Energy-efficient monitoring system for fire extinguishers

FREDRIK BERG
LUDVIG LIFTING
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Lunds universitet

Lunds Tekniska Högskola

Energy-efficient monitoring system for fire extinguishers

Fredrik Berg
his15fbe@student.lu.se

Ludvig Lifting
lu8023li-s@student.lu.se

Supervisor:
Niklas Gustafson, Mikrodust

Fredrik Tufvesson, Lund University

Examinator:
Ove Edfors, Lund University

February 5, 2023

LTH, Department of Electrical and Information Technology

Abstract

Fire extinguishers are a requirement for fire safety in every type of building today. The effectiveness of a fire extinguisher is however dependent on it not leaking and keeping the pressure. To assure that this is not the case, maintenance is recommended at regular intervals, by the owners themselves at monthly intervals, and once per year by professionals. This however leaves the fire extinguishers unmonitored for long periods of time, up to a year at worst, which could prove to be a fatal problem.

The goal of the work presented in this report is to successfully automate the regular maintenance of fire extinguishers such that only regular service is needed.

Different methods of monitoring the pressure of fire extinguishers were explored, such as mounting a small camera module and using computer vision to detect changes in the pressure gauge. The use of strain gauges to continuously weigh the fire extinguisher to detect any changes was also examined. To monitor tampering the use of an inertial measurement unit as well as other types of sensors like a reed switch were explored.

The final solution, using a small camera module combined with a low-power MCU (Micro Controller Unit) and a reed switch, was able to detect small changes in pressure accurately. It was also efficient enough to be operational for at least ten years off of a single standard AAA battery.

Keywords: **I**oT, **L**oRa, **M**onitoring, **F**ire safety, **W**ireless, **F**ire extinguisher, **E**nergy-efficient, **I**mage processing.

Preface

This thesis has been carried out at Mikrodust AB and commissioned by CE-BON, through the Faculty of Engineering (LTH), Department of Electrical and Information Technology, at Lund University.

We want to express our sincere gratitude to Mikrodust AB and all of its employees for accommodating us, allowing us access to their lab when needed, as well as answering our numerous questions. An additional thanks to Niklas Gustafson, Noroz Akhlagi, and Melker Lang for their extra efforts in helping us achieve a successful thesis.

We would also like to thank our supervisor and examiner, Fredrik Tufvesson and Ove Edfors. Their guidance and support have been essential during this thesis.

Contents

Abbreviations

- MCU - Microcontroller unit
- IMU - Inertial measurement unit
- LPWA - Low power wide area
- LpWAN - Low power wide area network
- LoRaWAN - Long range wide area network
- IoT - Internet of Things
- LTE CAT-M1 - Long Term Evolution (4G), category M1
- eDRX - Extended Discontinuous Reception
- PSM - Power Saving Mode
- NB-IoT Narrowband Internet of Things
- IR - Infrared

Chapter 1

Introduction

1.1 Motivation and background

Fire extinguishers are, for fire safety, a requirement in every commercial and public building. They are pressurized canisters containing a fire extinguishing agent, with their ability to put out fires heavily depending on pressure. Therefore, regular service and maintenance must be performed to ensure that the canisters are holding the correct pressure. The maintenance is performed manually at regular intervals, usually once per year and the service is typically performed once every five to ten years. Maintenance implies inspection of pressure and placement, where the pressure must be in the acceptable range (10-12 bar) and the placement of the fire extinguisher is such that it is clearly visible, unblocked, and in the correct attachment.

The relatively long time periods in between regular maintenance increase the risk of critical failures going undetected, such as the pressure dropping below the specified range or the fire extinguisher being misplaced, both of which greatly reduce its efficiency. Manual maintenance also introduces a possibility of human error which, when inspecting hundreds of fire extinguishers each day, only grows more probable.

In light of this, automating regular maintenance seems like a viable way forward. The regular service, however, cannot as of yet be automated and is required to be performed at the specified intervals regardless of the condition of the fire extinguisher.

1.2 Purpose and aims

There is much to be gained by automating the maintenance process, both in regards to more frequent checkups of pressure and placement, bringing more control and confidence in fire safety, as well as mitigating human error.

The goal of the work presented in this report is to successfully automate the regular maintenance of fire extinguishers such that only regular service is needed.

This means that the IoT device needs to be operable for the entire expected life cycle of the fire extinguishers.

More specifically, this report is aiming to answer the question of whether such a system, automating the process of regular maintenance, is viable to design and implement, with the constraints of it being non-invasive and having a maintenance-free run-time of 10 years.

1.3 Scope and limitations

This report investigates possible solutions for monitoring fire extinguishers, with the limitations of it being an "add-on" product for the fire extinguishers currently circulating on the market. This entails that the solutions should be non-invasive and not block or modify any features of the already existing fire extinguishers. Monitoring, in this case, implies keeping track of the canister pressure as well as detecting some forms of tampering.

There are many different models of fire extinguishers, and this report covers any model with an attached manometer. The design of manometers differs greatly around the world, therefore this report will focus exclusively on manometers made for the European market, specifically the EU.

The final system should be battery-powered and have a battery life of 10 years (Maximum service interval) with a single charge. It should also be able to communicate wirelessly with a monitoring application.

1.4 Method and outline

The necessary theory will be presented, which will include key technologies that are processed in this report as well as relevant concepts. This is followed by an introduction to various possible solutions, aiming to solve the essential problems of designing the proposed system. This section also presents implementational details of the solutions. The solutions are then compared to each other and discussed. This is to determine the most promising approaches and eliminate inferior paths. The experimental setup used for experiments and testing is then presented, and a baseline of its performance is calculated. This is to achieve some real experimental data for theoretical comparison. The final system is presented and its performance and power efficiency are evaluated, followed by a brief cost evaluation. Lastly, conclusions are made followed by a thorough discussion of future work and various optimizations.

1.5 Related work

There has been considerable progress in the field of automation in the last few decades, where more and more labor-intensive and time-consuming tasks have been automated. Everything from measuring water and electricity to multiple chores at home. IoT technologies have also started to make their way into security solutions. Products like smoke alarms, vibration sensors for monitoring windows, climate sensors, and much more are all interconnected using a hub

provided by the security company. The gains are obvious from the increased security of constant monitoring, but by eliminating menial tasks workers have also been freed to focus on more pressing problems at hand.

Academically there has also been a lot of work done in the last years comparing different LpWAN technologies, implementations of fire security systems as well as multiple overviews of the state of IoT and Industry 4.0.

In the papers "IoT-Based Fire Alarm System" published in 2019 [1] and "Design and implementation of the mobile fire alarm system using wireless sensor networks" published in 2016 [2], two different groups examine different methods to implement a wireless sensor network that improves fire safety in residential buildings. While some components are different, the underlying method is the same. They both want to construct a wireless sensor network consisting of nodes, where one node uses multiple different sensors to detect changes in the surrounding environment like temperature, gas, and much more. These sensors are then connected to a microcontroller which processes and communicates with a central node, and if any sensor values are outside of the allowed span the central node sends an alert to the homeowner.

In the master's thesis "Smart Environment - Early Wildfire Detection using IoT" [3], done at LTH in the spring of 2019, the writers are investigating what an IoT device for the detection of wildfires could look like. Another master's thesis, "The User Experience Perspective of Internet of Things Development" [4], has looked at the development of IoT devices from a UX perspective. It did so by interviewing 11 manufacturers of IoT devices and doing a literary overview to find different perspectives of the development process, both positive and negative. In October 2017 the paper "Energy Efficiency across Programming Languages" [21] the writers compared 27 of the most popular programming languages to find which one is the most efficient in terms of memory usage, energy consumption as well as runtime.

"Low Power Wide Area Network (LpWAN) Technologies for Industrial IoT Applications"[40] is a master's thesis written at LTH in the spring of 2018 by students at the department of Electrical and Information Technology. In it, they studied multiple popular LpWAN alternatives to compare them and find out which were best suited for a wide variety of applications. In November 2021 Gilles Callebaut published his doctoral thesis "Advances in Single and Multi-Antenna Technologies for Energy-Efficient IoT" [8] where the energy efficiency of IoT technologies was studied and techniques for improvement were explored.

So, in this thesis, the available source material, especially [40], [21] and [8], will be used to create an IoT application that is as energy efficient and accurate as possible.

Chapter 2

Theory

This chapter presents the theory necessary to understand the content in subsequent chapters. It has been structured in a way that the necessary information regarding fire extinguishers is presented first followed by the theory regarding the different parts of an IoT device.

2.1 Fire Extinguishers

Figure 2.1: A standard fire extinguisher

Figure 2.1 shows how a standard fire extinguisher looks. The current standard for fire extinguishers in Sweden is based on the European standard EN3, SS-EN3 here in Sweden. According to this standard, all the various types of fire extinguishers are classified like this:

1) Class A, for fires in solid materials where there will be embers. The suitable extinguishing agents for this class are water, foam, or powder. There is also a standard procedure for certifying extinguishers within the A-class. The different classes within the A-class tell us how large a test fire, consisting of a $0.5 \times 0.5 \times X$ meter large wood pyre as seen in Figure 2.2, that can be extinguished.

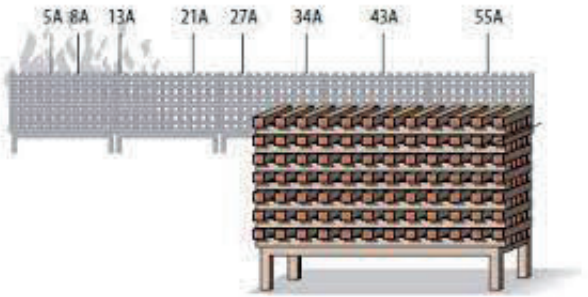


Figure 2.2: Test fire for class A fire extinguisher. Image from [7]

For example, if a fire extinguisher is classified as 13A it can put out a test fire of volume $0.5 \times 0.5 \times 1.3$ meters.

2) For fires in liquids one should use a fire extinguisher from the B-class and the standard extinguishing agent within this class is carbon dioxide. There is also a particular class of extinguishers specifically for grease fires. The F-class and the extinguishing agent is a special carbonic acid for this application. For B-class extinguishers, there is also a standardized test to classify them. The test is performed by filling a circular vessel with heptane and water (2/3 heptane and 1/3 water) as seen in Figure 2.3 and then using the extinguisher to put out the resulting fire. The efficiency classification is derived from how many liters of burning liquid there are. This same test is done to classify F-class extinguishers, but the liquid is then greased instead.

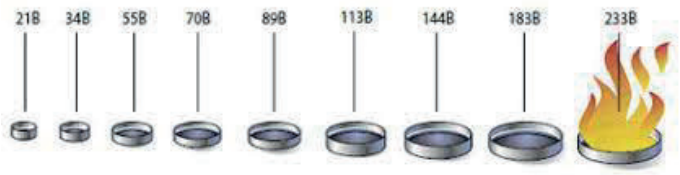


Figure 2.3: Test fire for class B and F fire extinguishers. Image from [7]

3) There is also the so-called C-class and extinguishers from this class are best for fires in gases.

4) Finally there is also the D-class which suggests that the extinguisher is suitable for fires in metals.
Neither the C-class nor the D-class has standardized efficiency testing. As both

these extinguishers are highly circumstantial and it is up to the manufacturer to make sure that the extinguishers are suitable for what they put in [5].

The maintenance of the fire extinguishers is then specified in the standard SS3656 [6]. It recommends the following maintenance at different time intervals. The standard says that the owners of the fire extinguisher should do some basic check-ups consisting of 1) That the extinguisher is visible in its designated position. 2) That the extinguisher is not blocked in any way. 3) That the user instructions are facing outwards and readable 4) There is no visible damage to the extinguisher 5) The pressure gauge is showing the correct pressure 6) That the extinguisher still has its seal. It is recommended that this maintenance is conducted each month, but the time period is flexible. It should be conducted at least once every three months though.

Once per year, there should be a trained service technician that comes to check all fire extinguishers on site. During this inspection, the extinguishers are inspected in accordance with the standard and any parts that are worn out will be replaced.

Finally, the extinguishers will have a thorough inspection, called a workshop inspection, done by a trained technician where all parts of the extinguishers are checked, but internally and externally. Beyond the checks performed during yearly check-ups, the technician will also: 1) Empty the fire extinguisher and inspect it internally (this is especially important for extinguishers that have a coating on the inside). 2) Deconstruct the valve and other loose parts on the fire extinguisher to change gaskets and any other worn-out or broken components according to the manufacturer's instructions. 3) Pressure test the hose and nozzle if they are installed on the fire extinguisher. 4) Change the extinguishing agent in accordance with the manufacturer's instructions. This workshop inspection is carried out every fifth year for liquid extinguishers while it is carried out every tenth year for powder and carbonic acid extinguishers. Once this type of inspection is carried out the extinguisher should be marked for it [6].

2.2 Internet of Things

According to Gilles Callebaut [8], the Internet of Things describes the exchange of information between different objects that have not been able to communicate before, for example, a fridge. This communication usually happens over the internet, but other communication protocols are available. According to this definition, IoT can be broken down into three separate components; the IoT device or the physical object referenced above, the communication link, and the gateway that is connected to the internet [8].

2.3 Computer vision and image processing

Computer vision is the field within computer science and artificial intelligence that enables computers to interpret and understand the natural world. Research within the field began in the early 1950s and following decades of progress in both computing power and algorithms, the accuracy of the systems has risen drastically [12].

The way computers identify objects in images starts with how images are represented digitally. Each image is made of multiple smaller elements called pixels which all have a value that corresponds to a specific part of the image. This value, usually in the range of 0-255, represents the intensity level of a given color. For a grayscale image, it corresponds to how white the pixel is with 255 being completely white and 0 being black. While a grayscale image only needs one layer of pixels, three layers of pixels are necessary to create color images. One layer each for the intensity of that pixel in red, blue, and green, as seen in Figure 2.4 [12].

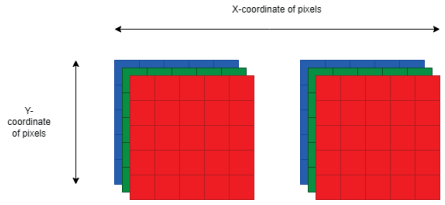


Figure 2.4: Color representation in digital images.

To detect objects in images computer vision uses pixel values to try and extract important features within the image. There are multiple techniques available to detect both edges as well as features. There are classic computer vision techniques like SIFT (Scale Invariant Feature Transform), SURF (Speeded Up Robust Features), Sobel Edge Detection, and many more. These techniques rely on detecting features and dominant pixels in images to determine the content of an image. These techniques are easy for humans to understand and will work on an arbitrary image, but the problem here is deciding which of these features is actually essential.

Then there is the newer field where neural networks are fed large datasets of labeled control data to train the network in order to classify objects in an image. This method lets the computer find the important features on its own. This means that it can become very efficient given large enough data sets, but how the classification is done might seem abstract to humans. Both of these groups of techniques have their own pros and cons, but for this thesis work, traditional methods will be the main focus due to the lack of a large dataset of images of manometers. Specifically, techniques for detecting edges will be highlighted [14].

2.3.1 Edge detection

Another way to detect features and areas of interest within images is to detect the edges of the image. An edge in an image is an area where the values of neighboring pixels are drastically different, which usually corresponds to a point of interest in the image. If this drastic change in pixel intensity follows some type of line or shape it can be described as an edge or a curve in the image. The reason to detect these sharp changes is that these areas usually capture essential events and changes in the image and by extension, the real world [13].

The different methods for detecting edges are described in Section 2.3 estimate

the gradients in different ways, where one such method is the so-called Sobel Filter. It estimates the gradients of the image in both the x-direction (from now called G_x) and in the y-direction (from now called G_y) by convolving it with the two matrices from now on called kernels [16]:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

These two kernels are then convolved with the original image one at a time according to the equations below. The convolution is numerically performed for each pixel by extracting a 3×3 section of the original image with the intended pixel at the center of the image. Then the dot product between the image section and the kernels S_x and S_y respectively, which is then the estimate of the gradient in the x-direction and the y-direction for that pixel [16], see

$$G_x = S_x * A \qquad (2.1) \qquad \text{and} \qquad G_y = S_y * A. \qquad (2.2)$$

Here follows an example calculation using the Sobel algorithm to further clarify.

Given the 6x6 pixel color image I_{in}, as seen in Figure 2.5, whose edges shall be identified using the Sobel operator. Before the algorithm can be started the image needs to be prepared. First, the color image has to be turned into a grayscale image, and second, the image has to be padded with an outer layer of zeros, as seen in Figures 2.6 and 2.7 [16].

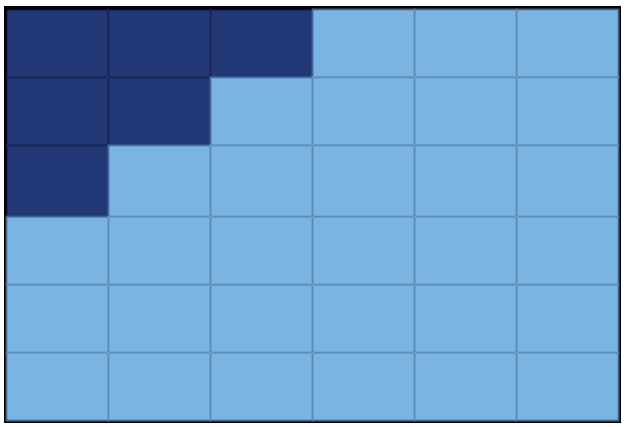


Figure 2.5: Original Image, inspired by [16]

150	150	150	255	255	255
150	150	255	255	255	255
150	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255

Figure 2.6: Pixel values for Greyscale Image. Inspired by [16]

0	0	0	0	0	0	0	0
0	150	150	150	255	255	255	0
0	150	150	255	255	255	255	0
0	150	255	255	255	255	255	0
0	255	255	255	255	255	255	0
0	255	255	255	255	255	255	0
0	255	255	255	255	255	255	0
0	0	0	0	0	0	0	0

Figure 2.7: Padded Greyscale Image. Inspired by [16]

The padding is done to increase the area that the kernel can cover in the image which leads to an increase in accuracy in the analysis. Turning the image from color into grayscale is also done to lessen the amount of data that needs to be analyzed, and the grayscale image contains also as much information regarding points of interest as the original color image. Now that the image has been prepared the two kernels, S_x and S_y, will be convolved with the image to approximate the gradients. The kernel S_x will be used to demonstrate how the calculation is done. To start S_x is overlayed with the upper left corner of the padded image, a 3×3 section of the padded image with the pixel at position (1, 1) at its center. The dot product is then calculated between this section of the image and the kernel by first performing an elementwise multiplication [16].

The resulting values are then added together and become the estimated gradient for the pixel at position (1, 1) [16].

$$0 + 0 + 0 + 0 + 0 + 300 + 0 + 0 + 150 = 450$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 300 \\ 0 & 0 & 150 \end{bmatrix}$$

Figure 2.8: Result of elementwise multiplication in Figure 2.9

This calculation is then repeated for the pixel at position (1, 2) and so on until the gradients have been estimated for all pixels in the original image in both the x- and y-directions [16].

0 * -1	0 * 0	0 * 0	0	0	0	0	0
0 * -2	150 * 0	150 * 0	150	255	255	255	0
0 * -1	150 * 0	150 * 0	255	255	255	255	0
0	150	255	255	255	255	255	0
0	255	255	255	255	255	255	0
0	255	255	255	255	255	255	0
0	255	255	255	255	255	255	0
0	0	0	0	0	0	0	0

Figure 2.9: Calculation of G_x. Inspired by [16]

When this has been done for both kernels the gradients have been calculated. Now to calculate the magnitude total gradient G of the image the following formula is used with the results shown in Figure 2.10 [16]:

$$\sqrt{G_x^2 + G_y^2} = G$$

.

0	0	0	0	0	0	0	0
0	636	713	968	1041	1020	1082	0
0	713	445	445	148	0	1020	0
0	968	445	148	0	0	1020	0
0	1041	148	0	0	0	1020	0
0	1020	0	0	0	0	1020	0
0	1082	1020	1020	1020	1020	1082	0
0	0	0	0	0	0	0	0

Figure 2.10: Calculation of G. Inspired by [16]

2.3.2 Thresholding

Thresholding is, in image processing, used for converting images from grayscale to binary. In other words, the grayscale values which are normally in a larger range of values are converted to either the minimum or maximum of the range. Doing this will reduce the complexity of the image by accentuating areas of interest and removing non-interesting areas. Areas of interest, in this case, refer to areas with large changes in values or areas which are not homogeneous. Another advantage is that it will significantly reduce the amount of data needed to represent the image, which in turn makes operations on it less computationally intensive [17].

Thresholding is done in either a global or adaptive way. Global thresholding implies using a single threshold value for the whole image with which the decision is made. The pixel value is compared to the threshold value and set to the maximum or minimum value depending on it being over or under the threshold value [17].

In adaptive thresholding, the threshold value is calculated individually for each pixel, this can be done in many ways with different functions but a common approach is the nearest neighbor mean. The threshold value is then calculated by computing the mean value of the neighboring pixels, followed by a comparison with the pixel value to make a decision [17].

2.3.3 Filtering

Noise is present in all digital images to some degree, at least those captured in an analog way. This usually does not hinder human's ability to determine the contents of an image, but it can cause trouble for computer vision algorithms. High-frequency noise can cause blurriness while low-frequency noise can make the contents of an image impossible to decipher. To handle this, a variety of filter techniques have been developed to smooth over high frequencies while also enhancing the lower ones to clear an image [18].

There are multiple ways to achieve this, the simplest being a mean filter. The

mean of the pixels surrounding the pixel at position (x, y) is calculated and used for comparison. The value of the pixel at (x, y) can either be set to the mean of its neighbors or can also be given a fixed value if it falls below or exceeds it [18].

2.4 Movement detection

One of the essential problems that have to be addressed by the autonomous monitoring system is how it is verified that the fire extinguisher has not been moved or tampered with in any way. There are many possible ways to monitor this.

2.4.1 Inertial Measurment Unit

An IMU is an electronic device consisting of multiple different sensors that can track the movements of an object. An IMU usually consists of three types of sensors: [19]

- An accelerometer is a sensor that measures the acceleration of an object in its own local three-axis cartesian coordinate system.
- A magnetometer is a sensor that measures the local cartesian coordinate's magnetic field. It is essential in telling how the orientation of the application has changed relative to the earth's magnetic field.
- A Gyroscope measures rotation and rotational rate around the same local three-axis cartesian coordinate system as the accelerometer.

Using an IMU any movement or change in orientation of the fire extinguisher could be detected, meaning total monitoring and protection against tampering. IMUs are however sensitive to noise and bias errors which accumulate while integrating data from the gyroscope. This causes what is called positional drift, where the estimated position of the IMU changes over time, without movement. This means that while IMUs can provide very accurate measurements, the computational power needed for compensation and processing is considerable [19].

2.4.2 Switches

Another way to detect the movement of the fire extinguisher is to mount a simple switch circuit to its mount. The circuit would either be open or closed when the extinguisher is in the mount which would symbolize a logical 1 or that everything is fine with the extinguisher at the moment, but if the extinguisher was to be moved then the circuit would either open or close which resulting in a logical 0 symbolizing that the extinguisher has been tampered with [20].

A simple switch circuit for this would be to use a so-called reed switch. Like a standard switch, they consist of two electrical contacts that are either connected or disconnected from one another. In reed switches the contacts are connected when a permanent magnet is nearby, closing the circuit, but when a permanent magnet isn't present then the circuit is open as the contacts no longer are connected [20].

In Figure 2.11 this basic function is shown:

Figure 2.11: Basic functionality of a reed switch

2.5 Force sensors

A strain gauge is a sensor whose resistance varies according to the the variance of strain. Strain is how a material is deformed as a result of applied stress. The stress on a material is the result of force applied to it divided by the area of the material's cross-section. It is known that the resistance of a wire can be calculated with the formula

$$R = \frac{\rho * L}{A}$$

where ρ is the resistivity of the wire, L is the length and A is the cross-sectional area [9]. So as the length of wires in the strain gauge varies depending on the type of pressure it is exposed to, so does the resistance of the wires in the strain gauge which can be measured. In Figure 2.12 the cross-section of a strain gauge is shown as also how the resistance will vary depending on the deformation [10].

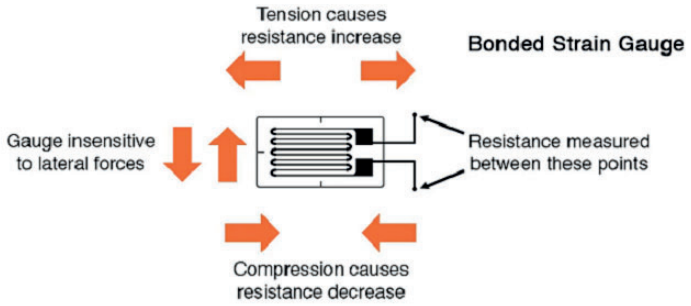


Figure 2.12: Cross section of a Strain Gauge [10].

2.6 Communication

Communication in some form is a critical part of any IoT system and enabling it to do so effectively is a challenging part of the system design. IoT devices are often needed in remote or inaccessible locations, which places restrictions on a stable power supply and wired communication. Therefore, the devices are usually battery-powered and communicate wirelessly. The environment in

which the devices are placed can introduce harsh conditions for radio signals, which means that the performance focus of the communication link usually lies in robustness and range rather than speed and latency. However, with battery-powered IoT devices, the primary focus is shifted toward not only robustness and range, but also power efficiency. The devices need to be able to operate for long periods of time for them to be flexible and low-maintenance enough to be a viable technology.

There are many ways to influence the robustness and range of a communication link, yet, transmission power will always be the main contributor. In cases where there are tough power restrictions, for example in a battery-powered IoT device, increasing transmission power is not an option, therefore other methods need to be applied. Some alternatives are to change the modulation scheme, usage of interleaving and coding, and transmit on different frequencies, all of which can reduce the bit error rate (BER). A reduction of the BER will lead to fewer retransmissions which, in turn, will lower the overall energy consumption.

2.6.1 LoRa and LoRaWAN

LoRa refers to the physical layer or the wireless modulation that is used to create the communication link for long range. Wireless systems have traditionally used frequency shifting keying (FSK) modulation, where the data is transmitted by discretely altering the carrier frequency, due to its high efficiency which helps in achieving low power. LoRa however is based on chirp spread spectrum modulation, sending out sinusoidal waves whose frequencies change over time over a wide band of frequencies, which maintains the same low power characteristics as FSK modulation but results in a drastic increase in range. A single LoRa base station can cover entire cities or hundreds of square kilometers in ideal cases [26, 8].

LoRaWAN is then the definition of the communication protocol and system architecture for the network which together with LoRa creates a long-range communication link. LoRaWAN is the portion of the communication link that has the most influence on a node's battery life, the capacity of the network as well as security. In Figure 2.13 the structure of LoRaWAN based on the OSI Model is shown.

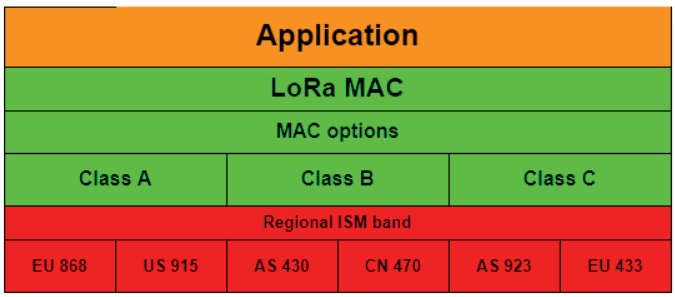


Figure 2.13: LoRaWAN in reference to OSI Model. Inspired by [27]

Unlike other networks, LoRaWAN networks are not associated with a specific gateway, but instead, the data from a specific node will be received and forwarded by each gateway to the network server. The server will handle all complex actions like filtering redundant packets, performing security checks, and so on. The nodes in LoRaWAN work asynchronously and send data whenever there is data to send, either at scheduled moments or as a result of an event. This is called the aloha method and it saves a tremendous amount of power compared to normal networks that have to wake up to synchronize with the network frequently [26, 8]. The gateways within a LoRaWAN network need to have a very high capacity for handling messages as they will receive and forward all messages they receive. This is achieved by utilizing adaptive data rates and a multichannel muli-modem transceiver in the gateway so that simultaneous messages on multiple channels can be received [26, 8].

To better handle different requirements for different types of nodes LoRaWAN has multiple different device classes. The different classes offer improvements within network downlink latency or battery life, at the cost of the other. The three different classes are [26, 8]:

- Class A allows bi-directions communications and has the lowest power consumption.
- Class B allows bi-directional communications, but with extra receive windows compared to class A devices at scheduled times. It is less energy efficient than class A but has a latency-controlled downlink.
- Class C is the class with the maximal amount of receive slots, with almost continuously open receive windows that are only closed when the device is transmitting. These devices have no latency but are the least energy efficient.

.

2.6.2 Wifi

Wi-Fi is a wireless networking technology that allows devices such as computers, mobile devices, and printers to communicate with one another [28, 29].

The technical aspect of Wi-Fi is governed by the IEEE standard 802.11, which defines the protocols that enable communications between wi-fi-enabled devices. It also defines which different bands of communications will be carried out. Communication over Wi-Fi is mainly carried out by radio waves on the two bands 2.4 GHz and 5 GHz, and these frequency bands are split further into multiple so-called channels to prevent interference and congestion. When data is to be transmitted over these channels there are multiple steps along the way. First, the data that shall be transmitted or received is transformed into binary. This binary representation is then modulated into the wave frequencies specified by the Wi-Fi embedded in the device. This wave travels between different levels of the global Wi-fi network until reaches the router of the receiving device, where the data is returned to a state that can be understood by humans [28, 29].

Wi-fi is very efficient when it comes to data throughput as it supported data rates of 9.6 Gbps in 2019, with a range of about 100 meters if there are no

obstacles. It is however not very energy-efficient. [28, 29, 30, 31]

2.6.3 Bluetooth

Bluetooth is a wireless technology standard designed to enable wireless communications between devices over short distances. It was developed in the 1990s in Sweden to eliminate the need for wires when transmitting data between mobile devices [32].

Bluetooth uses radio waves to transmit data between devices directly. It operates on 79 different frequencies around the 2.4 GHz band, which is one of the bands that Wi-Fi uses. To avoid interference with Wi-Fi Bluetooth transmits data with much less power than Wi-Fi. This also means that Bluetooth communications are generally far more energy-efficient than Wi-Fi. The low transmission power also means that the range of Bluetooth is relatively small, around 9 meters during normal conditions. When data shall be sent using Bluetooth the two devices pair with one another, during which a random frequency out of the 79 is chosen for the connection. The devices will also hop between the different frequencies if needed to maintain transmission, but only as long as they are within the range of one another [32, 33].

2.6.4 Narrowband IoT

Narrow-Band Internet of Things (NB-IoT) is a LpWAN technology proposed by 3GPP for data acquisition and transmission by low-power intelligent devices. NB-IoT supports massive connections, low power consumption, wide coverage, and bidirectional communications [34, 8].

NB-IoT has been in development since 2005 by 3GPP when the research for machine-type communication service (also known as Machine to Machine communication) on cellular networks was started. In 2017 the first standard for NB-IoT was released in collaboration with Vodafone and Neul based on LTE technology introduced by Nokia, Ericsson, and Intel. Due to this, it can coexist with 2G-4G networks and is supported by all major mobile equipment as it inhabits its own narrow band [34, 8].

The main features of NB-IoT are low power consumption, high connection density, and low costs. To achieve low power consumption NB-IoT uses the following techniques; power saving mode (PSM) and expanded discontinuous reception (eDRX). PSM means that the terminal is still registered, but not available for signaling. eDRX on the other is a method of extending the sleep device's sleep cycle, extending its time in low power mode which reduces the power consumption. According to a technical report by 3GPP the service life of a device powered by a 5 Wh battery is up to 12.8 years when a message of 200 bytes is sent once daily using NB-IoT [34, 8].

To achieve high coverage, up to 15 km, NB-IoT utilizes mechanisms like retransmissions and low-frequency modulation. There are also 3 different modes in which an NB-IoT node can operate; Stand-alone mode, which utilizes a frequency outside the LTE band. Guard-band deployment where uses the edge frequency band of LTE. Finally, there is In-band deployment, which uses a frequency band in LTE and takes up a section of its resources [34, 8].

The NB-IoT network consists of 5 parts [34]:

- NB-IoT end device, most IoT devices have access to an NB-IoT network, as long as the corresponding SIM card is installed.
- NB-IoT base station, usually an already deployed base station from a telecom company is used.
- NB-IoT core network. This is the connection point between the NB-IoT cloud platform and the base station.
- NB-IoT cloud platform. The NB-IoT cloud platform can process the data and then either forward the results to the vertical business center or the end node.
- Vertical business center. It can receive data from an end device and take control of it if need be.

2.6.5 LTE CAT-M1/LTE-M

LTE CAT-M1 is one of two LPWA technologies presented by 3GPP in 4G LTE, the other is NB-IoT. LTE CAT-M1 was developed with the objectives of supporting low-cost devices, large-scale deployment, 10-year battery life, and latency of at most 10 seconds. LTE CAT-M1 can also operate within any LTE-system bandwidth, which reduces costs greatly since it can utilize existing hardware for cellular communications. The only addition necessary is that the device is provided with a SIM card to enable transmissions [35, 37].

To achieve a 10-year battery life at a minimum, two performance modes have been added, LTE eDRX and LTE PSM. LTE eDRX is a technique that allows the UE (User equipment) to set how long the period between synchronizations is to maximize the time spent in sleep mode. LTE PSM, or LTE power saving mode, is a mode that the UE can enter where it will be seen as dormant within the network. So when the UE enters PSM it signals that it will be dormant, before it enters sleep mode. IF then the UE has data to send it will wake, either driven by an event from the host device or due to a timer, and then transmit the data. The UE will then remain awake in RX mode for four frames in case it is needed. Together these two techniques can help a smart metering solution communicating by LTE CAT-M1 achieve a battery life of more than ten years off of just two AA batteries [37].

2.7 Energy efficient systems

How to design and construct energy-efficient systems is a highly debated topic, suggestions vary greatly depending if it comes from a software perspective or a hardware perspective, and to achieve the best of both worlds is a science in itself. IoT systems put tough restrictions on the energy-efficiency and the restrictions keep getting tougher, while the applications get more complex.

2.7.1 Software

There are many ways to make software more energy efficient, but generally, it boils down to choosing a programming language suitable for the task, specializing the software to the task as much as possible, and then making the software as fast as possible [21, 8].

With the software being very task-specific, the development can be focused on a single or a few cases which will create more time for optimization, however, sacrifices in customizability and adaptability must be made. By reducing the adaptability, the software needs fewer checks and adjustments for the data it is handling as well as needing fewer instructions overall [21].

Making the software as fast as possible is all about reducing the number of instructions needed to achieve the end result. A compiled programming language, like C for example, is by orders of magnitude faster than an interpreted one, like Python for example. This is because a compiled language is compiled beforehand into machine code and does not need to be translated to machine code during runtime. Another aspect of the programming language is how near it is to machine code, i.e. how high/low abstraction level it has. For example, an object-oriented language (higher abstraction level) needs to, as an extra step, translate its objects into machine code which will result in more machine code and, in turn, more instructions [21].

2.7.2 Processors

The processor, or the central processing unit (CPU) in a computer, is the component that interprets and executes program instructions as well as performs any arithmetic operations for the IoT device. They are essential for all IoT devices, but there are different types of processors suitable for different tasks. The processor needs to be powerful enough to be able to handle any operations necessary for its application, but since the application will be battery-driven energy efficiency is important as well.

Bit size refers to the maximum size of information that a microcontroller can carry over the data bus. This means that all memory addresses, instructions, variables, or registers can at most be of the same size as the bit size. The standard bit sizes are 8 bits, 16 bits, and 32 bits. So, it is necessary to choose a microcontroller with a large enough bit size, but too large will just result in unnecessary energy consumption [22, 23, 24].

The available I/0 also matters as necessary peripherals must be capable of connecting to the processor, but too many I/0 pins are detrimental to the energy consumption and form factor of the microcontroller. The amount of available memory is also critical when it comes to processors. Having sufficient memory for operations is critical, but consumes more energy than necessary [22, 23, 24].

Finally, the most important feature of creating an energy-efficient device is that the processor can be placed in so-called sleep mode. In sleep mode, energy consumption is a fraction of what the MCU requires in any other mode, increasing the longevity of the device tremendously. So, to minimize energy consumption the device should be designed to be in sleep mode as much as possible and whenever it was active it should race back to sleep [22, 23, 24, 8].

2.7.3 Batteries

A battery is any type of device that turns chemical energy into electrical energy. In every battery, there is a positive plate called the cathode and a negative plate called the anode. For the battery to function properly these electrodes have to be separated and they are therefore often immersed in an electrolyte that allows ions to pass between the two electrodes. Usually, it is the anode that gives up electrons due to the potential difference between the two electrodes, and it becomes ionized as result. The cathode, consisting of a metal oxide, then accepts the electrons from the anode, becoming less ionized in the process. For this exchange, a conducting link has to be externally provided, as well as there being sufficient electrolytes to balance the electric flow within the battery by transporting electrons and chemical matter in the opposite direction of the external flow. The material of both the electrodes and the electrolyte has to be chosen so that sufficient voltage and current are developed by the battery to power numerous different electrical devices. The material also decides what capacity a certain size of battery will have. The interested reader is referred to [25].

There are as previously stated numerous types of batteries, but some of the most common compositions have zinc, nickel, or lithium as an anode combined with carbon, silver oxide, manganese dioxide, chloride, air, iron-sulfide, or cadmium as the cathode and potassium hydroxide, zinc chloride, organic, or sulfuric acid as the electrolyte [25].

As previously stated the size of the battery and the contents of the battery determine the battery capacity given in Ah (ampere-hours). Some of the most common sizes of batteries are AA with a battery capacity of 2600 mAh, AAA with a capacity of 800 mAH, and finally, CR2032 with a capacity of 225 mAh.

Chapter 3

Possible Solutions

A few things all solutions have in common are that they all require an MCU, a radio transceiver, and a portable power source. The MCU will be responsible for processing the specific data for that solution. Then the RF transceiver will be responsible for transmitting the data over the chosen wireless protocol for monitoring. Finally, a portable power source with the necessary capacity to guarantee 10 years of battery life is needed. Each solution then has its own components which will be mentioned in the corresponding section.

3.1 Pressure sensing

Pressure sensing is the most important feature of the proposed system, as checking that the pressure is in the correct range is a basic demand for the system. Below are some different ways to monitor the pressure in the fire extinguisher.

3.1.1 Integrating an electric pressure sensor into the Manometer

Installing an electric pressure sensor in series with or instead of the manometer. This solution would be the most elegant, as you could essentially replace the analog manometer with a "smart" manometer. Current off-the-shelf manometers are analog and are mounted in a threaded pressure-test slot. A smart manometer could probably be packaged within the same or similar footprint and provide the supervisor with accurate pressure data, proof of tampering, and other status metrics. However, this would require a re-certification of all models of fire extinguishers as the certification is made as a whole with the manometer included. The idea of a smart manometer is therefore outside the scope of this report.

3.1.2 Weight Measurement using Strain Gauges

With the limitations of only being able to measure the pressure of the fire extinguisher from the outside the next most elegant solution would be to measure the

weight of the fire extinguishers to detect any possible leaks. The most energy-efficient and elegant solution found for this was to use strain gauges which have been described in Section 2.5.

Using a strain gauge to measure the change in weight, a way to accurately measure the change in resistance of the strain gauges was needed. This is especially important since the changes in resistance might of the order of Milliohms. An efficient method of measuring these Minute changes in resistance is to use a so-called Wheatstone bridge. In Figure 3.1 the basic design of a Wheatstone bridge is shown.The bridge is said to be balanced when the ratio R_1/R_3 and R_2/R_4 is equal to one another.If the bridge is balanced then the measured voltage V_{out} will be zero. This is the basic state that every Wheatstone bridge is based upon. To then measure the variance in resistance of a resistor it replaces either R_1 or R_3 while all other resistors are known. Then the difference in voltage can be measured in V_{out} which gives us the change in resistance and as a result a change in strain [11].

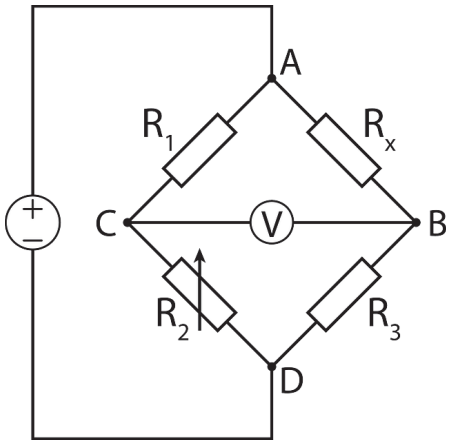


Figure 3.1: Working principle of a Wheatstone bridge [49].

This is the method to measure the change in resistance using a Wheatstone bridge, but there are some different options available for how the bridge should be constructed. The three different options that were considered for this project were the quarter-, half-, and full-bridge circuits, which are shown in Figure 3.2 where the simplest version, which has been described already, would be the quarter-bridge circuit farthest to the left. This is the simplest bridge circuit to construct which is also its strongest selling point. The problem with this simple design is that it is sensitive to external error factors such as wire resistance, temperature, and other external disturbances which may cause a change in strain on the sensors. To combat these external factors a second strain gauge can be introduced which creates a so-called half-bridge, as seen in the middle of Figure 3.2. Here the two gauges have been mounted in a way so that they will experience opposite deformations in response to the same type of strain, so one will be compressed while the other will be stretched. This configuration is both less sensitive to temperature as the ratio between the two sensors remains the same independent of the temperature, it is also more accurate than the

quarter-bridge due to the change in the ratio between the two strain gauges being that much bigger compared to the quarter-bridge. Finally, there is the third possible configuration, which is the full bridge seen on the far right in Figure 3.2. Here all resistors in the Wheatstone bridges have been replaced by strain gauges to achieve the best possible accuracy. The main drawback of both the half- and the full-bridge circuit is whether or not it can be mounted in the real world. In most applications, the space for mounting sensors on the test specimen is limited which makes mounting multiple gauges in a complementary composition a challenge. For this project, a full-bridge circuit was deemed to be unfeasible, so both the quarter- and half-bridge would be tested to see if the extra accuracy was needed or not [10].

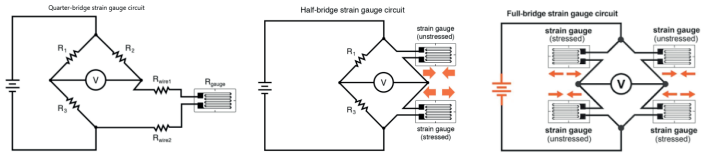


Figure 3.2: The three potential Bridge circuits [10].

The weighing construction would be built into the mount so that the weight could be checked in real-time. Then an algorithm would be developed which would find the weight by average so that disturbances such as vibrations would be minimized.

3.1.3 Local image analysis

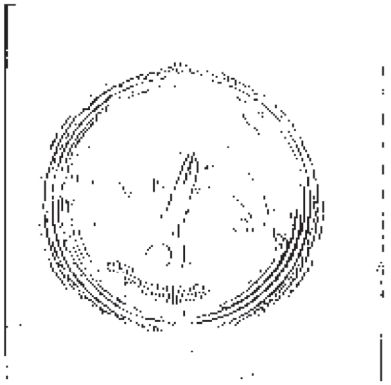
Another possible solution is to make use of a camera to monitor the pressure gauge. The idea is to mount a small camera module in front of the pressure gauge which in regular intervals will take a photo. The resulting image will then be analyzed using computer vision techniques to find the location of the pointer and determine if it has moved from its previous position.

As previously stated, this solution revolves around capturing an image of the manometer, therefore, an image sensor is needed. The sensor will need to be energy-efficient, which means, among other things, that the amount of data the sensor outputs will need to be limited. One way to do this is to choose the lowest possible resolution the image processing algorithm can handle (200×200 in this case). The images should be in grayscale and the pixel intensity values should have a limited range, however, the range will still need to be high enough such that sufficient details are kept, in this report 8-bit pixel intensities have been chosen.

The algorithm will receive the test image, seen in Figure 3.3, as an 8-bit 200×200 matrix followed by padding the matrix with zeroes to simplify the filtering process.

After that, the algorithm filters the image with a Sobel filter, obtaining an edge representation of the image, as seen in Figure 3.4. The edge representation of the image will contain interesting features and points of interest, namely edges, and because of the high contrast between most manometer backgrounds and

(a) Original reference and test image



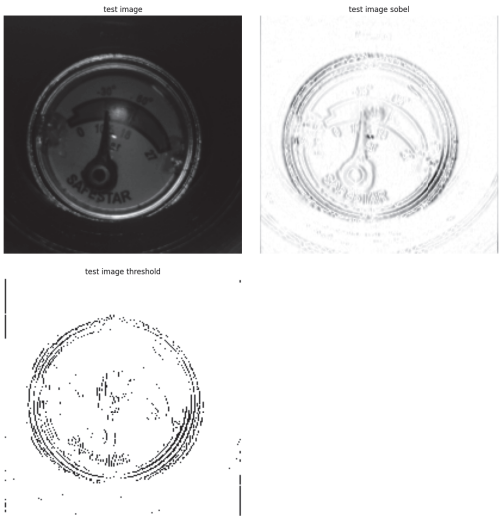
(b) Processed reference image saved in the system during installation (inverted colours)

Figure 3.3

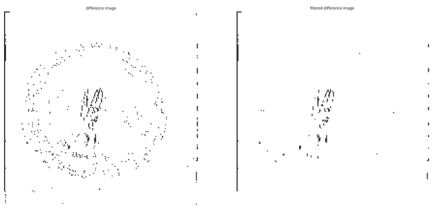
the pointer, the pointer will register as a distinct edge which will be crucial for reading the gauge.

When the edge representation has been obtained the image matrix is further processed by a thresholding algorithm converting the 8-bit pixel intensity values into binary values, as seen in Figure 3.4. The threshold for converting to a "1" or a "0" is calculated in accordance with the adaptive thresholding described in Section 2.3.2.

After thresholding, the padding is removed and the resulting image matrix is compared to the stored reference image, as seen in Figure 3.3. The comparison is done by computing a difference image, as seen in Figure 3.4, by subtracting the test image from the reference image and calculating the absolute value of each pixel. The resulting difference image normally contains some noise which



(a) Processing of the test image (inverted colours in processing steps)



(b) Difference image unfiltered (left) and filtered (right) (inverted colours)

Figure 3.4

is filtered out, the filtering involves, for each pixel, looking at its neighboring pixels and if each of those pixels has the value zero, the center pixel is also changed to zero. This filtering process will eliminate most of the "solo white pixels" generated by noise to give a clearer difference image. The program is then ready for making the decision of whether to send a fault signal or an "OK" signal.

The decision is made based on the percentage of white pixels among all pixels, the specific value has been experimentally determined to be $\approx 5\%$, but will vary depending on noise levels. If the number of white pixels exceeds this value, the algorithm decides that the pointer has been moved. A status signal will then be created by the application and sent to the application, whereby the system supervisor will be alerted in accordance.

The main challenge will be how to make it energy efficient. More specifically, reducing the amount of energy that is consumed by performing the image analysis on an image, followed by sending the results via radio to an application. It is necessary that the image processing algorithm is designed with energy efficiency in mind and also that the power consumption of the communications link is kept at a minimum.

Another challenge with this solution is the packaging, the footprint of the device should be minimized while still maintaining a sturdy casing resistant to physical impacts and vibrations. It is unclear whether the current regulations require the system to provide the possibility to read the manometer manually, if that is the case, this will become another challenge.

3.1.4 Cloud-based Image analysis

This solution is similar to Local image analysis but instead of performing the image processing locally and sending a few status information bits, the whole image is sent to a cloud application where advanced image processing can be used.

In this solution, advanced image processing can be used without any extra burden on the battery power supply of the device. However, as you need to send the whole image to the cloud, there will be more data to send which might lead to a higher power consumption overall depending on the efficiency of the communications link compared to local computation. This solution will also offer more flexibility when changing the image processing algorithm, as it is a matter of updating the software in the cloud compared to updating the software on each device.

Without the power constraints on image processing and with more computational power, more advanced algorithms can be used for feature extraction, for example, Hough transforms or more advanced edge detection. With these more advanced algorithms, the system could extract the whole pointer as well as the marked limits of the manometer to give very exact measurements of the pressure.

3.2 Tampering

The system's ability to detect tampering in different forms is an important feature. For full functionality of the fire extinguishers, they can not be obscured, displaced, or maliciously tampered with, which means that ways of detecting such events will be critical to ensure that the extinguishers can operate as intended.

The following alternatives suggest different ways of detecting that tampering has occurred, followed by alerting some kind of system administrator that a manual checkup is needed.

3.2.1 IMU

An Inertial Measurement Unit (IMU) is placed on the extinguisher, which can pick up changes in acceleration and orientation to determine whether any form of tampering is ongoing.

More specifically, normal patterns for acceleration and orientation can be experimentally obtained beforehand, and the system can compare the current acceleration and orientation against these normal values to determine whether tampering is taking place. The advantage of this solution is that very detailed information about the extinguisher's physical state can be obtained and processed, which means that it might be able to detect, for example, if someone is trying to remove the factory seal, which would void the functional guarantee. The drawbacks of this solution are that it would require a great number of experiments and testing to get it to function correctly and not to give false alarms, false alarms being, for example, a door in close proximity being slammed shut which can register as a movement in the IMU, or if the fire extinguisher is placed in a high vibration environment such as a factory. In other words, it would increase the complexity of the system, burden the MCU with more computations, and might give false alarms if it is not calibrated to its environment. However, if calibrated correctly it would produce valuable data for the system user which would give a greater insight into the state of the extinguishers.

3.2.2 Switch

A switch that can detect if the fire extinguisher is currently placed in the correct attachment.

More specifically, a magnetic switch or another low idle-power switch is installed between the attachment and the fire extinguisher, which produces a logical 0 if the extinguisher has not been moved and a logical 1 if it has been moved. The advantages of this solution are that it is inexpensive, both economically and energy-wise, and it is low in complexity while still providing system-critical data. Disadvantages of this solution are that you might need to install one part of the switch to the attachment which can introduce longer installation time and it needs to support multiple types of attachments. Another disadvantage is that you will not be able to determine the specific nature of tampering that is going on, only that it has been moved from the attachment, giving an incomplete picture of what is occurring.

Chapter 4

Comparisons and discussion

In the following section, the different solutions presented in Section 3 are compared, to determine which solutions will be explored further within this thesis.

4.1 Wireless communication

Wifi is great for high-performance communication, but its range is quite lacking for this system. As this system is aiming to send as little data as possible to preserve energy, and the fact that it is not time-critical in any way, the high-performance aspect is irrelevant. Wifi, however, is broadly available in almost every building which would decrease the upfront cost of accessories drastically. Nevertheless, with many buildings being residential, this advantage is somewhat mitigated. Wifi also does not have the required energy efficiency needed for this system, thus Wifi is discarded as an alternative.

Bluetooth is similar to Wifi with high-performance communication, but its range is even worse than that of Wifi. Bluetooth coverage is not broadly available and many accessories and access points would need to be set up to achieve good enough coverage, which would greatly increase the cost and installation time. Its energy efficiency is better than wifi, and low-energy Bluetooth (BLE) is even comparable to other LpWAN alternatives, but the minimal range and need for large investment into surrounding infrastructure means that Bluetooth is also discarded as an alternative.

Now on to the LpWAN alternatives (LoRa, NB-IoT, LTE CAT-M1), each of these alternatives is designed to achieve a long battery life and long range. Theoretically, each of the following alternatives should be able to achieve a battery life of 10 years with a range of at least 10 kilometers [40].

NB-IoT and LTE CAT-M1 both use existing LTE infrastructure, and can handle larger traffic volumes than LoRaWAN. However, LoRaWAN has been shown to be the most energy-efficient alternative for payloads lower than 150 bytes [8, 40]. This means that LoRaWAN can be much more energy efficient than the other alternatives for use cases such as local image processing. The fact that NB-IoT and LTE CAT-M1 use the existing LTE infrastructure means that they have

very high coverage both locally and globally, however, using this infrastructure has a cost, a potential system running NB-IoT or LTE CAT-M1 will have to subscribe to a cellular company to use their frequency bands, which will turn into a recurrent cost. LoRaWAN, on the other hand, is "free" to use, as in, if there exists coverage for LoRa it can be used freely as the infrastructure is usually installed by the government. The "free" coverage, however, is not yet well developed, which means that until then, the cost and execution of infrastructure installation will fall on the user. Where there is coverage LoRa would be considerably cheaper in the long run [40]. Considering the cost as well as the energy consumption, LoRa is the most viable option in this thesis.

This choice has a direct effect on which pressure-sensing techniques that are viable for further development. For cloud image analysis (Section 3.1.4) the entire image has to be sent from the device. This means that the device has to be able to send (worst case) a 200×200 image or 40,000 bytes via the chosen protocol if the image is not compressed. Callebaut, G found in his doctoral thesis that the average energy consumed per byte sent for both LoRaWAN and NB-IoT was at best 1 mJ per byte and that LoRaWAN was considerably more efficient with payload sizes smaller than 150 bytes [8]. However, if the whole image is sent, around 40 J/day would be spent at best. A normal AA battery usually has 2600 mAh of capacity (C) which translates to approximately 28.08kJ, see (4.1). In other words, the battery would be drained in only $\frac{28.08kJ}{40J \cdot 365} \approx 1.92$ years, which does not conform with the goals of this thesis

$$E = C \cdot v \cdot 3.6 = 2600 \cdot 3 \cdot 3.6 = 28.08\text{kJ}. \tag{4.1}$$

4.2 Pressure sensing

Four alternative solutions for pressure sensing have been introduced in this report (Section 3.1).

The first solution (Section 3.1.1) describes the use of an electrical pressure sensor in series with or instead of the manometer. Although this is arguably the best solution overall, with the lowest power consumption, highest reading accuracy, simplest installation, and most cost-effective, it is an invasive solution that should be preferred in newly produced fire extinguishers rather than those currently in circulation. This being an invasive solution also reaches outside of the scope of this thesis and the established boundaries. Thus, this solution is discarded as an alternative in the proposed system.

The second solution (Section 3.1.2) describes the use of strain gauges to accurately measure the weight of the whole fire extinguisher. This solution has several advantages, it could solve both pressure sensing and tamper detection simultaneously, strain gauges are relatively cost-effective, and the computational power needed is minimal which would increase the system's power efficiency. However, the sensitivity combined with the capacity needed in the weight measurement is tough to implement, making it slightly unrealistic for this system.

For example, according to Cebon in a 12 kg powder extinguisher nitrogen is used as the propellant to spread the extinguishing agent. In a 12 kg extinguisher, the

pressure would be at 12 bar or 13 atm from the factory. The volume of the extinguisher is 10.6 liters with 1.6 liters of free volume. Using the density of nitrogen, which is 1.2506kg/m^3, the total amount of nitrogen in the extinguisher can be calculated to be 26 g, see 4.2

$$N_{weight} = \rho \cdot pressure \cdot V_{free} = 1.2506 \cdot 13 \cdot 0.0016 = 26\,\text{g}. \tag{4.2}$$

Due to there only being 26 grams of nitrogen that can leak from this type of fire extinguisher, it creates high demands on the resolution of the weight measurement. If for example, the pressure were to drop by 0.5 bar due to leakage, the amount of nitrogen that had leaked would only be 1 g, see (4.3)

$$N_{old} - N_{new} = 26 - (1.2506 \cdot 12.5 \cdot 0.0016) = 26 - 25 = 1\,\text{g}. \tag{4.3}$$

This means that for a pressure drop of 0.5bar, which is a much larger drop than what would be accepted in this system, the resulting mass change would only be around 1g of propellant gas. This, in turn, means that the system would have to be able to measure a mass change of one 12000th which corresponds to a sensitivity of $\frac{1\,\text{g}}{12\,\text{kg}} \approx 0.000083\text{g} = 0.083\text{mg}$. If this result is compared to top-of-the-line laboratory scales at for example vendor [39], it is an order of magnitude smaller. If this is combined with the fact that it needs to measure more than 12 kg in total mass, while the laboratory scales are capable of measuring in the order of 500 g, this solution becomes slightly unrealistic to implement. Another problem is that gas leakage usually happens gradually, which means that worst case, it would need to measure this small mass change over a 10-year period. Another aspect is the temperature variations, especially for resistive measurements which are heavily temperature dependent. There is also the problem where scales with this kind of precision have to be calibrated at regular intervals, especially if the scales are used daily and placed in environments with multiple sources of interference, for example, vibrations and dust.

Taking all of these challenges into consideration, the weighing solution is unfeasible to implement for pressure sensing.

The third solution (Section 3.1.3) describes using a camera mounted on the manometer, which can read the gauge through on-chip image processing. The advantages of this solution are that it is quite easily realized, it could be made cheaply, and it is resistant to vibrations and other environmental aspects. It does not need to be certified as a pressure-withstanding part and, as such, is suitable as an add-on product. It is however more energy-demanding than the previously described solutions because of the higher processing demands, which means that the system needs to be highly optimized. The system would need a microprocessor capable of storing the whole image in memory and processing it in a reasonable time, which would exclude the most energy-efficient MCU:s available. The high requirements for optimization give rise to a longer development time and require many testing hours. The solution offers flexibility as most of the functionality is contained in the software, however, this flexibility remains untapped if an energy-efficient and effective way of updating the software is not developed alongside it. With the higher energy consumption, this

solution will also require a larger energy source, which will bring a bulkier mechanical design. The information sent to the application can be quite minimal in size as a few status bits would be enough to convey the status of the system. The mechanical fastening of the camera needs to ensure that the rotation of the camera always remains the same, as it otherwise would require a new reference image, which in turn, would disrupt the validity of the system's status reports.

The fourth solution (Section 3.1.4) describes, similar to the third solution, the use of a camera but with the image processing moved to a cloud application. With the image processing moved off-chip, the whole image needs to be sent wirelessly which is a large cost energy-wise. An advantage of this solution is that very advanced image processing could be used, as the cloud would have "infinite" resources compared to processing on-chip. Another advantage is that the MCU does not need to be able to process the image, which means that a more energy-efficient and slower MCU could be used. If the image is compressed before sending, which might be needed from an energy perspective, the latter advantage will be lost. With this solution, the possibility of convenient manual inspection of the images is possible, as the images are already stored in the application, this combined with the ability to directly compare images over time will give a very detailed picture of the connected systems. However, as previously discussed in Section 4.1, the energy needed to transmit the entire image daily is too large to achieve 10 years of battery life.

The third and fourth solutions are feasible to realize and are the most promising solutions for pressure monitoring with the constraints of this report. They have different approaches as to where the energy consumption is focused, the third solution is consuming the most energy by increased processing of the data, while the fourth solution transfers the processing onto a cloud platform and consumes the most energy by sending more data. However, the cloud-based solution would not achieve a battery life of 10 years if an uncompressed image is sent daily. An argument could be made for compressing the image and sending it at longer time intervals, but ultimately this solution is discarded as an alternative in this thesis. In other words, this report is only further investigating the third solution.

4.3 Tampering

In Sections 3.2.1 and 3.2.2 different solutions for monitoring against tampering were proposed. The pros and cons of both solutions were also discussed and after taking all this into consideration, it was decided that the solution using the switches would suit the system's needs.

The main reasons for deciding against an IMU were the drift in measurement values and the energy budget. Without routine calibration of the IMU, its measurement values can start to drift quite severely, leading to rather inaccurate readings. This could be amended by storing initial known values offline and using these to verify the offset. The most significant problem with using an IMU for tampering was the energy budget. Since the system needs to have a battery life of at least 10 years it is not possible to keep the IMU on at all times. It needs to be powered off almost always and only wakes to take a measurement. This implementation removes some of the biggest strengths in using an IMU

since we cannot take advantage of any of its real-time measurements. So the idea of using an IMU for this part of the process was put on pause and other solutions were explored.

Instead, the switch was the tampering solution that was deemed to be most suitable for implementation. While it doesn't provide the same amount of data as the IMU, the low complexity it brings to the system as well as that it can run on extremely low power is why the switch is the tampering solution we will move forward with.

Chapter 5

Experimental Setup

Throughout the entire thesis, the same experimental setup has been used, expanding it as new functionality has been added. In this chapter, a review of the system will be provided, both for the hardware and the software.

5.1 Tools and hardware

The foundation for the experimental setup has been a Raspberry Pi 4 Model B as shown in Figure 5.1. It has been used for both computation and communications and it is equipped with 8 GB ram, a quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz processor, 40 GPIO pin header, 2 USB 3.0 and 2 USB 2.0 ports, 2-lane MIPI CSI camera port and much more. While it is a lot more powerful and has a lot of unnecessary I/O, it was deemed a suitable device for proof of concept as programming would be done in both Python and C, as a quick setup for use in both languages was prioritized.

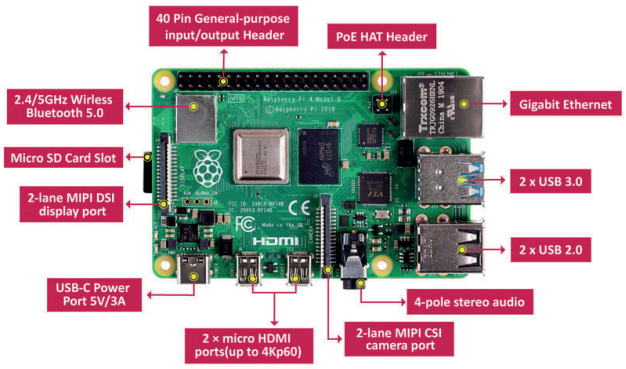


Figure 5.1: A Raspberry Pi 4 model B [41]

For image capture, the Raspberry Pi NoIR Camera V2 has been used. It is equipped with the Sony IMX219 8-megapixel sensor and there are also prewritten libraries for it, which lessens the start-up time. The camera is mounted at the end of a cylinder representing the casing of the system, in the other end of the cylinder, the manometer is mounted as can be seen in Figure 5.2. Observe that the cylinder is completely closed in the prototype, as seen in Figure 5.3.

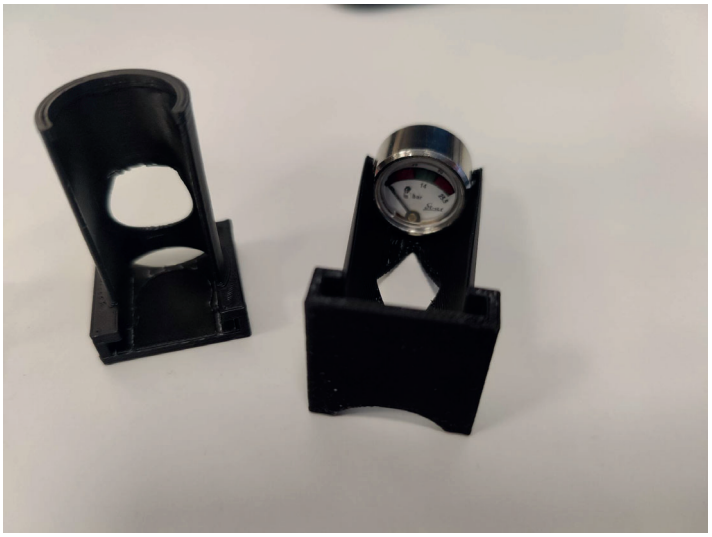


Figure 5.2: Cross section of camera mount

For communications, the Seeed Studio LoRa-E5 mini-development board has been used, as it is easy to integrate into the experimental setup due to it having a quick setup mode where it communicates over UART.

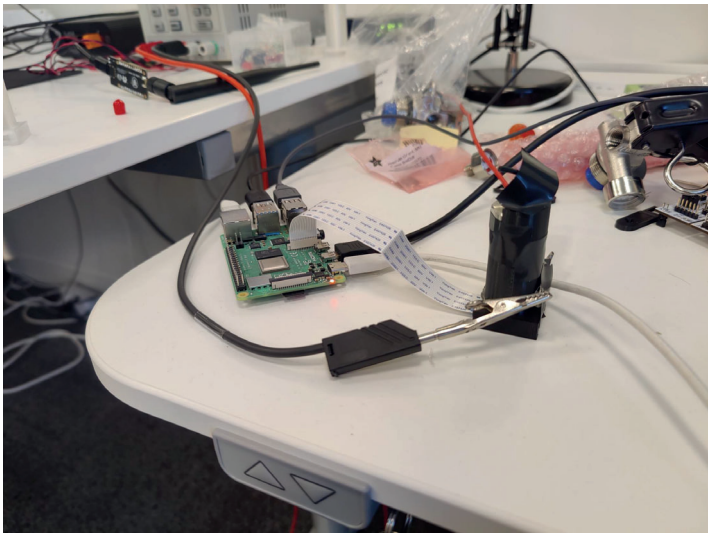


Figure 5.3: Complete prototype setup

For power measurements, the Otii Arc Pro by Qoitech has been used. It is a two-quadrant source measurement unit (SMU) with a constant voltage or constant current source and sinks together with a multichannel multimeter. It provides accurate power measurements, and together with its own software, it provides a clear visualization of the measurement. The Otii Arc Pro was provided by Mikrodust, whose lab we had access to during the entire thesis. Their 3D printers and soldering equipment among other things were also used.

When it comes to software programming was done in both Python and C and these are the libraries that were used:

- In Python, the libraries: Matplotlib, OpenCV, Numpy, and Pyserial were used.
- In C the libraries stdio.h, stdlib.h, string.h, unistd.h, time.h, stdbool.h, and errno.h were used.
- Matlab was also used to create the figures in this thesis.

For interested readers, the C-code can be found in Appendix 7.6.

5.2 Performance

Below are measurements for how the different components of the current experimental setup perform.

First off in Figure 5.4 we see how much measurements of the current consumption for the image processing algorithm on the raspberry pi. Each peak is an individual cycle of the algorithm. The averages were calculated from 10 unique runs and the results were that the average current consumption for the algorithm was 94.3mA and that one cycle took upon average 69.61ms to complete.

The voltage during the runs was 4.55V which means that one cycle consumed on average:

$$E = V \cdot I \cdot T = 4.55V \cdot 0.0943A \cdot 0.06961s = 29.8\text{mJ}. \tag{5.1}$$

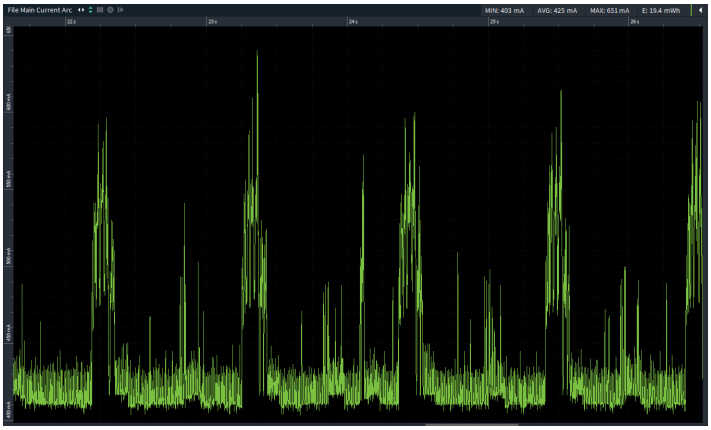


Figure 5.4: Measurment of the execution of the image-processing algorithm.

Then in Figure 5.5 a measurement of the Seeed Studio LoRa-E5 mini-development board transmitting a packet with one byte of the payload is shown. The module is configured for class A device with an adaptive scheme where it will transmit with the settings it considers optimal at that given moment. If one looks at one singular peak the different stages of transmission can be seen. First, the very small peak is when the module is in transmission mode, and it was in this mode for 49 ms on average with an average current of 93.4mA. Then for the following 3 seconds, it goes through the two receiving windows RXWIN1 and RXWIN2. During this time has an average current of 16.9mA.

This means that a message transmission lasted for a little more than 3 seconds and the energy consumption per transmission is:

$$E = V((I_{tx} \cdot T_{tx}) + (I_{rx} \cdot T_{rx})) = 4.55 \cdot ((0.0943 \cdot 0.049) + (0.0169 \cdot 3)) = 251.7\text{mJ}. \tag{5.2}$$

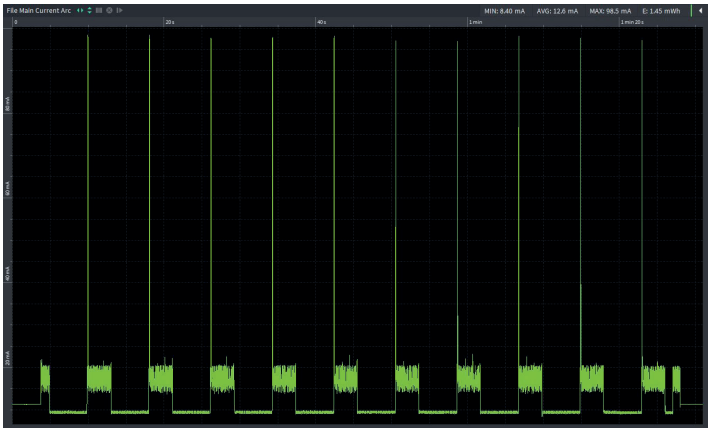


Figure 5.5: Measurment of the LoRa-module transmitting.

Finally, shown in Figure 5.6 is a measurement of the Raspberry Pi NoIR Camera V2 taking pictures. The middle bump is where 50 images are taken consecutively with a 0.2-second pause in between the capture of the images. It took 35.3 seconds to take all 50 images, which means that it took 25.2 seconds for 50 images or $25.2/50 = 0.504$ seconds per image. The average current consumption during the image capture was 134 mA. So, the energy for taking one image is:

$$E = V \cdot I \cdot T = 4.55 \cdot 0.134 \cdot 0.5 = 304.9\text{mJ}. \tag{5.3}$$

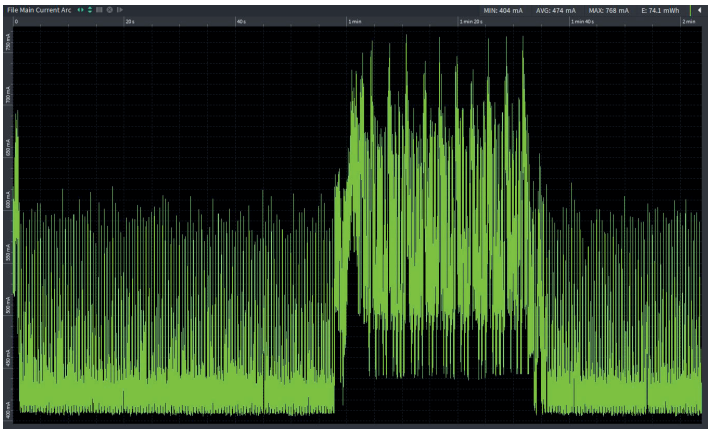


Figure 5.6: Measurment of 50 images taken consecutively by the Picamera

Together one cycle with the experimental setup would consume 304.9 + 251.7 + 29.8 = 594.5mJ = 0.5945J. This means that taking one picture per day would consume roughly 2170 J over ten years, this is however without taking into account any power consumption between cycles, since a raspberry pi does not have a low-power mode.

Chapter 6

Final system and recommendation

The final system will have the ability to monitor the pressure shown on a manometer using a camera and the image processing will be done locally, see Section 3.1.3. It will have tamper protection in the form of a magnetic reed-switch, see Section 3.2.2, which will account for tampering consisting of the fire extinguisher being removed from its attachment. The system will be able to communicate with an application, which will display the status of the connected fire extinguishers in a user-friendly interface. The communication protocol will be LoRa, see Section 4.1, which might require additional gateways in cases where the coverage is insufficient. The system will be powered by batteries and will have a battery life of 10 years.

6.1 System architecture

As can be seen in Figure 6.1, the central part of the system is the MCU, it is responsible for the image processing of the data captured by the camera, power management of the whole system, all logic and decisions, as well as preparing the data for transmission with LoRa. The entire system is battery-powered and the power management done by the MCU includes switching states and turning on/off subcircuits not currently in use, this is to conserve power.

6.2 Image processing

The image processing algorithm developed in this thesis is responsible for one of the main functions in the system, namely, pressure sensing. It was programmed in C and can be found in Appendix 7.6. As described in Section 3.1.3, the algorithm uses a pre-processed reference image to compare to a newly taken image, after the newly taken image has been processed and compared, it decides whether the pressure has changed.

The decision is made depending on whether the number of white pixels is above

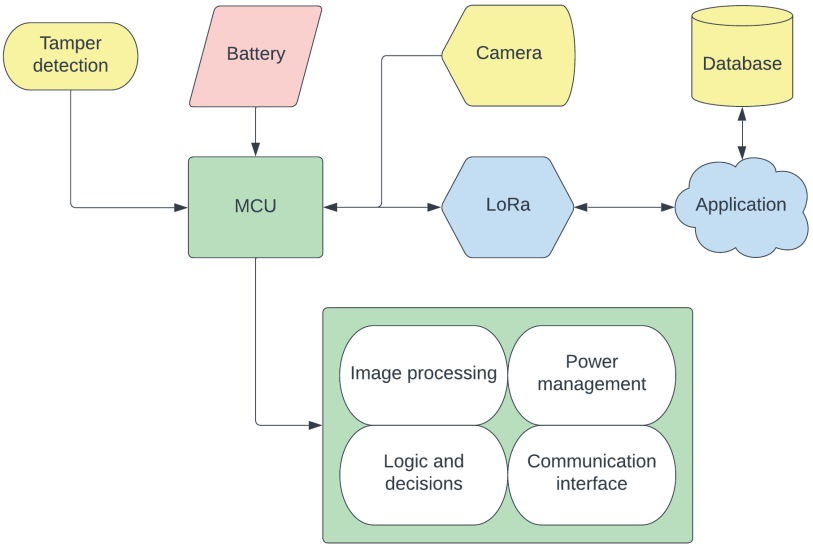


Figure 6.1: Block diagram of system architecture

a certain threshold. This threshold value has been experimentally acquired with the setup described in Section 5. Figure 6.4 shows the prediction accuracy depending on the threshold value, the two plots represent the result of comparing 20 different images using the two reference images in Figure 6.2. From Figure 6.4 it could seem that the best threshold value would lie somewhere between 0 and 270 as the accuracy is 100% in this range, however, this would be a bad conclusion. This test was made in a controlled lab environment where the noise level is roughly known, as can be seen in Figure 6.3. As the noise level oscillates around 242, choosing a threshold value below this noise value will result in the algorithm only detecting the noise, which would render the result meaningless. Thus, depending on the noise levels in the setup, the threshold value should be chosen to have a good enough margin from the noise while still achieving an acceptable accuracy. A thing to take note of in Figure 6.3 is that the noise levels are trending upward as time passes, this is probably due to the image sensor heating up. This will not become a problem in the final version of this system as the image sensor will be turned on to only acquire a single image, followed by turning the image sensor off again, thus it will not have time to heat up substantially.

The noise levels were acquired by generating 42 images, 21 for each reference image, of the same motive, taken at different time instances, followed by processing and calculating the noise level in the produced difference images. The noise level can be calculated simply by counting the number of white pixels, as the images are of the same motive the difference images should ideally contain no white pixels. The noise levels could probably be greatly reduced with better lighting conditions, the current setup uses a standard 5mm yellow led light which is not mounted in an optimal way.

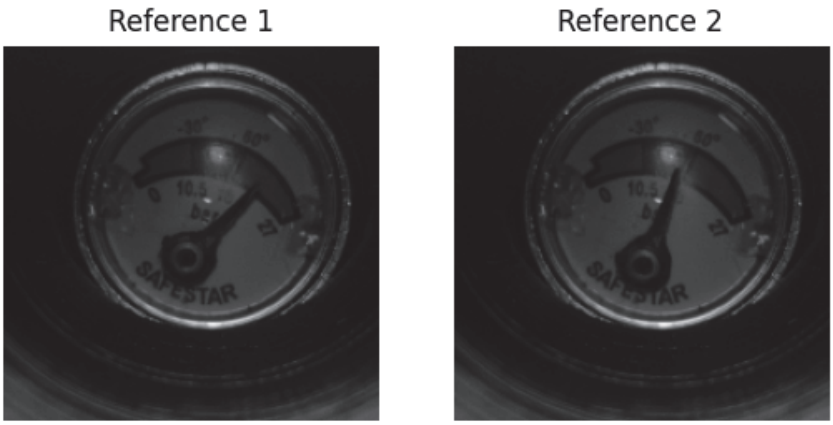


Figure 6.2: Reference images

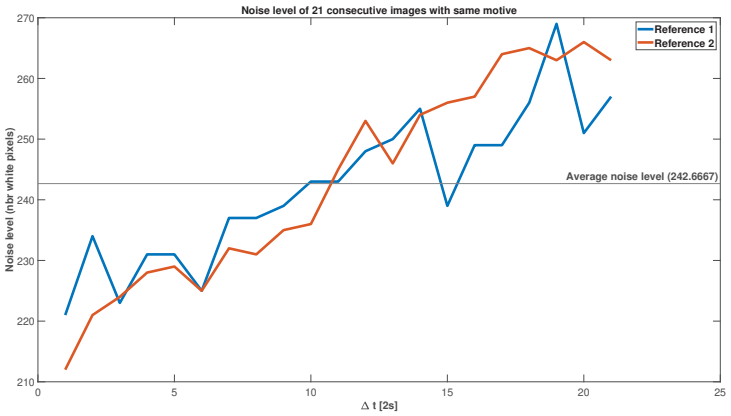


Figure 6.3: Noise level for consecutive images

In this system, with the current setup, a threshold value of 280 has been deemed suitable, as it has a good enough margin from the noise and achieves an acceptable accuracy of 0.75 and 0.85 respectively. A thing to note with a threshold value of 280 is that it is somewhat pessimistic as it is based on the average noise level for 21 consecutive images. A more realistic value would probably be around 230 as can be seen in Figure 6.3. Another thing to note in this experiment is that all (20) images from Figure 6.4 had different motives.

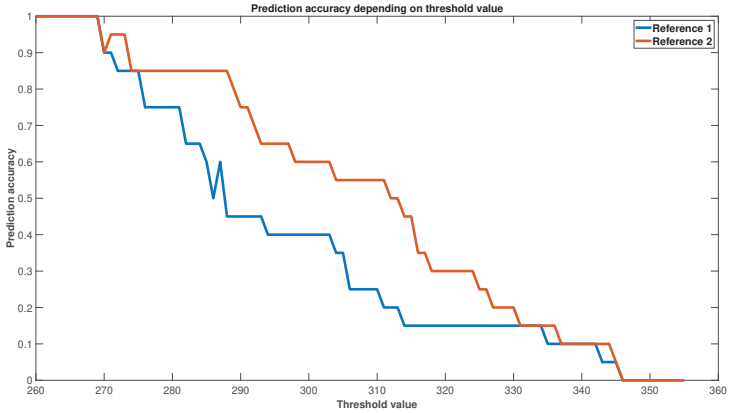


Figure 6.4: Prediction accuracy depending on threshold value

6.3 Energy and power supply

Minimizing energy consumption is one of the largest challenges in this system. The different ways to optimize are too many for all to be examined in this thesis. Therefore, the focus will be on choosing components that are as low power as possible.

There are many viable low-power MCU alternatives, ranging from sleep currents in the tens of nanoamperes to hundreds of microamperes. There are also many complete devices that have a sub-GHz radio transceiver built into it combined with an MCU. The devices chosen for this comparison are; CC430F512x MSP430 SoC With RF Core from texas instruments, SAM R34/R35 Low Power LoRa Sub-GHz SiP from Microchip, RL78/G14 SX1261 from Renesas, and Multiprotocol LPWAN 32-bit Arm Cortex-M4 MCU from STMicroelectronics. These devices will be in focus in this section as the need for only one IC will greatly reduce the form factor. The chosen device's current consumption for; sleep mode, active mode, and transmitting mode [42, 43, 44, 45] is shown in Figures 6.5, 6.6, and 6.7. A suitable image sensor was found to be, for example, the ASX340AT by OnSemi. With this sensor, it takes approximately 20 ms to take an image including power on and power down. It also uses approximately 25 mA while active [46].

The program for performing the image analysis in the local image processing solution (Section 3.1.3) consists of approximately 192 million instructions when built on an ARM64 platform (Raspberry Pi 4). The total amount of clock cycles summarizes approximately 107 million, the number of cycles is lower than the number of instructions because of parallelization in the CPU. This number is to be taken with a grain of salt as it can differ significantly depending on program implementation and platform-specific parameters. This calculation does not take into account the added instructions for loading the image from the image sensor, instructions for setting the MCU in different operational modes, or the added instructions for transmitting the status bits. With the number of clock cycles taken for the program to execute, a rough approximation of the runtime

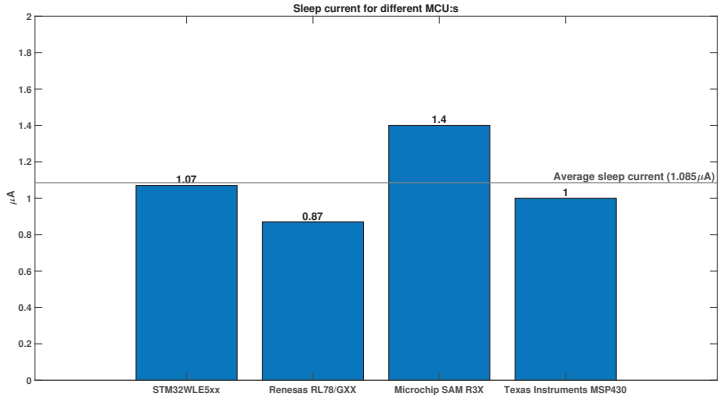


Figure 6.5: Sleep current for different MCU:s

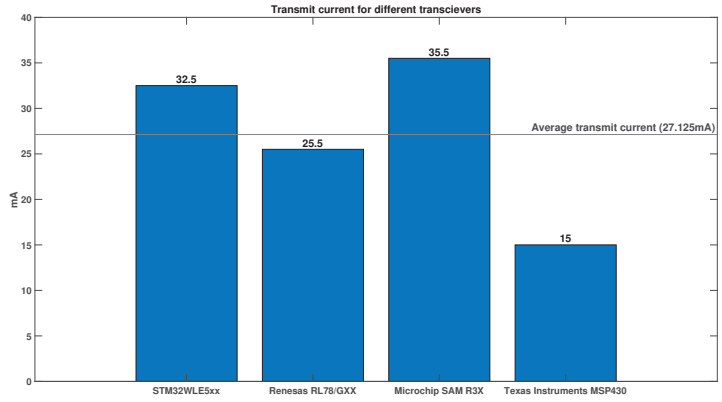


Figure 6.6: Transmit current for different LoRa transcievers

depending on clock frequency can be estimated (Figure 6.8). With a core clock of 1.6GHz the theoretical runtime becomes approximately 66.8ms which can be seen in Figure 6.8, this correlates well with experimental results which produced an average runtime of 69.6ms on the same platform.

However, in the final version of this system, a Raspberry Pi far exceeds the computational power needed as well as having a very large footprint, thus the MCU alternatives discussed in Figure 6.7 are better suited. These MCU:s operate with a frequency of around $40\,\text{MHz} - 50\,\text{MHz}$, which will produce an approximate runtime of 2.1356s, this will vary depending on internal MCU architecture, and other platform-specific parameters, but it is a rough estimation.

It is estimated that one wake-up cycle takes approximately 5229 ms as can be seen in Figure 6.1 (t_{cycle}), where the individual runtimes for each operation are listed in Figure 6.9. Also listed in the Figure are typical current consump-

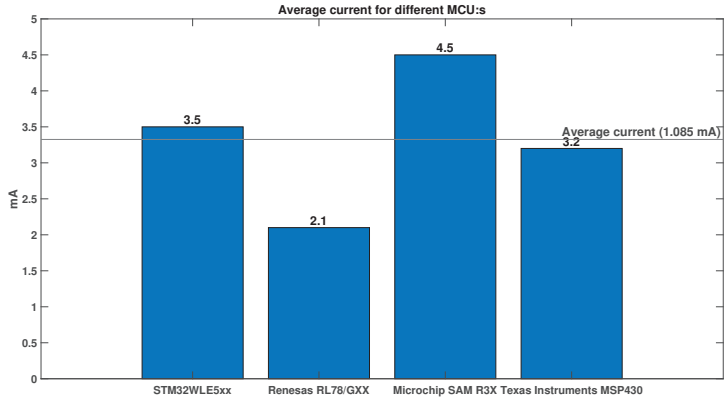


Figure 6.7: Average current for different MCU:s

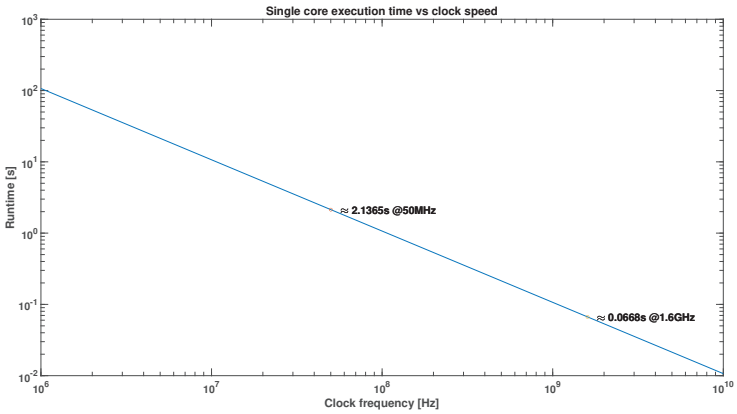


Figure 6.8: Single core runtime for different core frequencies

tions during the different operations. There are some expected losses in the system, especially battery inefficiencies, temperature variations, as well as regulator losses, whereby an efficiency factor $\eta = 0.65$ is used. Each of the values in Figure 6.9 has been acquired both through experiments as well as from the specific component's data sheets. The values are implying normal operating conditions, with a stable power supply of 3.0V and operating at room temperature (25°C).

$$t_{cycle} = t_{tx} + t_{rx} + t_{process} + t_{image} = 49 + 3000 + 20 + 2160 = 5229\,\text{ms}. \quad (6.1)$$

In (6.2), the total energy consumption for one wake-up cycle is calculated using the values from Figure 6.9. The calculation

$$E_{cycle} = V(2-\eta)(I_{MCU} \cdot t_{tot} + I_{tx} \cdot t_{tx} + I_{rx} \cdot t_{rx} + I_{image} \cdot t_{image}) \approx 0.153\text{J} \quad (6.2)$$

Time taken by the operations:

- $t_{tx} = 49\,\text{ms}$ (transmit time)
- $t_{rx} = 3\,\text{s}$ (time in receive state)
- $t_{image} = 20\,\text{ms}$ (time to capture an image)
- $t_{process} = 2.16\text{s}$ (execution time for the image processing algorithm)

Current during operations:

- $I_{tx} = 27\,\text{mA}$ (average transmit current)
- $I_{rx} = 20\,\text{mA}$ (average receive current)
- $I_{image} = 25\,\text{mA}$ (average current of capturing an image)
- $I_{MCU} = 3.325\text{mA}$ (average current draw of MCU in active mode)
- $I_{sleep} = 1.085\mu\text{A}$ (average current draw of MCU in sleep mode)
- $V = 3.0\text{V}$ (Voltage provided to the system by a voltage regulator)
- $\eta = 0.65$ (Efficiency factor)

Figure 6.9: Estimated computation times and current draw

shows that the total energy for one wake-up cycle amounts to 153 mJ, which would be considered a pessimistic result due to the low efficiency η

In (6.3), the total sleep mode energy consumption is calculated using the values from Figure 6.9. The calculation calculates the energy required for the full operational time of the system (10 years), with the active time subtracted. The total energy is calculated to be 1385.7J, which could be considered an optimistic result due to the low sleep current of 1.085μA, but investigations have shown that there exists several MCU:s that can approach this number and even go below it, as can be seen in Figure 6.5

$$E_{sleep} = V(2-\eta)(I_{sleep}((3600 \cdot 24 - 5.229) \cdot 365 \cdot 10)) \approx 1385.7\text{J}. \tag{6.3}$$

In (6.4), the total amount of energy consumed during the 10-year lifetime of the system is calculated. The total amount of energy amounts to 1944,1 J, which also could be considered quite optimistic

$$E_{tot} = E_{sleep} + E_{cycle} \cdot 365 \cdot 10 \approx 1944.1\text{J}. \tag{6.4}$$

As described in Section 2.7.3 there are multiple viable options when it comes to batteries, but a lithium-ion battery will most likely be used as they are produced in a wide variety of shapes and forms, which will increase the flexibility in a future design. As the device will have to be mounted on the fire extinguisher the size should be kept as small as possible. It should at most be of the same size as a standard AA battery cell, but smaller if possible to reduce the footprint. This all depends on what the power consumption of the final system will be, as the device needs a lifespan of at least ten years. If the result in (6.4) is used, the battery capacity needed can be calculated as in (6.5) (180 mAh), which,

for reference, approximately corresponds to a standard button cell battery. It would, however, be rather unwise to choose a battery with the exact capacity needed, as the power consumption will fluctuate with varying temperatures and conditions, as well as the fact that smaller batteries (especially button cells) are not very resistant to rapid current peaks, which are likely to occur when switching power modes

$$C = \frac{E_{tot}}{3600s \cdot V} = \frac{1944.1\text{J}}{3600s \cdot 3\,\text{V}} \approx 0.18\text{Ah} = 180\,\text{mAh}. \tag{6.5}$$

6.4 Application

The application can be implemented in a number of ways with many different technologies, in this thesis, a simple react app was developed to demonstrate a bare-bones interface that would convey the general idea behind the application. A sample of the application can be seen in Figure 6.10. The important features of the application are; registering a fire extinguisher with a monitor system connected, generating a unique id for each extinguisher as well as adding a location code. Another important feature is that the application needs to be able to communicate with each monitor system to acquire data about the respective extinguisher, including battery status, tamper signal, pressure fault signal, and any other fault signals. The application should also keep track of time and send a wakeup signal to each connected device once every 24 hours, this signal would wake the devices from their low-power mode and make them go through one cycle of the program, meaning, taking a picture and running it through the algorithm, followed by responding with the appropriate status message.

The application should also be connected to a database where the status of the connected devices would be saved, it would also save historical data to enable statistics and analytics. The interface should be intuitive and easy to use and should also include different views depending on the user's privileges.

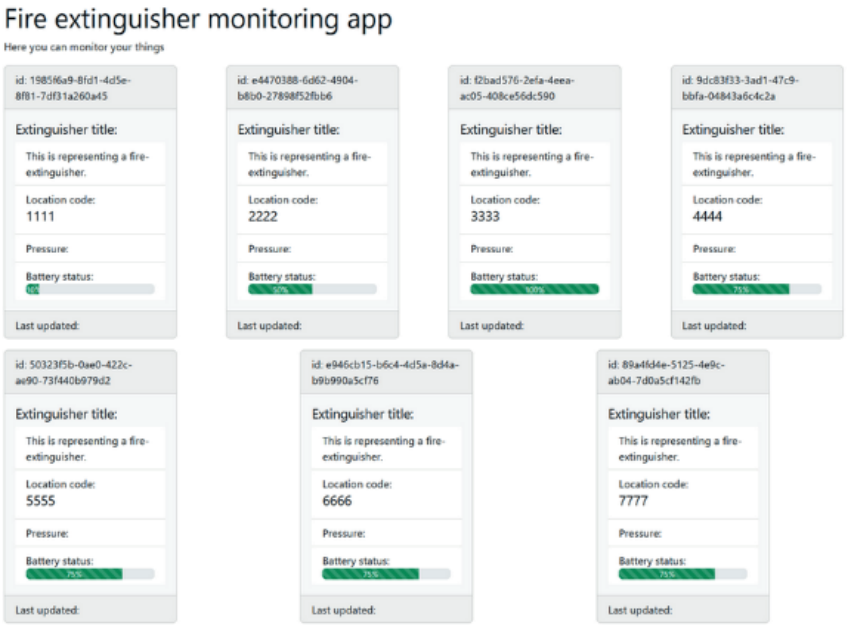


Figure 6.10: Prototype interface in application

6.5 Cost evaluation

The cost of a final system is difficult to accurately determine as it is highly dependent on how and where the product will be produced as well as which exact parts are chosen. It is also highly dependent on production volume, component sourcing, and what deals have been made with PCB production companies. However, an approximate estimation can be made from off-the-shelf components, which is presented in Table 6.1, the higher value in the table intervals is an estimated prototyping material cost for a single unit.

Component	price
MCU	$ 2-20
Batteries	$ 1-10
Camera	$ 5-80
Reed switch	$ 1-10
PCB	$ 0.1-5
Casing and packaging	$ 2-40
Total:	$ 11-165

Table 6.1: Estimated system cost

Chapter 7

Conclusions and Future Work

In this thesis, solutions for an energy-efficient monitoring system for fire extinguishers have been explored. The various solutions have been thoroughly investigated, which have boiled down to a prototype and a recommendation for the final system. The prototype as well as the recommended system uses a camera solution with onboard image processing for pressure monitoring. It uses a reed switch for tamper protection and communicates via a LoRa transceiver with an application. The recommended system is battery-powered and energy efficient enough to last for 10 years on a single standard button-cell battery (180 mAh).

The system has been successfully designed with the constraint of it being non-invasive, as no measurements are taken inside the canister which could result in leakage. The goal of having a maintenance-free run-time of 10 years has also somewhat been achieved. However, implementing a system prototype fulfilling both of these goals has been unsuccessful. The prototype became more of a proof of concept for the pressure monitoring solution. The reason for this is because of time constraints, which meant that the primary focus became the completion of a design plan for a realizable system.

The energy calculated in Section 6.3 is somewhat unreliable as it relies on typical values from datasheets. Another prototype based on one of the proposed MCU:s should be made to properly evaluate the energy consumption. However, an efficiency of 60 % was used in the calculations which should, to some degree, reduce the gap between the theoretical and real energy consumption. Experimental values obtained from the prototype (excluding sleep current) summed up to roughly 2170 J, compared to the theoretical value of 1944.1J (including sleep current). This difference in energy can probably be, in part, attributed to the Raspberry Pi running an entire OS and not being very application specific. However, it also shows that it is difficult to give a realistic approximation of the final energy consumption.

The image processing algorithm for monitoring the pressure has performed well

in controlled lab environments, but it has yet to be tested in more realistic scenarios with fire extinguishers from different manufacturers. As discussed in Section 6.2, an accuracy of 0.75-0.8 can be steadily achieved with the current setup. The accuracy could be further increased if the noise floor is lowered, which should be easily achievable as the current setup is made out of 3d-printed plastic with low mechanical accuracy.

The reed switch has also worked well in lab environments, but this is not very representative as the main challenge of this solution lies in designing the mechanical construction. This solution is also somewhat lacking as a tamper monitor as it can only detect displacement of the fire extinguisher, and is not able to detect tampering in the form of impeding the functionality of the fire extinguisher.

For communications, a LoRa architecture seems to be the most suited for this system. It performs very well in terms of energy efficiency with small payloads, as is the case in this system, and it can also be very cost-effective where there already is good coverage. The wireless range is long which means that the amount of accessories needed is minimal even when the wireless coverage is suboptimal.

As shown in the theoretical energy calculations, a standard button-cell battery (180 mAh) seems to be sufficient as a power supply. However, this does not take into account energy losses incurred by temperature variations in the environment nor does it take into account the voltage regulator limits. The voltage regulator limits meaning the necessary supply voltage provided to the regulator to output a stable voltage within the MCU supply range. This limit will increase the battery capacity needed as the system will stop working when the voltage falls below the limit. Thus, to be on the safe side with a 4 times safety margin, a standard AAA battery ($\approx 800\,\text{mAh}$) should be more than sufficient to ensure a 10-year lifetime. A case can also be made for a custom-made battery cell, which would make the system more compact as well as make the capacity more exact.

With the above points in mind, it seems, as previously stated, that the system will be viable to implement with a battery life of 10 years. The final system also has the required functionality of monitoring the pressure and basic tamper protection.

The prototype system still needs much work to be ready for production, especially the mechanical and packaging aspects as well as the electrical design to accommodate an MCU. In the following sections, possible optimizations and improvements will be discussed.

7.1 Optimizations

Optimizations of the system will ultimately come down to different ways of saving energy, as this is currently the biggest hindrance to a complete product. As the system in this report is only a prototype running on a seriously overdimensioned platform (Raspberry Pi 4), there are many optimizations to be made.

7.1.1 Embedded software

For the embedded software, further development should focus on implementing and optimizing the power management as well as reducing the number of instructions for the image-processing part.

The power management will be MCU-specific and should dynamically turn on and off subcircuits when they are not in use, this includes the camera, LoRa transmission circuits as well as different parts of the MCU. It should also dynamically change the MCU state to always strive for a transition to the low power modes as fast as possible.

The image processing algorithm can be further optimized by reducing the amount of memory it needs as well as reducing the number of instructions needed. This might open up for even more energy efficient MCU:s to be used that have limited memory and limited instruction sets, it will also decrease execution time, which will increase time spent in the low-power sleep mode. Another optimization could be to reduce the resolution of the camera even further, which would linearly reduce the number of instructions needed.

7.1.2 Electrical

The prototype system in this report has not been developed on a custom PCB, thus there are many circuit elements in the different modules that can be removed during further development. The custom-designed circuits should include components with very low idle current and every component should be appropriately dimensioned to avoid over-dimensioning. Every component should also have a temperature rating similar to that of the fire extinguishers, as they are normally placed in a wide variety of environments. The tamper monitor switch circuitry should be made in a normally open way, meaning that the circuit would only close if the extinguisher is removed, and remain open circuit otherwise. All subcircuits except for the tamper monitor should have the ability to be turned off by the MCU when not in operation, as the system is most of the time.

The transmit power of the LoRa transceiver should be based on the channel conditions and should be calculated during installation to keep the transmit power as low as possible. The camera needs to be as energy efficient as possible, however, it must have acceptable noise levels for the specific environment. The MCU should have some form of TPM (Trusted platform) solution, which makes it harder to reverse engineer and exploit.

7.2 Application

As stated in Section 6.4, the application is currently a simple prototype created in react.

There should be proper measures taken for the security and integrity of the app as this is currently not implemented. The app should have proper access control to ensure that only people with specific privileges can change and maintain the app. There is also a need to establish an identification scheme for the connected devices to prevent impersonation attacks and ensure that only trusted devices are able to connect. Measures also need to be taken against DDOS (Denial of

service) attacks, as such attacks could be very detrimental to the users of the system.

A database should be maintained with proper redundancy in case of failure, which holds the current status as well as a status history of all connected extinguishers.

Proper channels for maintenance and support should also be included, for software updates, data collection, and troubleshooting.

The interface is currently very bare-bones and should be further developed into a more modern and intuitive interface. An admins view should also be implemented.

7.3 Mechanical envelope

There are many mechanical challenges yet to be solved, the system cannot block any functionality of the fire extinguisher but is located close to the handle.

The system needs to be shock- and vibration-resistant, as the extinguishers could be placed in harsh environments.

The camera should always have the same orientation towards the manometer, otherwise, a new reference image needs to be taken each time the camera has moved and if it is not detected the system will send false fault signals.

The system needs to be securely mounted and not easily removed, otherwise tampering would become a greater problem.

If lithium battery cells are used, proper precaution needs to be taken into designing the battery casing, as lithium batteries if damaged can react violently, this is important to ensure the safety of people in close proximity as well as for fire safety.

The mechanical casing for the other part of the system needs to be IP-classed at least to the same standard as the fire extinguishers, as they will be placed in a variety of environments with dust, moisture, heat, and cold.

7.4 Factory seal and proof of not being used

The factory seal is the manufacturer's way of ensuring the functionality of the fire extinguishers, which is why it would be a good idea to verify that the seal is still in place and intact. Doing so would flesh out the tampering monitor of the system and give a more complete picture of the current state of a fire extinguisher. If the fire extinguishers are reused, the manufacturer can no longer guarantee their functionality, which is why this could be an important addition to the system.

7.5 Accessibility Control

Accessibility control has previously been identified as non-system critical, but an area suitable for further development of the system. This should not be

neglected, as doing so, in some cases, could lead to hindering the use of fire extinguishers, which in turn, would put accessibility control on an equal level of importance as tamper detection. The accessibility control of the system has not been prioritized in this report, which is primarily because of the scope as well as this feature not being system critical. However, the following solutions should be further looked into and studied as they might become excellent additions to the system.

7.5.1 Radar

The radar solution involves using radar sensors to acquire a map of the surrounding environment, which in turn can be processed to detect any blocking structures. This can be done more or less precisely with different amounts of processing power, which in turn will require different amounts of energy. The main idea is to send out radar pulses and process the reflections, which means that a substantial amount of development time needs to go towards calibration and making sure that it will not trigger movement or temporary blockades. The amount of processing the system can do will depend on the available surplus energy, as the system still must be operational for 10 years.

7.5.2 Ultrasonic

An ultrasonic sensor could also be used to detect any objects blocking access to the fire extinguisher. The idea behind an ultrasonic sensor is the same as for a radar sensor, but the difference lies in the frequencies of the pulses. For these types of sensors, the frequency range is usually 25-50 kHz. The pros of ultrasonic sensors are that they are not very susceptible to different light conditions or other things dispersed in the air like smoke or vapor. It is however susceptible to environments where the pulses are absorbed rather than reflected [47].

7.5.3 Infrared

Infrared sensors are also a viable option for detecting any blocking objects. The difference compared to ultrasonic and radar sensors is that an IR sensor utilizes reflecting light waves to detect objects. The sensor transmits infrared light and waits for any reflecting light. The time of flight for the reflected light is then used to estimate the distance to the object. An IR sensor is capable of detecting movement while also granting a greater resolution than for example an ultrasonic sensor, it is, however, more sensitive to in-air blockage [48].

7.6 System extensions

There are multiple ways this system could incorporate other existing systems, as well as various extensions that can be added to expand the functionality. Following are a few suggestions that could be worthwhile exploring.

The system could incorporate already existing fire safety equipment such as fire alarms, smoke detectors, and evacuation lights among other things. It can then be sold as a package, giving complete control of all critical devices for

fire safety. The system could incorporate them in the application where all these devices are monitored, further simplifying maintenance and preventing unnessacary checkup rounds.

The system could be extended to measure environmental variables, for example; temperature, humidity, and air quality. This would give a more detailed view of the climate, with the possibility of detecting unhealthy work environments and possibly preventing favorable conditions for ignition. This could also be used to extend the connected device's life expectancy, by ensuring that they operate in their specified temperature and humidity ranges.

Bibliography

[1] A. Mahgoub, N. Tarrad, R. Elsherif, A. Al-Ali, L. Ismail. 2019. IoT-Based Fire Alarm System. pp. 162-166. http://dx.doi.org/10.1109/WorldS4.2019.8904001 [accessed on 2023-01-15]

[2] K. Muheden, E. Erdem, S. Vançin. 2016. Design and implementation of the mobile fire alarm system using wireless sensor networks. pp. 243-246. http://dx.doi.org/10.1109/CINTI.2016.7846411 [accessed on 2023-01-15]

[3] Hidestål, Carl and Zreik, André. 2020. Smart Environment - Early Wildfire Detection using IoT. http://lup.lub.lu.se/student-papers/record/9022505 [accessed on 2023-01-15]

[4] Bergman, Johanna and Johansson, Isabelle. 2017. The User Experience Perspective of Internet of Things Development. http://lup.lub.lu.se/student-papers/record/8919050 [accessed on 2023-01-15]

[5] Brandexperten AB. 1996. EN3 - EUROPASTANDARD FÖR BRANDSLÄCKARE, https://www.brandexperten.se/images/stories/pdf/produktblad/BSSTND_bxp.pdf [accessed on 2022-09-11]

[6] Brandexperten AB.2001. SS 3656 - Underhåll och omladdning av handbrandsläckare. https://www.brandexperten.se/images/stories/2013/2013-Servicestandard_brandslackare_bxp.pdf [accessed on 2022-09-11]

[7] MSB. Brandsläckare. https://www.msb.se/sv/amnesomraden/skydd-mot-olyckor-och-farliga-amnen/brandskydd/brandskyddsutrustning/brandslackare/ [accessed on 2022-12-15]

[8] Callebaut, G. et al. .2021. Advances in Single and Multi-Antenna Technologies for Energy-Efficient IoT. pp. 23-101. https://kuleuven.limo.libis.be/discovery/fulldisplay?docid=lirias3556334&context=SearchWebhook&vid=32KUL_KUL:Lirias&search_scope=lirias_profile&tab=LIRIAS&adaptor=SearchWebhook&lang=en [accessed on 2022-12-22]

[9] Nave, Carl. Resistance and Conductivity. 2017. http://hyperphysics.phy-astr.gsu.edu/hbase/electric/resis.html [accessed on 2022-11-27]

[10] Kuphaldt R., Tony. Lessons In Electric Circuits Volume I – DC, pages 321-328. 5th ed. 2006.https://www.allaboutcircuits.com/textbook/direct-current/chpt-9/strain-gauges/ [accessed on 2022-10-19]

[11] Kuphaldt R., Tony. Lessons In Electric Circuits Volume I – DC, pages 289-296. 5th ed. 2006.https://www.allaboutcircuits.com/textbook/direct-current/chpt-9/strain-gauges/ [accessed on 2022-10-19]

[12] Mihajlovic, Ilija. 2019. Everything You Ever Wanted To Know About Computer Vision. https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e [accessed on 2022-12-16]

[13] Ashish. 2018. Understanding Edge Detection (Sobel Operator). https://medium.datadriveninvestor.com/understanding-edge-detection-sobel-operator-2aada303b900 [accessed on 2023-01-03]

[14] Roboticsbiz. 2022. Deep Learning Vs. Traditional Computer Vision – A Comparison. https://roboticsbiz.com/deep-learning-vs-traditional-computer-vision-a-comparison/ [accessed on 2022-12-08]

[15] LoRa-Alliance. 2022. About the LoRaWAN Standards. https://lora-alliance.org/lorawan-for-developers/ [accessed on 2022-11-07]

[16] Sears-Collins, Addison. 2019. How the Sobel Operator Works. https://automaticaddison.com/how-the-sobel-operator-works/ [accessed on 2022-11-30]

[17] Dilhani, Shashika. 2021. Digital Image Thresholding Techniques. . . . https://medium.com/@shashikadilhani97/digital-image-thresholding-techniques-969b006c9e07 [accessed on 2022-12-13]

[18] Dilhani, Shashika. 2021. Digital Image Processing Filters. https://medium.com/@shashikadilhani97/digital-image-processing-filters-832ec6d18a73 [accessed on 2022-12-13]

[19] Or, Barak. 2021. What is IMU? https://towardsdatascience.com/what-is-imu-9565e55b44c [accessed on 2023-01-02]

[20] Kuphaldt R., Tony. 2007. Lessons In Electric Circuits Volume IV – Digital, pages 103-112. 4th ed. https://www.allaboutcircuits.com/assets/pdf/digital.pdf [accessed on 2023-01-04]

[21] Pereira Rui, Couto Marco, Ribeiro Francisco,Rua Rui,Cunha Jácome,Fernandes Paulo João, Saraiva João. 2017. Energy Efficiency across Programming Languages. In Proceedings of SLE'17, Vancouver, BC, Canada, October 23–24, 12 pages. https://doi.org/10.1145/3136014.3136031 [accessed on 2022-12-15]

[22] Britannica, T. Editors of Encyclopaedia. 2022. Microprocessor. https://www.britannica.com/technology/microprocessor [accessed on 2023-01-02]

[23] Odunlade, Emmanuel. 2018. How to Select the Right Microcontroller for Your Embedded Application. https://circuitdigest.com/article/how-to-select-the-right-microcontroller-for-your-embedded-application [accessed on 2023-01-02]

[24] Odunlade, Emmanuel. 2019. Minimizing Power Consumption in Microcontrollers. https://medium.com/nerd-for-tech/minimizing-power-consumption-in-microcontrollers-b3eb9593c893 [accessed on 2023-01-02]

[25] Schumm, Brooke. 2022. battery. https://www.britannica.com/technology/battery-electronics [accessed on 2023-01-03]

[26] LoRa Alliance Technical Marketing Workgroup. 2015. A technical overview of LoRa and LoRaWAN https://www.tuv.com/content-media-files/master-content/services/products/1555-tuv-rheinland-lora-alliance-certification/tuv-rheinland-lora-alliance-certification-overview-lora-and-lorawan-en.pdf [accessed on 2023-01-02]

[27] Odunlade, Emmanuel. 2019. Introduction to LoRa and LoRaWAN: What is LoRa and How Does It Work? https://circuitdigest.com/article/introduction-to-lora-and-lorawan-what-is-lora-and-how-does-it-work[accessed on 2023-01-05]

[28] Lohnes, Kate. 2017. How Does Wi-Fi Work?. Encyclopedia Britannica. https://www.britannica.com/story/how-does-wi-fi-work [accessed on 2023-01-04]

[29] Britannica, T. Editors of Encyclopaedia. 2022. Wi-Fi. Encyclopedia Britannica. https://www.britannica.com/technology/Wi-Fi [accessed on 2023-01-04]

[30] MOKOLoRa. 2021. Comparison between LoRa and other wireless technologies. https://www.mokolora.com/lora-and-wireless-technologies/ [accessed on 2022-12-21]

[31] Deutsche Telekom AG. 2021. NB-IoT, LoRaWAN, Sigfox: An up-to-date comparison. https://iot.telekom.com/en/downloads/mobile-iot-network-comparison-nb-iot-lorawan-sigfox [accessed on 2022-12-21]

[32] Britannica, T. Editors of Encyclopaedia. 2023. Bluetooth. https://www.britannica.com/technology/Bluetooth [accessed on 2022-12-21]

[33] Raikar, S. Pai. 2022. How Does Bluetooth Work? https://www.britannica.com/story/how-does-bluetooth-work[accessed on 2022-12-21]

[34] M. Chen, Y. Miao, Y. Hao, K. Hwang. 2017. Narrow Band Internet of Things. in IEEE Access, vol. 5, pp. 20557-20577, 2017. http://dx.doi.org/10.1109/ACCESS.2017.2751586 [accessed on 2023-01-02]

[35] R. Ratasuk, N. Mangalvedhe, D. Bhatoolaul, A. Ghosh. 2017. LTE-M Evolution Towards 5G Massive MTC. IEEE Globecom Workshops. pp. 1-6. http://dx.doi.org/10.1109/GLOCOMW.2017.8269112 [accessed on 2023-01-03]

[36] A. Díaz-Zayas, C. A. García-Pérez, Á. M. Recio-Pérez and P. Merino. 2016. 3GPP Standards to Deliver LTE Connectivity for IoT. IEEE First International Conference on Internet-of-Things Design and Implementation. pp. 283-288. http://dx.doi.org/10.1109/IoTDI.2015.26[accessed on 2023-01-03]

[37] Link Labs. 2016. LTE eDRX and PSM Explained for LTE-M1. https://www.link-labs.com/blog/lte-e-drx-psm-explained-for-lte-m1[accessed on 2023-01-03]

[38] Precision solutions. 2022. Scale Calibration Guide. https://www.precisionsolutionsinc.com/scale-calibration-guide/ [accessed on 2023-01-02]

[39] Vågexperten. 2022. Precisionsvåg KERN EW-N / EG-N. https://www.vagexperten.se/precisionsvag-kern-ew-n-eg-n.html [accessed on 2023-01-05]

[40] Tsavalos, Nikolaos. Abu Hashem, Ahmad. 2018. Low Power Wide Area Network (LPWAN) Technologies for Industrial IoT Applications. Department of Electrical and Information Technology, Faculty of Engineering, LTH, Lund University. pp. 51-63. http://lup.lub.lu.se/student-papers/record/8950859 [accessed on 2023-01-02]

[41] Singh Abhishek. 2021. Raspberry Pi 4 Specifications Pin Diagram and Description. https://www.hackatronic.com/raspberry-pi-4-specifications-pin-diagram-and-description/ [accessed on 2023-01-15]

[42] Texas Instruments. 2018. CC430F614x, CC430F514x, CC430F512x MSP430™ SoC With RF Core, Datasheet.[accessed on 2023-01-05] https://www.ti.com/lit/ds/symlink/cc430f5143.pdf?ts=1673022316831&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC430F5143

[43] Renesas. 2022. RL78/G14 Datasheet. [accessed on 2023-01-05] https://www.renesas.cn/us/en/document/dst/rl78g14-data-sheet-rev340?r=1635701

[44] Microchip. 2019. SAM R34/R35 Low Power LoRa® Sub-GHz SiP Datasheet.[accessed on 2023-01-05] https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/SAM-R34-R35-Low-Power-LoRa-Sub-GHz-SiP-Data-Sheet-DS70005356C.pdf

[45] STMicroelectronics. 2022. Datasheet - STM32WLE5xx STM32WLE4xx - Multiprotocol LPWAN 32-bit Arm Cortex-M4 MCUs, LoRa, (G)FSK, (G)MSK, BPSK, up to 256KB flash, 64KB SRAM. [accessed on 2023-01-05] https://www.st.com/resource/en/datasheet/stm32wle5c8.pdf

[46] Semiconductor component industries. 2018. ASX340AT - 1/4-inch Color CMOS NTSC/PAL Digital Image SOC with Overlay Processor. [accessed on 2023-01-05] https://eu.mouser.com/datasheet/2/308/ASX340AT_D-1540035.pdf

[47] Jost, Danny. 2019. What is an Ultrasonic Sensor? https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor [accessed on 2023-01-06]

[48] Burnett, Roderick. 2017. Ultrasonic vs Infrared (IR) Sensors – Which is better? https://www.maxbotix.com/articles/ultrasonic-or-infrared-sensors.htm [accessed on 2023-01-06]

[49] Pixabay, Open source images. https://pixabay.com/vectors/wheatstone-bridge-electric-circuit-40465/

Appendix

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h> //For getcwd
#include <time.h>
#include <stdbool.h>
#include <errno.h>

#define PATH_MAX    4096

struct Size {
    int rows, cols;
};

typedef struct Size Size;

void delete_arr(unsigned char** arr, Size size){

    for(int i = 0; i < size.rows; i++){

        free(arr[i]);
    }
    free(arr);
}

unsigned char** create_arr(Size size, bool mal){

    //printf("Creating array with size: %dx%d\n", size.rows, size.cols);

    unsigned char** arr;

    if(mal){
        arr = malloc(size.rows * sizeof(char*));
        for(int i = 0; i < size.rows; i++){

            arr[i] = malloc(size.cols * sizeof(char));
    }
    }
    else{
        arr = calloc(size.rows, sizeof(char*));
        for(int i = 0; i < size.rows; i++){

            arr[i] = calloc(size.cols, sizeof(char));
        }
    }

    return arr;
}
```

```
50
51  void export_csv(unsigned char** arr, Size arrsize, char fileName[]){
52
53      char cwd[PATH_MAX];
54      getcwd(cwd, sizeof(cwd));
55      strcat(cwd, fileName);
56      printf("Exporting file \"%s\"\n...\n", cwd);
57
58      FILE *csv = fopen(cwd, "w");
59
60      if( csv == NULL ){
61          printf("Export: file open error.. %s\n", strerror(errno));
62          exit( -1 );
63      }
64
65      for(int i = 0; i < arrsize.rows; i++){
66          for(int j = 0; j < arrsize.cols; j++){
67
68              fprintf(csv, "%d", arr[i][j]);
69              if( j < arrsize.cols - 1 ){
70                  fprintf(csv, " ");
71              }else{
72                  fprintf(csv, "\n");
73              }
74          }
75      }
76      fclose(csv);
77  }
78
79  void print_arr(unsigned char** arr, Size arrsize){
80
81      printf("[");
82      for(int i = 0; i < arrsize.rows; i++){
83
84          printf("[");
85
86          for(int j = 0; j < arrsize.cols; j++){
87
88              if( j < arrsize.cols - 1 ){
89                  printf("%d ", arr[i][j]);
90              }
91              else{
92                  printf("%d]", arr[i][j]);
93              }
94          }
95          if( i < arrsize.rows - 1 ){
96              printf("\n ");
97          }
98      }
99      printf("]\n");
100 }
101
```

```
102 unsigned char** load_file(Size imsize, char fileName[]){
103 
104     char cwd[PATH_MAX];
105     getcwd(cwd, sizeof(cwd));
106     strcat(cwd, fileName);
107 
108     printf("Loading file \"%s\"\n...\n", cwd);
109 
110     FILE *fptr = fopen(cwd, "r");
111 
112     if ( fptr == NULL ){
113         printf("Load: File open error..(%s)\n", strerror(errno));
114         exit(-1);
115     }
116 
117     unsigned char** image_matrix = create_arr(imsize, true);
118 
119     int i = 0;
120     int j = 0;
121     int number;
122 
123     //Load all integers from the csv
124     while ( fscanf(fptr, "%d", &number) != EOF ){
125 
126         if( i > imsize.rows-1 ){
127             //Matrix full
128             break;
129         }
130 
131         image_matrix[i][j] = number;
132 
133         if( j != 0 && j % ( imsize.cols - 1 ) == 0 ){
134             //New row
135             i++;
136             j = 0;
137         }else{
138             j++;
139         }
140     }
141 
142     fclose(fptr);
143 
144     return image_matrix;
145 }
146 
147 unsigned char** pad(unsigned char** image, Size *imsize){
148 
149     //Update imsize and allocate new expanded array
150     Size newSize = { .rows = imsize->rows + 2, .cols = imsize->cols + 2 };
151     unsigned char** expanded = create_arr(newSize, false);
152 
153     for(int i = 0; i < imsize->rows; i++){
```

```
154
155         memcpy(&expanded[i + 1][1], image[i],
156             (size_t)(imsize->cols*sizeof(char)));
157     }
158
159     delete_arr(image, *imsize);
160     imsize->rows += 2;
161     imsize->cols += 2;
162
163     return expanded;
164 }
165
166 unsigned char** unpad(unsigned char** image, Size *imsize){
167
168     Size newSize = { .rows = imsize->rows - 2, .cols = imsize->cols - 2 };
169     unsigned char** reduced = create_arr(newSize, false);
170
171     for(int i = 0; i < newSize.rows; i++){
172
173         memcpy(reduced[i], &image[i + 1][1], newSize.cols*sizeof(char));
174     }
175
176     delete_arr(image, *imsize);
177     imsize->rows -= 2;
178     imsize->cols -= 2;
179
180     return reduced;
181 }
182
183 unsigned char** sobel(unsigned char** image, Size imsize, bool xy){
184
185     unsigned char** sobeld = create_arr(imsize, true);
186     int x = 0;
187     int y = 0;
188
189     for(int i = 1; i < imsize.rows-1; i++){
190         for(int j = 1; j < imsize.cols-1; j++){
191
192             x = 0.25*(-(image[i - 1][j - 1] + image[i + 1][j - 1]) +
193                 image[i - 1][j + 1] + image[i + 1][j + 1] +
194                 2*(image[i][j + 1] - image[i][j - 1]));
195             if(xy){
196                 y = 0.25*(-(image[i + 1][j - 1] + image[i + 1][j + 1]) +
197                     image[i - 1][j - 1] + image[i - 1][j + 1] +
198                     2*(image[i - 1][j] - image[i + 1][j]));
199                 sobeld[i][j] = (unsigned char) (abs(x) + abs(y));
200             }
201             else{
202                 sobeld[i][j] = (unsigned char) (abs(x));
203             }
204         }
205     }
```

```
206
207     delete_arr(image, imsize);
208     return sobeld;
209 }
210
211 unsigned char** threshold(unsigned char** image, Size imsize, int offset){
212
213     unsigned char** thresholded = create_arr(imsize, true);
214     int mean;
215
216     for(int i = 1; i < imsize.rows - 1; i++){
217         for(int j = 1; j < imsize.cols - 1; j++){
218
219             mean = 0.125*(image[i-1][j-1] + image[i-1][j] +
220                 image[i-1][j+1] + image[i][j-1] + image[i][j+1] +
221                 image[i+1][j-1] + image[i+1][j] + image[i+1][j+1]);
222
223             thresholded[i][j] = (unsigned char)
224                 ((image[i][j] < ( mean - offset )) ? 255 : 0);
225             mean = 0;
226         }
227     }
228
229     delete_arr(image, imsize);
230     return thresholded;
231 }
232
233 unsigned char** diff(unsigned char** ref, unsigned char** test, Size imsize){
234
235     unsigned char** difference = create_arr(imsize, true);
236     for(int i = 0; i < imsize.rows; i++){
237         for(int j = 0; j < imsize.cols; j++){
238
239             difference[i][j] = (unsigned char) abs(ref[i][j] - test[i][j]);
240         }
241     }
242
243     return difference;
244 }
245
246 unsigned char** filter_dots(unsigned char** image, Size size){
247
248     unsigned char** filtered = create_arr(size, false);
249     int sum = 0;
250
251     for(int i = 1; i < size.rows - 1; i++){
252         for(int j = 1; j < size.cols - 1; j++){
253
254             if(image[i][j] < 255){
255                 //Nodot
256                 filtered[i][j] = image[i][j];
257             }
```

```
            else{
                //dot
                sum = image[i-1][j-1] + image[i-1][j] + image[i-1][j+1] +
                    image[i][j-1] + image[i][j+1] + image[i+1][j-1] +
                    image[i+1][j] + image[i+1][j+1];
                if(sum <= 255){
                    //unimportant dot
                    filtered[i][j] = 0;
                }
                else{
                    //important dot
                    filtered[i][j] = image[i][j];
                }
            }
        }
    }

    delete_arr(image, size);
    return filtered;
}

bool decide(unsigned char** diff, Size size, int floor){

    int nbr = 0;
    bool decide = false;

    //Count white pixels
    for(int i = 0; i < size.rows; i++){
        for(int j = 0; j < size.cols; j++){

            if(diff[i][j] > 0){
                nbr++;
            }
        }
    }

    if(nbr > floor){
        decide = true;
    }

    return decide;
}

bool process(char file[], char ref[], char output[]){

    Size imsize = { .rows = 200, .cols = 200 };
    unsigned char** reference;
    unsigned char** image;
    unsigned char** difference = create_arr(imsize, true);
    bool decision;
    int thresh_val = 5;

```

```
310     //printf("File: %s\nReference: %s\nOutput: %s\n", file, ref, output);
311 
312     image = load_file(imsize, file);
313     image = pad(image, &imsize);
314     image = sobel(image, imsize, false);
315     image = unpad(image, &imsize);
316     //export_csv(image, imsize, "/figures/sobeled.csv");
317     image = pad(image, &imsize);
318     image = threshold(image, imsize, thresh_val);
319     image = unpad(image, &imsize);
320     //export_csv(image, imsize, "/figures/thresholded.csv");
321 
322     reference = load_file(imsize, ref);
323     reference = pad(reference, &imsize);
324     reference = sobel(reference, imsize, false);
325     reference = threshold(reference, imsize, thresh_val);
326     reference = unpad(reference, &imsize);
327     //export_csv(reference, imsize, "/figures/reference.csv");
328 
329     difference = diff(reference, image, imsize);
330     //export_csv(difference, imsize, "/figures/diff_unfiltered.csv");
331     difference = pad(difference, &imsize);
332     difference = filter_dots(difference, imsize);
333     difference = unpad(difference, &imsize);
334     decision = decide(difference, imsize, 280);
335     //printf("Decision is: %d\n", decision);
336 
337     export_csv(difference, imsize, output);
338 
339     if(image != NULL){
340         delete_arr(image, imsize);
341     }
342 
343     return decision;
344 }
345 
346 int main(int argc, char** argv){
347 
348     clock_t start;
349     double elapsed_time;
350 
351     start = clock();
352 
353     //CODE HERE//
354     process(argv[1], argv[2], argv[3]);
355 
356     elapsed_time = ((double) (clock() - start) / CLOCKS_PER_SEC);
357 
358     if((int)elapsed_time > 0){
359         printf("Execution time: %.3fs\n", elapsed_time);
360     }
361     else if((int)(elapsed_time*1000) > 0){
```

```
362         printf("Execution time: %.3fms\n", 1000*elapsed_time);
363     }
364     else{
365         printf("Execution time: %.3fus\n", 1000000*elapsed_time);
366     }
367 
368     return 0;
369 }
```

Printed by Tryckeriet i E-huset, Lund 2023

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2023-908
http://www.eit.lth.se