

DEPARTMENT OF
INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Semester 2021/2022

Smart Capacitive Sensing Wearable for Detecting Wrist & Finger Kinematics

Master Thesis



Elio Reinschmidt
elior@student.ethz.ch
23.05.2022

Supervisor: Dr. Michele Magno, magnom@iis.ee.ethz.ch

Dr. Christian Vogt, christian.vogt@pbl.ee.ethz.ch

Professor: Prof. Dr. Sebastian Kozerke, kozerke@biomed.ee.ethz.ch

Abstract

The average daily screen time worldwide is slightly less than seven hours per day [1]. This number is steadily increasing as we enrich our daily lives more and more with electronic devices. It is therefore important that we address the way we interact with these devices, as this has a direct impact on our quality of life. We need new, intuitive ways to interact with these devices that ensure a seamless experience. This work introduces such a novel input device. It is a bracelet that measures wrist and finger movement using charge variation (QVAR) on the skin of the wrist. To increase user adoption of this new technology, it was designed in the shape of a wristwatch so that it can be integrated into any smartwatch. Six QVAR channels are implemented on the bracelet and their differential electrodes are integrated into the wristband. Thus, it is capable of transmitting the raw QVAR data via Bluetooth Low Energy in real time. A custom gesture dataset was recorded with 20 participants consisting of 10 gestures and used to train a machine learning model. This model runs directly on the microcontroller in real time and has an overall accuracy for the 10 recorded gestures of 87.17%. The device has a low power consumption of less than 6.52 mA at 3.7 V, which guarantees a battery life of 11h 40min with the built-in 75mAh battery.

Acknowledgments

I would like to take this opportunity to thank the following people who have supported me in this master's thesis. First, I would like to thank my two supervisors, Dr. Michele Magno and Dr. Christian Vogt. Their direct support and guidance were of great help during this thesis. Furthermore, I would like to thank Federico Villani for sharing his knowledge on various topics with me, which was very helpful in troubleshooting the various problems that arose. I would like to thank Julian Moosmann and Marco Giordano for supporting me with their expertise on machine learning. Finally, I would like to thank all the volunteers who participated in the acquisition of the required gesture dataset.

Content

Abstract.....	2
Acknowledgments	3
1 Introduction	7
2 Theoretical Background	9
2.1 Electromyography (EMG).....	9
2.1.1 sEMG Signal Recording.....	9
2.1.2 Noise and Filtering	10
2.2 Electrostatic Charge Variation (QVAR)	11
2.2.1 Triboelectric effect	11
2.2.2 Sensing Electrostatic Charge Variation (QVAR).....	11
2.3 Deep Learning Classification	13
2.3.1 Overview	13
2.3.2 Artificial Neural Networks (ANN).....	13
2.3.3 Recurrent Neural Networks (RNN)	14
2.3.4 TensorFlow Lite for Microcontrollers	15
2.4 Zephyr.....	16
2.5 Bluetooth Low Energy (BLE)	16
2.5.1 Overview	16
2.5.2 Generic Access Profile (GAP).....	16
2.5.3 Generic Attribute Profile (GATT).....	17
3 Related Work.....	18
3.1 Gesture Recognition	18
3.1.1 Overview	18
3.1.2 Myo Bracelet from Thalmic Labs	19
3.1.3 Capacitive Based Recognition.....	19
3.2 Capacitive Electrodes	21
4 Developing Process of the QVAR Bracelet	23
4.1 Design of the EMG & QVAR Development Board	23
4.1.1 Overview	24
4.1.2 QVAR Electrodes and Test Setup	24
4.1.3 COMSOL Simulation of Capacitive Electrodes.....	26

4.1.4 Experimental Results QVAR Electrodes.....	28
4.2 Improved Design	30
4.2.1 Overview	30
4.2.2 Electrode Strap	32
5 Final System Architecture	35
5.1 Overview	35
5.2 Electrical System	35
5.2.1 nRF5340 / NORA-B106.....	35
5.2.2 LSM6DSV16X IMU with QVAR.....	36
5.2.3 Power	37
5.3 Hardware Architecture	38
5.3.1 PCBs & Capacitive Electrodes	38
5.3.2 Mechanical Construction.....	39
5.4 Firmware Architecture.....	40
5.4.1 Firmware Overview	40
5.4.2 Bluetooth Low Energy (BLE)	40
5.4.3 TensorFlow Lite for Microcontrollers	41
6 Gesture Dataset.....	42
6.1 Gestures	42
6.2 Recording Application.....	43
6.3 Recording Setup	44
6.4 Dataset Structure.....	45
7 Gesture Recognition	46
7.1 Data Preprocessing, Augmentation & Feature Extraction	46
7.2 Neural Network Architecture (SideflickNN)	49
8 Experimental Results.....	51
8.1 QVAR Bracelet Performance	51
8.1.1 Power Consumption	51
8.1.2 QVAR Measurements of the Final Bracelet.....	51
8.2 Gesture Recognition	54
8.2.1 Accuracy Evaluation.....	54
8.2.2 SideflickNN Performance.....	55

8.2.3 Quantization	57
8.2.4 Number of QVAR channels vs Accuracy.....	57
8.3 Comparison to the Myo Bracelet.....	58
9 Conclusion & Future Work	61
9.1 Conclusion	61
9.2 Future Work.....	62
List of Figures.....	63
List of Tables	65
List of Acronyms	66
Bibliography	67

1 Introduction

We live in a time when electronic devices have become an integral part of our daily lives. The last decade has ensured that everyone has a smartphone. It has become our daily companion and is so important to our lives that most people no longer leave home without it. Globally, the average person's daily screen time is a little less than 7 hours [1]. With so much time spent with electronic devices, the way we interact with them is becoming increasingly important, as it has a major impact on the quality of our lives.

One technology that promises to improve the user experience is augmented reality smart glasses. The first generation of these glasses is already on the market, and major companies like Apple, Google, Microsoft and Facebook are researching them intensively. They promise a solution where we can leave the phone in our bag, but still be able to see the screen and interact with the digital world. The problem with most current smart glasses is that the user still needs their phone as a remote to interact with the screen. If we want to improve the user experience and lower the barrier for adoption, we need to be able to interact seamlessly with them. This is only possible if we find a more intuitive way of interacting than a touchscreen.

Gestures are well suited for interacting with such devices because we perform them naturally when we communicate. To recognize a gesture, we first need to record our movements before we can analyse them and extract the gesture. Over the past year, several technologies have been proposed for this task. Broadly speaking, there are two ways to accomplish this. The first option is to use an external system that observes us, as is the case with computer vision or radar. The second option uses wearables to measure bio signals such as electromyography or to measure motion with accelerometers. The former has the great advantage that the user does not need to wear an additional device. However, he is limited to the field of view of the camera/radar and therefore cannot move freely. The opposite is true for the latter, which requires an additional wearable device, such as an EMG bracelet, but allows the user to move independently.



Figure 1: The Myo Bracelet from Thalmic Labs [2]

The goal of this work is to develop such a wearable system that can serve as a novel input device for such applications. It should be able to recognize various gestures performed by the user, which can be used to interact with various devices such as smart glasses or prosthetics. As electrical technology is rapidly advancing, it should be equipped with novel hardware and sensors to enhance the already proven methods such as EMG. A good candidate is electrostatic charge variation (QVAR), it is a novel electrical potential sensing channel developed by STMicroelectronics. With it, we can measure the deformation of our skin and draw conclusions about the gesture that caused it. The main advantage over EMG is that it is capacitively coupled and therefore does not require galvanic skin contact. If we want to measure the movement of our hand, we can attach the electrodes to the wrist, while EMG electrodes must be placed directly over the muscle on the back of the forearm near the elbow. The latter is an important advantage as we can integrate this technology into any watch, which will increase ease of use and adoption. As the user does not need to wear any additional devices.

The final device should meet the following criteria:

- It should be small, compact and convenient for the user to carry.
- To be portable, it should be battery powered and the battery life should be at least 7 hours.
- It should be able to transmit the raw data in real time via Bluetooth Low Energy.
- A gesture data set of at least 20 people should be recorded and a machine learning model that can recognize different gestures should be trained on it.
- This machine learning model should be implemented on the device itself and run-in real time.

In this master thesis the design of such a novel input device is presented. Chapter 2 presents the necessary theoretical background. Chapter 3 discusses the current state of the art and presents similar research on this topic. Chapter 4 explains the detailed development process of the device. Chapter 5 presents the final device and its specifications. The acquisition of the dataset and the gestures used are described in Chapter 6. The machine learning model and the features used are discussed in Chapter 7. The performance of the device is analysed in Chapter 8. The conclusion is drawn in Chapter 9 and some important points for future work are discussed.

2 Theoretical Background

2.1 Electromyography (EMG)

Stimulation of a muscle by a nerve causes it to contract. This generates a current that flows through the muscle. Since the muscle has a certain resistance, the flow of charge causes a voltage drop across it. This voltage drop is measured in EMG using electrodes [3]. With EMG, we can measure the muscle activity of a person, which opens up the possibility for many applications, such as detecting hand movements to perform gesture recognition.

2.1.1 sEMG Signal Recording

There are two main types of electrodes when it comes to EMG recording. The first type consists of thin needles that are inserted directly into the muscle. This is called nEMG. By inserting the needle directly into the muscle, we can isolate the signal from individual muscle fibers. This results in much higher accuracy. When compared to muscle, skin has a much higher resistance, so this method gives a much better signal-to-noise ratio. The only drawback to this technology is that it is invasive. The alternative method is the use of surface electrodes. This method is referred to as sEMG. The main advantage of this technology is that it is non-invasive, as only gel pads or metal plates are placed on the skin. Since the electrode is further away from the muscle, it can only measure the sum of all muscle fiber activity below the electrode, resulting in less accuracy. As mentioned earlier, the skin has a much higher impedance, which reduces the signal-to-noise ratio. Although the signal is poorer compared to the nEMG, the non-invasiveness of the sEMG makes it a perfect candidate for daily use in wearable devices.

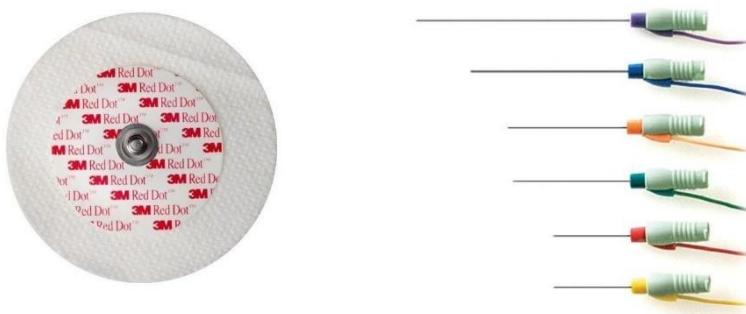


Figure 2: Gel electrodes (left) [4], needle electrodes (Right) [5]

The sEMG signal is recorded using a differential amplifier. This measures the voltage across two electrodes. This signal is passed to an analog-to-digital converter (ADC), which quantizes it before it is further filtered and processed. Filtering out noise is important in sEMG measurement; this is discussed in section 2.1.2. [6]

2.1.2 Noise and Filtering

When measuring sEMG signals we'll record our wanted signal, that is overlayed with additional noise. In order to distil the wanted information, the signal must be first filtered. The desired signal will have an amplitude of 0-10mV, and it has a frequency range of 20Hz to 500Hz, but the most power lies in the range of 50Hz to 150Hz [6 - 8]. Figure 3 shows an sEMG recording with the electrodes placed over the flexor carpi ulnaris muscle and the wrist was bent upwards twice. The signal peaks correspond to the activation of the muscle. It was recorded with a sampling frequency f_s of 500 Hz

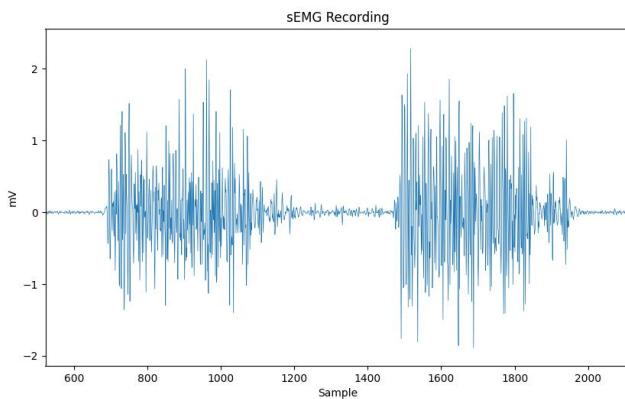


Figure 3: sEMG recording of the flexor carpi ulnaris muscle ($f_s = 500$ Hz)

Most of the noise comes from our environment, and our skin acts like a large antenna, picking up electromagnetic radiation that is superimposed on the signal. The predominant source of noise comes from our power lines and therefore has a frequency of 50/60 Hz. This noise must be filtered out as it can be up to three orders of magnitude larger than our desired signal. The best way to remove this noise is to use a 50/60 Hz notch filter. In addition, the frequencies between 0 and 20 Hz should also be removed, as the sEMG signal is inherently unstable in this range and drifts greatly. This is due to the random activation of the individual muscle fibers. Since we live in a wireless world, we are exposed to a lot of radio communication. Fortunately, this has a much higher frequency than 500 Hz and can easily be removed with a low-pass filter. [8]

2.2 Electrostatic Charge Variation (QVAR)

2.2.1 Triboelectric effect

When a person rubs a balloon against their sweatshirt or walks across a carpet, they pick up charges. This accumulation of charge is best felt when touching something made of metal, and a small electrical spark jumps from the body. We know from experience that rubbing certain materials together generates static electricity. This is the triboelectric effect.

When two materials are close together, the atoms on the surface interact with each other. At the atomic level, we can observe an overlap of the electron clouds. Now, if we add relative motion like rubbing, one material, which is more attractive to electrons, can pull the electrons out of the other material. After separating the two materials, we find that one material has accumulated more electrons and is now negatively charged. Meanwhile, the other is positively charged.

Since we are permanently affected by the triboelectric effect, our bodies are constantly charged and discharged by our movement and our interaction with the environment. These variations in charge can be measured and used to predict the motion that caused them. [9] [10]

2.2.2 Sensing Electrostatic Charge Variation (QVAR)

When two unequally charged materials come into contact with each other, they influence one another and try to establish equilibrium. This can be measured using an electrode and an electronic signal conditioning circuit. If we use an insulated electrode and thus no direct contact is made, the sensing process is achieved by electrostatic induction. When a positively charged material is brought near the electrode, the electrons in it are attracted to it. This results in an electron flow inside the electrode. A conditioning circuit connected to such an electrode with a high input resistance R_i measures this internal electron flow and generates a proportional voltage that can be sampled.

This type of sensor is called an electrostatic sensor. The application of this sensor in this work is to detect various hand movements. It is best explained by the function of a corresponding capacitive sensor. Our charged object in this case is the skin of our hand and forms one plate of the capacitor, and our electrode in this case is an insulated copper foil and forms the other plate. The capacitance C of a plate capacitor is inversely proportional to the distance d between the two plates and is proportional to the area A of them. The formula 1 below describes the capacitance of a plate capacitor, where ϵ_0 is the electrostatic constant. Since the electrode is on our skin and is much smaller, we can assume that the area between our skin and our electrode remains constant and only the distance changes. When we move our hand, we deform the skin and the distance between

it and the electrode changes. Therefore, the change in capacitance correlates with the movement of the hand. At the same time, our body is capacitively coupled with the environment. When we interact with it by moving and touching things, we constantly gain and lose charges. The change in the charges leads to a change in the voltage across the capacitor plates.

$$C = \frac{\epsilon_0 * A}{d} \quad (1)$$

From Section 2.1, we know that sEMG measures the activation of individual muscles and that the flexion and extension of muscles directly correlates with the movement of our hand. We can therefore use sEMG to detect hand and wrist movements. Electrostatic sensors, on the other hand, measure the change in distance caused by the deformation of the skin and the change in charges due to interaction with our environment. This makes it more difficult to detect and isolate hand movements compared to sEMG but has the advantage of not requiring direct skin contact since it is capacitively coupled. Since interaction with our environment plays a major role, we can take advantage of this effect by being able to measure movements at a distance from the electrode.

QVAR stands for electric charge (=Q) variation (=VAR). It is a novel electrical potential sensing channel developed by STMicroelectronics specifically for measuring quasi-electrostatic potential changes. This technology is further explored in this work as it is capable of detecting the differential electric potential change induced at the connected electrodes. The following figure 4 shows the block diagram of the QVAR sensor. [9] [11]

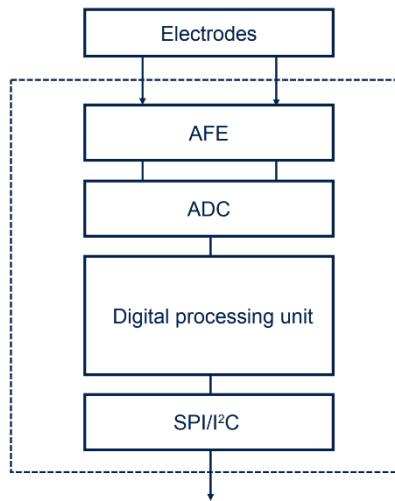


Figure 4: QVAR sensing channel architecture [9]

As described in section 2.1.2 the skin of our bodies acts as an antenna and picks up therefore electric radiation from our environment. 50/60Hz noise and high frequency noise from radio communication causes the same problem as in sEMG. We need therefore a 50/60Hz notch filter to remove power line noise and a low pass filter to reduce the noise from radio communication. An additional high pass filter is not needed, as the capacitive electrode and the input resistance of the sensor, form already a 1st order high pass filter.

The QVAR uses capacitive electrodes, which are also capacitively coupled to the environment. It is therefore more susceptible to disturbance from the environment and it must be ensured that the electrode and cables are well shielded by a faraday cage, as the induced noise can drown out the signal.

2.3 Deep Learning Classification

2.3.1 Overview

The main goal of Deep Learning is to simulate an approximate behaviour of the human brain. It is a subfield of machine learning that consists of "deep" networks with more than three layers. The goal is to train neural networks to "learn" from a large amount of data and to predict or classify new data that has not yet been seen. This is done through a combination of data inputs, weights, and biases that are wired together.

2.3.2 Artificial Neural Networks (ANN)

An artificial neural network (ANN) consists of interconnected nodes called neurons. Each neuron multiplies the inputs by the weight of the connection before adding them together. This value is fed into a nonlinear activation function before serving as input to subsequent neurons. An ANN is organized into a combination of layers composed of a series of neurons.

Deep Learning makes it possible to recognize complex information and patterns. This strength comes from stacking and combining several different layers. When it comes to detecting patterns in noise or generalizing certain features, ANNs are theoretically able to approximate the problem [12]. The issue is that there is no closed-form solution for determining the values for the individual weights and biases of the neurons. Therefore, the network must be trained iteratively with a large set of data. This is done by initializing the parameters with random values. After each training iteration, the network parameters are adjusted and optimized to minimize loss and increase accuracy. A common algorithm for determining the changes to be made is called gradient descent [13] [14].

This description applies only to the simplest type of deep neural network, the so-called artificial neural network. As the tasks for neural networks become more complex, so do the networks. Different types of networks have emerged to serve different applications and needs. Convolutional neural networks (CNNs) are used primarily in computer vision because they are optimized for processing images. Recurrent neural networks (RNN) are used for sequential or temporal data series, such as for speech and natural languages.

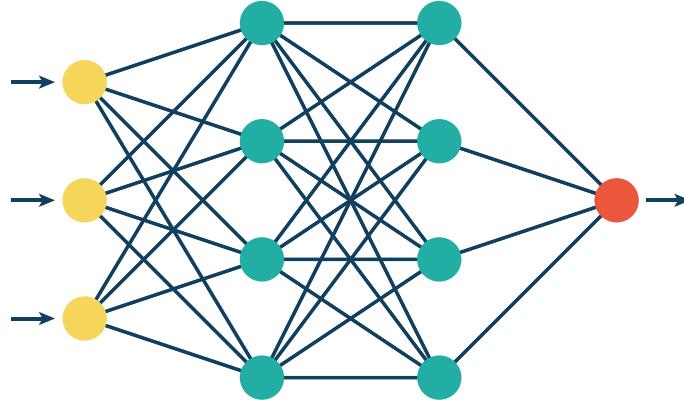


Figure 5: Simple ANN model, with input, hidden and output layer

2.3.3 Recurrent Neural Networks (RNN)

A gesture is a movement over time; therefore, a single snapshot is not sufficient; the entire sequence must be considered. In a pure feed-forward network such as an ANN, the entire sequence of data must be applied to the input at once. Thus, one must wait until the sequence is complete before passing it to the network. RNNs also have a feed-back input in addition to a feed-forward path. This feed-back path allows the network to store previous data and take it into account. They basically have a built-in memory that allows them to remember sequences. This architecture allows the network to be continuously fed data, so it can update predictions on the fly, rather than having to record the entire sequence beforehand. Long Short-Term Memory (LSTM) neural networks are a type of RNNs and have been shown to have good recall [15], making them well suited for time-series classification such as text, speech, and gesture/sign language recognition.

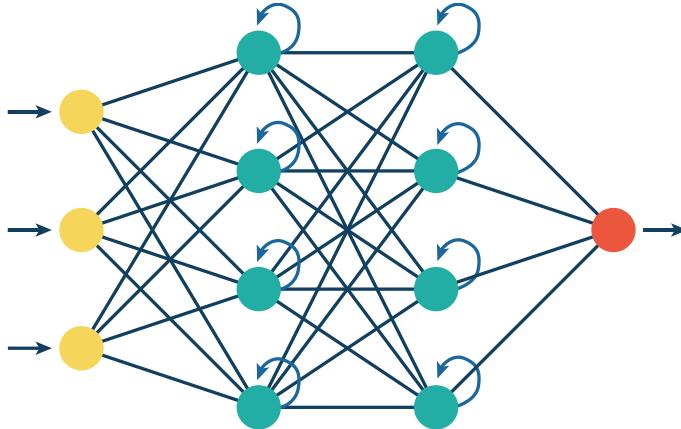


Figure 6: Simple RNN model with input, hidden and output layers

2.3.4 TensorFlow Lite for Microcontrollers

TensorFlow is a machine learning software library that is free and open source. It was released by the Google Brain team on November 9, 2015. Its focus is on training a deep neural network and running it. With its strength in ease of use, it provides a set of intuitive features for building, training, and running various models. [16]

The limited resources available on a microcontroller make the deployment of a machine learning model in an embedded system a non-trivial task. The two main limiting factors are the size of the model and the time required to compute an inference. A proven method to create a small model that runs fast is to use post-training quantization. The goal is to take the trained network and quantize the input, output, and weights of the model. Wu et al. showed that it is possible to quantize all the weights of a model from Float32 to Int8 while maintaining accuracy within 1% of the Float32 baseline. [17] This has the major advantage of drastically shrinking the model size, since a weight is represented by only a single byte. In addition, the latency for the inference is reduced since Int8 operations are faster than Float32.

To bridge the gap between training on a GPU and running the model on a microcontroller, TensorFlow provides a ready-to-use solution. TensorFlow Lite Micro is an open-source library designed to run machine learning models on microcontrollers with limited memory. The core runtime has a size of 16 KB and implements the most common deep learning model types. Loading the model from memory and running an inference is handled by this runtime, making it platform independent. [18] [19]

2.4 Zephyr

The Zephyr project [20] is a real-time operating system (RTOS) optimized for networked, resource-constrained embedded devices. It is a collaborative project of the Linux Foundation that has been running since 2016. As of January 2022, it is the RTOS with the largest number of contributors and commits [20]. Zephyr comes with a built-in kernel, a growing number of libraries, and a full TCP/IP and Bluetooth stack. In addition, TensorFlow Lite Micro is natively supported, so no additional libraries need to be imported. Together, this makes Zephyr a good candidate for embedded IoT devices. The device tree is a system adapted from the Linux kernel. There, common hardware components such as GPIOs, SPI, I2C, ADC and PWM are described and linked to drivers. These components can later be accessed via abstract APIs. As a result, Zephyr is hardware-independent, which enables easy integration into future research. [21] [22]

2.5 Bluetooth Low Energy (BLE)

2.5.1 Overview

Bluetooth Low Energy (BLE) was integrated into Bluetooth 4.0 in December 2009. It is a communication standard developed for wireless personal networks. It is independent of Bluetooth Classic and not compatible with it. The significantly lower power consumption compared to Classic makes it suitable for IoT, wearables and smart beacons. [23]. Two complementary communication protocols are used. GAP is essential and defines a device as BLE-enabled. GATT defines how data is formatted and exchanged between devices.

2.5.2 Generic Access Profile (GAP)

The first step in establishing a BLE connection is for one device to advertise itself so that the second device can scan it and establish a connection. This dynamic is usually split between two roles. The first role that can be assigned to a device is that of the central device. This is usually a more computationally powerful device such as a laptop, smartphone, or gateway. The second role is that of the peripheral device. This is usually a device with limited power consumption, such as a mouse, thermostat, or wearable. When the client receives an advertisement packet from the peripheral device, it can respond to it and establish a connection. The client selects the connection parameters, such as connection interval and packet length, and transmits them to the peripheral device. Role assignment (e.g., client or peripheral) and advertising are done via the GAP protocol. [24]

2.5.3 Generic Attribute Profile (GATT)

In BLE, data is stored in characteristics. A characteristic consists of its properties and a value (e.g. the value of a sensor). The data stored in characteristics are called attributes. Multiple characteristics that are related to each other are grouped into a service. If we take an accelerometer as an example, we would have an accelerometer service that contains three characteristics. Each of the three characteristics corresponds to the x, y and z axis and contains the acceleration value for the respective axis.

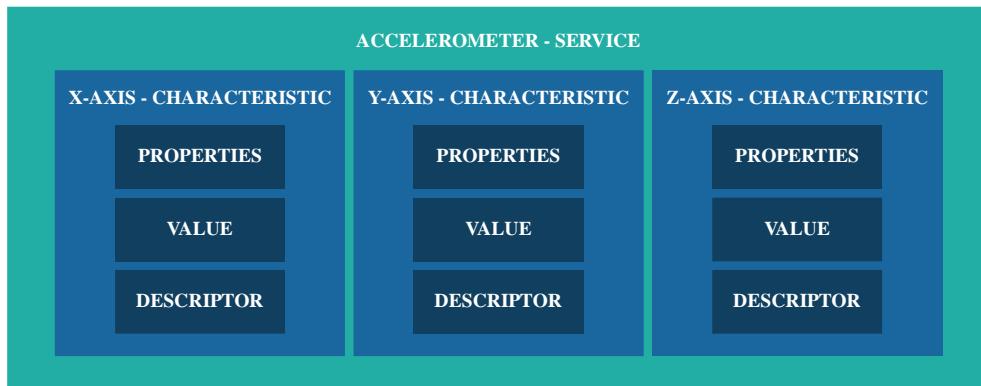


Figure 7: Example for an accelerometer service

The GATT protocol defines a way to discover these services and exchange characteristics. A peripheral device typically assumes the role of a GATT server that defines one or more of these services. A client connected to a peripheral device can read, write, and notify to the services defined by the peripheral device. The read/write command allows the client to read/write a value in a characteristic. When a client activates notification for a feature, it is automatically notified by the peripheral when a new value is written to the feature. [24]

3 Related Work

Gesture recognition is a promising technology. It enables intuitive interaction and communication in various applications, from input devices for smart glasses and prostheses to sign language recording. It is therefore not surprising that this topic is an active research area where much has already been done. This chapter presents the relevant theory on gesture recognition and the use of capacitive electrodes.

3.1 Gesture Recognition

3.1.1 Overview

Gesture recognition methods can be divided into two categories. The first consists of technologies that do not require direct contact with the user. They are mostly based on vision and radar, which allows good observation of the user's movements, and they do not require the user to wear anything. Pisharady et al [25] reviews 16 years of visual gesture recognition based on computer vision. The main problems cited are poor performance on complex backgrounds, especially in low light conditions, and that the user is constrained to the camera's field of view.

The second option is for the user to wear a device that records their movements. Various methods have become established, such as IMU, EMG and capacitive sensors. They have the great advantage that the user can move freely and is not limited to the field of view of a camera. The biggest hurdle is implementing the machine learning model in a wearable device that is small and convenient to wear.

Schlömer et al [26] used a Wii controller to use the built-in accelerometer to detect gestures. They achieved an accuracy of over 85%. An accelerometer can capture a lot of information about the gesture since most gestures involve large hand movements. One drawback is that it can only capture finger movements well when it is attached to the finger itself.

Chen et al [27] propose to combine accelerometer and sEMG data. They demonstrated that sensor fusion increases the achieved accuracy of gesture recognition by 5-10%. Zhang et al [28] built on this method and present a method that combines EMG and IMU data to perform sensor fusion. They are able to classify 72 Chinese characters with 95.3% accuracy. In addition, Zhang et al [28] proposes a machine learning method in which the start and end points of gesture recording are automatically determined by observing the intensity of the EMG signal.

3.1.2 Myo Bracelet from Thalmic Labs

The Myo bracelet from Thalmic Labs is an eight-channel sEMG wearable with an IMU on it. It is very popular for gesture recognition and many datasets of recorded gestures are available [29] [30]. Since it is well understood, it can serve as a good reference for a wearable capable of performing gesture recognition.

Zhang et al. used the Myo bracelet to record five gestures with 12 healthy subjects who repeated each gesture 30 times. Figure 8 below shows the five selected gestures. They propose a method of dividing the sEMG data into individual segments with a sliding window approach where features can be extracted. The five features in the time domain are: mean absolute value (MAV), slope sign change (SSC), waveform length (WL), root mean square (RMS), and Hjorth parameter (HP). Using a three-layer ANN model, they were able to achieve an accuracy of 98.7% and a model response time of 227.8 ms. [31]



Figure 8: Five chosen gestures [31]

3.1.3 Capacitive Based Recognition

Bian et al. show that one of the strengths of capacitively coupled electrodes is that they can measure the activity of the body in different areas where the electrode is not present. They use a human body capacitance-based coupling between the body and the environment. Here, the body acts as one plate of a capacitor, and the environment forms the second. Movement of the body changes the capacitance, resulting in a flow of charge that can be measured. They developed a bracelet that they used to collect a dataset of 11 subjects performing seven fitness exercises. The resulting bracelet is shown in Figure 9 below. Using their trained machine learning model, they were able to perform exercise counting with an average accuracy of 91% on body parts where the sensor was not directly involved. [32]

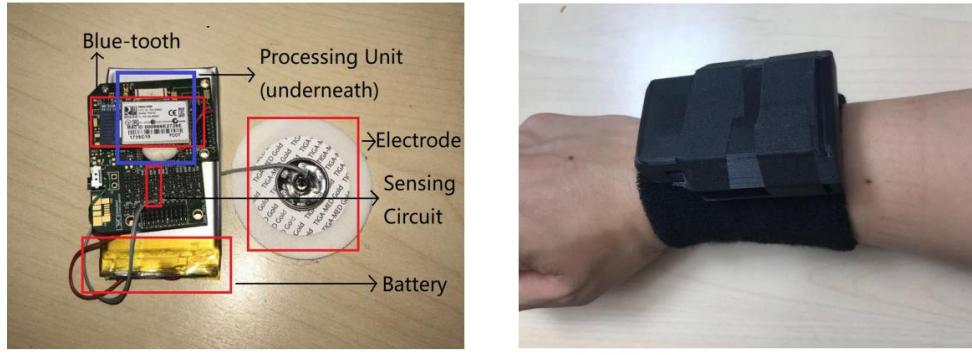


Figure 9: Hardware including sensing, processing, wireless transmission, and battery units (left); Hardware attached to wrist (right) [32]

Due to the triboelectric effect and the changes in our environment, our body potential is constantly changing. If our system is purely capacitively coupled, the problem can arise that the potential of our body changes over time compared to the potential of the system. This causes the signal to change even when the body is not moving. Nguyen et al. propose as a solution to use a ground electrode that galvanically couples the potential of the body to the potential of the system [33].

When we move our hand and perform our gestures, the tenon in our wrist deforms accordingly. Wang et al [34] propose a capacitive bracelet that takes advantage of this effect. The skin of our wrist forms one plate of the capacitor and the electrodes on the bracelet form the other. As the tenons deform, the distance between the skin and the electrode changes, resulting in a change in capacitance. Using this method, two sets of data were collected from five healthy subjects, the first containing nine wrist gestures and the second containing ten finger gestures. Figure 10 below shows the waveform for each gesture. Figure 12 shows the capacitive bracelet.

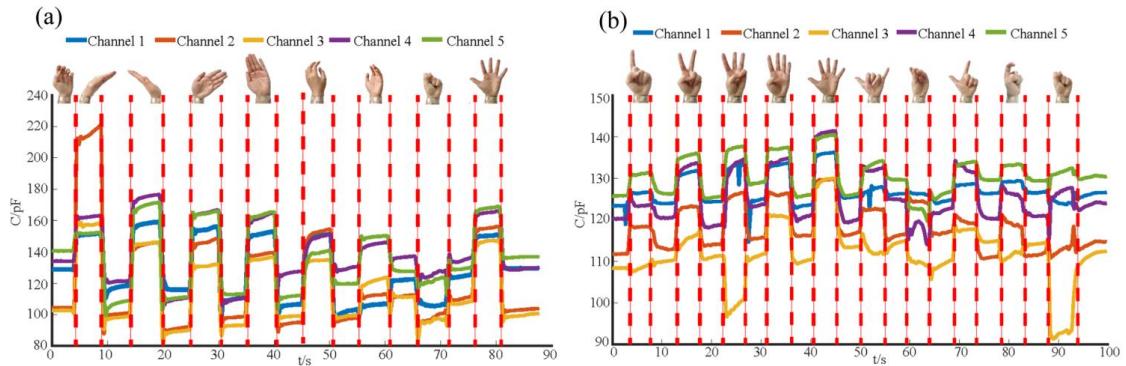


Figure 10: Time domain waveform of: (a) gesture set 1 (wrist gestures) and (b) gesture set 2 (finger gestures) [34]

3.2 Capacitive Electrodes

As described in Section 2.2.2, capacitive electrodes are required for the QVAR to function. Designing suitable capacitive electrodes is not trivial, as they directly affect the performance of the system. It is well known that capacitive electrodes are strongly affected by motion artifacts and ambient noise [35], since they are not only capacitively coupled to the skin but also to the environment. Portelli et al. present a method to shield the electrodes by building a Faraday cage around the electrode [36]. The Faraday cage is shown in the following figure 11. Building capacitive electrodes with printed circuit boards (PCB) has the advantage of being easy to fabricate. PCBs allow for detailed design and ease of fabrication.

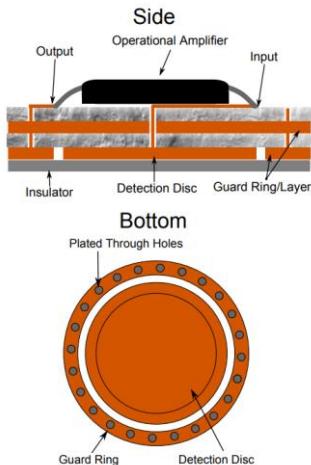


Figure 11: Diagram of the layers of the basic design of a capacitive sensor on a PCB [36]

Since the goal is to develop a wearable device, the electrodes must be flexible and comfortable to wear. Flexible printed circuit boards are a possible candidate to fulfil this. B. Babusia et al. present another method for integrating capacitive signal sensing into textile electrodes, using conductive fabrics as the sensing element and cotton as the dielectric. [37] This method opens up the possibility of flexible and stretchable electrodes, which would allow the QVAR to be integrated into gloves where direct measurements on the fingers are possible.

Wang et al [34] demonstrated a method in which a capacitive bracelet was fabricated from a 0.1 mm thick sheet of thermoplastic polyurethane (TPU) on which the electrodes were printed with conductive ink. The resulting bracelet is shown in Figure 12 below.

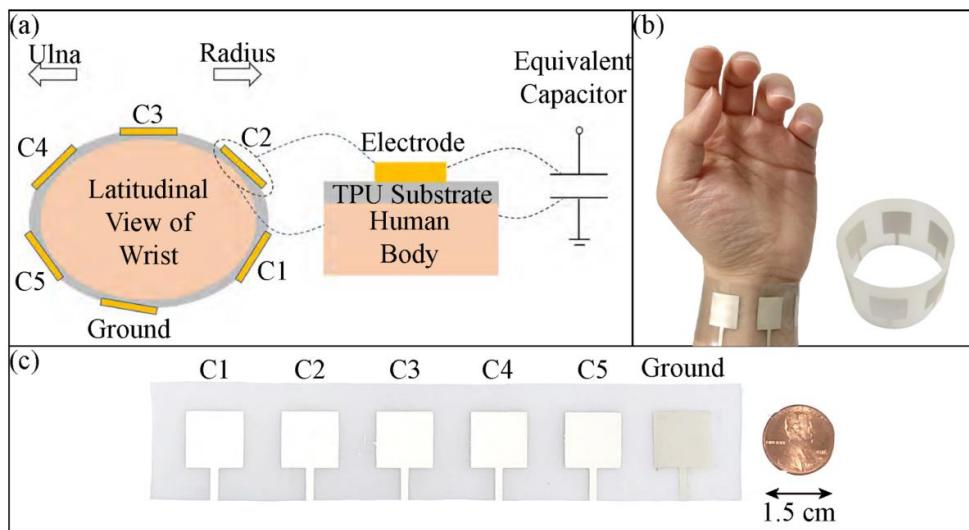


Figure 12: Overview of the flexible capacitive sensing system: (a) The sensing principle of the flexible capacitive sensing system. (b) the sensing system is placed on the wrist of one subject and (c) the optical image of the sensing system. [34]

4 Developing Process of the QVAR Bracelet

The initial goal of this master's thesis is to develop a novel, compact, low-power, and wearable system that collects sEMG, IMU, and electrostatic charge variation (QVAR) data. These data are to be streamed in real time via BLE. This new device will be used to collect a dataset to train a machine learning model capable of detecting and recognizing various hand and finger movements. At the time of writing, the QVAR sensor is nowhere to be found in a product, as production has not yet started, and only small engineering samples provided by the company are available. As a result, there is little to no knowledge about this sensor and its best application. The goal is to conduct an initial investigation into the strengths and weaknesses of this novel sensor. These findings will then be used to decide on the final application and system design.

In a first step, a development board was designed that meets the above requirements. This development board will be discussed in more detail in section 4.1. It allows an initial investigation of the sensors used. This development board and the QVAR sensor will be studied in detail, and the results will be presented in Sections 4.1.3 and 4.1.4. Based on these results, a final application will be selected. The resulting system is discussed in Section 4.2 and explained in detail in Section 5.

4.1 Design of the EMG & QVAR Development Board

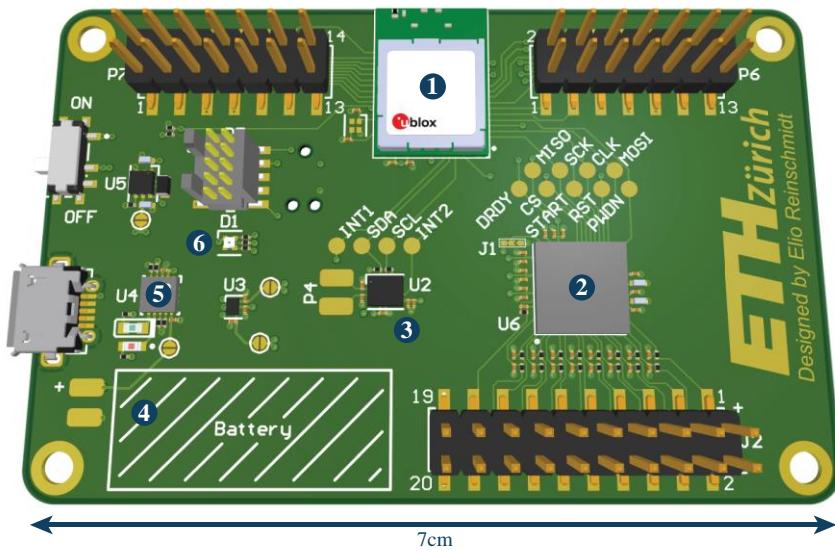


Figure 13: Rendering of the EMG and QVAR development board

4.1.1 Overview

The starting point of this master thesis was to design a development board that can be used to test both the sEMG sensor and the novel QVAR sensor. In addition, this board should be as close as possible to the final design in order to gain as much insight as soon as possible. The NORA-B106 (**1**) module from u-Blox was chosen for the microcontroller. It combines two processor cores. One is a fast application core that runs at 128 MHz with 1 MB of memory and is capable of running pre-trained machine learning models. The other core is a low-power network core that can handle all data streaming over BLE in real time.

The ADS1298 (**2**) from Texas Instruments was chosen for the sEMG sensor. It is an eight-channel delta-sigma ADC with 24-bit resolution designed for medical applications such as electrocardiogram (ECG) and electromyography (EMG).

The QVAR sensor is located in the LSM6DSV16X (**3**) IMU from ST Microelectronics. Additionally, it has a 3D digital accelerometer and a gyroscope, which will allow for future sensor fusion. The features and functions of this sensor are described in more detail in Section 5.2.2. To increase functionality, an 75 mAh Li-Po battery (**4**) and the LM3658 (**5**) Li-Po charging IC were added. This allows for portability of the system. An RGB LED (**6**) has been added to provide feedback to the user.

Overall, it is a portable, low-power wireless system capable of measuring 8 sEMG channels simultaneously with 24-bit resolution and transmitting the data via BLE at 500 samples per second. In addition, it is complemented by the integrated 6-axis IMU with a QVAR channel that can stream IMU and QVAR data with 16-bit resolution at 240 samples per second. This development board forms the basis for future investigation of sEMG and QVAR signals.

4.1.2 QVAR Electrodes and Test Setup

The first step in testing the QVAR sensor is to determine the shape and characteristics of the electrodes. A capacitive electrode is not only coupled to the skin, but also capacitively coupled with the environment. As a result, the electrode will not only pick up signal form the skin, but it is also picking up noise from the environment. To counteract this a faraday cage can be build around the electrode and connected to the systems ground. This ensures that the electrode is more strongly coupled with the shielding than the environment [36], which should reduce noise. The cables from the electrodes to the sensor are capacitively coupled with the environment. They should be shielded as well. In addition, the sensitive of the electrode should correlate with the size of the area. This can be explained by the fact that a larger electrode has a larger capacity, which can store more charges.

In order to test this assumption, a test setup has been designed that allows the development board to be attached to the wearer's forearm. It consists of a platform with two Velcro straps to which the development board can be attached. There are two solder pads on the board to which the differential electrodes can be soldered to. The electrodes are attached to the back of the hand using adhesive tape. The following figure 14 shows the test setup and the mounting position mentioned above.

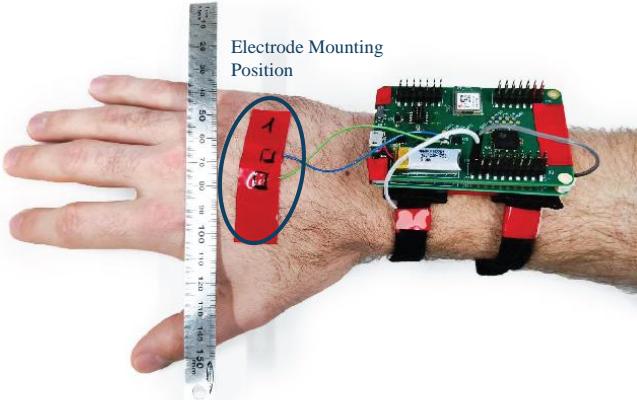


Figure 14: QVAR development board test setup

In its simplest form, a capacitive electrode consists of a conductive material (e.g., copper) connected to the input of the sensor that is electrically isolated by a non-conductive material (e.g. air, plastic, etc.). Flexible printed circuit boards (PCBs) are a good candidate for this job because they are easy and cheap to produce and allow fine and complex details in the design. Simply put, they allow for slim and flexible custom electrodes. To first get a qualitative understanding of how the electrodes should be designed, several capacitive electrodes were made by hand. They are made of copper tape stuck to a clear PVC sheet. The copper tape has a thickness of 0.1mm and the PVC film has a thickness of 0.1mm. This results in the stack-up shown in figure 16 below. They are not as refined as professional flexible PCBs, but they allow for quick testing because they can be made by hand in a few minutes, whereas a flexible PCB has a manufacturing and shipping time of at least a week. Figure 15 below shows a set of some handmade electrodes.



Figure 15: Set of handmade electrodes

The electrodes were made in different shapes and sizes and in a shielded and a non-shielded variant. The shielded type is made of the same stack as the unshielded one but has a stack of copper and PVC foil stuck on top with a double-sided 3M VHB tape. This copper foil is connected to ground and serves as a shield. The stack-up of a shielded and unshielded electrode is shown in figure 16 below. In addition, some of the electrode cables were shielded using a thin braided copper sleeve, which is also grounded.

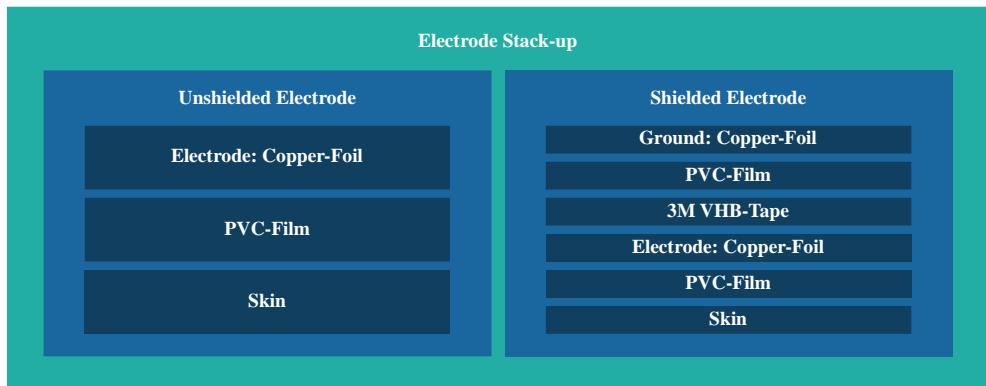


Figure 16: Stack-up of shielded and unshielded electrodes

4.1.3 COMSOL Simulation of Capacitive Electrodes

Portelli et al [36] have shown that the electrode can be shielded by surrounding the electrode with a grounded guard ring with via stitching around the electrode. Figure 17 below shows a possible stack-up. To get a better understanding, the design was simulated in COMSOL Multiphysics. An electrostatic simulation of the electrode was performed where the electrode and shield are modelled as perfect electric conductors. Figure 18 and table 1 shows the setup with all dimensions and model parameters. They are of fixed value.

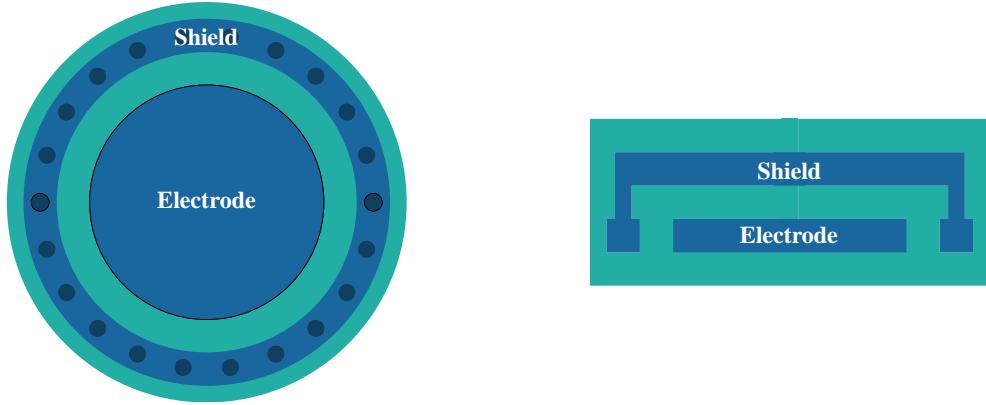


Figure 17: Electrode stack-up with shield ring; bottom view (left), side view (right)

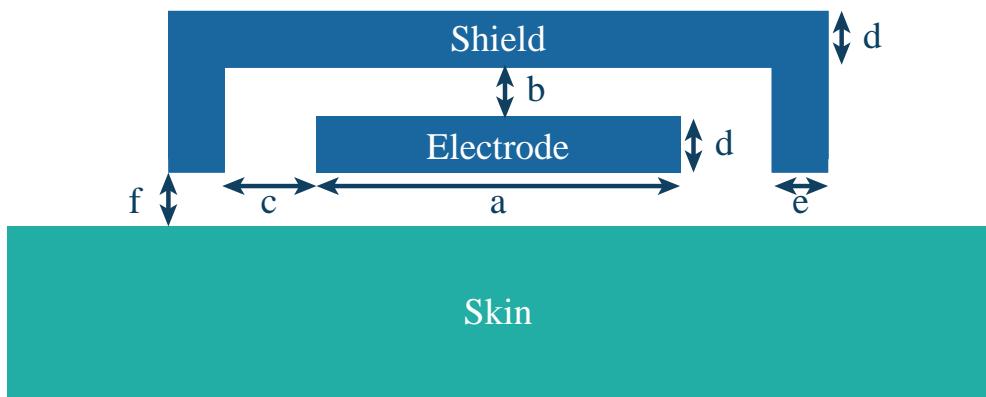


Figure 18: Diagram of the simulation setup

Name	Value	Description
a	5mm	Inner Electrode Radius
b	0.6mm	Distance Shield to Electrode (top)
c	0.5mm	Distance Shield to Electrode (side)
d	0.2mm	Copper thickness
e	0.6mm	Shield Ring Width
f	0.5mm	Electrode to Skin Distance

Table 1: Dimension values for the simulated electrode

4.1.4 Experimental Results QVAR Electrodes

To obtain real values, the different electrode designs were tested as described in section 4.1.2 by sticking them on the back of the hand as shown in figure 14. The electrodes have a length of 8cm and are 0.05mm^2 . In a first step, the background noise of all electrodes was measured. To get consistent results the ground of the development board was galvanically connected to the skin. The hand was still during the recording and all noise values are the RMS QVAR ADC values of a 60-second recording. The following table 2 shows the noise values of three circular electrodes, each are 18 mm in diameter. One is not shielded, one has only the electrode shielded, and one has the electrode and cable shielded.

	Noise [rms]	ADC LSB [bits]
Unshielded	178.24	7.48
Electrode Shielded	46.02	5.52
Electrode & Cable Shielded	34.65	5.11

Table 2: Noise values of different electrode shielding of QVAR test setup

Table 2 clearly shows how important shielding is when it comes to reducing noise. In addition, the shielding helps to ensure that the electrode only picks up the charges of the hand and not those of the environment. With an unshielded electrode, external motion was clearly measured in the 20 cm range, like a second person waving their hand.

To obtain a better insight on how the shielding works and get a visual indication for it. The shielded electrode was simulated as described in section 4.1.3. Figure 19 shows the simulation results. The left image shows three charges set to 10V placed around the electrode to simulate an external noise source. It can be clearly seen that the shielding helps to protect the electrode from this influence. The right image shows the same simulation, but the shield was moved above the noise charges, so it can't shield the electrode. Figure 20 shows the simulated region of sensitivity of the electrode

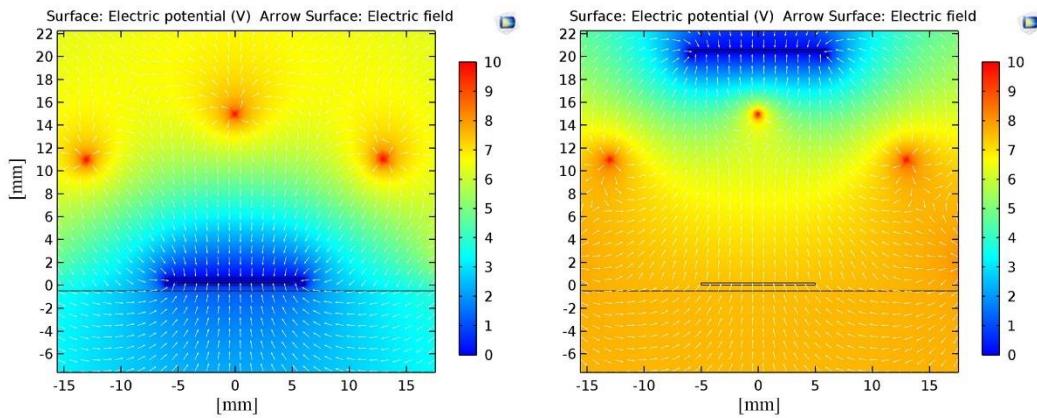


Figure 19: Electrode with shield between disturbance (left), electrode with no shield (right)

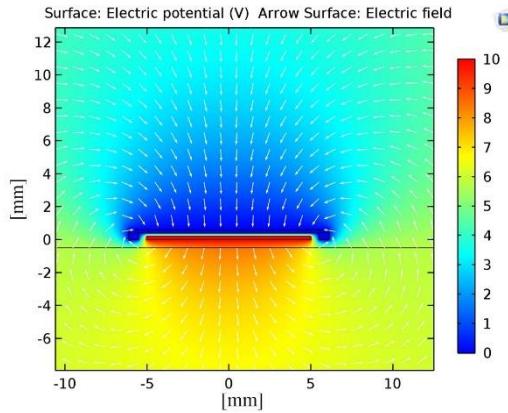


Figure 20: Simulated region of sensitivity of the electrode

Shielded electrodes of different sizes were measured. All electrodes were placed on the back of the hand as shown in figure 10 above. The wrist was bent to measure the signal strength at each electrode. For each electrode, the signal peak was measured 10 times and the average was taken. Table 3 shows the average QVAR ADC values for three different electrode sizes.

Electrode Diameter	Signal Strength	ADC LSB [bits]
8mm	386.61	8.59
14mm	664.98	9.38
18mm	827.36	9.69

Table 3: Signal strength for various electrode sizes

There is a clear correlation between the electrode size and the signal strength. This can be explained by the fact that a larger electrode has a larger capacity, which can store more charges. Therefore, the most important conclusion is to make the electrodes as large as possible and shield them as well as possible.

4.2 Improved Design

QVAR has the great advantage over sEMG that it is a capacitive measurement. This means that the electrodes do not need to have direct skin contact. With this advantage, the electrodes can be easily integrated and hidden in different materials. sEMG, on the other hand, requires good skin contact. This has the drawback that dry skin or hair will decrease the signal quality. To measure wrist movement with sEMG, the bracelet must be directly over the muscles, i.e., it must be on the back of the forearm near the elbow. With QVAR, we can bring the electrodes close to the wrist. This has the great benefit that it can now be easily integrated into any smartwatch, which increases the user experience and adoption.

Based on the results from Section 4.1, a final application was selected together with the two supervisors. The sEMG and QVAR parts are now split into two separate parts. The circuitry of the development board from section 4.1 will be implemented in a sEMG bracelet similar to the Myo bracelet by a research assistant at the Center for Project Based Learning at ETH Zurich and is not part of this thesis. The new goal is now to develop a stand-alone QVAR bracelet with the form factor of a watch. The goal is to have multiple QVAR channels that measure the movement of the hand. A machine learning model will be trained and implemented on the bracelet that will allow it to recognize multiple gestures performed by the user.

4.2.1 Overview

To meet the abovementioned criteria, additional QVAR channels must be added. The sEMG part is removed and the whole design is miniaturized to fit into a wristwatch. The resulting system design is explained in detail in section 5.

It consists of two parts. The first part is a rigid circuit board on which all the circuits and components are located. The second part is a wristband that is wrapped around the wearer's wrist and contains all the electrodes. The two parts are soldered together using castellated holes.



Figure 21: Rendering of the rigid PCB with 6x QVAR channels

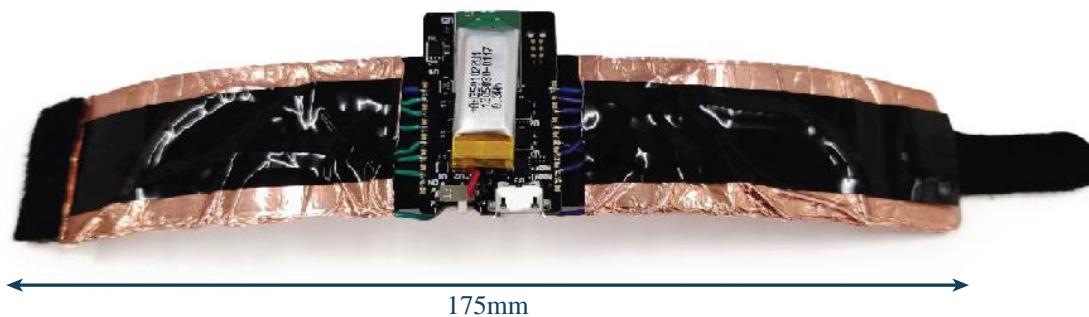


Figure 22: Image of the first bracelet prototype with 12 hand crafted electrodes

4.2.2 Electrode Strap

QVAR is a differential measurement, so each QVAR channel has one positive and one negative electrode. With six QVAR sensors, three electrode configurations are of interest. The first is when the positive and negative electrodes are adjacent to each other (**a**). The second is when the positive and negative electrodes are on the opposite side of the wrist (**b**). The third option is to combine all the negative electrodes into one and keep only the positive electrode separate (**c**). The three configurations are shown in Figure 23 below. Electrodes labelled as $Qx+$ are the positive electrode of QVAR channel X and $Qx-$ are the corresponding negative ones.

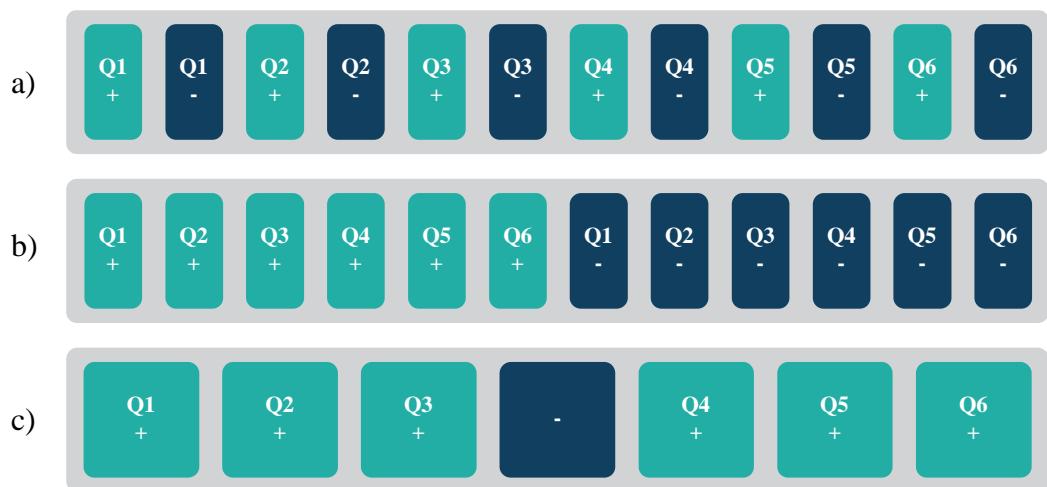


Figure 23: The three electrode configurations



Figure 24: Handmade electrode strap, with configuration b)

To test these three configurations, they were built by hand in the same way as described in Section 4.1.2. The resulting armband is shown in Figure 24. The electrode configurations were tested by performing the gestures that would later be used in the machine learning part. The goal was to find the configuration that provided a mixture of the most information and the best signal strength. The configuration where all negative electrodes are connected had the best signal quality. This can be explained by the fact that it has only seven electrodes compared to the 12 of the other two. This allows the electrodes to be larger, resulting in a stronger signal, as shown in Section 4.1.4. The disadvantage of this configuration is that the individual channels strongly influence each other and are therefore highly correlated. This means that a lot of features and information is lost. Figure 25 below shows a recording where the participant made a fist wearing configuration (b). Figure 26 shows a recording of the same gesture, but the participant was wearing configuration (c). The recordings show the raw sensor values for the six QVAR channels and are sampled with 240Hz. The signal in configuration (c) correlates between the channels, where for configuration (b) a different pattern in each channel is visible. No distinct exclusion criteria were found for configuration (a) and (b). Based on these findings, it was decided to choose both the electrode configuration (a) and (b). A flexible printed circuit board with these two configurations was designed, but in the end only configuration (b) was used. It is described in section 5.3.1.

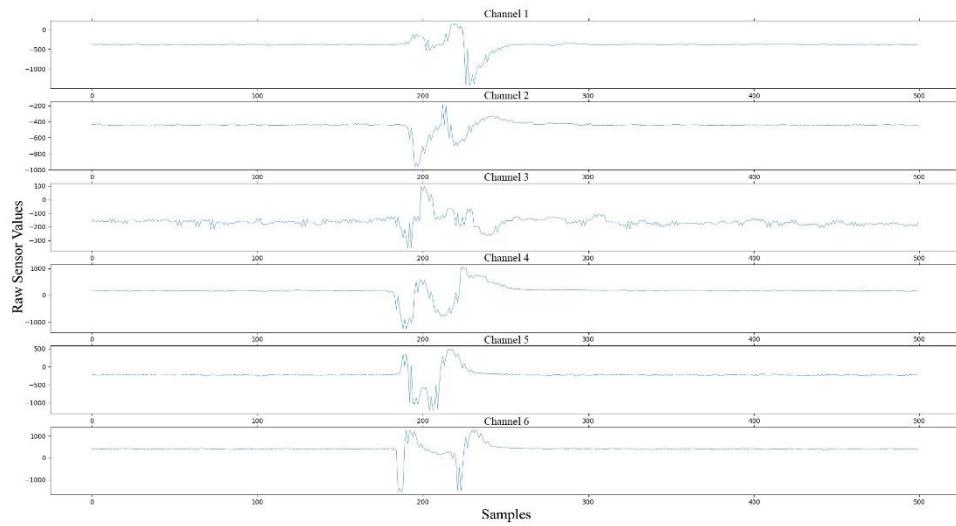
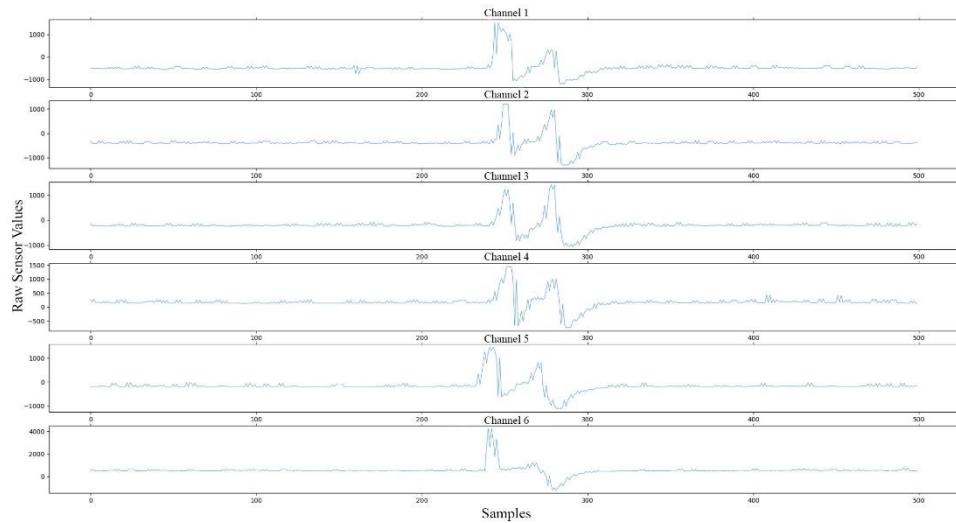


Figure 25: 2 second recording of participant making a fist, wearing electrode configuration (b)



*Figure 26: 2 second recording of participant making a fist,
wearing electrode configuration (b)*

5 Final System Architecture

5.1 Overview

In this section, the final system design of this wearable is presented. The overall form factor is designed to mimic a watch. This is one of the strengths of this design as it allows for simple integration with existing products. Making it easy and seamless to adopt into daily life.

5.2 Electrical System

5.2.1 nRF5340 / NORA-B106

The ability to stream raw data in real time over BLE while running a machine learning model in parallel with low power consumption are two requirements that the NORA-B106, a standalone BLE 5.2 low energy module from U-blox, meets perfectly. It is the brain of this system. It is equipped with the nRF5340 microcontroller from Nordic Semiconductor. This consists of two ARM Cortex-M33 processor cores with integrated flash and RAM memory. It has a footprint of 14.3 mm x 10.3 mm. And it does not require any external components.

The primary core is optimized for high performance and can operate at up to 128MHz. This core collects the data from the sensors and performs the inference on it. The secondary core is optimized for low power consumption and efficiency and runs at 64MHz. The main purpose of this core is to run the BLE stack, handling all wireless communication. This allows the main core to run without interruptions from BLE communication while maintaining high throughput. The main core has 1 MB of flash memory and 512 kB of RAM. Thereby allowing the core to run a TensorFlow Lite model on it. The advertised power consumption of the module at 3 V is shown in the table 4 below [38]. The application core of the nRF5340 runs at 128Mhz for reading sensor data and running the inference. The rest of the time the application core will be in a low power idle state. The network core runs the BLE stack at 64Mhz in 1Mbps phy.

Mode	Current Consumption
System Off, 0 kB application RAM, wake on reset	1.0 μ A
System ON, 64 kB network RAM, wake on network RTC	1.5 μ A
Application core running CoreMark benchmark @ 128 MHz	8 mA
Network core running CoreMark benchmark @ 64 MHz	2.6 mA
Radio RX only @ 1 Mbps Bluetooth LE mode	3.7 mA
Radio TX only, 0 dBm output power	4.1 mA

Table 4: Advertised current consumption of the NORA-B106 at 3V [38]

5.2.2 LSM6DSV16X IMU with QVAR

The LSM6DSV16X from ST Microelectronics is a novel system-in-package sensor. It integrates a digital 3D accelerometer, a digital 3D gyroscope and electrical charge variation detection (QVAR). It also has several features that enable on-device data filtering. It includes a finite-state machine and an integrated machine learning core.

Unfortunately, only single-channel QVAR sensors are available at the time of writing. Therefore, six of these sensors are present in the final design. This allows for six parallel QVAR channels. Since all IMUs are mounted on the same rigid PCB, they are all subject to the same acceleration and rotation rate changes. Therefore, only the accelerometer and gyroscope data from one IMU is used. For the other five, only the QVAR and accelerometer channels are turned on to save power. It is important to note that the QVAR functionality only works when the accelerometer is set to high power mode. In addition, QVAR is only supported in I2C and there are only two selectable addresses. This means that if more than two LSM6DSV16X are used, the microcontroller must have multiple I2C peripherals, or a bus expander must be used.

The LSM6DSV16X is running the accelerometer, gyroscope and QVAR in high-power mode. The acceleration measurement range was set to $\pm 2g$ and the rotation rate sensitivity to ± 250 dps. This results in a sensitivity of 0.061 mg/LSB for the accelerometer and 8.75 mdps/LSB for the gyroscope. The QVAR channels are configured to have an input impedance of 300 M Ω . All values are transmitted via I2C in 16bit and with a sampling rate of 240Hz. The following table 5 shows the advertised current consumption of the LSM6DSV16X in various operating modes at 1.8V [39].

Mode	Current Consumption
Accelerometer & Gyroscope in high-power mode	0.65 mA
Accelerometer, Gyroscope & QVAR in high-power mode	0.665 mA
Accelerometer in high-power mode	0.19 mA
Accelerometer & QVAR in high-power mode	0.205 mA
Accelerometer in normal mode	0.1 mA
Accelerometer in low-power mode	0.02 mA

Table 5: Advertised current consumption of the LSM6DSV16X at 1.8V [39]

5.2.3 Power

The system is powered by the 75mA/h Li-Po battery ICP501022UPM, which can be charged with the on-board charging IC LM3658SD via a micro-USB port. The NORA-B106 and LSM6DSV16X are both powered at 1.8V. The analog circuits of the QVAR channels are very sensitive to noise at the supply level. Since digital circuits are known to generate a lot of noise, two separate power domains must be supplied. This means that each LSM6DSV16X must be powered by a digital and an analog power supply. Both the digital and analog supply voltages are generated by two NCP110AMX180TBG low drop-out (LDO) regulators. Since the efficiency of an LDO depends on its voltage drop across, this must be minimized to ensure high efficiency. High efficiency is achieved by the MAX38640A low-noise buck converter. It steps down the battery voltage to 2 V, resulting in a voltage drop of only 200 mV across the two LDOs. The entire system can be switched on or off by the user via a slide switch.

5.3 Hardware Architecture

5.3.1 PCBs & Capacitive Electrodes

The bracelet consists of two printed circuit boards (PCB). These are soldered together via castellated holes. The first is a rigid 4-layer PCB that houses all the electronic components. The second is a 4-layer flexible PCB that forms the band that is wrapped around the wrist. The bottom of the flex PCB houses all the electrodes, while the top has solder pads for connection to the rigid PCB and additional ground pads (GND) for any future modifications. Figures 27 and 28 below show the rendering of the two PCBs. In figure 28, Q1+ corresponds to the positive electrode for QVAR channel 1, while Q1- is the corresponding negative electrode.

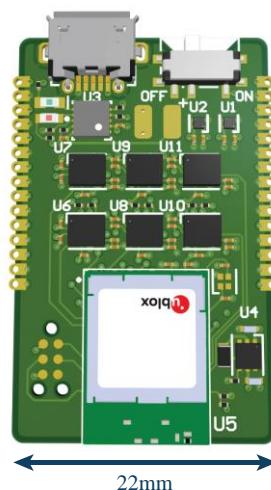


Figure 27: Rendering of the rigid PCB with 6x QVAR channels

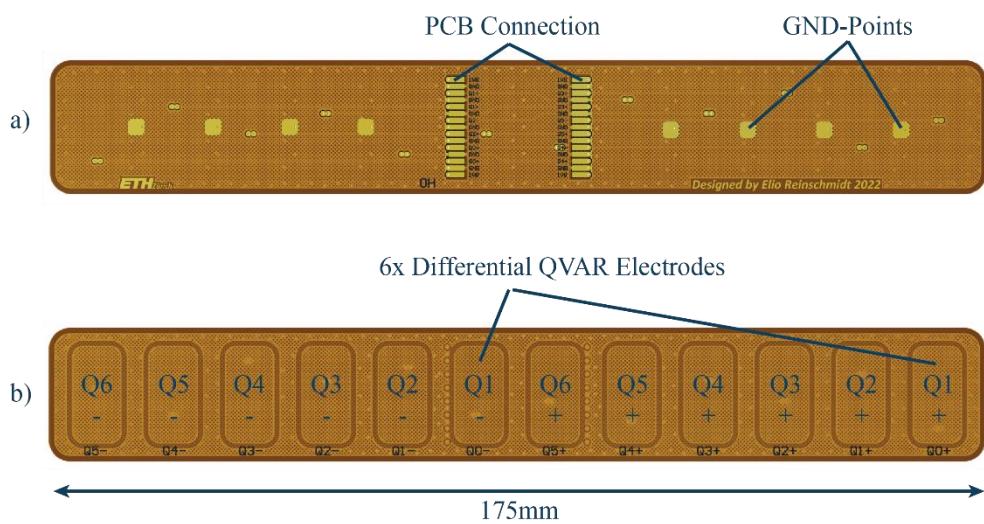


Figure 28: Top (a) and bottom (b) rendering of the flex PCB

As described in section 4.1.4, the QVAR electrodes are huge "antennas" that are very susceptible to external influences. Therefore, the electrodes and the traces must be properly shielded. This is achieved by the 4-layer stack up of the flexible PCB. The top layer consists of a large ground plane, which serves as a shield for the traces. All traces run on the second layer below. The third layer is again a large ground plane to shield the individual traces from the electrodes below. The electrodes are located on the 4th and last layer. They are surrounded by an additional ground plane, as described in section 4.2.2, to direct the sensitivity of the electrodes. The flexible PCB is subjected to mechanical stress due to constant bending when the bracelet is put on and taken off. This has the disadvantage that the copper in the PCB becomes hard and brittle and breaks due to work hardening. To reduce this effect and increase the longevity of the component, all copper ground planes and electrodes were hatched. It was ensured that all traces are mainly perpendicular to the bend. All vias are doubled and the connection to the trace is teardrop shaped.

5.3.2 Mechanical Construction



Figure 29: Top view of final bracelet with the 3D printed enclosure

The mechanical design is simple and consists of four main components. The first component is a rigid PCB that contains all the electronic components and circuits. The second component is a flexible PCB that serves as a strap and holds all the electrodes. These two components have already been described in the previous section. The third component is a 3D-printed enclosure that covers all the electronic components and protects them from the environment. It has two cut-outs to allow access to the on/off switch and the micro-USB charging port. Finally, the armband is attached to the user's wrist with a Velcro strap. The final bracelet is depicted in figure 29.

5.4 Firmware Architecture

5.4.1 Firmware Overview

One of Zephyr's strengths is that it is a full RTOS that allows easy scheduling between multiple threads. Seven threads are of particular interest. The first six threads each represent an LSM6DSV16X sensor. Their task is to read and preprocess the data from it. The data is continuously read and compressed before being stored in a ring buffer to be used later by another thread for gesture recognition. The data from the six LSM6DSV16X are independently polled. In addition, the thread sends the raw data over BLE when a client enables notification for the specific sensor characteristic. The seventh thread is responsible for gesture recognition. It takes the compressed data from a ring buffer every 16ms and checks if an event has occurred, by checking if the new value is 5% greater than the moving average of the previous eight values. This is done to check if enough movement for a gesture is present. If such an event is detected, it extracts the features from the data and runs an inference on them. Otherwise the thread will pause to save power. When a gesture is detected, it sends it to the client via BLE.

The final firmware uses 338,632 KB of flash and 176,913 KB of SRAM, accounting for 32.29% of the total flash and 38.56% of the total SRAM. When continuously reading the QVAR data from the six IMUs and performing an inference every two seconds, the CPU utilization is about 5% and 95% is spent idle.

5.4.2 Bluetooth Low Energy (BLE)

The bracelet takes the role of the peripheral and the user application takes the role of the client. When the bracelet is turned on, it will continuously advertise until a client connects to it. After the connection is established, the client can enable notification for multiple characteristics. Three services are available to the client. The first is the IMU service, which consists of six individual characteristics for each axis of the accelerometer and gyroscope. The second is the QVAR service, which consists of one characteristic for each of the six channels. When the client enables notification for one or more of these characteristics, the bracelet starts sending the raw data to the client continuously via BLE. The third service is the gesture service, which sends the predicted gesture label when a gesture is detected.

A sent packet of the gesture service consists of a single byte containing one number. It is sent only when a gesture has been detected by the inference thread. For characteristics in the IMU and QVAR service, a packet consists of 30 samples of 16-bit integers. This results in a packet length of 60 bytes. The data is sent with notifications and if enabled, the client is notified every 125ms. The following table 6 lists the characteristic UUIDs.

Characteristic	UUID
X-Axis Accelerometer:	3da22dc6-70d7-4217-9bb2-de5d79560001
Y-Axis Accelerometer:	3da22dc6-70d7-4217-9bb2-de5d79560002
Z-Axis Accelerometer:	3da22dc6-70d7-4217-9bb2-de5d79560003
X-Axis Gyroscope:	3da22dc6-70d7-4217-9bb2-de5d79560011
Y-Axis Gyroscope:	3da22dc6-70d7-4217-9bb2-de5d79560012
Z-Axis Gyroscope:	3da22dc6-70d7-4217-9bb2-de5d79560013
QVAR Channel 1:	d4eb1a81-2444-4d16-993e-4d28fe2c0001
QVAR Channel 2:	d4eb1a81-2444-4d16-993e-4d28fe2c0002
QVAR Channel 3:	d4eb1a81-2444-4d16-993e-4d28fe2c0003
QVAR Channel 4:	d4eb1a81-2444-4d16-993e-4d28fe2c0004
QVAR Channel 5:	d4eb1a81-2444-4d16-993e-4d28fe2c0005
Gesture Label:	b2d8ee01-5e36-4d96-86f7-5269e4820363

Table 6: Characteristic UUIDs

5.4.3 TensorFlow Lite for Microcontrollers

As introduced in Section 2.3.4, TensorFlow Lite is a precompiled module in Zephyr that allows it to work out of the box. Since TensorFlow Lite is based on C++ and the rest of the code is based on C, these two need to be merged. To make development easier, Zephyr has a HelloWorld example that presents the user with a project already set up that merges the two. This example was the starting point for the firmware. The trained machine learning model described in Section 7 is quantized from FLOAT32 to INT8 this reduces the model size to 47.99kB. It was converted to a Flatbuffer binary and saved in an .h file. TensorFlow Lite for Microcontrollers is interpreter-based, meaning that it is responsible for loading the model from the Flatbuffer and performing inference on it. The machine learning model and the TensorFlow Lite conversion are described in more detail in Section 7.

6 Gesture Dataset

To train a machine learning model, a custom data set must be collected. Participants wore the bracelet described in Section 5. Their movements were recorded with the bracelet while they performed the gestures described in Section 6.1. The QVAR dataset consists of a set containing the recordings of 20 individual participants, where every gesture was performed 30 times.

6.1 Gestures

To get a good understanding of performance, 10 gestures were defined. The goal was to find a mix of gestures where some are large movements, such as bending the wrist, while others are more subtle small finger movements, also known as micro-gestures, such as pinching the index finger with the thumb. For the gestures involving mainly wrist movement, we also made sure that the movements in both directions (e.g., left and right circles) were different gestures to see if the wristband could distinguish the direction. Figure 30 below shows the start, path and end positions of the gesture, followed by a detailed description of the gestures. All gestures were performed with the participants' right hand.

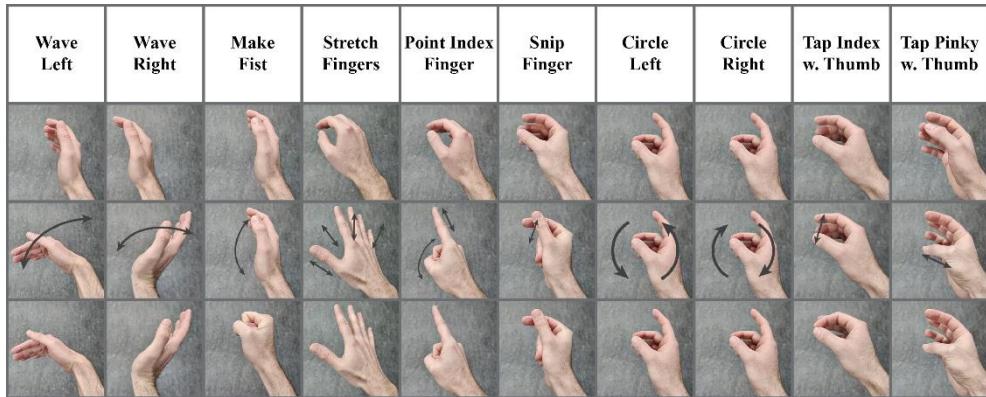


Figure 30: Start, path and end positions of the 10 gestures

Wrist Extension & Flexion (wave left, wave right): Extending and flexing the wrist are large movements that produce a strong signal. Participants were asked to remain in a neutral position and perform the movement once (e.g., extend the wrist) and then return to the neutral position.

Open, Close Hand & Extending Index Finger (make fist, stretch fingers, point index finger): Participants started with a neutral/relaxed hand position before and after the

gesture. When opening the hand, participants were asked to slightly hyperextend the fingers. When closing the hand, they were asked to tighten the fist slightly. This leads to a stronger and clearer signal for training. Since closing the hand with the index finger extended is a mixture of opening and closing the hand, they were asked to make sure to extend the index finger and tighten the partial fist.

Make Left & Right Circle (*circle left, circle right*): For these gestures, participants were asked to move their wrist in a circle. Drawing a left circle once and a right circle once. Participants begin and end in a neutral/relaxed position.

Snip Finger: Participants snapped their thumb and middle finger. They were asked to minimize the other movements of the hand in order to isolate the "snapping". The hand was in a neutral/relaxed position prior to snapping. To isolate the snapping, the return to a relaxed position was not recorded.

Pinch Index, Little Finger & Thumb (*tap index w. thumb, tap pinky w. thumb*): These two gestures are also called micro-gestures. The reason for this is that they only require small movements of the finger and are usually performed faster because the path of the finger is shorter compared to the other gestures. Therefore, in theory, these two gestures should be the most difficult to recognize. The hand is in a neutral/relaxed position before and after the movement. In between, only a short, small pinch was performed. Participants were asked to pinch with slight force to produce a clean, strong signal.

6.2 Recording Application

To reduce user errors when recording and labelling data, a custom GUI recording application was written in Python. The application consists of a dropdown menu (1) to select the COM port of the BLE dongle (nRF52840-Dongle by Nordic Semiconductor) that connects to the bracelet. A connect button (2) to start pairing with the bracelet and subscribing to all characteristics. A live recording (3) of QVAR channel 1 is displayed on the top left. This is mainly to ensure a good connection and for troubleshooting. In the middle are the gesture name (4) and the images (5) for start, path and end of the gestures. They always show the currently selected gesture. An additional drop-down menu (6) is used to select the gesture to be recorded. Two recording modes are available. Auto mode (7) starts a recording sequence that allows 10 consecutive recordings. When the button is pressed, a three-second timer on a red background starts counting down. When it ends, the countdown (8) turns green, and the recording begins. It will record for three seconds before turning red again. This cycle is now repeated nine more times. Single mode (9) works like Auto mode, but only records once.

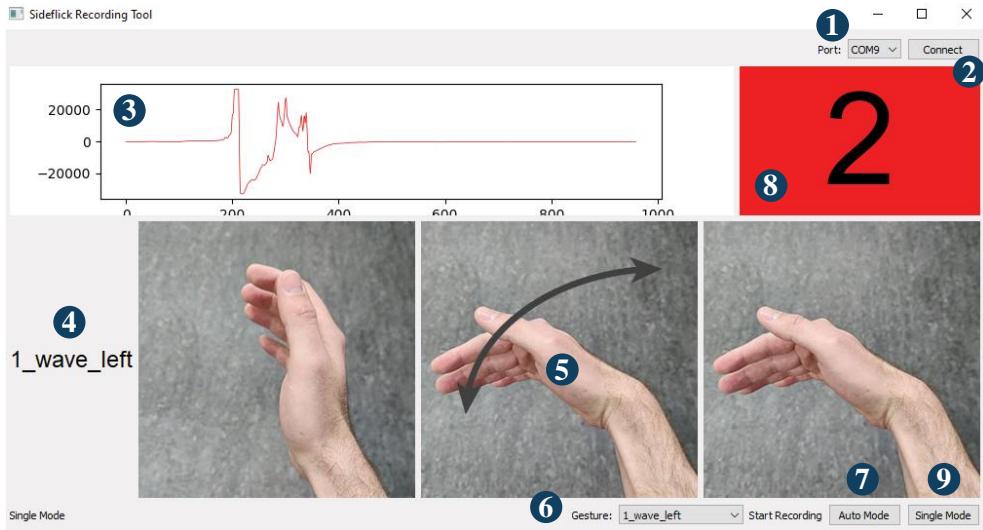


Figure 31: Screenshot of the Recording application

Since BLE is asynchronous by nature, the data packets are not received in the correct order and at the same time. This means that all data must be synchronized afterwards after recording. Each packet sent by the bracelet consists of the system time of the bracelet immediately before it was sent. After a three second recording, all received packets are synchronized with this time stamp. If a packet is lost, the recording is discarded, and the user is informed that they must repeat this one gesture. The synchronized recording is stored in a JSON file along with the date, time, and gesture name.

This recording application allows a large number of gestures to be recorded quickly with minimal chance of user error. The only user error that can occur is if the bracelet is worn incorrectly, the wrong gesture label is selected, or the participant forgets to perform the gesture while in auto mode.

6.3 Recording Setup

Before the data set was collected, participants were informed and guided through the process. The bracelet was disinfected and placed around the wearer's right wrist. It is worn like a watch. The supervisor explains and practices each gesture with the participant. After verifying that the gesture is performed correctly, the recording procedure begins. Each gesture is recorded 30 times for each participant using the custom recording application described in Section 6.2. If no errors have occurred, the next gesture is explained and practiced before recording continues. The fastening position for the QVAR bracelet is shown in Figure 32.

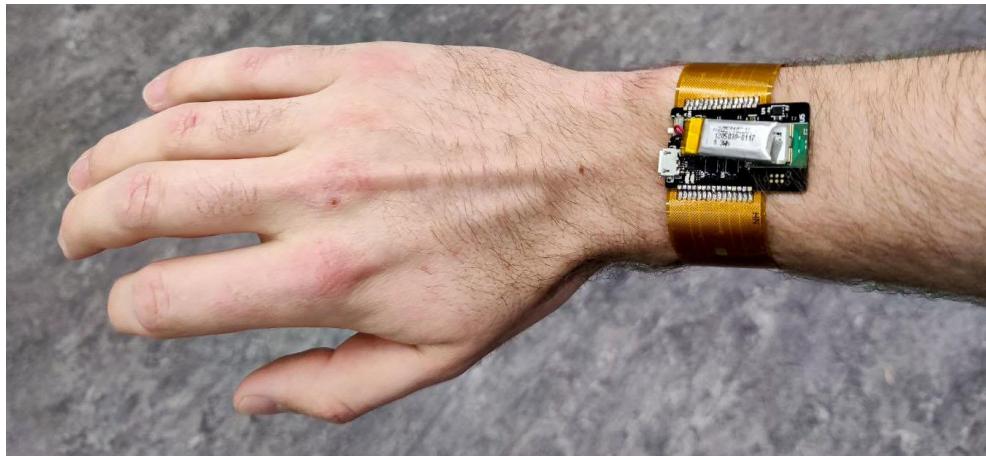


Figure 32: Fastening position of the QVAR bracelet

6.4 Dataset Structure

The QVAR data set includes 10 gestures recorded 30 times per participant. There is a total of 20 individuals. The recordings are stored in a JSON file format. Each participant is stored separately in a folder. The naming scheme for the files is the following:

```
"[YearMonthDay]-[HourMinuteSeconds]_recording_[GestureName].json"
```

The JSON file is structured the following way:

```
JSON
  file_name:          // Name of this file
  gesture:            // Name of the gesture
  recording_time:    // Time of recording
  file_path:          // Path where original file is stored
  recording_length:  // Length of recording in seconds
  error_found:        // True, if data is missing in one channel
  acc_x:              // Data of Accelerometer X-Axis
  acc_y:              // Data of Accelerometer Y-Axis
  acc_z:              // Data of Accelerometer Z-Axis
  gyro_x:             // Data of Gyroscope X-Axis
  gyro_y:             // Data of Gyroscope Y-Axis
  gyro_z:             // Data of Gyroscope Z-Axis
  qvar_ch0:            // Data of QVAR channel 1
  qvar_ch1:            // Data of QVAR channel 2
  qvar_ch2:            // Data of QVAR channel 3
  qvar_ch3:            // Data of QVAR channel 4
  qvar_ch4:            // Data of QVAR channel 5
  qvar_ch5:            // Data of QVAR channel 6
```

7 Gesture Recognition

For gesture recognition, Deep Learning has gained popularity in the last decade. It is an efficient method to classify data and find relationships and patterns in it. To increase the efficiency and effectiveness of the model, the data should be pre-processed first. The idea is to clean the recordings by filtering it and to extract the gesture event by cropping it out of the recording. After pre-processing, the desired features can be extracted to train the model.

7.1 Data Preprocessing, Augmentation & Feature Extraction

The dataset described in Section 6 consists of three-second recordings stored in individual JSON files. Theoretically, we could simply use the entire raw three-second recording directly as input to our model. However, this would be ineffective because it contains a lot of redundant data that can be removed. This reduces the model size and the time to compute an inference. At the beginning and end of each recording, the participant's hand was in a rest position and in between was the movement of the gesture. This period of motion is about one second long, depending on the gesture, and forms the window of interest. Figure 33 shows a recording of the wave left gesture with the window of interest highlighted.

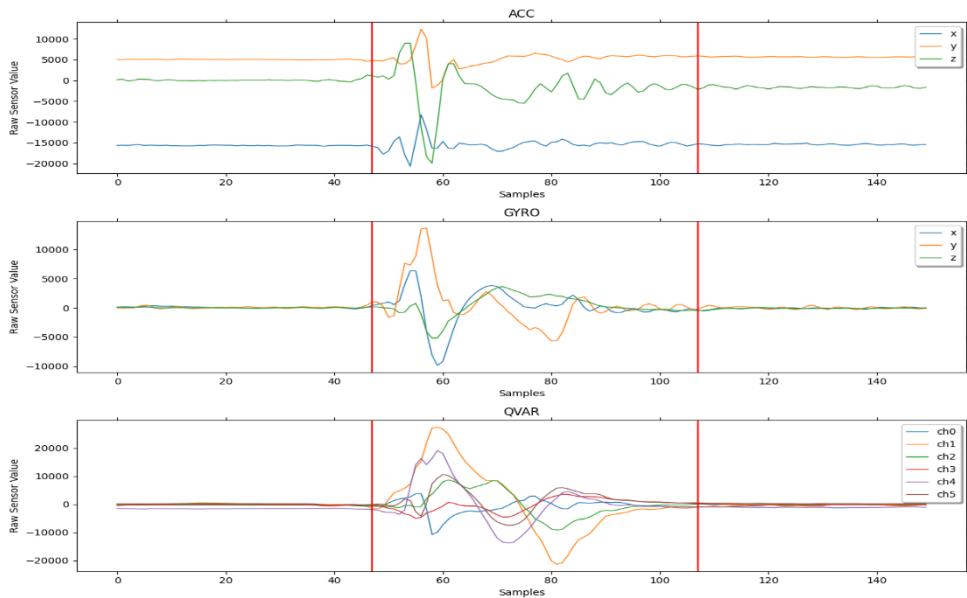


Figure 33: Raw recording of wave left gesture with highlighted window of interest

The first step of pre-processing is to extract this window of interest and remove the beginning and end of the recording. Since the beginning and end of the recording do not contain motion, a threshold detection algorithm was used. The moving average of the QVAR signal is taken, and the threshold is triggered when in one of the six channels the value of the signal above the threshold trigger value is. The moment after the threshold is crossed forms the recording window of interest and it is one second long, and everything before and after is deleted.

Our windowed signal consists of 12 channels (3x accelerometer, 3x gyroscope, 6x QVAR), each with 240 sample points. To reduce computational demand for the feature extraction on the microcontroller and to remove high frequency components the signal is down sampled by a factor of four by taking the average over four samples. The resulting signal now consists of 12 channels of 60 samples each. Figure 34 shows the final signal used for feature extraction. It is the window of interest from figure 33 and shows the wave left gesture.

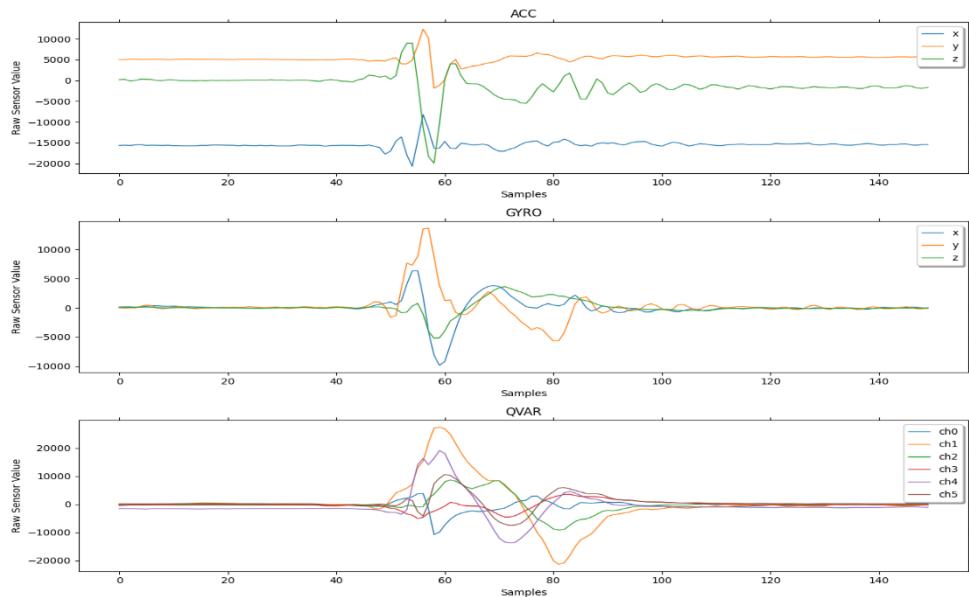


Figure 34: Window of interest, cut out from figure 33

In machine learning, the amount of training data plays an important role. With 20 participants, the network will have the problem of overfitting. This happens when the network is large enough to remember the records instead of generalizing to learn what a gesture is. There are three ways to address this problem. The most trivial solution would be to reduce the size of the model. This avoids overfitting, but at the same time reduces the number of features and gestures that a network can learn. The second option is to include bias regularization and dropout layers. The third option is to artificially increase the amount of training data available. This artificial increase in the size of the training

data is called data augmentation, where the augmented data are lightly modified copies of the original recordings. Each recording is different in shape and size from the others because each user performs the gestures differently and in a different environment. For example, an optimal network should not be affected by a slight change in the amplitude of the signal or if it is a little noisier. We can use this to generate new data for our model by slightly changing the shape and size of the original recording.

Each recording is augmented in the following way. First, the amplitude is increased or decreased by multiplying it by a constant factor. The factor is chosen randomly and is between 0 and 3. The second augmentation consists of adding random noise to the signal. The third augmentation is to speed up the signal by compressing it, or to speed it up by stretching it. The last augmentation is to move the window of interest forward or backward in the recording. A possible example of an amplified recording might be, "*Amplify by a factor of 2, add noise, speed up by compressing by a factor of 2, and shift the window of interest backward by 100 ms.*" These enhancements are combined multiple times with different random factors to produce new data that is slightly different from the original. Thus, the training dataset can be enlarged from 4,000 individual recordings to 1,680,000 individual recordings.

Several features were tested and compared, and in the end two features in the time domain were selected to be used. These are the mean absolute value (MAV) and the mean value (MV). They are defined in the formulas 2 and 3 below, where N is the length of the segment and x_i is the signal at point i.

The window of interest is split into 10 segments and for every segment the MAV and MV are calculated. This is done separately for the individual accelerometer and QVAR channels. The gyroscope is not used. This results in 9 channels used, one for each axis of the accelerometer and the six QVAR ones. This results in a final model input vector size of 180 (9 channels * 10 segments * 2 features). Figure 35 shows the structure of the input vector in more detail.

$$MAV = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (2)$$

$$MV = \frac{1}{N} \sum_{i=1}^N x_i \quad (3)$$

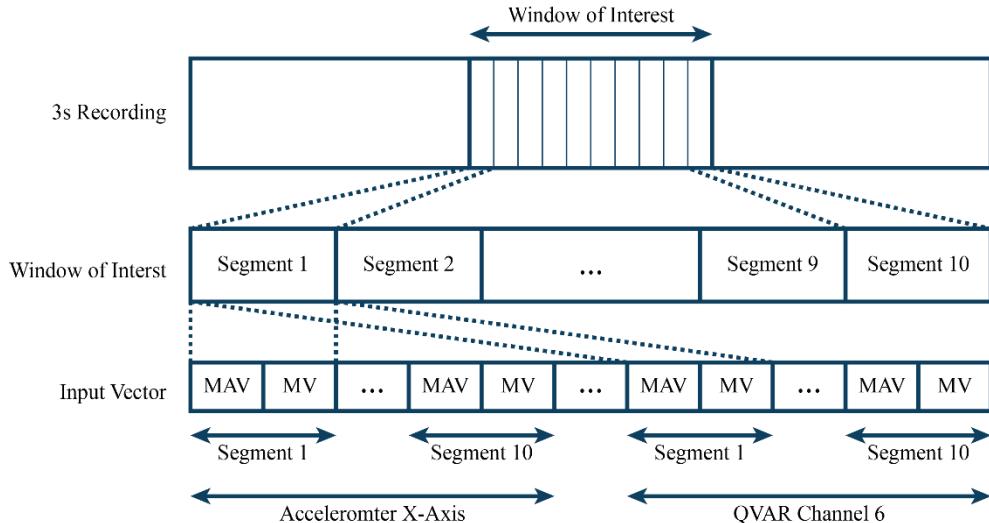


Figure 35: Detailed structure of the input vector

7.2 Neural Network Architecture (SideflickNN)

The universal approximation theorem [12] states that an artificial neural network (ANN) is theoretically a universal function approximator and therefore should be able to classify our gestures. However, it says nothing about the number of neurons and layers a network should have to achieve such results. To find a good approximation for our network parameters, several configurations were tested and evaluated with 10-fold cross-validation. The resulting network consists of an input layer, followed by three dense layers and a final output layer. The ReLU activation function was used in all layers and the SoftMax was used for the output layer. Since only a small dataset with 20 participants was available for training, overfitting is a major problem. To address this, all layers have a bias regularization of 0.01 and a dropout layer with a dropout rate of 25% was used.

The first component of our network is the input layer, which has an input size of 180. It is followed by a dense layer with 100 neurons. To increase the stability of the network, a batch normalization layer is used followed by a dense layer that consists of 100 neurons and one with 80 neurons. A dropout layer is used before adding one dense layer with 80 neurons. The final component of our network is the output layer, which consists of 10 neurons, one neuron for each gesture class. The following figure 36 shows the structure of the network.

This results in a network size of 43'570 parameters. Before the model was used on the microcontroller, it was converted to a TensorFlow Lite model and the network was quantized to INT8. The quantized model has a final size of 47.99kB.

Sequential Model - Fully Connected

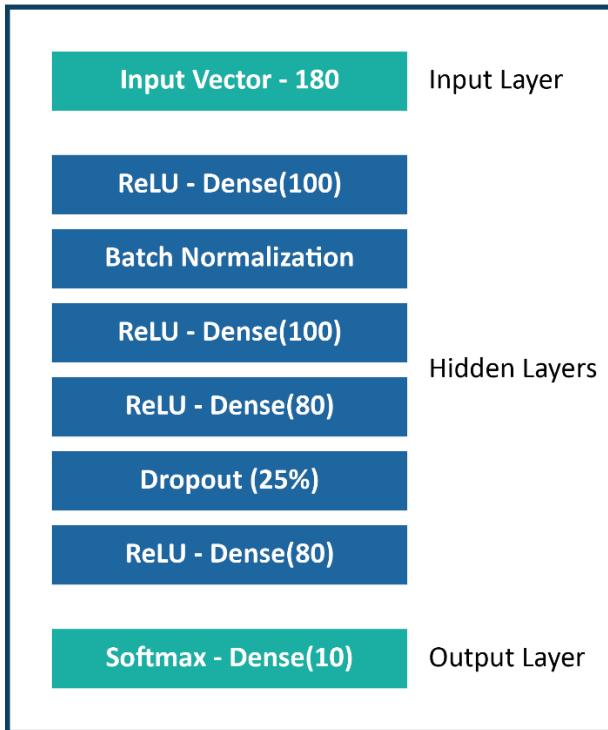


Figure 36: Model architecture

8 Experimental Results

8.1 QVAR Bracelet Performance

8.1.1 Power Consumption

Power consumption and thus battery life is an important parameter for any portable device. To measure the current consumption, the battery was removed, and the device was connected to the Nordic Power Profiling Kit II [40]. It is capable of up to 100k samples per second and can continuously measure from sub- μ A to 1A current. This allows continuous measurement from sleep mode to streaming data over BLE. Table 7 below shows two average currents. The first value is for continuous streaming of raw QVAR and IMU data over BLE. In the second case, the device performs an inference every second and sends the detected gesture label over BLE. Both measurements are performed at a nominal battery voltage of 3.7 V and represent the average over 60 seconds. Battery life was calculated for the built-in battery with a capacity of 75 mAh.

	Voltage	Average Current	Battery Life
Data Streaming	3.7 V	5.47 mA	13h 40min
Running Inference	3.7 V	6.52 mA	11h 30min

Table 7: Average current consumption

8.1.2 QVAR Measurements of the Final Bracelet

The signal-to-noise ratio (SNR) is defined as the ratio of the signal power to the noise power. It provides a good understanding of the quality of the signal by comparing the desired signal to the noise floor. To calculate the SNR of the system, the noise floor N_{rms} must first be measured. To obtain meaningful values from the field, all tests of the QVAR bracelet were performed while the bracelet was worn. Figure 32 shows the test setup of the QVAR bracelet. The measured noise for each channel is shown in Table 8 below. All noise values are the root mean square (RMS) of a 60 second measurement where the bracelet was worn but there was no movement. The strength of the QVAR signal is more difficult to determine because it depends on the motion performed. Large movements produce a stronger signal and therefore have a better SNR than small movements. Therefore, two values are used, one is the average RMS signal strength $S_{avg,rms} = 3488.39$ observed in the gesture dataset, and the other is the maximum signal strength $S_{max,rms} = 17675.14$ that can be measured. All values are the raw ADC values of the sensor and are not converted to a voltage. The signal-to-noise ratio for each channel is calculated using

the below formula, where S is the signal strength and N is the noise floor. They are depicted in table 8 below.

$$S_{\text{avg,rms}} = 3488.39, S_{\text{max,rms}} = 17675.14$$

$$SNR = 20 * \log_{10} \frac{S}{N}$$

	Noise [rms]	SNR_{avg} [dB]	SNR_{max} [dB]
Channel 1	30.92	41.05	55.14
Channel 2	26.12	42.51	56.60
Channel 3	17.71	45.89	59.98
Channel 4	22.82	43.69	57.78
Channel 5	21.75	44.10	58.19
Channel 6	41.93	38.40	52.49

Table 8: QVAR channel noise level and SNR

As described in Section 4.1.4, the noise and SNR of a QVAR channel are strongly dependent on the type of electrode and its shielding. With an average noise floor $N_{\text{rms,avg}}$ of 26.88 over all channels, we can conclude that the 4-layer flex PCB is sufficient to shield all electrodes and cables from the environment. With an average SNR of 42.61 over all six channels, we can further conclude that no additional digital filtering of the signal is required for this application, as the characteristics of the signal are clearly visible. If an application requires a higher SNR, additional digital filters are recommended.

Figure 37 below gives a visual indication of the noise floor of the bracelet. It is a 4-second recording window of the noise measurement. Figure 38 gives a visual indication of the signal strength. It is a recording of the left wave gesture from the data set described in Section 6.

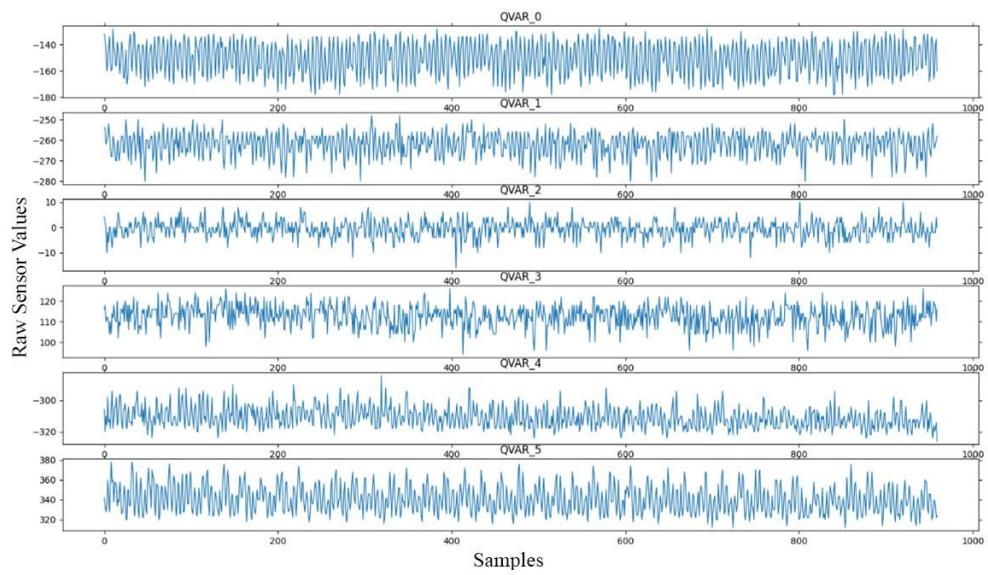


Figure 37: 4 second recording of the QVAR noise floor

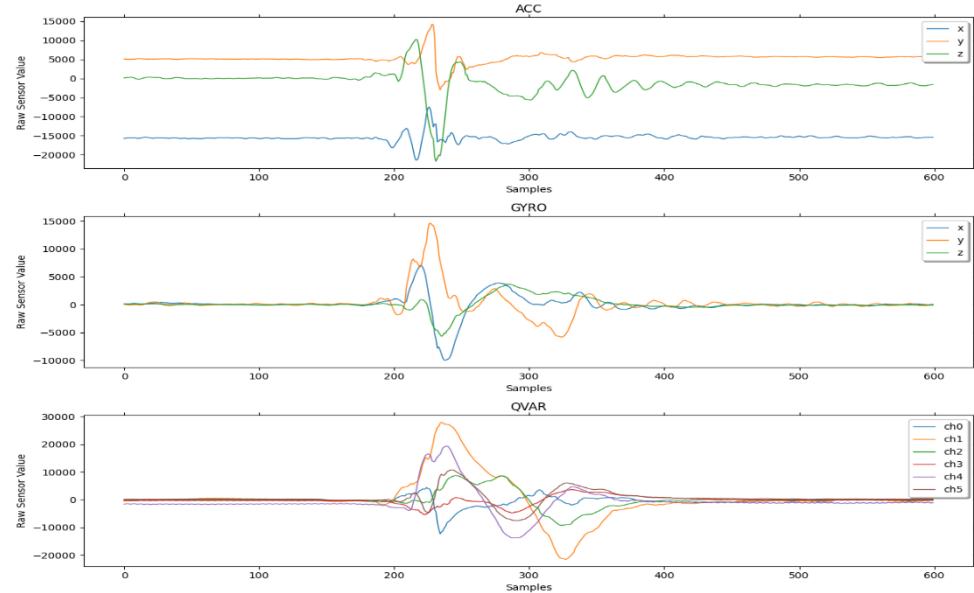


Figure 38: A raw 3s recording of the gesture wave left made with the bracelet

8.2 Gesture Recognition

8.2.1 Accuracy Evaluation

To get a feel for the performance of a neural network, accuracy is an important metric to consider. In a classification problem, accuracy is described as the number of correct predictions divided by the number of total predictions and is described by the following formula. In a multiclass prediction, the confusion matrix provides a good illustration of the problem. It compares the predictions of the machine learning model to a ground truth and provides information about which classes are "confusing" and are misclassified as others.

$$\text{Accuracy} = \frac{\text{number of Correct Predictions}}{\text{number of Total Predictions}}$$

In general, the accuracy of a model depends on the number of parameters that can be trained. A model with more parameters is theoretically able to learn more features than a similar model with fewer. Since there is no free lunch, a larger model also means that it takes up more space in memory and may take longer to compute. A balance must be struck between model size and accuracy to ensure optimal performance. In an embedded system, the size that a model can occupy is usually given by the available memory of the microcontroller, which provides an upper bound on the size. The nRF5340 has enough memory with 1MB available, in this case the processing power of the processor plays a bigger limit as it affects the inference time. The time needed for the execution of an inference is limited by the application.

In this case, classification should be possible in real time, i.e. the execution time should be so fast that the user does not notice it. This is because a noticeable delay would impair immersion and the user experience. The human reaction time for visual input is about 200 ms [41]. As a guideline for an upper limit for the inference time, 50 ms was chosen, since this is significantly below the human reaction time. This guarantees a fast response time with an acceptable inference time for the user. In addition, lower inference time means that the device can spend more time in a low-power state, which reduces the overall power consumption of the system.

As of writing no work has yet been done, which combines QVAR and machine learning. Therefore, no network architecture exists for reference. To find an optimal network topology different combination of ANNs were tested ranging from 3 to 5 layers. Each layer can have values ranging from 10 to 150 neurons. Every network is trained on the

augmented training data set with 10-fold cross validation and a batch size of 100 for 250 epochs, where 25% of the training set was used for validation. The results are depicted in figure 39. In figure 39 below, a clear trend can be seen where network size correlates positively with accuracy. Initially, a small increase in the size of the network can produce a large increase in accuracy, but as the size of the network increases, a point is reached where the yield decreases.

The final model size is a trade-off between speed and accuracy. A balance is reached at about 45,000 parameters, resulting in a model accuracy of 87.24%. Increasing the size of the model from this point only yields diminishing returns, since even a model three times as large only yields an accuracy increase of less than 1%, and the problem of overfitting the network increases with the size of the network. Moreover, the unquantized model has an inference time of 33.267 ms on the microcontroller, which satisfies our condition. The performance of the selected network is described in Section 8.2.2. To further increase the performance, the network was quantized before being deployed on the microcontroller, which further reduced the inference time; this is described in Section 8.2.3.

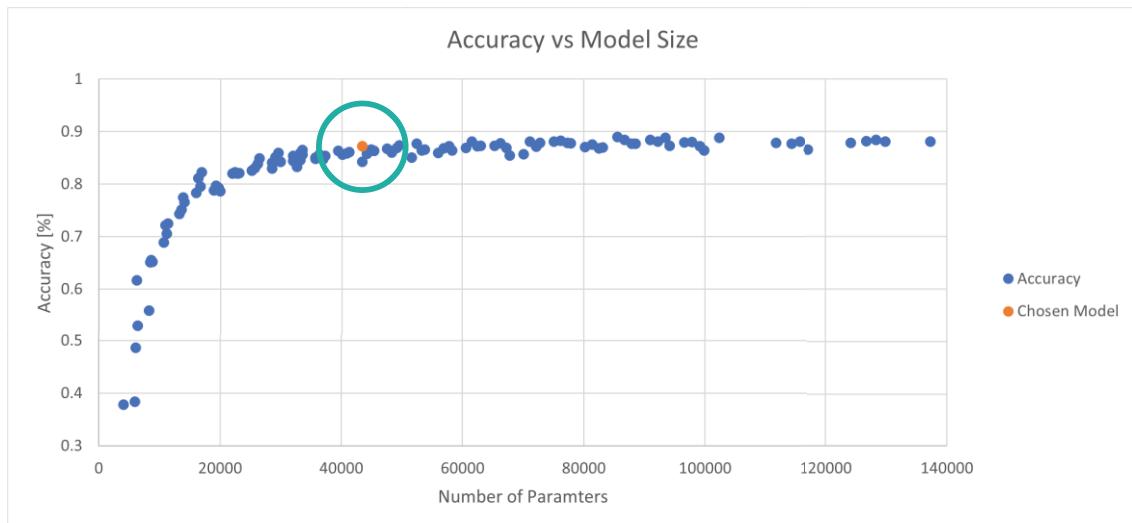


Figure 39: Accuracy vs model size, with highlighted chosen model

8.2.2 SideflickNN Performance

The final artificial neural network described in Section 7.2 has a final size of 43'570 parameters and consists of four dense, interconnected layers. It was trained using the recorded data set, with one-third of the data used for testing and the rest for training. The network was trained with 10-fold cross validation and a batch size of 100 for 250 epochs, where 25% of the training set was used for validation. It is able to recognize gestures with an overall accuracy of 87.24%. Figure 40 shows the confusion matrix for this network on

the test set containing 200 recordings per gesture. Table 9 lists the key with the gesture names for the confusion matrix. It can be clearly seen that gestures involving the wrist have much higher accuracy than micro gestures consisting of only small finger movements. If we consider these two types of gestures separately, we can calculate an accuracy of 76.5% for micro gestures and an accuracy of 94.33% for larger gestures. With an accuracy of 76.5%, it shows that it is possible to recognize micro gestures with this system, but the machine learning algorithm needs further investigation to increase the accuracy.

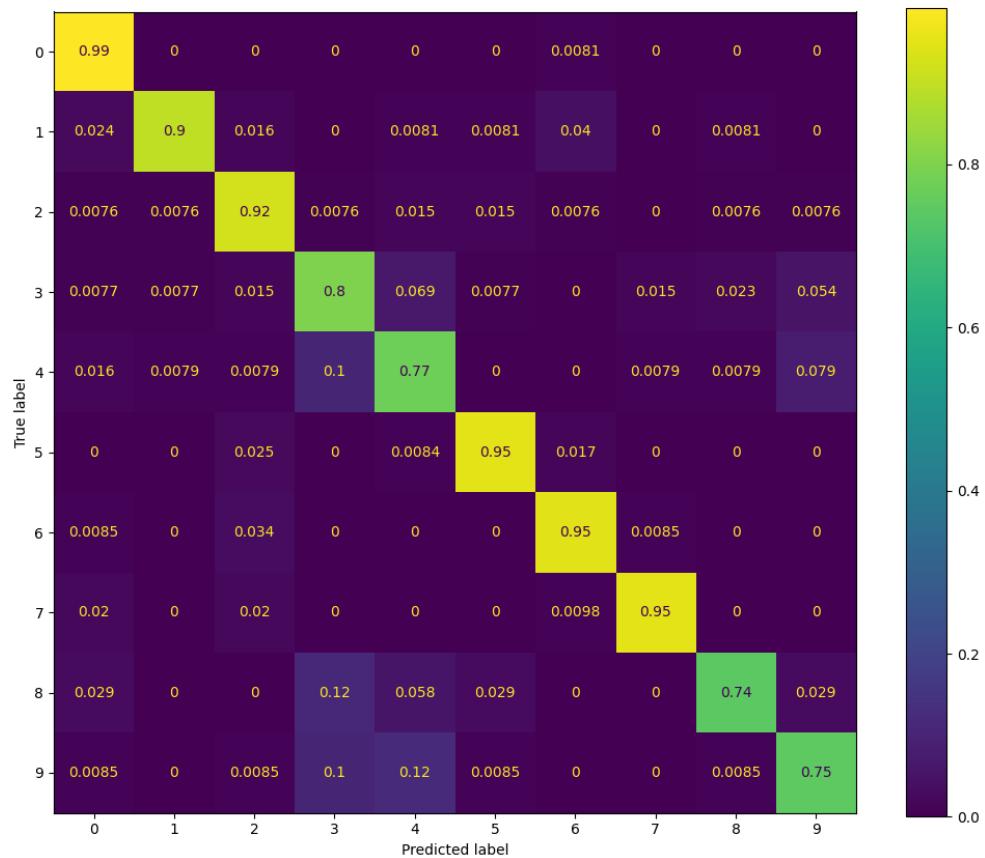


Figure 40: Confusion matrix of test set with 200 recordings per gesture

Gesture	Number	Gesture	Number
Wave Left	0	Snip Finger	5
Wave Right	1	Circle Left	6
Make Fist	2	Circle Right	7
Stretch Fingers	3	Tap Index w. Thumb	8
Point Index Finger	4	Tap Index w. Pinky	9

Table 9: List of gestures and corresponding label

The final model was quantized to INT8, which is discussed in section 8.2.3. The execution of an inference for the quantized model on the microcontroller needs 945'408 cycles and the threshold detection, data pre-processing and feature extraction need 270'522 cycles. The nRF5340 runs at 128MHz and therefore the total time to detect a gesture is 1'231'306 cycles or 9.619ms.

8.2.3 Quantization

The final model has a size of 176.84kB. This would still be able to be stored in the microcontroller's memory but would not be efficient. In section 2.3.4, post-training quantization was introduced. This quantizes the parameters of the trained network from FLOAT32 to INT8, while allowing them to perform with similar accuracy. This brings two major advantages: first, the space in memory is reduced, since a parameter now requires only one byte. Second, it speeds up the computation of the net since integer operations on the microcontroller are inherently faster than floating-point operations. To quantize the network, the trained TensorFlow model was converted to a TensorFlow Lite model and quantized to INT8. After quantization, the model size was reduced by a factor of 3.68 and the time to compute an inference was reduced by a factor of 4.5 Table 10 below shows the size and inference time before and after INT8 quantization.

	Type	Inference Time	Size	Accuracy
Original Model	FLOAT32	33.267ms	176.84kB	87.24%
Quantized Model	INT8	7.386ms	47.99kB	87.17%

Table 10: Model metrics before and after INT8 quantization

8.2.4 Number of QVAR channels vs Accuracy

The bracelet was designed with six QVAR channels in mind. This number represents the maximum number of QVARs that the nRF5340 can support without having to add more circuitry. That is why it does not reflect the optimal number and it is consequently of interest to investigate the minimum number of QVAR channels required. A smaller number of channels reduces the bill of materials and lowers the cost. Additionally, the relationship between electrode size and signal strength was discussed in Section 4.1.4,

where it was shown that larger electrodes provide a stronger signal. With fewer QVAR channels, fewer electrodes are needed, so the electrode size can be increased, e.g., with three channels, the electrode size can be doubled. When the bracelet is worn, it rotates slightly around the wrist as it moves. Since a larger electrode has less relative movement compared to a small/thin electrode, the rotation error is also reduced. Fewer channels bring many advantages to the system design, but with fewer channels, less information is acquired, which has an impact on accuracy. To see this effect on the network, the final model was trained several times with different numbers of QVAR channels. For each number of channels, the network was trained on all possible combinations of the channels (e.g., there are $\binom{6}{2} = 15$ different combinations that can be made with 2 channels), the other channels are simply ignored and discarded. In addition the network was trained once only using the accelerometer data, to see the impact of the QVAR sensor fusion. The network was trained with the entire augmented training set with 10-fold cross-validation and a batch size of 100 for 250 epochs, where 25% of the training set was used for validation, and the accuracy is calculated for the test set with 200 recordings per gesture. The average accuracy across all combinations is shown in Table 11 below.

Number of QVAR Channels	Accuracy
0, Only Accelerometer	78.23 %
1	82.87 %
2	83.25 %
3	84.63 %
4	85.71%
5	85.63 %
6	87.24 %

Table 11: Number of QVAR channels vs model accuracy

Model accuracy decreases with fewer QVAR channels. This can most likely be explained by the fact that less information is available from which to extract features. It is interesting to see that the accuracy decreases only slightly with fewer channels and the network is still able to recognize gestures with 84.63 % accuracy with only 3 channels. In other words, it is possible to reduce the number of QVAR channels without sacrificing much accuracy.

8.3 Comparison to the Myo Bracelet

Gesture recognition with EMG is the technology that comes closest to QVAR gesture

recognition. It is therefore interesting to compare those two technologies, about their strengths and weaknesses. In the beginning the idea was to collect an EMG dataset in parallel, which allows for a close comparison in performance. The Myo bracelet form Thalmic Labs is a low power and wearable eight channel EMG bracelet, that is often used in research for EMG gesture recognition. Unfortunately, the Myo bracelet isn't available on the market anymore and no other commercial EMG bracelet exists at the time of writing. With these circumstances only a general comparison can be made, as no access to such a bracelet was available. The QVAR bracelet itself will be first compared to an EMG bracelet in application and user friendliness and in a second stage the machine learning model will be compared to previously published papers which use the Myo bracelet for gesture recognition.

The biggest difference in application is that QVAR is a capacitive based measurement, while EMG is a resistive based method. With EMG the electrode needs to make a direct electrical connection with the skin. This has the downside that the electrical connection between electrode and skin can influence the measurement, especially with the dry electrodes found in wearable devices. If the wearer has dry skin or a lot of hair, the signal quality will degrade. With direct skin contact the electrodes have to be made of stainless steel to reduce the risk of corrosion.

QVAR on the other hand doesn't need direct skin contact, it can be even worn over clothing. This opens new possibilities and allows for a better integration of the system in different existing devices as the electrodes are not visible. Dry skin and hair won't influence the reading which is an added benefit. The downside of a capacitive method is that the capacitance between the human body and the environment can influence the reading. The electrodes and cables therefore need proper shielding from the environment. EMG on the other hand doesn't require any shielding of the cables or the electrodes, which simplifies the design.

Since EMG measures the activation of the muscles, the electrodes therefore must be placed directly over the muscle of interest. Since the main muscles for wrist movement are placed at the back of the forearm close to the elbow, the bracelet must be worn there. Contrary to QVAR which allows to measure the hand and finger movements directly at the wrist. This makes it possible to integrate easily into a smartwatch or fitness bracelet, to enable the possibility of gesture recognition, which is not possible with EMG. Being able to integrate into already existing wearables will inevitably help users to adopt this new technology as they don't have to wear additional devices just to use gesture recognition.

Comparing the performance in gesture recognition is more difficult to make. It is only made to gather an understanding but not draw any conclusions. Gesture recognition with EMG is an active research area, while QVAR is a novel technique. Therefore, the models and features used for EMG are better researched and understood, while the machine learning model described in this work is not. In addition, the model described in section

7.2 is optimized for embedded devices with memory and computational constraints, while the papers presented which use the Myo bracelet optimize their models to be run on computers with dedicated hardware.

The following table presents a selection of papers and the achieved accuracies that use the Myo bracelet for gesture recognition. The model presented in this thesis is also included as reference.

Author	Technology	Number of Gestures	Accuracy	Inference Time
	y			
This Thesis	QVAR	10	87.14%	7.386ms
M. Benalcázar et al. [42]	sEMG	5	89.5%	194.1ms
Zheng et al. [43]	sEMG	200	>90%	-
Karapinar Senturk et al [44]	sEMG	7	96.43%	-
Xie et al [45]	sEMG	17	83.61%	-

Table 12: Selection of papers that use the Myo bracelet for gesture recognition

9 Conclusion & Future Work

9.1 Conclusion

In this master thesis, a novel, low-power, wearable device was designed and built that can measure hand and finger movements. It focuses on the new LSM6DSV16X, a six-axis IMU from ST Microelectronics, which incorporates the new QVAR technology capable of measuring the local change of charges due to the movement and deformation of the skin. To increase user acceptance, a form factor that can be easily integrated into any smartwatch is introduced. Only a minimal increase in the number of parts is required, and it has been shown that the electrodes can be embedded simply into the wristband. This allows for a variety of possible functions, one possibility being the recognition of gestures performed by the user, which was further explored in this work.

A custom data recording application was developed, and this device was used to collect a dataset of 20 participants who performed the 10 gestures 30 times each. A machine learning model was trained on this dataset, quantized, and implemented on the bracelet. The final model has an overall accuracy of 87.24%. The dataset contains two main types of gestures. Six of the 10 gestures include finger and wrist movements. These are classified with an accuracy of 94.33%. In addition, four micro gestures were examined, which involve only small finger movements and are more difficult to detect. The network is able to recognize these with an accuracy of 76.5%. The final quantized model has a size of 47.99kB and it takes 7.386ms to execute an inference. This allows the bracelet to recognize gestures in real time.

The device can operate in two modes. The first is streaming the raw QVAR and IMU data in real time via BLE. With constant streaming, the device has an average power consumption of 5.47mA, resulting in a battery life of 13h 40min, thanks to the built-in 75mAh battery. The second mode is real-time gesture detection, where the device waits for events, and when a gesture is detected, it sends it to the client via BLE. Since everything happens on the bracelet, it is independent of any external system. This allows it to serve as a general input device for multiple applications. In this mode, it consumes an average current of 6.52mA, resulting in a battery life of 11h 30min, which is around 1.5 workdays.

As far as I know, this work introduces the first wearable charge variation (QVAR) system capable of recognizing gestures in real time. As a result, this novel device can be worn comfortably around the user's wrist and can be easily integrated with existing products such as smartwatches.

9.2 Future Work

This thesis laid the foundation for a new type of technology. It was shown that it is possible to integrate it into a watch capable of detecting the wearer's movements and recognizing the gestures performed. The technology is promising but still in its infancy, and numerous opportunities for improvement have been discovered throughout this work. These findings and insights are presented in this section.

- **Wrist Strap:** The bracelet was made of a 4-layer flexible printed circuit board. This is a process that allows fine features to be realized and provides good shielding. Therefore, it is suitable for a prototype, but not for a product, as it would not survive the mechanical stress of constant wear and tear. Besides, it is only bendable, but cannot stretch. So it has a fixed size, and users would benefit if it could be adjusted to fit different wrist sizes. A method should be developed for stretchable electrodes or individual parts that can move relative to each other.
- **Number of QVAR channels:** In section 8.2.4 it was shown that the number of QVAR channels can be reduced. This has the advantage that larger electrodes can be produced, and the unit costs are reduced. A special wrist strap should be developed that allows a direct comparison between different numbers of electrodes.
- **Machine Learning Features:** The MAV and MV were chosen as the final two features in the time domain. Features in the frequency domain were not investigated in this work. The robustness of the machine learning model would greatly benefit from a thorough investigation of optimal feature selection in the time and frequency domains.
- **Recognizing Micro Gestures:** The model is able to recognize micro gestures with an accuracy of 76.5%. This shows the ability to recognize fine finger movements, but a method should be developed to increase the reliability of the model for micro gestures.
- **Comparison to EMG Gesture Recognition:** Unfortunately, no direct comparison could be made between this bracelet and an EMG bracelet. A gesture data set where both systems record simultaneously would be beneficial for future evaluation. In addition, this would allow an investigation of the fusion of QVAR and EMG sensors.

List of Figures

- Figure 1: The Myo Bracelet from Thalmic Labs [2]
- Figure 2: Gel electrodes (left), needle electrodes (Right) [4] [5]
- Figure 3: sEMG recording of the flexor carpi ulnaris muscle ($f_s = 500$ Hz)
- Figure 4: QVAR sensing channel architecture [9]
- Figure 5: Simple ANN model with input, hidden and output layers
- Figure 6: Simple RNN model with input, hidden and output layers
- Figure 7: Example for an accelerometer service
- Figure 8: Five chosen gestures [31]
- Figure 9: Hardware including sensing, processing, wireless transmission, and battery units (left); Hardware attached to wrist (right) [32]
- Figure 10: Time domain waveform of: (a) gesture set 1 (wrist gestures) and (b) gesture set 2 (finger gestures) [34]
- Figure 11: Diagram of the layers of the basic design of a capacitive sensor on a PCB [36]
- Figure 12: Overview of the flexible capacitive sensing system: (a) The sensing principle of the flexible capacitive sensing system. (b) the sensing system is placed on the wrist of one subject and (c) the optical image of the sensing system. [34]
- Figure 13: Rendering of the EMG and QVAR development board
- Figure 14: QVAR development board test setup
- Figure 15: Set of handmade electrodes
- Figure 16: Stack-up of shielded and unshielded electrodes
- Figure 17: Electrode stack-up with shield ring; bottom view (left), side view (right)
- Figure 18: Diagram of the simulation setup
- Figure 19: Electrode with shield between disturbance (left), electrode with no shield (right)
- Figure 20: Simulated region of sensitivity of the electrode
- Figure 21: Rendering of the rigid PCB with 6x QVAR channels

- Figure 22: Image of the first bracelet prototype with 12 hand crafted electrodes
- Figure 23: The three electrode configurations
- Figure 24: Handmade electrode strap, with configuration b)
- Figure 25: 2 second recording of participant making a fist, wearing electrode configuration (b)
- Figure 26: 2 second recording of participant making a fist, wearing electrode configuration (b)
- Figure 27: Rendering of the rigid PCB with 6x QVAR channels
- Figure 28: Top (a) and bottom (b) rendering of the flex PCB
- Figure 29: Top view of final bracelet with the 3D printed enclosure
- Figure 30: Start, path and end positions of the 10 gestures
- Figure 31: Screenshot of the Recording application
- Figure 32: Fastening position of the QVAR bracelet
- Figure 33: Raw recording of wave left gesture with highlighted window of interest
- Figure 34: Window of interest, cut out from figure 33
- Figure 35: Detailed structure of the input vector
- Figure 36: Model architecture
- Figure 37: 4 second recording of the QVAR noise floor
- Figure 38: A raw 3s recording of the gesture wave left made with the bracelet
- Figure 39: Accuracy vs model size, with highlighted chosen model
- Figure 40: Confusion matrix of test set with 200 recordings per gesture

List of Tables

- Table 1: Dimension values for the simulated electrode
- Table 2: Noise values of different electrode shielding of QVAR test setup
- Table 3: Signal strength for various electrode sizes
- Table 4: Advertised current consumption of the NORA-B106 at 3V [38]
- Table 5: Advertised current consumption of the LSM6DSV16X at 1.8V [39]
- Table 6: Characteristic UUIDs
- Table 7: Average current consumption
- Table 8: QVAR channel noise level and SNR
- Table 9: List of gestures and corresponding label
- Table 10: Model metrics before and after INT8 quantization
- Table 11: Number of QVAR channels vs model accuracy
- Table 12: Selection of papers that use the Myo bracelet for gesture recognition

List of Acronyms

QVAR	Electrostatic Charge Variation
EMG	Electromyography
ADC	Analog to Digital Converter
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
RTOS	Real-Time Operating System
BLE	Bluetooth Low Energy
GAP	Generic Access Profile
GATT	Generic Attribute Profile
ECG	Electrocardiogram
PCB	Printed Circuit Board
LDO	Low Dropout Regulator
MAV	Mean Absolute Value
MV	Mean Value
SNR	Signal-to-Noise Ratio
RMS	Root Mean Square

Bibliography

- [1] ComperiTech: “Screen Time Statistics: Average Screen Time in US vs. the Rest of the World”, accessed April 24th 2022; [Online] Available: <https://www.comparitech.com/tv-streaming/screen-time-statistics/#:~:text=Worldwide%2C%20the%20average%20person%20spends,for%20internet%2Dconnected%20activities>

[2] Myo Bracelet: <https://ict.swisscom.ch/wp-content/uploads/2016/07/armband.jpg>

[3] Johns Hopkins Medicine: “Electromyography (EMG)”, accessed May 24th 2021; [Online] Available: [Electromyography \(EMG\) | Johns Hopkins Medicine](#)

[4] Needle Electrodes:
<https://neurowebshop.webshopapp.com/en/technomed-disposable-hypodermic-botox-emg-naald-el.html>

[5] Gel Electrodes: https://www.kardiodepot.ch/s/46503_319553_elektroden-3m-red-dot-2238-beutel-mit-50-stuck?part=googleMC&part=google-shopping-de-ch&gclid=Cj0KCQjw5auGBhDEARIsAFyNm9G7qEz6OkRNhRP_bNlFVSZb5dzD_vcUdYzzo0-qjX8cxcWyinxPfY4aAoBFEALw_wcB

[6] M. Raez, M. Hussain, F. Mohd-Yasin, “Techniques of EMG signal analysis: detection, processing, classification and applications”, accessed February 21st 2022; [Online] Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1455479/>

[7] S.Day, “Important Factors in Surface EMG Measurements”, accessed February 22th 2022; [Online] Available: https://www.researchgate.net/publication/308365391_Important_factors_in_surface_EMG_measurement

[8] Delsys: “Surface Electromyography: Detection and Recording”, accessed February 25th 2022; [Online] Available: <https://www.delsys.com/downloads/TUTORIAL/semg-detection-and-recording.pdf>

[9] ST Microelectronics: “QVAR Sensing – Application Note”, accessed April 27th 2022; [Online] Available: https://www.st.com/resource/en/application_note/an5755-qvar-sensing-channel--stmicroelectronics.pdf

- [10] C. Xu, A. Wang, H. Zou, B. Zhang, C. Zhang, Y. Zi, L. Pan, P. Wang, P. Feng, Z. Lin, Z. Wang, “Raising the Working Temperature of a Triboelectric Nanogenerator by Quenching Down Electron Thermionic Emission in Contact-Electrification”, accessed April 27th 2022; [Online] Available: <https://pubmed.ncbi.nlm.nih.gov/30091484/>
- [11] E. Purcell, D. Morin, “Electricity and Magnetism”, Cambridge univ. Press pp 127-128, 2013; [Online] Available: https://books.google.ch/books?id=A2rS5v1SFq0C&pg=PA127&redir_esc=y#v=onepage&q&f=false
- [12] K. Hornik, “Multilayer Feedforward Networks are Universal Approximators”, accessed May 5th 2022; [Online] Available: https://cognitivemedium.com/magic_paper/assets/Hornik.pdf
- [13] IBM: “What is Deep Learning?”, accessed April 27th 2022; [Online] Available: <https://www.ibm.com/cloud/learn/deep-learning#:~:text=Deep%20learning%20is%20a%20subset,from%20large%20amounts%20of%20data.>
- [14] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, accessed April 27th 2022; [Online] Available: <http://www.deeplearningbook.org>
- [15] S. Hochreiter, J. Schmidhuber, “Long Short-Term Memory”, accessed April 25th 2022; [Online] Available: <https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory?redirectedFrom=fulltext>
- [16] TensorFlow: “TensorFlow”, accessed April 19th 2022; [Online] Available: <https://www.tensorflow.org/>
- [17] H. Wu, P. Judd, X. Zhang, M. Isaev, P. Micikevicius, “Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation”, accessed April 14th 2022; [Online] Available : <https://arxiv.org/pdf/2004.09602.pdf>
- [18] TensorFlow: “TensorFlow Lite for Microcontrollers”, accessed April 19th 2022; [Online] Available: <https://www.tensorflow.org/lite/microcontrollers>
- [19] R. David, J. Duke, A. Jain, V. J. Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natrajan, S. Regev, R. Rhodes, T. Wang, and P. Warden, “TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems”, accessed April 19th 2022; [Online] Available: <https://arxiv.org/pdf/2010.08678.pdf>
- [20] GitHub: “Zephyr GitHub page”, accessed April 19th 2022; [Online] Available: <https://github.com/zephyrproject-rtos/zephyr>

- [21] Zephyr Project: “Introduction”, accessed April 19th 2022; [Online] Available: <https://docs.zephyrproject.org/latest/introduction/index.html>
- [22] MethodPark: “What can the Zephyr RTOS do for you as a developer?”, accessed April 19th 2022; [Online] Available: <https://www.methodpark.de/blog/what-can-the-zephyr-rtos-do-for-you-as-a-developer-part-1/>
- [23] Bluetooth SIG: “Bluetooth Technology Overview”, accessed April 19th 2022; [Online] Available: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>
- [24] C. Gomez, J. Oller, J. Paradells, “Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology”, accessed April 19th 2022; [Online] Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3478807/pdf/sensors-12-11734.pdf>
- [25] P. Pisharady, M. Saerbeck, “Recent Methods and Databases in Vision-based Hand Gesture Recognition: A Review”, accessed January 20th 2022; [Online] Available: <https://www.sciencedirect.com/science/article/abs/pii/S1077314215001794>
- [26] T. Schröder, B. Poppinga, N. Henze, S. Boll, “Gesture Recognition with a Wii Controller”, accessed January 20th 2022; [Online] Available: <https://dl.acm.org/doi/pdf/10.1145/1347390.1347395>
- [27] X. Chen, X. Zhang, Z. Zhao, J. Yang, “Hand Gesture Recognition Research Based on Surface EMG Sensors and 2D-accelerometers”, accessed February 3rd 2022; [Online] Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4373769>
- [28] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, J. Yang, “A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors”, accessed January 22nd 2022; [Online] Available: <https://ieeexplore.ieee.org/abstract/document/5735233>
- [29] NinaPro: “DB5 – 10 Intact Subjects – Double Myo Armband”, accessed December 15th 2022; [Online] Available: http://ninapro.hevs.ch/DB5_DoubleMyo
- [30] S. Pizzolato, L. Tagliapietra, M. Cognolato, M. Reggiani, H. Müller, M. Atzori, “Comparison of six electromyography acquisition setups on hand movement classification tasks”, accessed December 15th 2022; [Online] Available: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0186132>

- [31] Z. Zhang, K. Yang, J. Qian, L. Zahng, “Real-Time Surface EMG Pattern Recognition for Hand Gestures Based on an Artificial Neural Network”, accessed January 22nd 2022; [Online] Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6679304/>
- [32] S. Bian, V. Rey, P. Hevesi, P. Lukowicz, “Passive Capacitive based Approach for Full Body Gym Workout Recognition and Counting”, accessed 10th December, 2022; [Online] Available: <https://ieeexplore.ieee.org/document/8767393>
- [33] V. Nguyen, S. Rupavatharam, L. Liu, R. Howard, M. Gruteser, “HandSense: Capacitive coupling-based Dynamic, Micro Finger Gesture Recognition”, accessed January 18th 2022; [Online] Available: <https://www.winlab.rutgers.edu/~gruteser/papers/sensys19-final195.pdf>
- [34] T. Wang, Y. Zhao, Q. Wang, “Flexible Non-contact Capacitive Sensing for Hand Gesture Recognition”, accessed December 8th 2022; [Online] Available: https://link.springer.com/content/pdf/10.1007%2F978-3-030-89095-7_58.pdf
- [35] Searle, A.; Kirkup, L. A direct comparison of wet, dry and insulating bioelectric recording electrodes. *Physiol. Meas.* 2000, 21, 271–283.
- [36] A. Portelli, S. Nasuto, “Design and Development of Non-Contact Bio-Potential Electrodes for Pervasive Health Monitoring Applications”, accessed January 24th 2022; [Online] Available: <https://www.mdpi.com/2079-6374/7/1/2>
- [37] B. Babusiak, S. Borik, L. Balogova, “Textile electrodes in capacitive signal sensing applications”, accessed January 20th 2022; [Online] Available: https://www.sciencedirect.com/science/article/pii/S0263224117305894?casa_token=4Q2ZDRIfpqoAAAAA:nyqAVxVnuSHLRBrICRtUGmu2ES7neUtkYPdYNriTxii8KndOPSTBBwDgYigcvZBE2pXD_CaBVh8
- [38] uBlox: ‘NORA-B1 series, datasheet”, accessed January 10th 2022; [Online] Available: https://content.u-blox.com/sites/default/files/NORA-B1_DataSheet_UBX-20027119.pdf
- [39] ST Microelectronics: “LSM6DSV16X Datasheet”, accessed January 10th 2022; [Online] Available: <https://www.st.com/en/mems-and-sensors/lsm6ds16x.html#documentation>
- [40] Nordic Power Profiler Kit II; [Online] Available: [Power Profiler Kit II - nordicsemi.com](https://nordicsemi.com)

- [41] V. Amini Visteh, A. Mirzahani, E. Jafarzadehpour, S. Darvishpour, “Evaluation of Simple Visual Reaction Time of Different colored Light Stimuli in Visually Normal Students”, accessed 19th May 2022; [Online] Available: <https://www.dovepress.com/evaluation-of-simple-visual-reaction-time-of-different-colored-light-s-peer-reviewed-fulltext-article-OPTO>
- [42] M. Benalcázar, C. Motoche, J. Zea, A. Jaramillo, C. Anchundia, P. Zambrano, M. Segura, F. Palacios, M. Pérez, “Real-time hand gesture recognition using the Myo armband and muscle activity detection”, accessed May 13th 2022; [Online] Available: <https://ieeexplore.ieee.org/abstract/document/8247458/authors#authors>
- [43] Z. Zheng, Q. Wang, D. Yang, Q. Wang, W. Huang, Y. Xu, “L-Sign: Large-Vocabulary Sign Gestures Recognition System”, accessed May 13th 2022; [Online] Available: <https://ieeexplore.ieee.org/abstract/document/9714154/authors#authors>
- [44] Z. Karapinar Snturk, M. S. Bakay, “Machine Learning-Based hand Gesture Recognition via EMG Data”, accessed May 13th 2022; [Online] Available: <https://revistas.usal.es/index.php/2255-2863/article/view/ADCAIJ2021102123136/26176>
- [45] B. Xie, B. Li, A. Harland, «Movement and Gesture Recognition Using Deep Learning and Wearable-sensor Technology”, accessed May 13th 2022; [Online] Available: <https://sci-hub.se/tree/47/2b/472b16112a00cb29560d6d8d94a4d2be.pdf>



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D-ITET center for Project-Based
Learning

Tasks Description For a Master project
At the Department of Information Technology and
Electrical Engineering

Elio Reinschmidt

Smart sEMG & Capacitive Sensing Wearable for Detecting Wrist & Finger Kinematics

Advisors: Dr. Michele Magno
Dr. Christian Vogt

Professor: Prof. Dr. Sebastian Kozerke

Handout Date: 22.11.2021
Due Date: 22.05.2022

1 Project Goals

The ability to measure muscle movements with wearables while performing rehabilitation exercises is of great benefit. Wearables such as these are of great interest because they provide vivid feedback on how well the movements are being performed. This allows for better monitoring of the patient when they are performing the exercises by themselves at home. Another advantage is that these wearables can be used as input devices to control various prostheses and applications.

One way to track the movements of our muscles is through surface electromyography (sEMG). When our muscles contract, a voltage drops across the muscle. In sEMG, electrodes are placed on the skin to measure this voltage drop. This voltage drop corresponds to the contraction of the measured muscle [1]. Another way to track motion is through capacitive sensing. Our body forms a capacitor with its environment, and by moving our body we change the shape and thus the capacitance of our body. This capacitance change can be used to detect the movements of different body parts such as fingers [2].

The main objective of this master thesis is to develop a novel, compact, and low-power wearable system that collects sEMG, IMU, and capacitive sensor data that are streamed in real time via Bluetooth Low Energy. In addition, a machine learning model will be trained and run on the microcontroller that combines the collected data to recognize different hand and finger movements.

The project will be split into four phases, as described below:

Phase 1: “Design & Implement Evaluation Board” (Week 1-6)

1. Choose final components to be used in the bracelet
2. Develop initial evaluation board to test the final hardware for the bracelet
3. Move the previous Project from the nRF52840 to the new nRF5340 microcontroller.
4. Test and evaluate the hardware
5. Move code from previous semester thesis to Zephyr RTOS

Phase 2: “Capacitive Sensing setup” (week 7-13)

1. Preliminary investigation and feasibility study of capacitive sensing to detect finger kinematics and identify possible application scenarios. Analysis and consolidation of previous works.
2. Short ~10 minute presentation about the literature review, metric discussion and the future roadmap of the thesis.
3. The student and the supervisor decide on an application scenario. A suitable method to attach capacitive sensing sensors to the hand will be developed and designed, that connects to the smart bracelet, in the form of a custom product.
4. Investigation and implementation of a suitable test setup.

Phase 3: “Capacitive Sensing implementation” (week 14-22)

1. Define a ground truth measurement to compare the model against.
2. Recording/generating a dataset to train a machine learning model. This should include more than 10 subjects.
3. Train a machine learning model to detect the wanted finger kinematics
4. In-field evaluation and concluding the performance of the capacitive sensing application

Phase 4 (week 23-26)

5. Finalize and clean up hardware schematics/layout and code base for hand in.
6. Write report and prepare for final presentation.

Milestones

By the end of **Phase 1** the following should be completed:

- Working evaluation board that reassembles the electronic circuit of the final sEMG and capacitive sensing system
 - Firmware is moved to Zephyr RTOS and adaptet to the nRF5340

By the end of **Phase 2** the following should be completed:

- Capacitive Sensing background study is finalized
- Decided on the final capacitive sensing application and test setup
- A working prototype to measure capacitive sensing

By the end of **Phase 3** the following should be completed:

- A recorded dataset to train a machine learning model
- A trained machine learning model to detect finger kinematics

By the end of **Phase 4** the following should be completed:

- Final report & presentation

2 Project Organization

2.1 Weekly Meeting

There will be a meeting at the end of every week. The main purpose of this meeting is to document the project's progress and should be used by the student to communicate any problems that arise during the week.

2.2 Final Report and paper

PDF copies of the report and presentation are to be turned in.

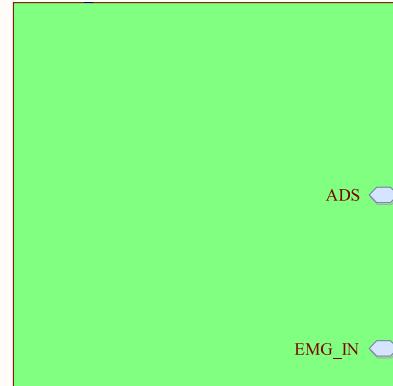
Bibliography

- [1] Johns Hopkins Medicine: "Electromyography (EMG)", accessed October 27th 2021; [Online] Available: <https://www.hopkinsmedicine.org/health/treatment-tests-and%20therapies/electromyography-emg>
- [2] Association for Computing Machinery: "HandSense: Capacitive coupling-based Dynamic, Micro Finger Gesture Recognition", accessed October 27th 2021; [Online] Available: <https://dl.acm.org/doi/pdf/10.1145/3356250.3360040>

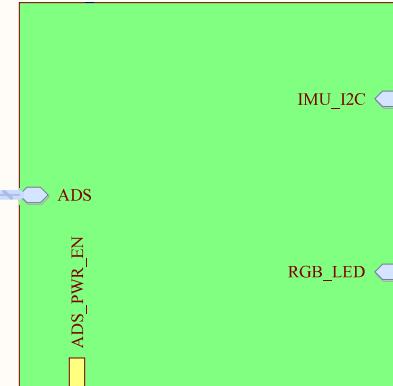
A

A

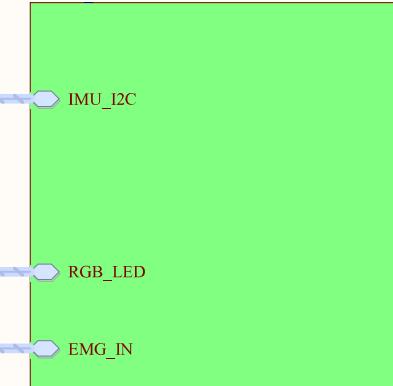
EMG CHIP
ADS1298_schem.SchDoc



MCU
nRF5350_schem.SchDoc



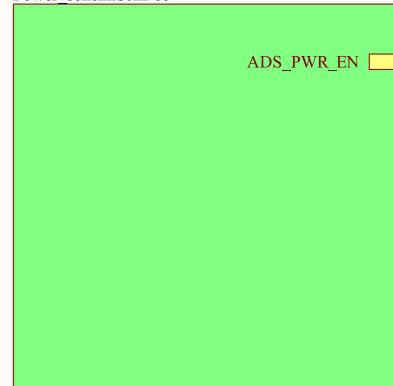
SENSORS
Sensors_schem.SchDoc



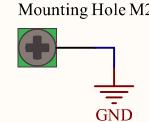
B

B

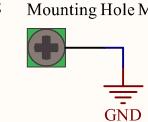
POWER
Power_schem.SchDoc



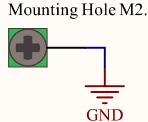
M1



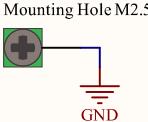
M2



M3



M4



ETHZurich
Logo ETH Zurich



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Project:

*

Drawing number: * Rev: *

Date: 28.03.2022 17:18:12

Format: Laboratory: *

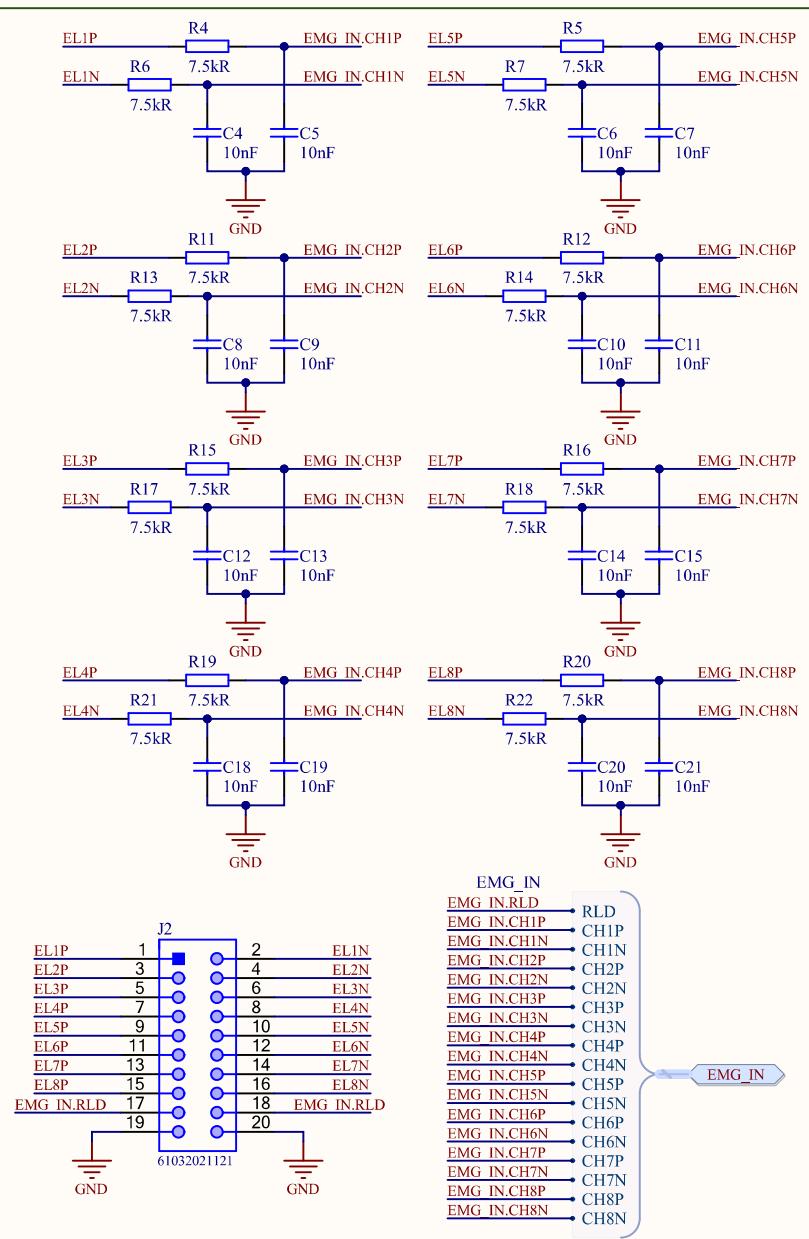
A4 Q Drawn by: *

Sheet: Main_schem.SchDoc

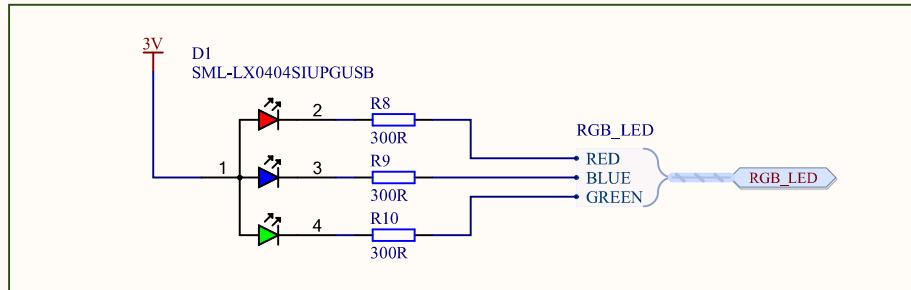
Page 1 of 5

File: C:\Users\roscos\Documents\Neurlet\DEV_BOARD\EMG_ADS1298_EVAL_BOARD_v2>Main_schem.SchDoc

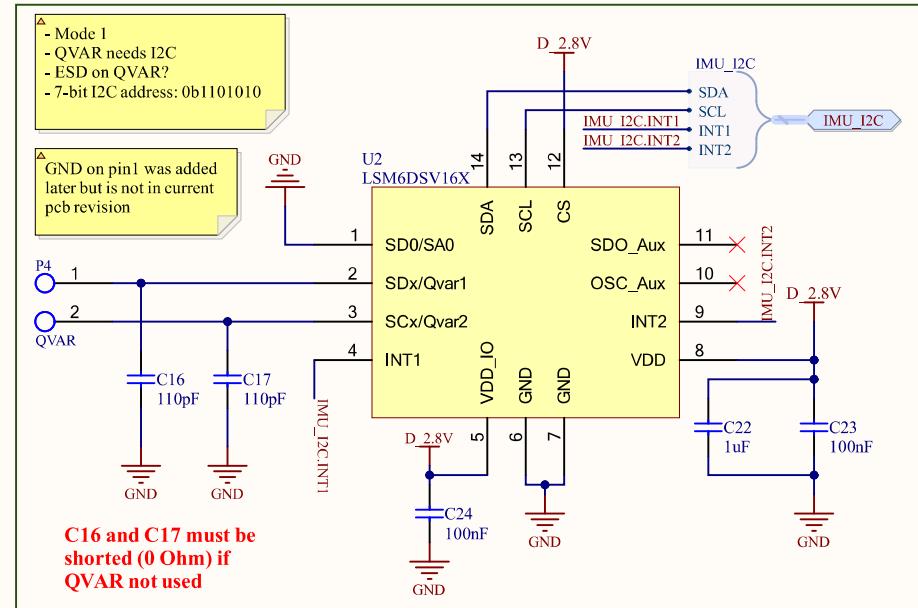
EMG Filter



RGB LED



IMU & QVAR



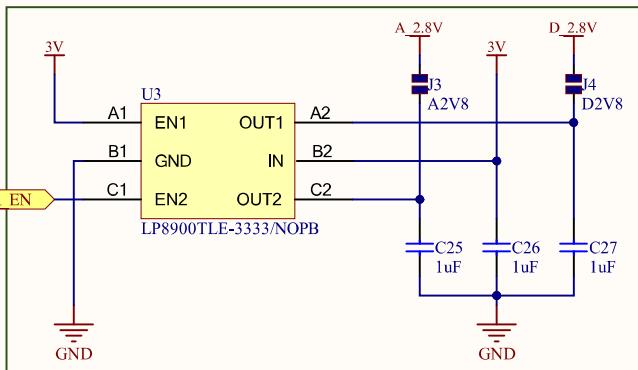
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Project:

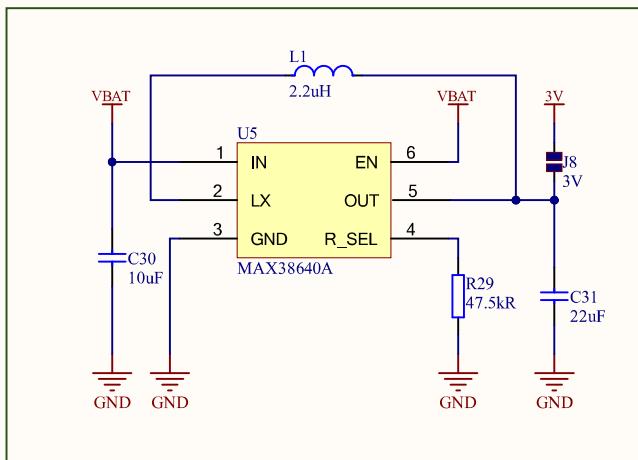
*

Drawing number:	*	Rev:	*	Format:	Laboratory:	*	Sheet:	Sensors_schem.SchDoc
Date:	28.03.2022	17:18:12		A4 Q	Drawn by:	*		
File:	C:\Users\rosco\Documents\Neurlet\DEV_BOARD\EMG_ADS1298_EVAL_BOARD_v2\Sensors_schem.SchDoc							Page 2 of 5

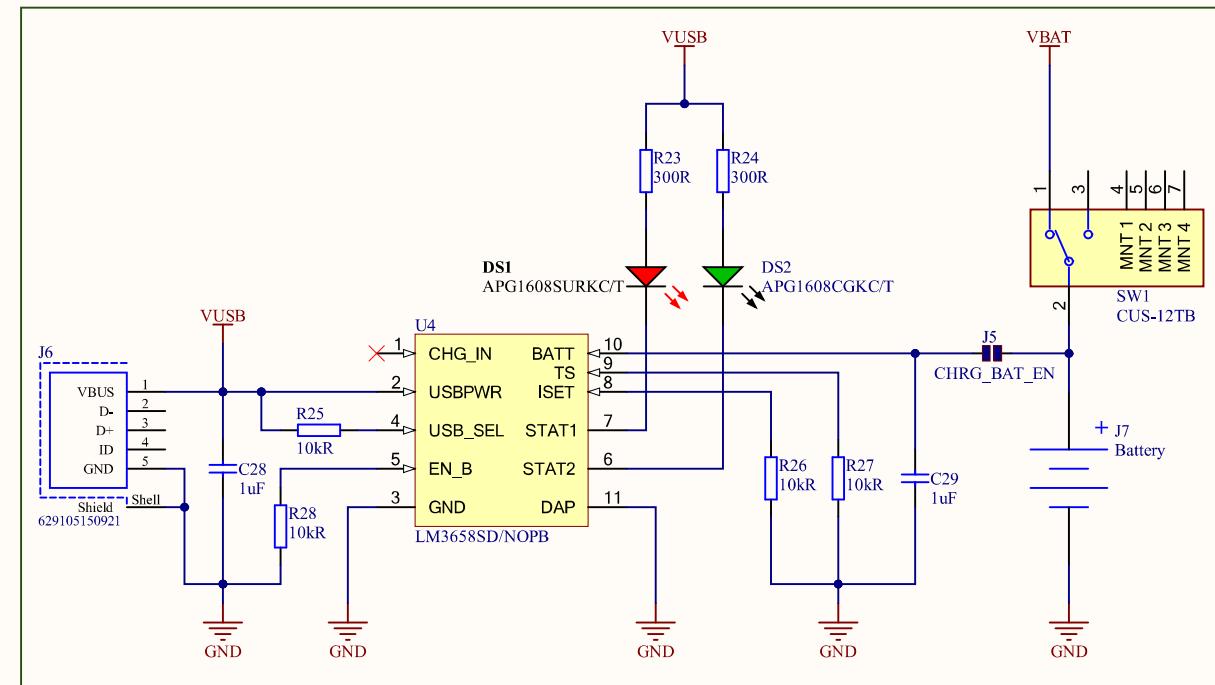
2x 2.8V LDO



3V Buck Converter



Battery Charger

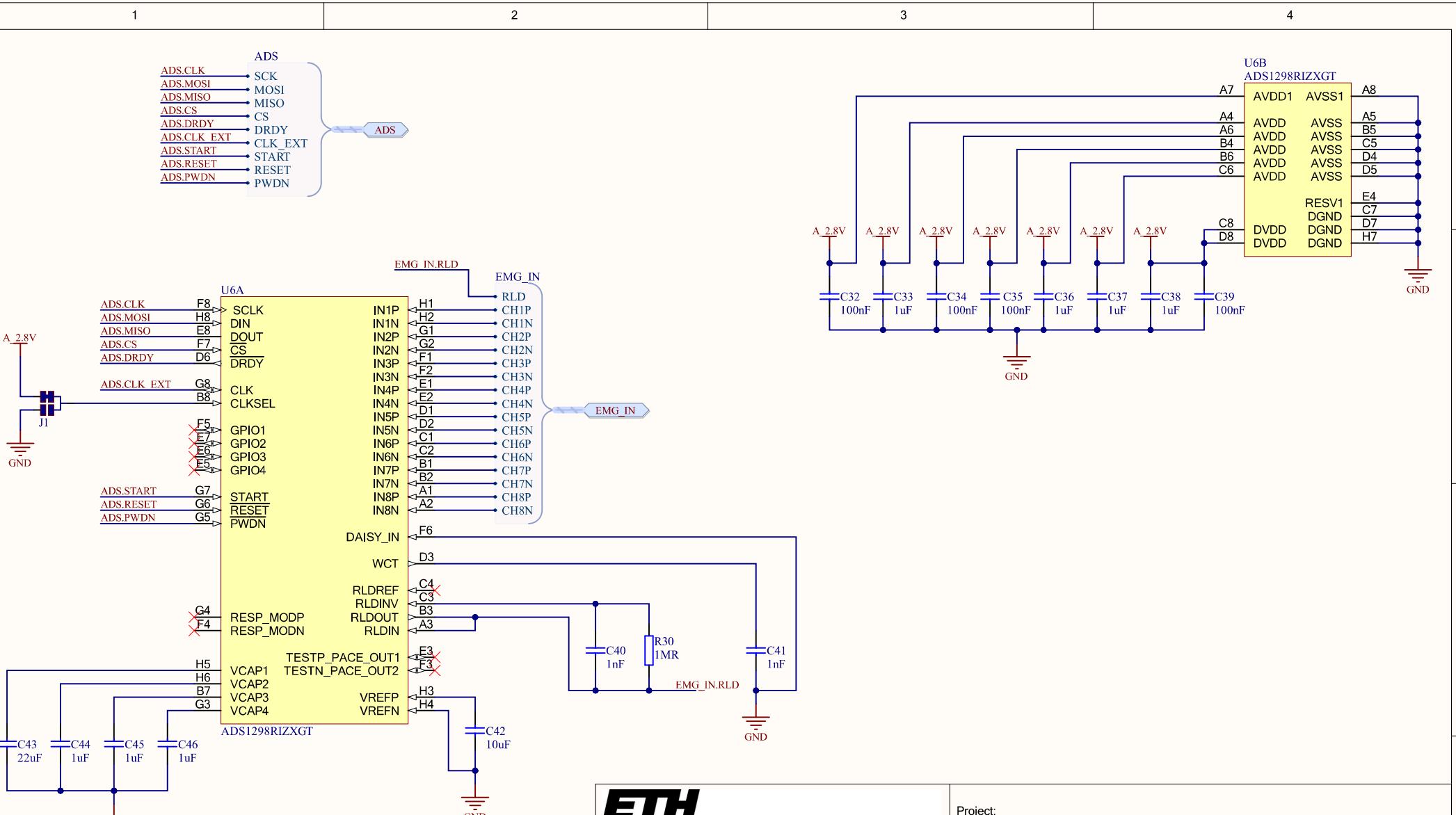


Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Project:

*

Drawing number: *	Rev: *	Format: A4 Q	Laboratory: *	Sheet: Power_schem.SchDoc
Date: 28.03.2022 17:18:12		Drawn by: *		Page 3 of 5
File: C:\Users\roscos\Documents\Neurlet\DEV_BOARD\EMG_ADS1298_EVAL_BOARD_v2\Power_schem.SchDoc				



ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Drawing number:	*	Rev:	*	Format:	Laboratory:	*	Sheet:	ADS1298_schem.SchDoc
Date:	28.03.2022	17:18:13		A4 Q	Drawn by:	*	Page	4 of 5
File: C:\Users\rosclo\Documents\Neurlet\DEV_BOARD\EMG_ADCS_EVAL_BOARD_v2\ADS1298_schem.SchDoc								

1 2 3 4

A

B

C

D

1 2 3 4

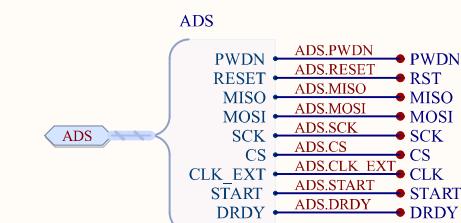
1

2

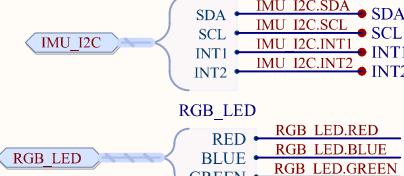
3

4

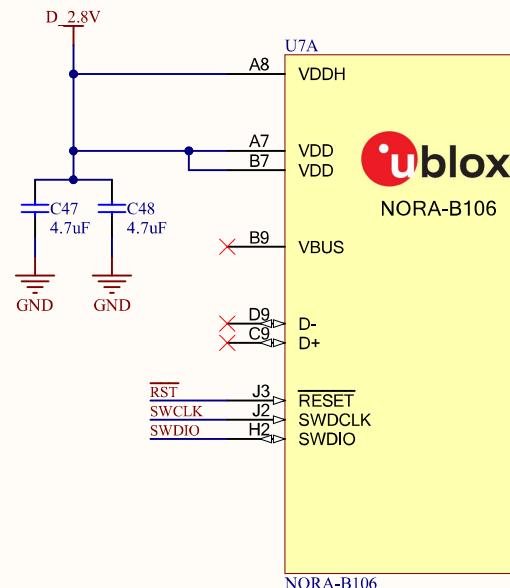
A



B



C



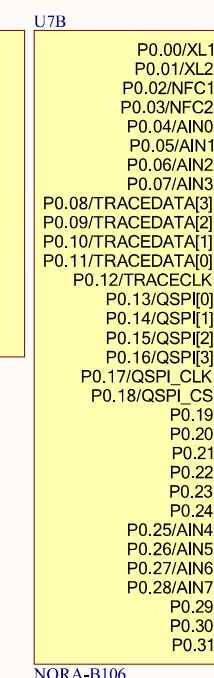
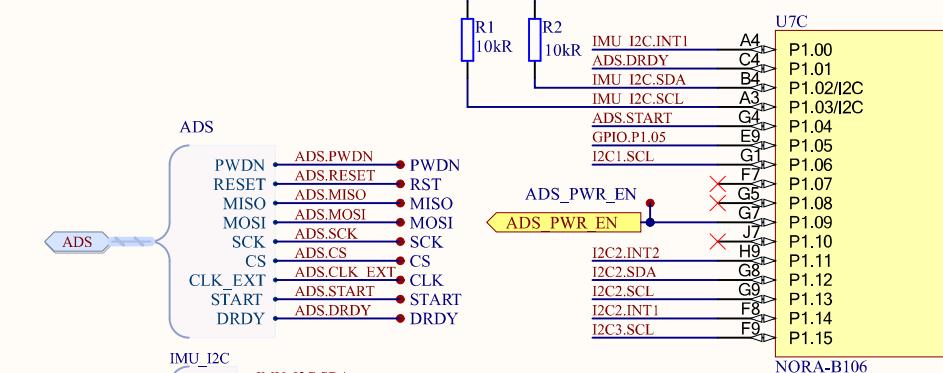
D

1

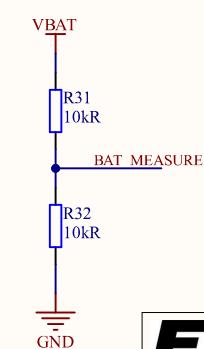
2

3

4



NORA-B106



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Drawing number: * Rev: *

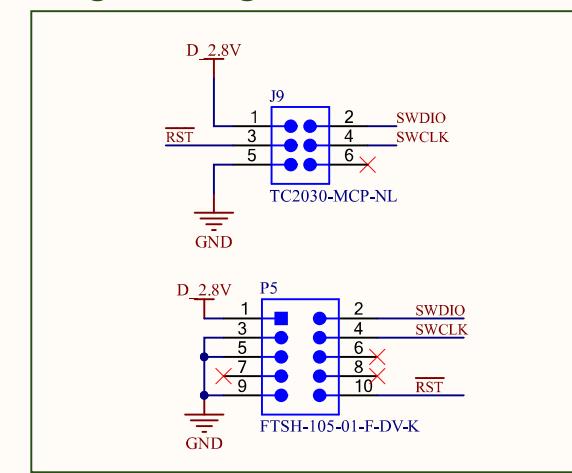
Date: 28.03.2022 17:18:13

Format: A4 Q Laboratory: * Sheet: nRF5350_schem.SchDoc

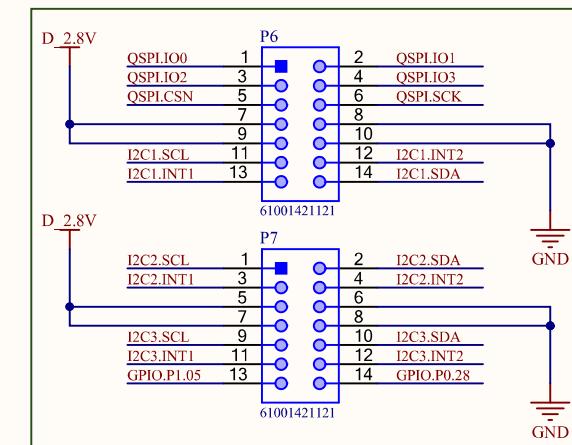
Drawn by: * Page 5 of 5

File: C:\Users\rosco\Documents\Neurlet\DEV_BOARD\EMG_ADS1298_EVAL_BOARD_v2\nRF5350_schem.SchDoc

Programming Connector



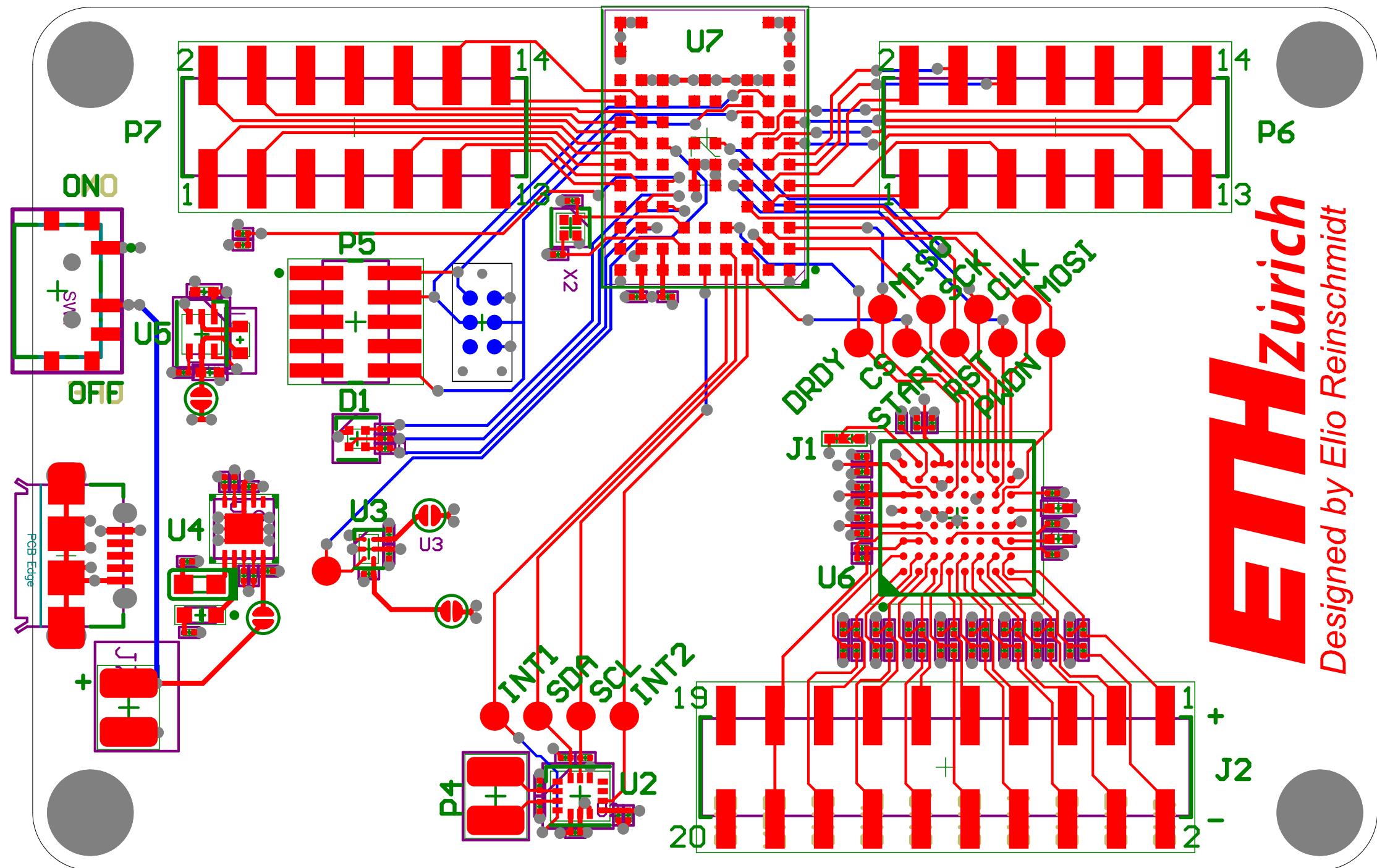
Programming Connector



61001421121 Pinout:

- 1: I2C3 SCL
- 2: I2C3 INT1
- 3: I2C3 INT2
- 4: I2C3 SDA
- 5: GND
- 6: I2C2 SCL
- 7: GND
- 8: I2C2 INT1
- 9: I2C2 INT2
- 10: I2C2 SDA
- 11: GND
- 12: I2C1 SCL
- 13: GND
- 14: I2C1 INT1
- 15: I2C1 INT2
- 16: I2C1 SDA
- 17: GND

ETH Zurich
Designed by Elio Reinschmidt



Comment	Description	Designator	Footprint	LibRef	Quantity
Test_Point	ADS_PWR_EN, CLK, CS, DRDY, INT1, INT2, MISO, MOSI, PWDN, RST, SCK, SCL, SDA, START	Test_Point	Test_Point	Test_Point	14
10nF	* CAP, 10nF, 10V, 5%, X5R, 0201	C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C18, C19, C20, C21	CAP0201	GRM033R61A103JA01D	16
110pF	* CAP, 110pF, 50V, 5%, COG/NPO, 0201	C16, C17	CAP0201	GRM0335CH111JA01D	2
1uF	* CAP, 1uF, 10V, X5R, 0201	C22, C25, C26, C27, C28, C29, C33, C36, C37, C38, C44, C45, C46	CAP0201	GRM033R61A105WE15D	13
100nF	* CAP, 0.1uF, 25V, 20%, X5R, 0201	C33, C24, C32, C34, C35, C39	CAP0201	GRM033R61E104ME14D	6
10uF	* Cap, 10uF, 10V, X5R, 20%, 0402, * Cap, 10nF, 10V, X5R, 20%, 0402	C30, C42	CAP0402	ZRB15X61A106ME01D	2
22uF	* CAP, 22uF, 6.3V, X5R, 20%, 0402	C31, C43	CAP0402	GRM158R60J226ME01D	2
1nF	* CAP, 1nF, 25V, X7R, 10%, 0201	C40, C41	CAP0201	GRM033R71E102KA03D	2
4.7uF	* CAP, 4.7uF, 6.3V, X5R, 20%	C47, C48	CAP0201	GRM035R60J475ME15D	2
16pF	* CAP, 16pF, 25V, 2%, 0201	C49, C50	CAP0201	GRM033SC1E160GA01D	2
LED, Rot, Grün, Blau, Oberflächenmontage SML-LX0404S1UPGUSB	120°, Quadratistik, 8 2mA, 6 2mA	D1	SML-LX0404S1UPGUSB	SML-LX0404S1UPGUSB	1
APG1608SURKC/T	Ultra Thin Hyper Red SMD Chip LED Lamp, 2 to 2.5 V., 40 to 85 degC, Body 1.6 x 0.8 mm, 0.28 mm Lead, RoHS, Tape and Reel	DS1	KING-LED0603-25-RED_V	OMP-0404-00004-1	1
APG1608CGKC/T	LED, SMT, 0603(1608), 0.25mm Thickness, Green	D52	KING-LED0603-25-GREEN_V	OMP-0239-00001-1	1
61032021121	SMT Vertical Pin Header WR-PHD, Pitch 2.54 mm, Dual Row, 20 pins	J2	61032021121	OMP-1502-01057-1	1
A2V8		J3	Small Solder Jumper	CMP-00005-5	1
D2V8		J4	Small Solder Jumper	CMP-00005-5	1
CHRG_BAT_EN		J5	Small Solder Jumper	CMP-00005-5	1
G29105150921	Micro USB 2.0 Type AB Receptacle WR-COM, Horizontal, SMT	J6	G29105150921	CMP-1502-03249-1	1
Battery	* Pad to solder on battery	J7	Battery_Pad	Battery_Pad	1
3V		J8	Small Solder Jumper	CMP-00005-5	1
TC2030-MCP-NL	CABLE, TAG CONNECT, WITH OUT LEG, FOR IC	J9	TC2030-MCP-NL	CMP-00723-1	1
2.2uH	HIND, 2.2uH, 1.05A, 20%	L1	IND0603	LQM18PN2R2MGHD	1
QVAR		P4	QVAR Pad	QVAR Pad	1
FTSH-105-01-F-DV-K	Male Header, Pitch 1.27 mm, 2 x 5 Position, Height 6.12 mm, RoHS, Tube	P5	SMTC-FTSH-105-01-L-DV_K_V	OMP-2000-05352-1	1
61001421121	SMT Vertical Pin Header WR-PHD, Pitch 2.54 mm, Dual Row, 20 pins	P6, P7	61001421121	OMP-1502-01031-1	2
10kR	* RES, 10kOhm, 1%, 0201	R1, R2, R25, R26, R27, R28, R31, R32	RES0201	ERJ-1GEF1002C	8
7.5kR	* RES, 7.5k Ohm, 1%, 0201	R4, R5, R6, R7, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22	RES0201	ERJ-1GEF7501C	16
300R	* RES, 300 Ohm, 1%, 0201	R8, R9, R10, R23, R24	RES0201	ERJ-1GEF3000C	5
47.5kR	* RES, 47.5k Ohm, 1%, 0201	R29	RES0201	ERJ-1GNF4752C	1
1MR	* RES, 1M Ohm, 1%, 0201	R30	RES0201	ERJ-1GNF1004C	1
CUS-12TB	SWITCH SUIDE SPDT 300mA, 4V	SW1	FP-CUS-12TB-MFG	OMP-71745-000005-1	1
LSM6DSV16K	#IMU_GVAR	U2	LGA14_STM	LSM6DSV16K	1
UP8900TLE-3333/NOPB	* Standard Regulator Pos 2.8VZ 8V 0.2A/0.2A 6-Pin Micro SMD T/R	U3	UP8900TLE-3333/NOPB	UP8900TLE-3333/NOPB	1
LM3658SD/NOPB	Dual Source USB/AC Li Chemistry Charger IC for Portable Applications, 10-pin TLP, Pb-Free	U4	DSC0010A_V	CMP-0063-00780-3	1
MAX38640A	# Buck Converter	U5	uDFN-6 MAXIM	MAX38640AEI+T	1
ADSL298RIZXGT	8-Channel, 24-Bit ADC with Integrated Reference	U6	Tl-ZXG64	OMP-0003-00996-1	1
NORA-B106	u-blox NORA-B106 Bluetooth 5.2 LE module - dual core	U7	NORA_LGA82R_14300	NORA-B106	1
32.768kHz	* Crystl 0.03768MHz ±20ppm [T0] 12.5pF FUND 800000hm 4-Pin Ultra Mini-CSMD T/R	X2	ECX-1210B	ECS-327-12.5-1210B-N-TR	1

A

A

B

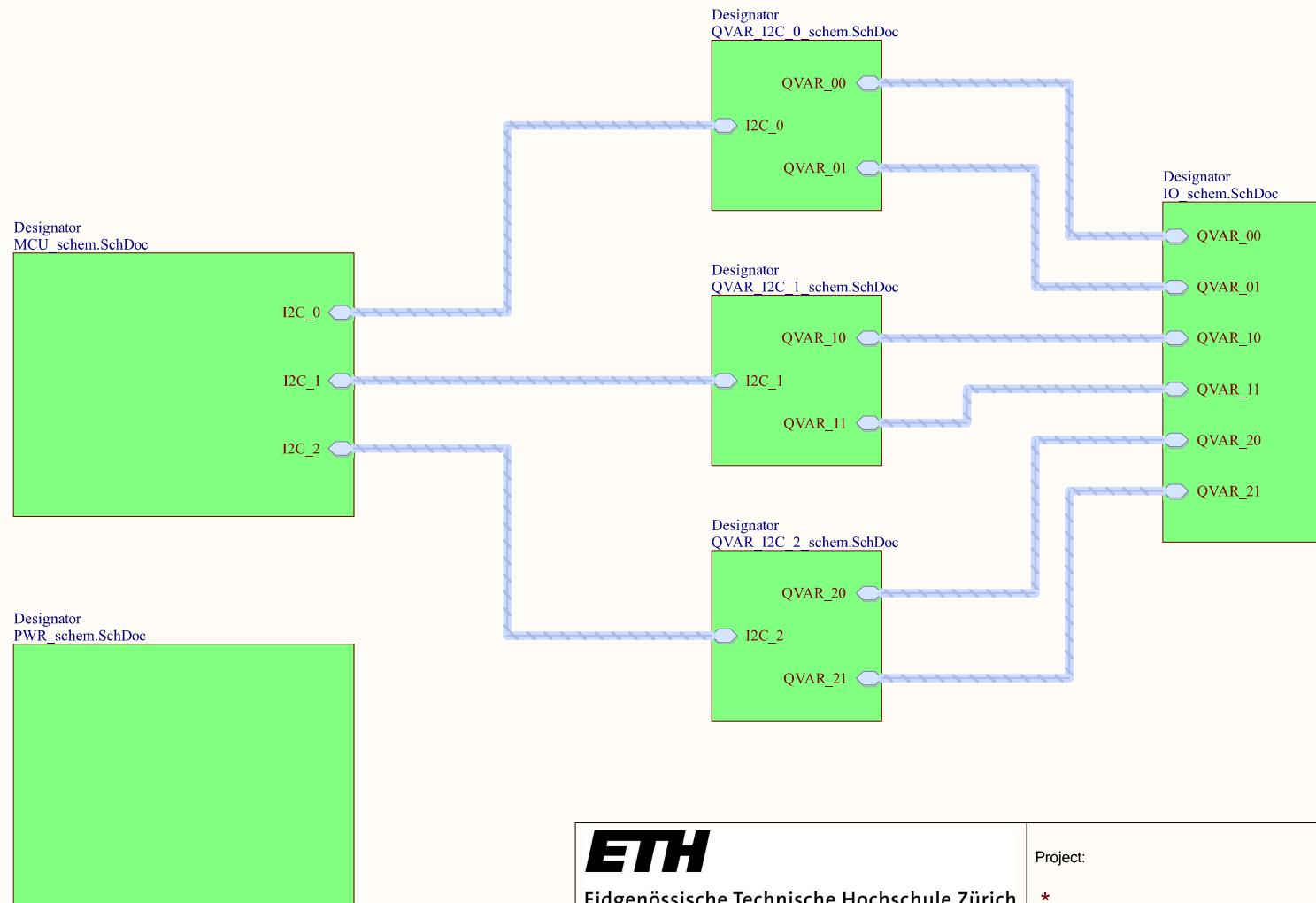
B

C

C

D

D

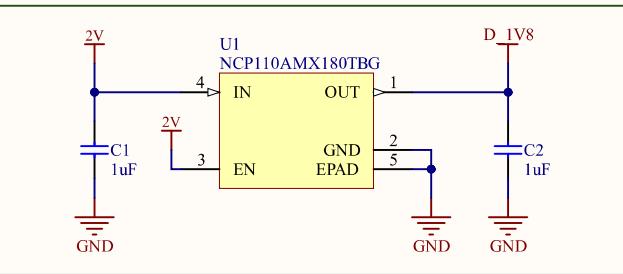


Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

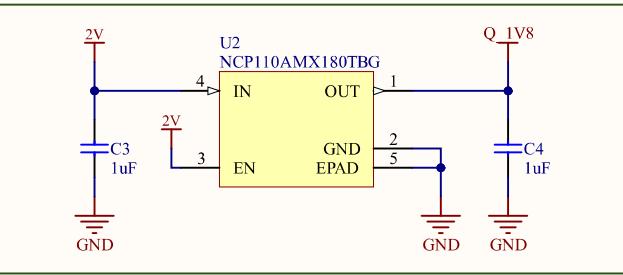
Project:
*

Drawing number: *	Rev: *	Format: A4 Q	Laboratory: *	Sheet: main_schem.SchDoc
Date: 20.01.2022 09:17:50		Drawn by: *		Page 1 of 7
File: C:\Users\rosco\Documents\Master_Thesis\QVAR_Bracelet\main_schem.SchDoc				

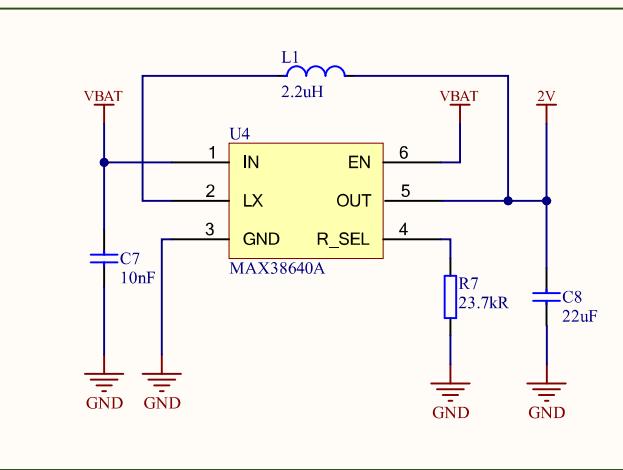
1V8 LDO MCU



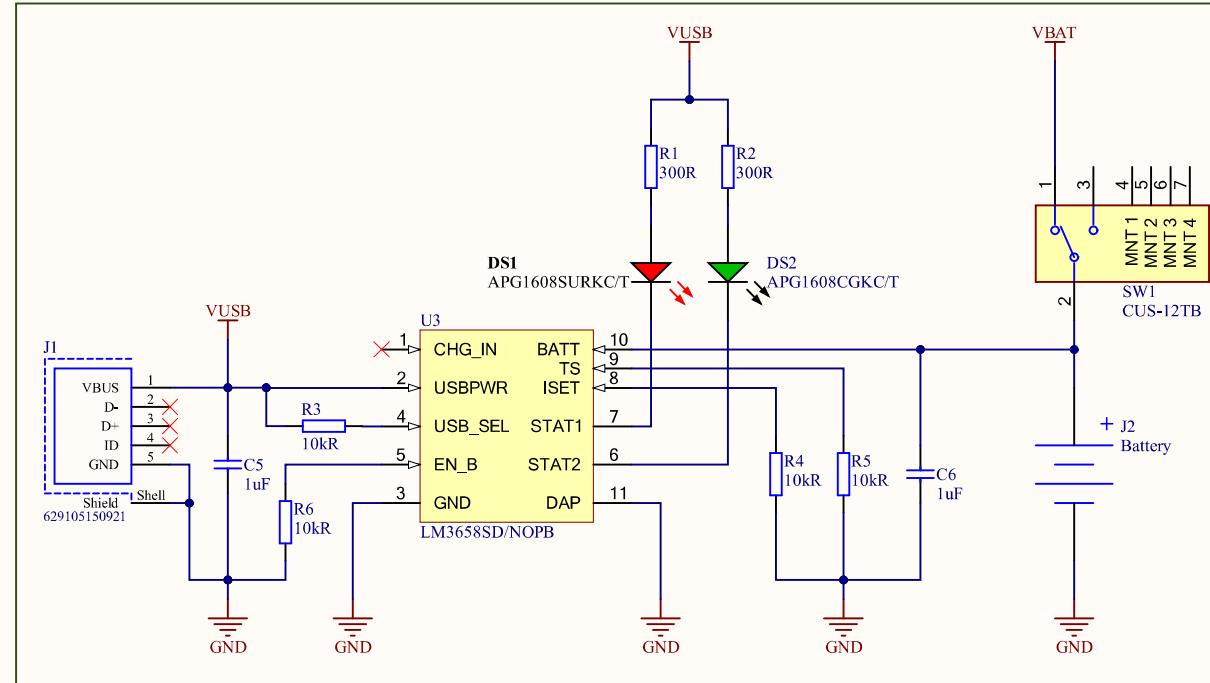
1V8 LDO QVAR



2V Buck Converter



Battery Charger



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Project:

*

Drawing number:	*	Rev:	*	Format:	Laboratory:	*	Sheet:	PWR_schem.SchDoc
Date:	20.01.2022	09:17:51		A4 Q	Drawn by:	*		
File:	C:\Users\rosco\Documents\Master_Thesis\QVAR_Bracelet\PWR_schem.SchDoc							

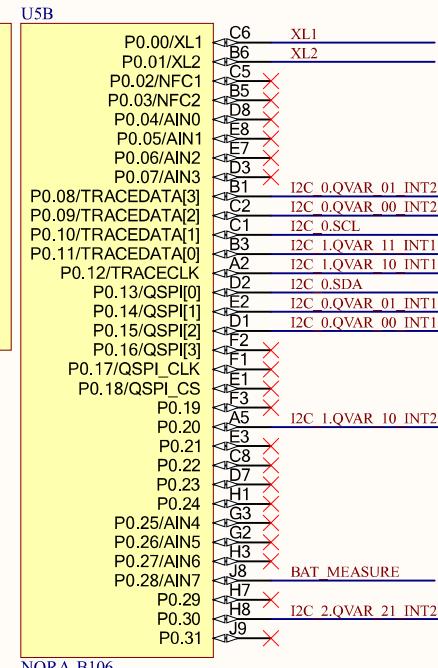
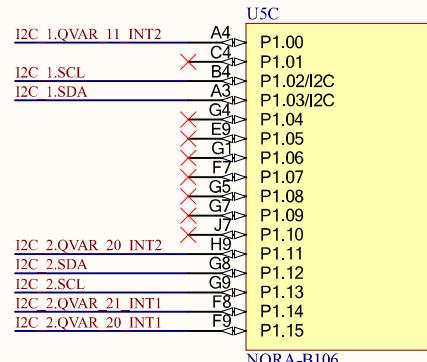
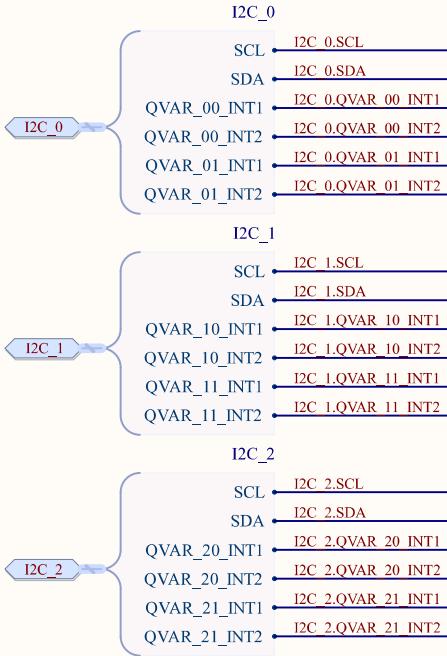
1

2

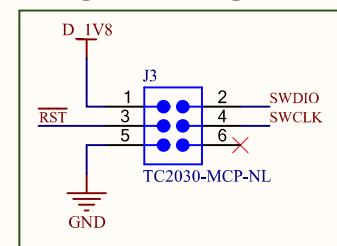
3

4

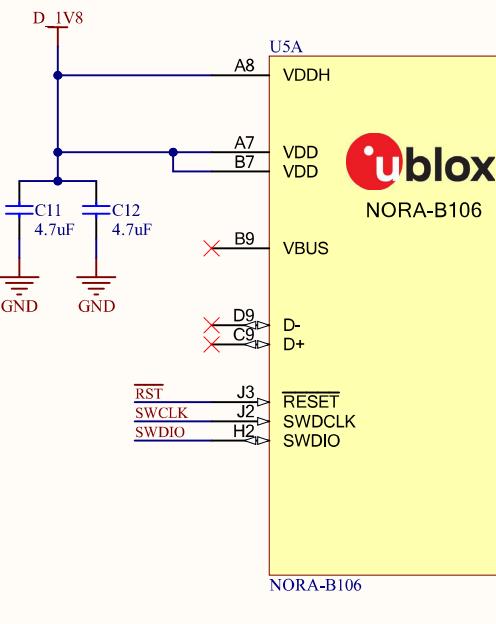
A



Programming Pad



B



NORA-B106

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Project:

*

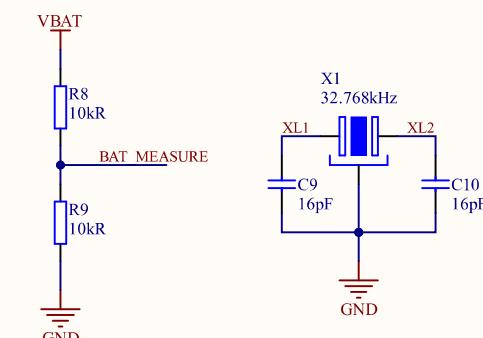
Drawing number:	*	Rev:	*	Format:	Laboratory:	*	Sheet:	MCU_schem.SchDoc
Date:	20.01.2022	09:17:51		A4 Q	Drawn by:	*	Page	3 of 7
File:	C:\Users\rosco\Documents\Master_Thesis\QVAR_Bracelet\MCU_schem.SchDoc							

1

2

3

4



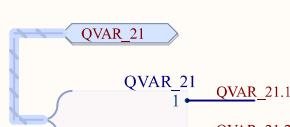
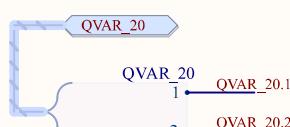
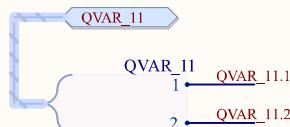
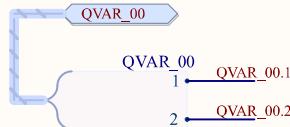
1

2

3

4

A



D 1V8

P?

- 1
- 2
- 3 QVAR_01.1
- 4
- 5 QVAR_01.2
- 6
- 7 QVAR_10.1
- 8
- 9 QVAR_10.2
- 10
- 11 QVAR_00.1
- 12
- 13 QVAR_00.2
- 14
- 15

Castellated Hole

P?

- 1
- 2
- 3 QVAR_11.1
- 4 QVAR_11.2
- 5
- 6
- 7 QVAR_21.1
- 8
- 9 QVAR_21.2
- 10
- 11 QVAR_20.1
- 12
- 13 QVAR_20.2
- 14
- 15

D 1V8

Castellated Hole

GND

GND



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Project:

*

Drawing number: * Rev: *

Date: 20.01.2022 09:17:51

Format: Laboratory: *

A4 Q Drawn by: *

Sheet: IO_schem.SchDoc

Page 4 of 7

File: C:\Users\rosco\Documents\Master_Thesis\QVAR_Bracelet\IO_schem.SchDoc

1

2

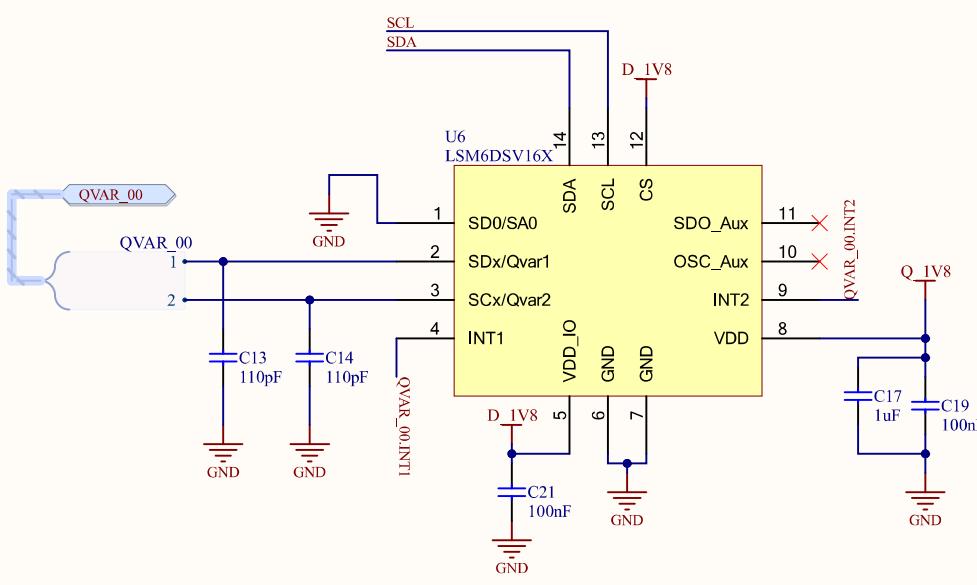
3

4

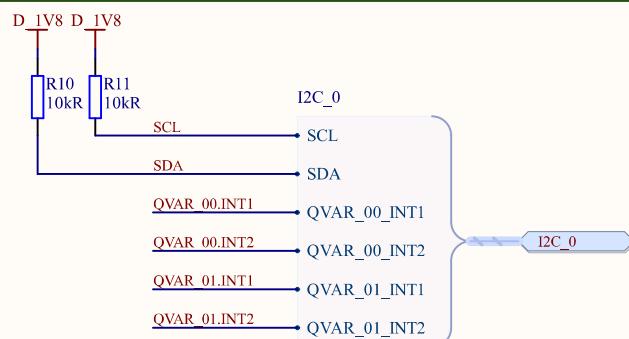
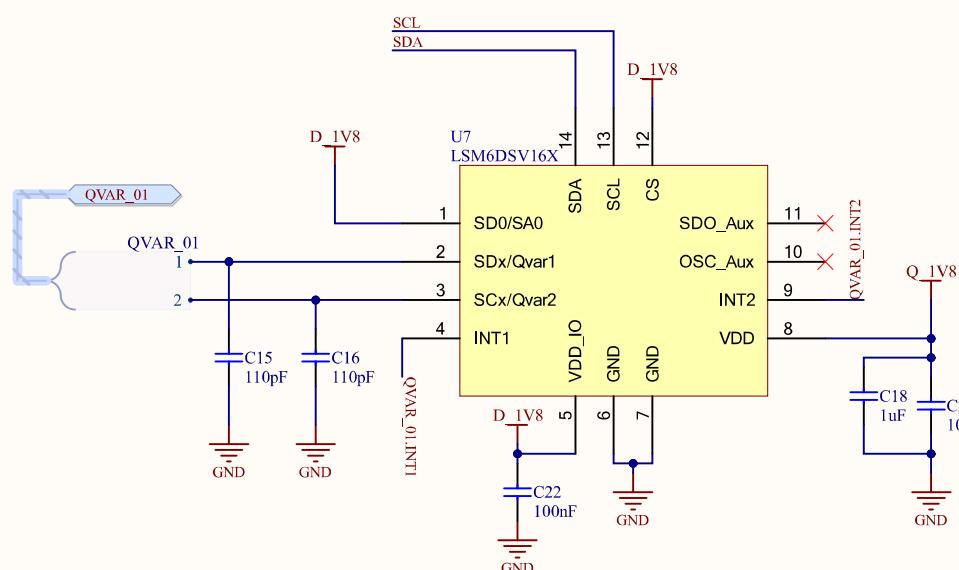
A

I2C_0_QVAR_0

- Mode 1
- 7-bit I2C address: 0b1101010

**I2C_0_QVAR_1**

- Mode 1
- 7-bit I2C address: 0b1101011



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Project:
*

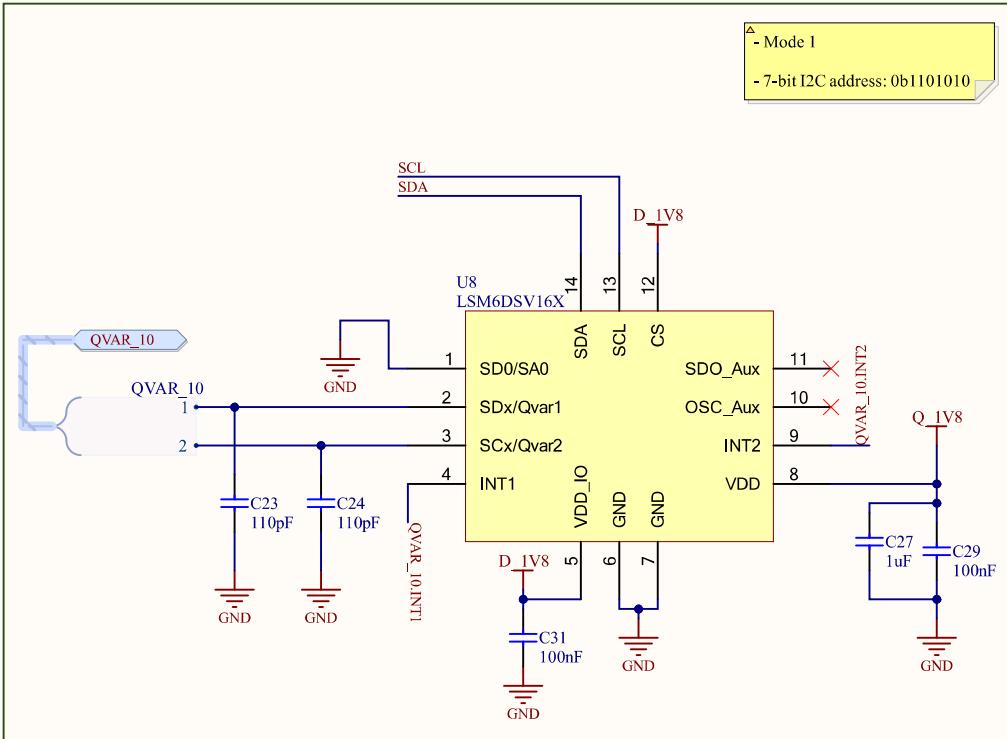
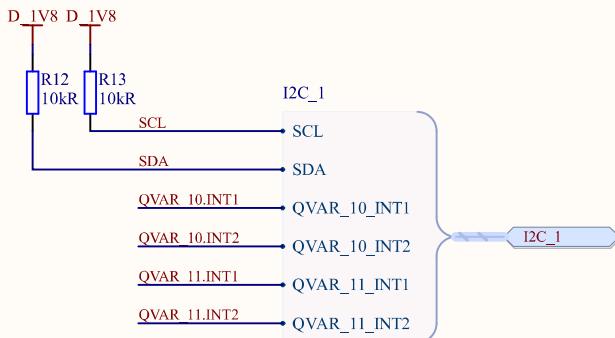
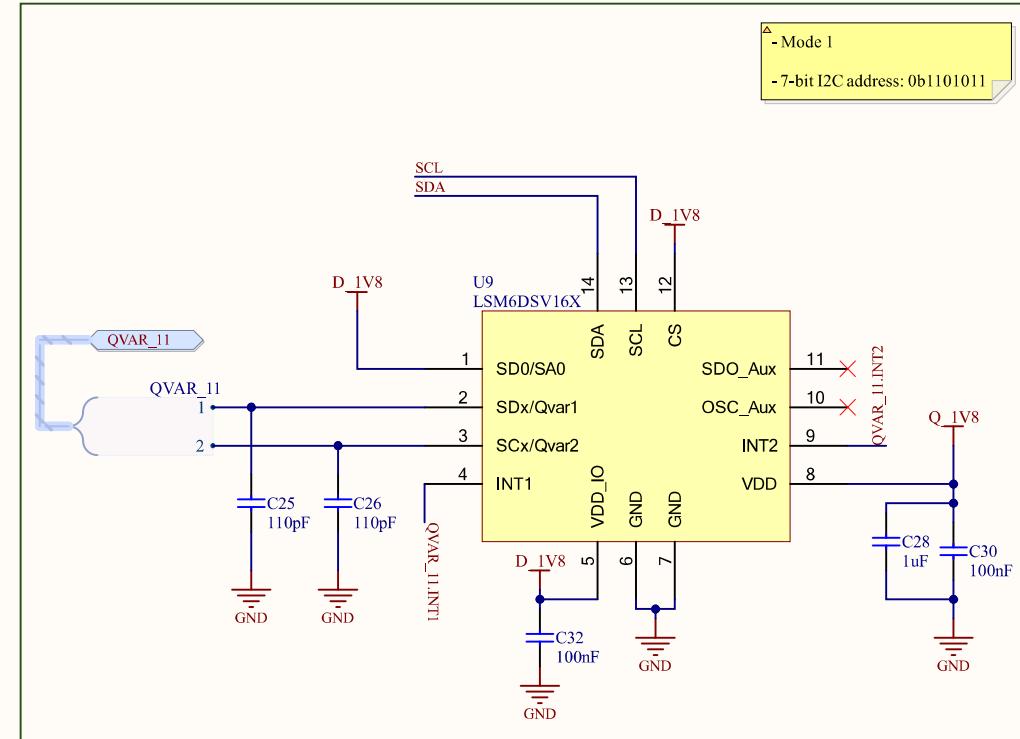
Drawing number:	*	Rev:	*	Format:	Laboratory:	*	Sheet:	QVAR_I2C_0_schem.SchDoc
Date:	20.01.2022	09:17:51		A4 Q	Drawn by:	*	Page	5 of 7
File: C:\Users\rosco\Documents\Master_Thesis\QVAR_Bracelet\QVAR_I2C_0_schem.SchDoc								

A

B

C

D

I2C_1_QVAR_0**I2C_1_QVAR_1**

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Project:

*

Drawing number: * Rev: *

Date: 20.01.2022 09:17:51

Format: Laboratory: *

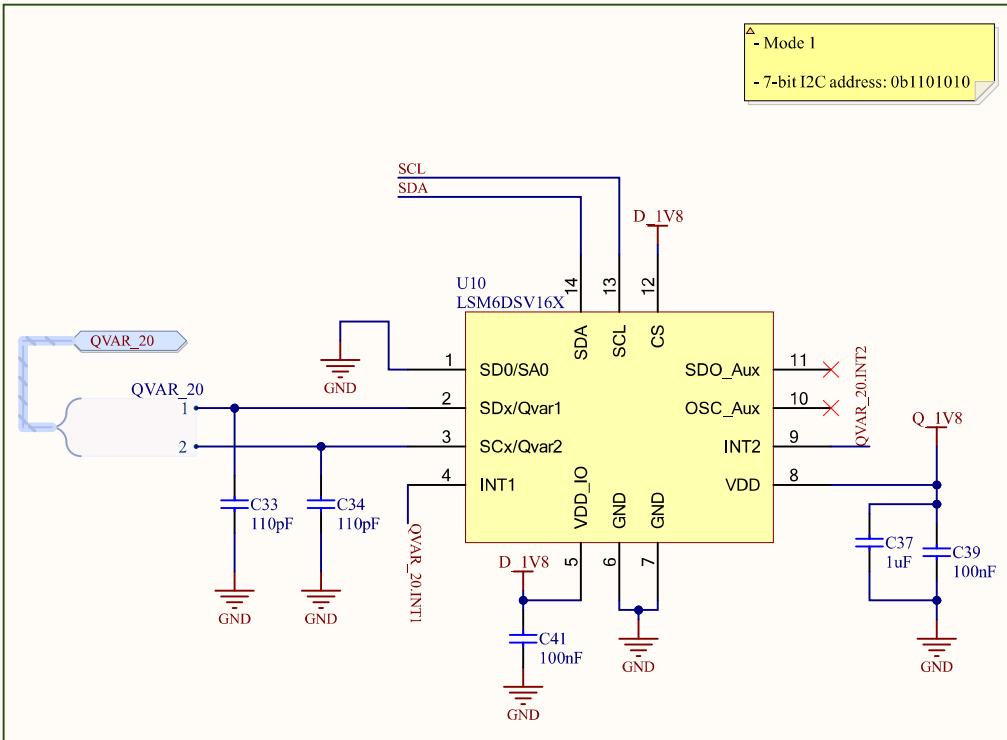
A4 Q Drawn by: *

Sheet: QVAR_I2C_1_schem.SchDoc

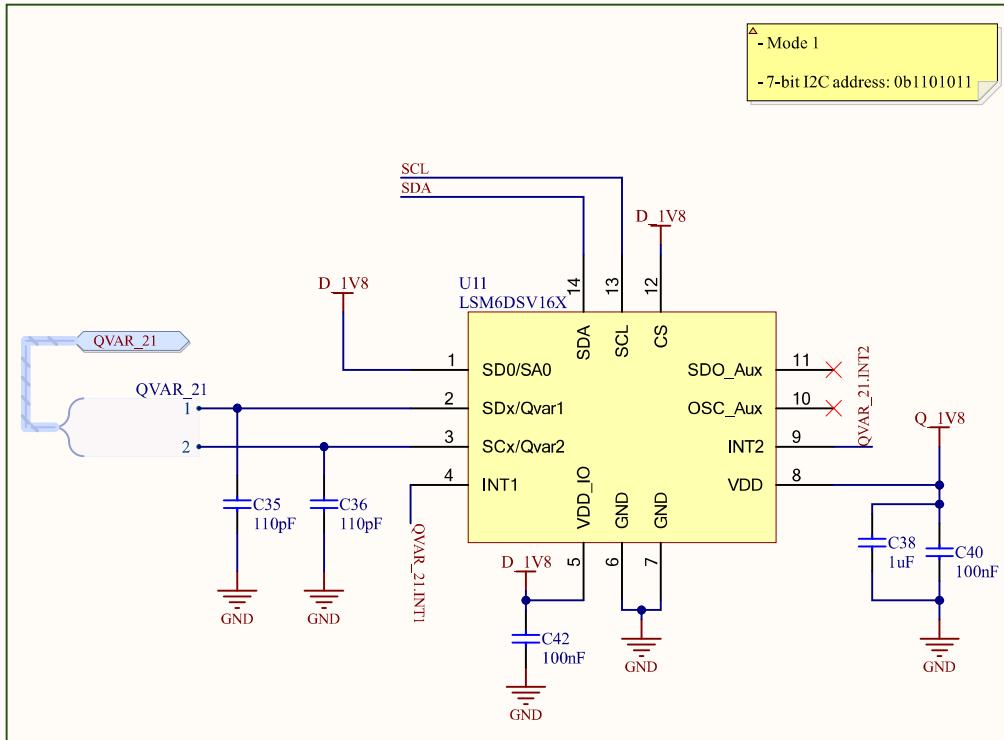
Page 6 of 7

File: C:\Users\rosco\Documents\Master_Thesis\QVAR_Bracelet\QVAR_I2C_1_schem.SchDoc

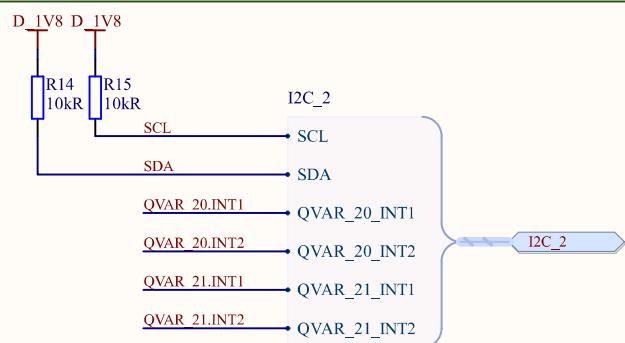
A

I2C_2_QVAR_0

B

I2C_2_QVAR_1

C



D



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Project:
*

Drawing number: * Rev: *

Date: 20.01.2022 09:17:51

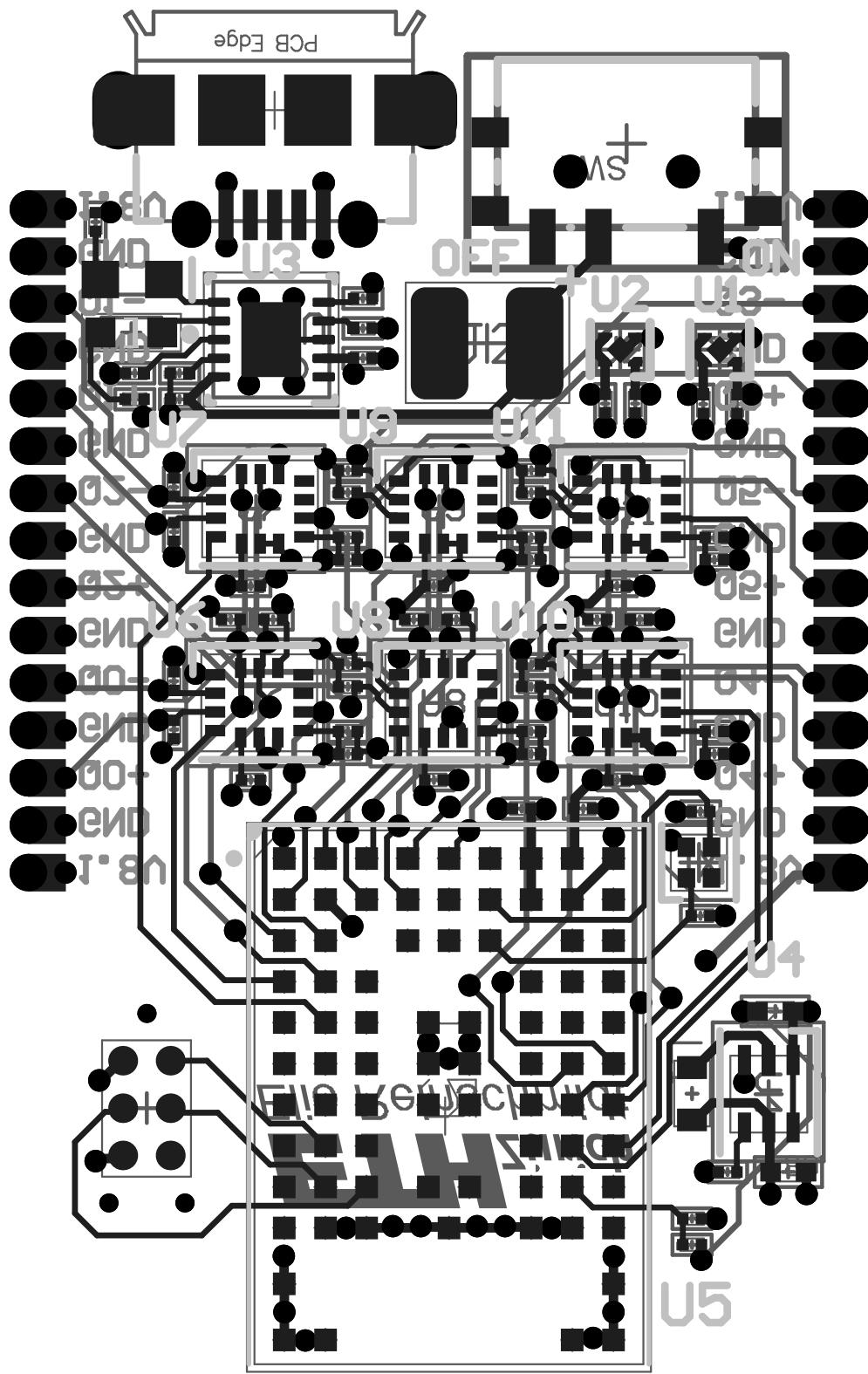
Format: Laboratory: *

A4 Q Drawn by: *

Sheet: QVAR_I2C_2_schem.SchDoc

Page 7 of 7

File: C:\Users\rosco\Documents\Master_Thesis\QVAR_Bracelet\QVAR_I2C_2_schem.SchDoc



Comment	Description	Designator	Footprint	LibRef	Quantity
1uF	* CAP, 1uF, 10V, X5R, 0201	C1, C2, C3, C4, C5, C6, C17, C18, C27, C28, C37, C38	CAP0201	GRM033R61A105ME15D	12
10nF	* Cap, 10nF, 10V, X5R, 20%, 0402	C7	CAP0402	ZRB15XR61A106ME01D	1
22uF	* CAP, 22uF, 6.3V, X5R, 20%, 0402	C8	CAP0402	GRM158R60J226ME01D	1
16pF	* CAP, 16pF, 25V, 2%, 0201	C9, C10	CAP0201	GRM033C1E160GA01D	2
4.7uF	* CAP, 4.7uF, 6.3V, X5R, 20%	C11, C12	CAP0201	GRM035R60J475ME15D	2
110pF	* CAP, 110pF, 50V, 5%, COG/NPO, 0201	C13, C14, C15, C16, C23, C24, C25, C26, C33, C34, C35, C36	CAP0201	GRM033C1H111JA01D	12
100nF	* CAP, 0.1uF, 25V, 20%, X5R, 0201	C19, C20, C21, C22, C29, C30, C31, C32, C39, C40, C41, C42	CAP0201	GRM033R61E104ME14D	12
APG1608SURKC/T	Ultra Thin Hyper Red SMD Chip LED Lamp, 2 to 2.5 V, -40 to 85 degC, Body 1.6 x 0.8 mm, 0.28 mm Thickness, RoHS, Tape and Reel	DS1	KING-LED0603-25-RED_V	CMP-0404-00004-1	1
APG1608CGKC/T	LED, SMT, 0603(1608), 0.25mm Thickness, Green	DS2	KING-LED0603-25-GREEN_V	CMP-0239-00001-1	1
629105150921	Micro USB 2.0 Type AB Receptacle WR-COM, Horizontal, SMT	J1	629105150921	CMP-1502-03249-1	1
Battery	* Pad to solder on battery	J2	Battery_Pad	Battery_Pad	1
TC2030-MCP-NL	CABLE, TAG CONNECT, WITH OUT LEG, FOR IC	J3	TC2030-MCP-NL	CMP-00723-1	1
2.2uH	# IND, 2.2uH, 1.05A, 20%	L1	IND0603	LQM18PN2R2MGHD	1
Castellated Hole		P?	Castellated Holes	Castellated Hole	2
300R	* RES, 300 Ohm, 1%, 0201	R1, R2	RES0201	ERJ-1GEF3000C	2
10kR	* RES, 10kOhm, 1%, 0201	R3, R4, R5, R6, R8, R9, R10, R11, R12, R13, R14, R15	RES0201	ERJ-1GEF1002C	12
23.7kR	* RES, 23.7k Ohm, 1%, 0201	R7	RES0201	ERJ-1GNF2372C	1
CUS-12TB	SWITCH SLIDE SPDT 300mA 4V	SW1	FP-CUS-12TB-MFG	CMP-71745-000005-1	1
NCP110AMX180TBG	LDO Regulator, 200 mA, Low Vin, Ultra-High PSRR 1.8V, Active Discharge	U1, U2	NCP110AMX180TBG	NCP110AMX180TBG	2
LM3658SD/NOPB	Dual Source USB/AC Li Chemistry Charger IC for Portable Applications, 10-pin LLP, Pb-Free	U3	DSC0010A_V	CMP-0063-00780-3	1
MAX38640A	# Buck Converter	U4	uDFN-6_MAXIM	MAX38640AELT+T	1
NORA-B106	u-blox NORA-B106 Bluetooth 5.2 LE module - dual core	U5	NORA_LGA82R_1430X1040X180_ANT	NORA-B106	1
LSM6DSV16X	# IMU, QVAR	U6, U7, U8, U9, U10, U11	LGA-14_STM	LSM6DSV16X	6
32.768kHz	* Crystal 0.032768MHz ±20ppm (Tol) 12.5pF FUND 80000Ohm 4-Pin Ultra Mini-CSMD T/R	X1	ECX-1210B	ECS-.327-12.5-1210B-N-TR	1