
myArray[i][j] = rtc.getDay();
j++;
myArray[i][j] = rtc.getMonth();
j++;
myArray[i][j] = rtc.getYear()-2000;
j++;
myArray[i][j] = rtc.getHour();
j++;
myArray[i][j] = rtc.getMinute();
j++;
// store values from sensors
myArray[i][j] = TEMP;
j++;
myArray[i][j] = DIST;
j++;
myArray[i][j] = GAS;

j++;
if (COVER >= 1){
    myArray[i][j] = 1;
}
else COVER=0;
j++;
//Conditions for setting the warning alarms
if (TEMP >= 85){
    myArray[i][j] = 1;
}
else myArray[i][j] = 0;
j++;
if (DIST <= 30){
    myArray[i][j] = 1;
}
else myArray[i][j] = 0;
j++;
if (GAS >= 30){
    myArray[i][j] = 1;
}
else myArray[i][j] = 0;
    
```

Figure 6.9: Sequence for storing values

First is necessary to get the data from the sensors. That is why at the beginning the functions are called for each sensor and stored in temporary variables (TEMP, DIST and GAS).

Secondly, it is time to store these values into the array with the time that they were checked, and see if everything is correct.

Before continue, it is necessary to know the “i” and “j” variables are created to be able to move positions along the entire array, they both are just index for define a pointer.

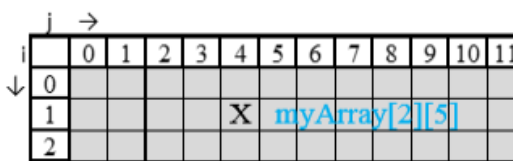


Figure 6.10: Pointer for moving along the array

For storing in the right sequence, it is used the variable “j” as an index for moving along the columns. It is initialized in zero, and then each time there is the command “j++” this variable adds one position in the columns. Therefore, the next command is for store data. In the code, first it is stored the time, later the sensors values, and finally, if something is wrong with them, is stored the alarm.

### 6.4.5 Convert and send the data

The microcontroller knows what time it is, then, by setting alarms, a part of checking every hour; it is possible to give an extra functionality to the code. In this case, what is wanted is to convert the array into a string and send it to the GSM module for transmitting the data to the server. It involves creating different conditions. Then, when an alarm matches, the code can run the part to convert and send the data.

In this case, as the unit has to do the same every time, the three alarms (firstAlarm, secondAlarm and thirdAlarm) are written when match with “hourIncrement”, and then they go to the same part of code. Figure 6.11 shows it.

```

if (hourIncrement == firstAlarm || hourIncrement == secondAlarm || hourIncrement == thirdAlarm || COVER == 1 ||
    myArray[i][9] == 1 || myArray[i][10] == 1 || myArray[i][11] == 1){ //set conditions to convert into a string and send

    valueChain(myArray, counter, previous); //convert the array into a string
    sim8001.begin(9600);
    delay(500);
    GSM_module(mystring); // send all the information as a string
    previous++; //set a previous point to send the information next time
    mystring=""; //clean string
    COVER = 0;
}
if (hourIncrement == 24) hourIncrement=0; //if the day is over, start counting from 0 again

```

Figure 6.11: Converter and sender module

Moreover, what also is wanted is that if something happens that is not part of the plan, specially a reading from a sensor, if its measurement is above set alarm value, it is needed to notice which sensor set the alarm and retransmit the data. This is the reason of adding more conditions, if there is a non-planned reading from the sensors; the data needs to be sent to the server.

For instance, myArray[i][10], is referred to the current line and the column 10 into the array, in other words, the alarm of the water level sensor. Therefore, when it is set, the condition matches and proceeds.

### 6.4.6 Converting the array into a string

```

//-----String converter-----
void valueChain (byte arg[][11], byte after, byte prev){
  for ( byte x = prev; x < after ; x++) {
    for (byte z = 0; z < 10; z++) {
      mystring += myArray[x][z]; //mystring is receiving all the values together
      mystring+=" "; //add an space
      delay(10);
    }
    mystring+="\n"; //add a new line
  }
}

```

Figure 6.12: String converter module

This module in the Figure 6.12 is the one that convert the array done before into a string ready to send. It receives the array as an argument, and two bytes called “after” and “prev”.



These two bytes are given because “after” is the number of last new line added to the array, and “prev” is corresponded to the number of the last line sent. Giving this information, the microcontroller knows from where to where it has to send the information. It is useful not to send old information from the array, and not to send random information not written yet. The idea is shown below.

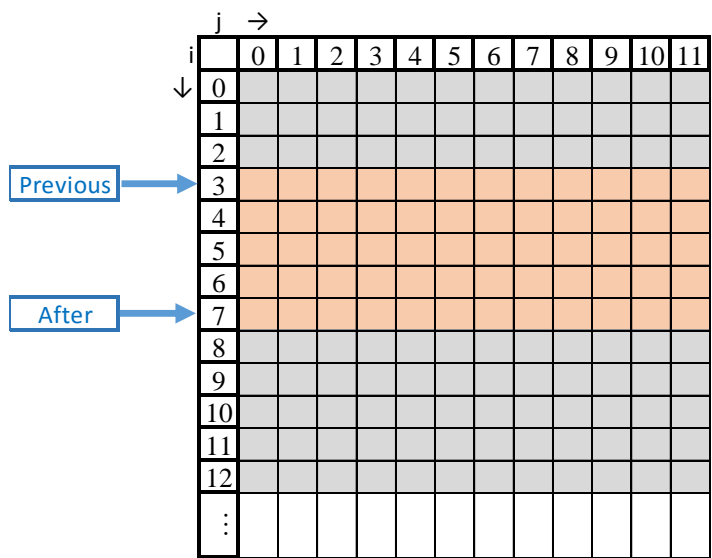


Figure 6.13: Information sent

As Figure 6.13 shows in, the information that will be send is the orange part, regardless of the old data and the random data.

Now it is time to convert these data into a string. It is necessary to take the string, copy it, adding a new value, and add a comma; and at the end of the line, add a new position line. It is produced a word, which is easier to manage than an array. Therefore, this sequence of characters is ready to be sent to the GSM module in one shot.

## 7 TESTING

Execution of tests was by following a test plan. This test plan is found in Attachment K: Test plan document

### 7.1 Unit test

This following chapter will explain the different tests, for the unit.

#### 7.1.1 Input

The program need some inputs for start running by itself.

A brief summary of the inputs needed:

- Current time
- Checking time
- Alarms

##### 7.1.1.1 Current time

The microcontroller has to know the time each moment, for checking when it has to wake up. How does the microcontroller know the time? Which part of microcontroller is active? Therefore, to set the time at the beginning to start counting, it can be done by following the next instruction in Figure 7.1.

```
rtc.begin(); // Start RTC calendar mode
rtc.begin( MONDAY, 4, 10, 2016, 8, 29, 0);
```

Figure 7.1. Setting the current time at Monday  
10 of April of 2016 at 8:29 in the morning

Now the unit is able to count from this start point of time.

##### 7.1.1.2 Checking time

When the system has to create the interrupt to wake the microcontroller, it has to be defined a time in seconds. In this case, it must be every hour, that is to say every 3600 seconds, but it can be predefined as the user wants.

```
int timetochek = 3600; //check every 1 hour
```

Figure 7.2: Setting the time to check every 1  
hour

##### 7.1.1.3 Alarms

For being able to send the data every a period, it is needed to define three alarms. Depending on the user, the number of alarms can be increased or decreased. Moreover, the time of the own alarm.

```
byte firstAlarm = 8;    //8:00
byte secondAlarm = 12; //12:00
byte thirdAlarm = 15;  //15:00
```

Figure 7.3: Setting the alarms to send the information

### 7.1.2 Expected output

A brief summary of the expected outputs:

- Array
- String
- Message sent when alarm

#### 7.1.2.1 Array output

After introducing all the inputs required, every hour, the microcontroller must wake up from sleep mode, check out all the sensors, store their data into the array, and then, go back to sleep. Therefore, the output of this segment of the code would be the one that Figure 7.4 shows.

	0	1	2	3	4	5	6	7	8	9	10	11
	Day	MoTh	Year	Hour	Minut e	Temp. value	Dist. value	Gas value	Cover Alarm	Temp. Alarm	Dist. Alarm	Gas Alarm
0	x	x	x	x	x	x	x	x	x	x	x	x
1	x	x	x	x	x	x	x	x	x	x	x	x
2	x	x	x	x	x	x	x	x	x	x	x	x
3	x	x	x	x	x	x	x	x	x	x	x	x
4	x	x	x	x	x	x	x	x	x	x	x	x
5	x	x	x	x	x	x	x	x	x	x	x	x

Figure 7.4: Expected output saving data into the array

The microcontroller must store the data and keep it until the array becomes full. Then, wipe out the array and start again.

#### 7.1.2.2 String output

When required in each alarm, the unit must convert part of the array into a string and send it to the server. Therefore, here is another two outputs, the string, and the sending.

```
mystring =
x, x, x, x, x, x, x, x, x, x, x, x,
x, x, x, x, x, x, x, x, x, x, x, x,
x, x, x, x, x, x, x, x, x, x, x, x,
x, x, x, x, x, x, x, x, x, x, x, x,
```

Figure 7.5: Expected conversion from the array to a string

Note that the string is created writing commas between each value, this is because when sent, the server cannot differentiate values, but .CSV-files (Comma Separated Values) are importable by most databases. The colors are just for the better understanding.

The message sent must be the same as the string, but as a SMS in a mobile phone.

### 7.1.3 Actual output

However, what can this unit really do?

For being able to see all the answers of the microcontroller, it is necessary to print all the values into the serial monitor that Energia offers. This is an easy way to see what is happening when the code is running.

#### 7.1.3.1 Real Array output

The next part of the code in the Figure 7.6 shows how to visualize the array.

```
//visualize array
for ( int x = 0; x < counter ; x++) {
  for (int z = 0; z < 12; z++) {
    Serial.print(myArray[x][z]);
    Serial.print("\t"); // add a space
    delay(10);
  }
  Serial.println("");
}
```

Figure 7.6: Code to visualize the array created

As it can be seen, it is used the same system to point the different cells into the array. By just defining two index, “x” and “z”, it is possible to move along the entire array. Thus, it is possible to read each value.

Using the command “Serial.print(myArray[x][z])”, when the “x” or “z” change, the pointer moves through the array into a different cell. Then, the value corresponding to that cell is printed. Afterwards, a space is added and a new line just for the better understanding. The Figure 7.7 is taken from the Serial monitor in Energia, and shows the real answer that the microcontroller is giving.

10	4	16	8	29	26	215	0	0	0	0
10	4	16	8	29	26	215	0	0	0	0
10	4	16	8	29	26	213	0	0	0	0
10	4	16	8	29	26	213	0	0	0	0
10	4	16	8	29	26	214	0	0	0	0
10	4	16	8	30	26	237	0	0	0	0

Figure 7.7 Real array output

All the values are correctly stored into the array as Table 7.1 shows.

Table 7.1: Values in the array

0	1	2	3	4	5	6	7	8	9	10	11
Day	Moht	Year	Hour	Minute	Temp. value	Dist. value	Gas value	Cover Alarm	Temp. Alarm	Dist. Alarm	Gas Alarm
Answer in:					Celsius degrees	Centimeters	%	0/1	0/1	0/1	0/1

The answer is given every 15 seconds; this is just for see how the array grows. In addition, the alarms are in zero because there are non-planned readings.

### 7.1.3.2 Real String output

When the string is created, it is so easy to visualize, because it is like a word. In other words, it is not a group of numbers; it is a sequence of characters.

```
Serial.println(mystring);
```

Figure 7.8: Command to visualize the string created

Therefore, Figure 7.9 shows what is possible to see into the Serial monitor.

```
10,4,16,8,29,26,215,0,0,0,0,0,
10,4,16,8,29,26,214,0,0,0,0,0,
10,4,16,8,29,26,213,0,0,0,0,0,
```

Figure 7.9: Real String output

Note that the string is represented with “new line” every new line because the Serial Monitor can understand when it has to create a new line (“\n”).

### 7.1.3.3 Message sent

After creating the string, the unit has to send the information using the GSM module. Then, the string is sent directly to the GSM module. The answer received in the Serial Monitor from the GSM module is shown below in Figure 7.10.

```
Sending Text...
Text Sent.
```

Figure 7.10: Answer received from GSM module

Time to check if the unit really has sent the message, Figure 7.11 proves it.



Figure 7.11: Real screen shot showing the output from the unit

#### 7.1.3.4 Comments

The program was written with performance in mind. Functionality and optimization were the two main characteristics while programming. Finally, as it can be checked, the outputs expected are the same as the real outputs, which that mean the goals are been accomplished and the unit is working well.

Of course this code can be improved and give more functionalities to the whole system, however, right now it is just a prototype, which proves that the manhole covers can be intelligent.

#### 7.1.4 Testing sensors

All components have been tested individually, all sensors worked as required. For more details on sensors test, see attachments:

- Attachment M: Level sensor test
- Attachment N: Cover sensor test
- Attachment O: Gas sensor test
- Attachment P: Temperature test

#### 7.1.5 Testing antenna/GSM and SD card

A simple mobile test was executed, along with test of GSM module and test of different SD-cards. The tests can be found in the attachments:

- Attachment L: Mobile test
- Attachment Q: GSM module test.
- Attachment R: SD-Card speed test

#### 7.1.6 Testing power consumption

Furthermore a test of the power consumption was done. See attachment for further information.

- Attachment S: Power consumption test

### 7.2 Software test

The test was executed under different scenarios to determine its level accomplishment in regards to “Test Plan Document”. It was tested on Windows 10 and can be concluded that it meets the major functional requirements of the Software. It can display data from the database and users can search through the data for desired results with specific searching criteria. Administrators can register new manhole information and edit information about manhole, administrator and company. It is also possible for users to save pages as pdf file and eventual printing of pages.

For more details about the Software test and Software see:

- Attachment T: Software test document
- Attachment U: Software document

How to use and install the program see:

- Attachment V: User manual
- Attachment W: Installation guide.

## 8 DISCUSSION

In this chapter, it will be discussed some of the components of the project that were not finished, and what is recommended for the future of the system.

### 8.1 Components not finished

Some of the components that are not completed are critical to the project, while others should be implemented for a better product. The components that are critical are marked.

#### 8.1.1 SD-card

In the research, it is stated that SD-card would be used to store information in the unit, but two factors stopped the progress of implementing the SD-card to the unit:

- As an atypical microcontroller was chosen, the MSP430FR5969, the libraries to the SD-card needed to be made from scratch. Meaning extensive knowledge in programming and libraries to make the SD-card work. One solution would be to use the Arduino, which has greater community support and integrated SD-card solutions, instead of the MSP430FR5969, but the group, when comparing the two microcontrollers ruled this out. As stated in research, the MSP430FR5969 was chosen over Arduino for multiple reasons.
- The second factor for excluding the SD-card is that the MSP430FR5969 has a unified memory, meaning one can store the code and other information simultaneously. As stated in the chapters “Research” and “Programming” it was created an array to store the values of the sensors directly in the microcontroller. There is also explained in the SD-card part of the research why the unified memory will last longer than the SD-card.

#### 8.1.2 GSM and database connection (CRITICAL)

To connect the GSM transmission of information directly and automatically to the database it was needed a GSM-module that one connects to the PC/server. The information was then needed to be automatically imported to the database to get “live-feed” of the alarms. Xepto probably has this technology, and it would be of great use for us.

As stated earlier, the group wanted to automatically store the information from the GSM to the database. Some of the groundwork has already been done. The message that the unit sends via GSM, is already have converted to a CSV-message (Comma Separated Values). Some research has been done, and as shown in some communities, one can easily import CSV-files to the database, and all the information will be sorted in the correct places.

Although, it was harder to understand how to automatically make the database import CSV-files when a new message was received. The most likely solution was to make a script that read the files, once new information came in.

#### 8.1.3 Gas proof box and connectors

As stated in the research visit to Skien municipality, and something they made us aware about was the issue with gas. The workers thought the gas seeped in through the cracks of the old unit and corroded the microcontroller.



The group did extensive research and found a gas proof box, but the water-resistance was too poor. The unit can at times be submerged in water, and to find a box that is gas proof and simultaneously waterproof was a harder task than what was thought in advance. At the same time, the group did not want to order a product of a great cost and which was uncertain met the right criterias.

The group were in contact with a company that sells gas proof connectors, which have a ceramic base. They were fair enough to tell that if the connectors were in a hazardous environment, and disconnected the connectors (for example to change the batteries), the gas could leak inside the connectors and the hazardous gas could end up being sealed inside the connector. Their warranty was just one week after disconnection, because the gas could end up corroding the connectors from the inside, with this procedure.

### 8.1.4 Battery misunderstanding

After a physical quick test, it was tried to mount the battery block and, suddenly, the pin headers became too hot. Thus, the batteries were removed from the socket.

Then, there was a small research of this fact. The answer it was simple, the three batteries in series have 3.6V and the one in parallel has 1.2. As the parallel requires the same amount of voltage, the pins became hot because there is a decompensation of energy in the parallel circuit, in other words, the battery alone is forced to drop the voltage the other three.

### 8.1.5 Sealing the box with polyurethane

A way to solve the issue with gases seeping inside the unit, a problem stated in the research, was to seal the entire box with polyurethane. This has the advantage that the unit becomes completely gas proof and waterproof. The downside of this solution is that the seal of the polyurethane would have to be broken, or the use of connectors would come in use (see 8.1.3 for problems with connectors). Furthermore, sealing the box with polyurethane was not completed, with the enclosure of the unit and batteries being completed too late in the project.

## 8.2 Implementations in the future

This chapter will explain all the features that the group thinks will enhance the product and make the unit more attractive to users. It is recommended implementing these features in the future.

### 8.2.1 GSM and database connection

As stated in chapter 8.1.2 (GSM and database connection (CRITICAL)) the connection between GSM and the database should be finished before deploying the product.

In addition, a feature that will be very useful is to make the possibility of updating the microcontroller remotely. Meaning through the GSM check the firmware and update to the newest version if necessary.

### 8.2.2 Use of GSM technology

As stated in the research part, the unit only uses 2G, because of small data and the need for greater speeds is not necessary. An idea is to make use of the 3G for greater speed, but the area of coverage will be smaller. In the future, the coverage for higher speed will be greater, and

when this technology is in place one can use faster speeds as 3G, 4G and others for transmitting information.

### 8.2.3 Implementing a SD-card

If one chooses to use 3G or other technologies, or chooses to store bigger files (see chapter 8.2.4 Implementing camera inside the manhole) in the unit, a greater storage-space will be needed. For the use of the unit, the 64 kB unified memory is enough as the information sent through GSM will be stored in a database, and long-term saving to the unit-memory is redundant. If one stores bigger files, or uses the unit in another way, a SD-card (or another technology with greater storage space) may have to be implemented.

### 8.2.4 Implementing a camera inside the manhole

Ulefos stated in the beginning the idea of implementing a camera inside the manhole to observe if there was some problems. This increases the need for a bigger storage-space if they want to store a picture when the alarm is triggered. As of now, there is no space for files bigger than 64kb (minus the code) in total. (See chapter 8.2.3 Implementing a SD-card)

In addition, for creating a live stream, one should use a technology greater than 2G, as the bandwidth is too slow. For live streaming video, or sending picture-by-picture, one should use 3G or greater.

Another point is that making the program for the camera and the implementation (placement in the manhole) was deemed too difficult to focus on in this project. If there is a great need for a camera, one should consider implementing it in the future.

### 8.2.5 Bluetooth

Originally planned, but turned to a future feature, was adding a Bluetooth module to the unit. There are multiple applications for using the Bluetooth:

1. When the cover is opened, and an interruption in the cover sensor appears, the Bluetooth is enabled. The workers can connect to the Bluetooth and see all the information in the manhole. This also come in handy if the GSM stops working, and the workers can connect with Bluetooth as a check if the unit in itself is working.
2. Connecting various components to the Bluetooth, i.e. a camera. When the alarm is enabled, the Bluetooth and camera are enabled to take a picture (see chapter 8.2.4 Implementing a camera inside the manhole)
3. The Bluetooth can be used with a simple interface for the workers to connect and mark that they have been in the manhole. I.e. same procedure as point 1.

### 8.2.6 Gas proof box and connectors

The current box that the group acquired is a standard and simple box from a Norwegian company called Clas Ohlson. This box is only an IP65 standard (meaning it cannot be submerged in water and is not gas proof) (See chapter 8.2.6 Gas proof box and connectors).

In the future, the box that is recommended is a gas proof and water-submerging proof box. In addition, there should be a solution for either connecting a separate box of batteries to the main

unit with gas proof connectors, or in an other case, using a gas proof and water-submerging proof box that the unit with batteries are inside.

Either, the sealing with polyurethane should be considered, but if one finds a box that meets the criteria stated before, this probably would not be necessary.

### 8.2.7 Considering separation or unification of batteries with unit

Either having the batteries in the same box or separating the batteries from the main unit in an own box, is a discussion on its own. The companies responsible should consider the necessity of separating components or unifying them, and the cost of both solutions.

Having all components in the same box means that when the batteries of one unit drains out, one has to take out the unit that is power-empty and replace with another to charge the one that is power-empty. Therefore, with choosing this solution, the companies always need two, or more, units at all times.

While separating the unit from the batteries, means one must find the correct connectors.

### 8.2.8 Safety measures

Another point, referring to the schematic design, is to have damage preventive analog electronics inside the unit to protect from short-circuiting or other electrical problems.

Also as another safety measure, one should consider separating the batteries from the unit, either inside the box or completely in an other box, in case of leakage.

### 8.2.9 Gas sensor readings

It is recommend in the future to trim the code, so that the gas sensor only reads when the cover is opened. This is for energy conservation, and unit active time. As of now, the gas sensor needs to heat up the resistors to make a correct reading. This operation takes 30 seconds, where the gas sensor draws 61 mA, and the whole unit is in active mode.

### 8.2.10 Smaller system

As of now, the unit consist of the microcontroller, sensors, GSM module, antenna and battery. In the future, it is recommend making the system smaller, as this easily can be done. Some parts of the microcontroller are not used, and the components that are used can be connected one-by-one, and by hand, summarizing in a smaller product.

Furthermore, when the unit is smaller, a smaller enclosure can be used, meaning it is easier to install, and can be used in smaller areas.

### 8.2.11 Switch sensor for water (CRITICAL)

For the future, it is recommended to have a sensor to detect sudden rise of water level. At this point, the unit can only read when it is in active mode. Meaning a sudden rise of water level will not be detected, only a difference from the last reading.

### 8.3 Comparison of Xepto unit vs. new unit

A short overview over what the group has accomplished, compared to the unit Ulefos got from Xepto. This is presented in

Table 8.1.

Table 8.1: Comparison of Xepto unit vs. new unit

	Xepto	Our unit
<b>Enclosure</b>		
Gas proof enclosure	No	Yes
Water-submerging proof	Yes	Yes
Small unit?	No	Yes
Need for bracket	Yes	No
Universal placement	No	Yes
<b>Communication and hardware</b>		
GSM hardware	Yes	Yes
GSM transmission	Yes	Yes
GSM reception	Yes	Yes
Antenna placed in enclosure	No	Yes
Antenna placed beneath cover	No	Yes
<b>Sensors</b>		
Wireless water-level sensor	No	Yes
Water-level alarm	Yes	No
Water-level readings	No	Yes
Water-level interruption	Yes	No
Temperature sensor	No	Yes
Temperature alarm	No	Yes
Gas sensor	No	Yes
Gas sensor alarm	No	Yes
Cover sensor	Yes	Yes
Cover sensor alarm	Yes	Yes
<b>Battery</b>		
Battery endurance	Medium	High
Rechargeable batteries	No	Yes
Size of batteries	Big	Small
<b>Storing information</b>		
Store information locally	No	Yes



## 9 SUMMARY

The old solution that Ulefos Jernværk had was assessed, researched, and furthermore improved. As stated in the research, the old solution only had a water-level alarm and a cover-alarm. The new solution can in addition measure temperature, dangerous gases, and show current water-level, with alarms for every sensor.

The new solution has all sensors, GSM, antenna and microcontroller placed in one box. The old solution was placed in the manhole-cover and needed a special bracket for the manhole-cover. The new solution can be placed anywhere in the manhole, but the cover-sensor needs to be placed near the manhole-cover, as needed in the old solution. With the new solution having the possibility to be placed anywhere in the manhole, and not needing brackets makes it less expensive.

Furthermore in the new solution, as stated in the report, features can be added later.

In a perspective for the future, this unit can collect data from all the manholes. Making it cheaper and more available, it can reduce the workload of the pump-stations, and show the state and problems that occur, directly in the manholes.

To summarize the project and report, the group feels that the new solution is better and cheaper than the old. The group has acquired new knowledge, in terms of programming and product-solution. As the solution is an early prototype, with more time, the group could produce an even better product, with more features, and of course a complete product.

## REFERENCES

- [1] "scitechnol," 2016. [Online]. Available: <http://www.scitechnol.com/embedded-system/microprocessor-microcontroller.php>
- [2] "maxkalavera," 2012. [Online]. Available: <http://maxkalavera.blogspot.no/2012/05/microcontroladores.html>
- [3] "Elprocus," 2013. [Online]. Available: <https://www.elprocus.com/microcontrollers-types-and-applications/>
- [4] "instructables," 2016. [Online]. Available: <http://www.instructables.com/id/How-to-choose-a-MicroController/?ALLSTEPS>
- [5] "Electronics.stackexchange," 2009. [Online]. Available: <http://electronics.stackexchange.com/questions/2324/why-are-atmel-avr-so-popular>
- [6] "Texas Instruments" 2016. [Online]. Available: [http://www.ti.com/lscds/ti/microcontrollers\\_16-bit\\_32-bit/msp/ultra-low\\_power/overview.page](http://www.ti.com/lscds/ti/microcontrollers_16-bit_32-bit/msp/ultra-low_power/overview.page)
- [7] "Texas Instruments" 2016. [Online]. Available: [http://www.ti.com/lscds/ti/microcontrollers\\_16-bit\\_32-bit/msp/ultra-low\\_power/msp430frxx\\_fram/what\\_is\\_fram.page#fram-base](http://www.ti.com/lscds/ti/microcontrollers_16-bit_32-bit/msp/ultra-low_power/msp430frxx_fram/what_is_fram.page#fram-base)
- [8] "Texas Instruments" 2016. [Online]. Available: <http://www.ti.com/product/MSP430FR5969>
- [9] "engineering " 2013. [Online]. Available: <http://www.engineering.com/ElectronicsDesign/ElectronicsDesignArticles/ArticleID/9245/LaunchPad-Development-Kit-Simplifies-Debugging.aspx>
- [10] "Texas Instruments\_microcontroller\_userguide," 2016. [Online]. Available: <http://datasheet.octopart.com/MSP-BNDL-FR5969LCD-Texas-Instruments-datasheet-26823934.pdf>
- [11] "Texas Instruments\_microcontroller\_datasheet," 2016. [Online]. Available: <http://www.ti.com/lit/ds/symlink/msp430fr5949.pdf>
- [12] "engineersgarage " 2012. [Online]. Available: <http://www.engineersgarage.com/articles/gsm-gprs-modules>
- [13] "IndianTelecomNews," [Online]. Available: <http://www.indiantelecomnews.com/mobile-internet-standards-2g-3g-4g-hspa-lte-evdo-gprs-edge-rtt>
- [14] "University of St Andrews," University of St Andrews, 2016. [Online]. Available: <https://www.st-andrews.ac.uk/itsupport/mobile/dataconnectionspeeds/>
- [15] "telenor\_dekningskart," 2016. [Online]. Available: [https://www.telenor.no/privat/dekning/dekning\\_data.jsp](https://www.telenor.no/privat/dekning/dekning_data.jsp)
- [16] "simcomm2m," 2015. [Online]. Available: <http://simcomm2m.com/En/module/?type=19>
- [17] "simcomm2m," 2015. [Online]. Available: <http://simcomm2m.com/En/module/detail.aspx?id=138>

- [18] "batterylogic," 2015. [Online]. Available: <http://www.batterylogic.co.uk/nimh-d-rechargeable-battery.asp>
- [19] "Sparkfun," Sparkfun, 2016. [Online]. Available: <http://cdn.sparkfun.com/datasheets/Components/General/SDSpec.pdf>
- [20] "Whatis.techtarget," 2016. [Online]. Available: <http://whatis.techtarget.com/definition/EEPROM-electrically-erasable-programmable-read-only-memory>
- [21] "Texas Instruments\_slaa334a," 2016. [Online]. Available: <http://www.ti.com/lit/an/slaa334a/slaa334a.pdf>
- [22] "LeifNorman," 2016. [Online]. Available: <http://www.leifnorman.net/how-big-is-it>
- [23] "makeuseof," 2014. [Online]. Available: <http://www.makeuseof.com/tag/extend-life-raspberry-pis-sd-card/>
- [24] "Honeywell," Honeywell, 2015. [Online]. Available: <http://sensing.honeywell.com/SSCEB31C-MICRO-SWITCH-Medium-Duty-Limit-Switches>
- [25] "Brokentoaster," 2006. [Online]. Available: <http://www.brokentoaster.com/arduinoomp3/files/Sandiskmanual-SecureDigital2.2.pdf>
- [26] "MaxBotix," 2015. [Online]. Available: [http://www.maxbotix.com/documents/XL-MaxSonar-WR\\_Datasheet.pdf](http://www.maxbotix.com/documents/XL-MaxSonar-WR_Datasheet.pdf)
- [27] "MaxBotix," 11 11 2013. [Online]. Available: [http://www.maxbotix.com/documents/MaxBotix\\_CE%20Declaration%20of%20Conformity.pdf](http://www.maxbotix.com/documents/MaxBotix_CE%20Declaration%20of%20Conformity.pdf)
- [28] "MaxBotix," 2015. [Online]. Available: [http://www.maxbotix.com/documents/MaxBotix\\_RoHS\\_Compliance\\_Statement.pdf](http://www.maxbotix.com/documents/MaxBotix_RoHS_Compliance_Statement.pdf)
- [29] "MaxBotix," MaxBotix, 5 12 2010. [Online]. Available: [http://www.maxbotix.com/documents/MaxSonar\\_Reliability\\_Demonstration\\_Test\\_RDT\\_and\\_MTBF\\_Report.pdf](http://www.maxbotix.com/documents/MaxSonar_Reliability_Demonstration_Test_RDT_and_MTBF_Report.pdf)
- [30] "ucpros," 2009. [Online]. Available: [http://pub.ucpros.com/download/tgs2602\\_product\\_information\\_rev02.pdf?osCsid=38r4q80vn31q41329blblkja5](http://pub.ucpros.com/download/tgs2602_product_information_rev02.pdf?osCsid=38r4q80vn31q41329blblkja5)
- [31] "Microcontrollershop," 2016. [Online]. Available: [http://microcontrollershop.com/product\\_info.php?products\\_id=6789](http://microcontrollershop.com/product_info.php?products_id=6789)
- [32] "Adafruit," Adafruit, 2005. [Online]. Available: <https://www.adafruit.com/datasheets/DS18B20.pdf>
- [33] "Marechal," 2016. [Online]. Available: [www.marechal.com](http://www.marechal.com)
- [34] "Wikipedia," 2016. [Online]. Available: [https://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio)
- [35] "Advantages & Disadvantages," 2016. [Online]. Available: <http://www.cs.iit.edu/~cs561/cs425/VenkatashSQLIntro/Advantages%20&%20Disadvantages.html>
- [36] "Nema7," 2013. [Online]. Available: <http://www.nema7.com/index.html>



- [37] "Hubpages," 2016. [Online]. Available: <http://hubpages.com/technology/Which-is-better-Nickel-Metal-Hydride-NiMH-or-Lithium-Ion-Li-ion-batteries>
- [38] "MichaelBluejay," 2016. [Online]. Available: <http://michaelbluejay.com/batteries/>
- [39] "Rei," 2016. [Online]. Available: <http://www.rei.com/learn/expert-advice/batteries.html>
- [40] "Energia," 12 09 2015. [Online]. Available: <http://energia.nu/>
- [41] "Maritex," 3 5 2015. [Online]. Available: [https://www.maritex.com.pl/media/uploads/PRODUKTY\\_PDF/wireless/Simcom/SIM800-Hardware-Design-V1.08.pdf](https://www.maritex.com.pl/media/uploads/PRODUKTY_PDF/wireless/Simcom/SIM800-Hardware-Design-V1.08.pdf)
- [42] "Honeywell," 2015. [Online]. Available: <http://sensing.honeywell.com/honeywell-sensing-micro->

## **ATTACHMENTS**

Attachment A: Assignment

Attachment B: Goal formulation

Attachment C: WBS

Attachment D: Progress plan

Attachment E: MSP430FR5969 Letter explanation

Attachment F: Gas proof connector offer

Attachment G: SRS & SDD

Attachment H: Software code

Attachment I: Unit code

Attachment J: Tables

Attachment K: Test plan document

Attachment L: Mobile test

Attachment M: Level sensor test

Attachment N: Cover sensor test

Attachment O: Gas sensor test

Attachment P: Temperature test

Attachment Q: GSM module test

Attachment R: SD-Card speed test

Attachment S: Power consumption test

Attachment T: Software test document

Attachment U: Software document

Attachment V: User manual

Attachment W: Installation guide