Bachelor Project

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

Sensorics equipment for posturometric platform

Adam Šťastný

Supervisor: doc. Ing. Tomáš Haniš, Ph.D.
Field of study: Cybernetics and Robotics
May 2023

Acknowledgements

I would like to thank my supervisor doc. Ing. Tomáš Haniš, Ph.D. for his advice and support through the bachelor's project.

I would also like to express my thanks to my family and friends for their support during my bachelor's project and the entire study.

I would like to thank the Střední průmyslová škola na Proseku, which provided the space and tools for realizing the project.

Last but not least, I would like to thank especially David Laušman and Josef Vágner who cooperated with me on the whole sensor treadmill project.

Declaration

I declare that the presented work was developed independently and I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague, ...

Abstract

The subject of this thesis is the design of a sensor platform for measuring patients' posturometric data. The thesis focuses on the selection of the appropriate sensoric equipment for the platform, the communication protocol for sending the measured data and a designed of printed circuit boards that integrate the mentioned parts – sensor data measurements, processing, and sending.

The thesis uses strain gauges to measure load at the assembly points. The measured data are then sent to the central control unit via CAN bus, which calculates the patient's position on the platform.

The main result is a functional sensor platform that measures the load. The measured data are then sent to the central control unit.

Keywords: sensor platform, posturometry, center of pressure, CAN bus, strain gauge

Supervisor: doc. Ing. Tomáš Haniš, Ph.D.
Katedra řídicí techniky,
Resslova 9
Praha

Abstrakt

Práce se zabývá návrhem senzorické platformy pro měření posturometrických dat pacienta. Práce se zaměřuje na výběr vhodného senzorického vybavení platformy, komunikačního protokolu pro posílání změřených dat a na návrh desek plošných spojů, které integrují zmíněné části – změřená data ze senzorů, jejich zpracování a odeslání.

Práce používá pro měření posturometrických dat tenzometry, kterými měření zatížení v místech montáže. Změřená data jsou následně pomocí sběrnice CAN odesílána do centrální řídicí jednotky, který provádí výpočet pro určení pozice pacienta na platformě.

Hlavním výstupem práce je funkční senzorická platforma, která měří zatížení. Změřená data jsou poté odesílá do centrální řídicí jednotky.

Klíčová slova: senzorická platforma, posturometrie, bod zatížení, sběrnice CAN, tenzometr

Překlad názvu: Senzorické vybavení pro posturometrickou platformu

Contents

Figures

Tables

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Šťastný Adam** Personal ID number: **499335**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Sensorics equipment for posturometric platform

Bachelor’s thesis title in Czech:

Senzorické vybavení pro posturometrickou platformu

Guidelines:

The injuries of the musculoskeletal system and especially of the lower limbs is critical issue in sociate, with high impact on life quality and productivity. Early indication and physiotherapy process monitoring dramatically reduce recovery time. The primary objective of this thesis is to prepare technical equipment for posturometric analysis.

1. Get familiar with posturometric platforms and measurements principles.
2. Develop hardware instrumentation for posturometric platform.
3. Sensorics platform tests execution and evaluation

Bibliography / sources:

[1] Helštýnová, Barbara. Analysis of Posturometric Data. Ostrava, 2012. VSB – Technical University of Ostrava.
[2] Bruštíková Jolana, Use of Tenzometric Platform in Ambulatory Therapy, CTU in Prague, Faculty of biomedical engineering, 2016
[3] Tichý, Lukáš. The effect of a six-week physiotherapeutic intervention in child patients with clubfoot evaluated with a change in ROM of ankle dorsiflexion and the quality of gait measured by 2D analysis. Prague, 2021. Charles Univerisity, Faculty of Physical Education and Sport, supervisor: Nováková, Tereza.

Name and workplace of bachelor’s thesis supervisor:

doc. Ing. Tomáš Haniš, Ph.D. Department of Control Engineering FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **31.01.2023** Deadline for bachelor thesis submission: **26.05.2023**

Assignment valid until:
by the end of summer semester 2023/2024

doc. Ing. Tomáš Haniš, Ph.D. Supervisor’s signature	prof. Ing. Michael Šebek, DrSc. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt	Student’s signature

List of notations

Symbol	**Meaning**
CCU	Central control unit
PCB	Printed circuit control
CoP	Center of pressure
IC	Integrated circuit
UART	Universal asynchronous receiver-transmitter
ADC	Analog to digital converter
SWD	Serial wire debugging
JTAG	Joint test action group
LDO	Low Dropout Regulator
CAN	Controller area network

Chapter 1

Introduction

1.1 Motivation

In life, people encounter a variety of injuries, from minor bruises to fractures and diseases, that can directly impact their mobility. For example, in the case of a broken your leg, attending rehabilitation after treatment becomes essential. Neglecting to do so many results in movement difficulties as the body will reduce strain on the injured leg until the broken leg fully recovers.

Presently, there are issues arising among individuals who took up jogging during the COVID-19 pandemic. The primary reason behind these problems is the improper use of techniques, leading to potential health complications. Improper technique can result in overloading one leg, which increases the risk of adverse effects.

To detect and solve those problems, we could identify the position of the center of pressure (CoP) during movement, which shows us if the person strains one leg more than the other. For that, we can use sensor treadmills that detect the CoP.

1.2 Posturometry

Posturometry, also known as posturography, is used to measure a patient's posture based on measuring his center of pressure on the posturometric platform [1].

1.2.1 Center of mass, center of gravity and center of pressure

The center of mass is a point where the body's mass is concentrated. The center of gravity is the projection of the joint center of gravity of the body into the plane of the support base. The center of pressure is the vector of force that a person creates. For our project, we will measure only the CoP. The main reason is that the center of pressure gives us enough information about the postural stability of the person that we need [2].

1.2.2 Static posturography

Static posturography measures the change in the patient's CoP while standing in one place, which is on the sensor platform. The measured data are then plotted on the graph called statokinesiogram, which shows a change in the CoP [3]. Fig 1.1 shows a measurement of the CoP. Control is human without diseases, and the patient has hemiplegia [4].

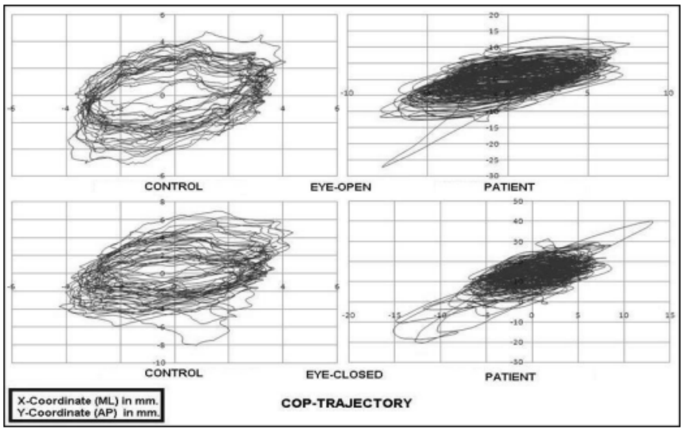

Figure 1.1: Statokinesiogram [4]

1.2.3 Dynamic posturography

This method of posturography uses a mechanical horizontal platform with motors. When the patient is standing on the platform, we measure his CoP. During the test, the platform tilts, and the computer measures patients' reactions. The platform can be combined with a display that simulates visual inclinations as shown in Fig. 1.2. This type of posturography can be used to detect dizziness and disturbance of equilibrium [3], [5].

Figure 1.2: Dynamic posturography platform [6]

1.3 Current offer of posturometric treadmills

The market offers many types of posturometric platforms, including those mentioned earlier for static or dynamic posturometry. Another option is posturometric treadmills.

1.3.1 Classic rehabilitation treadmills

Classic rehabilitation treadmills allow setting belt speed. The treadmill then measures the time of exercise, the patient's pulse, traveled distance, burned calories, and more. Rehabilitation treadmill often includes extra features, like slow start and reverse belt. Those treadmills do not have a sensor platform.

For example, the treadmill from PhysioMill in Fig 1.3 disposes of all features mentioned above and comes with handles for the patient to hold, which helps them with walking.

Figure 1.3: PhysioMill treadmill [7]

1.3.2 Rehabilitation treadmills with a CoP measurement

These treadmills often come with standard treadmill features mentioned in section 1.3.1 and a sensor platform under the belt to measure the CoP.

As an example, Axelo Gait & Balance has all mentioned features and has as well wireless control of a treadmill and wireless tablet to record patients' data and show their progress. The patient and doctor can see the changes in the patient's CoP and other important data such as the number of steps, stride length, average speed, and more.

Figure 1.4: Gait & balance treadmill [8]

Figure 1.5: Gait & balance user interface [8]

1.3.3 Underwater rehabilitation treadmills

Underwater treadmills come in two types - free-standing or integrated into the pool. For example, an underwater treadmill from Hydro Physio is free-standing, which you can see in Fig 1.6.

Figure 1.6: Hydro Physio treadmill [9]

Ewac Medical has an integrated Pooltrack underwater treadmill, which must be directly built into a pool. The downside is the cost of build and the space it will take up in the pool. You can see the treadmill in Fig 1.7.

The solution to this problem is Ewac medical Drop-in underwater treadmill, which can be used as a normal or as an underwater treadmill. The downside is the absence of a sensor platform for measuring the posturographic data.

Figure 1.7: Ewac medical integrated treadmill [10]

Figure 1.8: Ewac medical Drop-in treadmill [11]

1.4 Our treadmill

Treadmill will be equipped with a sensor platform and evaluation equipment like a tablet (central control unit, CCU). The sensor platform must be able to measure the load on the platform and send the data to the evaluation equipment, which will be a CCU. The CCU must be able to read data transmitted by the sensor platform, evaluate the CoP, and show the change of CoP in the display. It must be able to measure the number of steps and the step width. We must create a database with statistics of the exercise of each patient and display statistics data like change in CoP over exercises.

In comparison to commercially available treadmills, our treadmill will be able to measure the static posturography data and CoP of a walking person. This means we can use one treadmill to perform two different exercise types. Our treadmill will be water resistant in the future, and it will be a drop-in treadmill. We also aim to make the treadmill more accessible than manufactured treadmills due to their high price.

1.5 Thesis goal

The objective of this thesis is to develop a sensor platform capable of measuring the CoP of patients while they walk or run on a treadmill equipped with the sensor platform.

My objective is to locate appropriate sensors capable of accurately measuring the center of pressure (CoP). Then, I will choose the necessary electronic components to create a printed circuit board (PCB) that will collect data from these sensors. To ensure the seamless transmission of measured data to the evaluation equipment, which is the CCU in this case, I will develop a suitable algorithm for processing the sensor data. Furthermore, I will research and implement a communication protocol that will facilitate the efficient transfer of the measured data from the PCB to the CCU.

Chapter 2

Architecture

This is a project of three people, Adam Šťastný, David Laušman, and Josef Vágner. In this chapter, we will establish a goal for each. Then we will discuss a selection of sensors for the sensor platform and communication between the board and CCU.

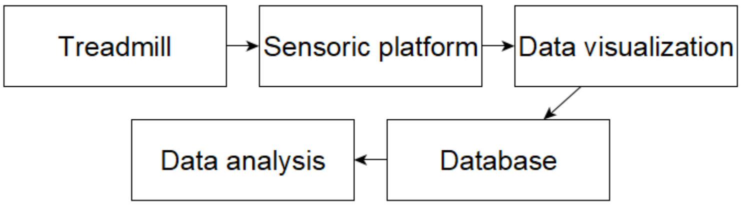

Figure 2.1: System architecture

As you can see in Fig. 2.1, we did the basic system architecture. For measuring CoP we have to do a sensor platform for the treadmill. After measuring CoP the data must be sent to the device that will visualize the data. After that, the measured data must be stored in the database, and for each patient, it must be possible to create statistical data of exercises.

For my part of the project, I will focus on the sensor platform and electronics. I will find the best sensor for our treadmill, create a measuring PCB and connect my PCB electrically to the CCU.

David Laušman will be responsible for the CCU and statistical analysis. He needs to make a PCB that will electrically connect my PCB to the CCU. Then he needs to create a database and calculate the statistics data of each patient.

Josef Vágner will handle the algorithm for measuring CoP. He will utilize data from the sensor board to compute the CoP and gather essential exercise data such as exercise duration, step count, and step dimensions (width or length). Additionally, he will develop a real-time display application for CoP.

2.1 Sensors for posturometric platforms

We require to be able to measure the patients' CoP on the platform. For this purpose we can choose strain gauges, a projected capacity panel, or a resistive panel.

2.1.1 Strain gauge

A strain gauge is used for measuring a change of mechanical strain (mechanical deformation) of material often via electrical resistance. The strain gauge uses a resistive foil on the gauge backing. The strain gauge is glued on a measured object. When the measured object deforms, the resistive foil deforms with it, which means it changes its resistance (resistive foil is bending or stretching)[12].

For measuring the deformation, the strain gauge is connected to a voltage supply, and when the resistance change occurs, the voltage drop across the strain gauge changes. The voltage can be measured, for example, with ADC. The strain gauge is shown in Fig 2.2 [12].

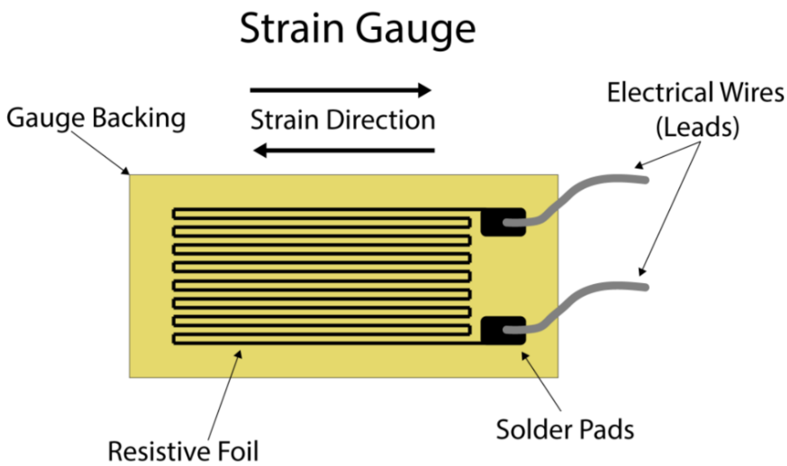

Figure 2.2: Strain gauge [13]

2.1.2 Resistive panel

The resistive panel uses two separated resistive plates with readout strips (see Fig. 2.3). Plates are made from resistive material and are separated by a gas gap and spacers. When we place something on the measuring platform, like a finger, two resistive plates touch, which creates electrical contact in an exact position read by the readout strips [14].

For measuring the position of the touch, there are resistive touch controllers which can be directly connected to the platform. Those drivers can output data with a position of touch through the bus like SPI or I^2C, for example, AR1021IS IC, which has mentioned features [15].

The advantage of a resistive platform is a reading of the position of the touch with electrically non-conductive materials like boots in the case of the treadmill in the exact position. This disadvantage is the lack of measuring force applied in the touch position and low durability due to the overtime deformation of resistive layers [16].

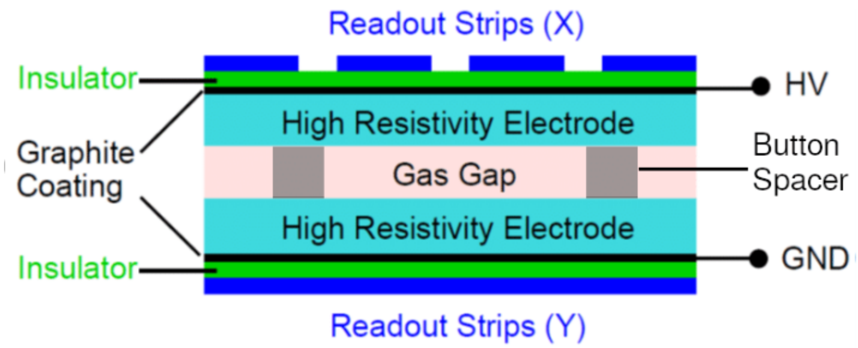

Figure 2.3: Resistive platform [14]

2.1.3 Projected capacitive panel

Projected capacitive panels use thin layers of conducting material. When an electrically conductive material such as a finger touches the platform, it draws a small electrical charge to the point of the touch, see Fig. 2.4 [17].

For measuring the position, there are controllers like resistive platforms. For example, already mentioned driver AR1021 can also be used for capacitive platforms [15].

The advantages of a capacitive platform are high touch sensitivity and accuracy. The disadvantage is the necessity of using an electrically conductive object to read the position and the lack of measuring force of the touch [16].

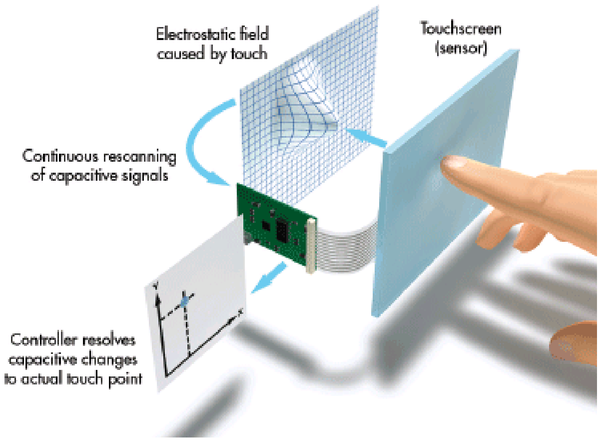

Figure 2.4: Capacitive platform [18]

2.1.4 Sensor for treadmill

For my sensor board, I choose a strain gauge. The main advantage is measuring the force applied on the sensor (resistive and capacitive platform measure only position), durability, and price.

Over time, the resistive display experiences damage due to the load applied to it, while capacitive platforms require a specific electrically conductive material. In addition, these platforms would require unique dimensions, necessitating the creation of a new platform for each treadmill. However, with strain gauges, we can easily modify their position within the program.

For example, strain gauges are also used in Nintendo Wii Balance Board, which can be used as a simple static posturography platform at home. It has four strain gauges in the corners and a load capacity of 150 kg. For measuring posturography on this Nintendo board, many software were developed as HomeBalance or StereoBalance [2].

Figure 2.5: Nintendo Wii Balance Board [19]

2.2 Communication between sensor platform and central control unit

We require communication which will be capable of communicating from CCU to all six strain gauges, which is roughly 10 meters of wire. The treadmill has a brushed DC electric motor which can cause significant interference so the communication must be immune to it. At least the communication must be capable of sending six messages at minimal frequency of 500 Hz where communication speed of 500 kbps will be enough.

Our requirements for communication:

- High-speed communication with a minimum speed of 500 kbps
- Length of bus approximately 10 meters
- Be immune or resistant to interference from the internal motor

2.2.1 UART

Universal asynchronous receiver/transmitter (UART) is a serial, full-duplex asynchronous protocol. UART is using ground signal and two data channels, TX (transmit) and RX (receive) [20].

The disadvantages are a maximum frame length of 8 bits (without parity), hard to use with more than two devices (unreliability), and desynchronization of baud rate between devices if the clocks of communication devices are not accurate enough [20].

2.2.2 RS-232 and RS-485

RS-232 is also known as EIA/TIA-232 standard. RS-232 uses symmetrical voltage levels. Binary 0 is represented with voltage levels +5VDC to +15VDC, and binary 1 is represented with a voltage levels -5VDC to -15VDC [21], [22].

RS-485 is also known as EIA/TIA-485. RS-485 uses differential signals for communication. The main advantage is increased resistance to interference due to differential signals. Physical layers must be terminated with 120 ohms resistors at the start and the end of the line [23], [24].

2.2.3 I^{2}C

I^{2}C uses two wires called SDA (Serial Data Line) and SCL (Serial Clock Line) for communication. I^{2}C uses master/slave communication. The master is sending a clock signal to all slaves. When the master communicates with the slave, the master sends to the SDA slave address, then sends the data [25], [26].

2.2.4 SPI

SPI uses four wires called MOSI (Master Out Slave In), MISO (Master In Slave Out), SCLK (Clock Signal), SS (Slave Select). Based on the number of slaves, the number of wires can be higher - every slave must have a separate SS wire. SPI uses master/slave communication, where the master chooses slaves by logic 0 on the SS wire to selected slave [27], [28].

2.2.5 IsoSPI

IsoSPI (isolated SPI) uses standard SPI communication but adds a differential pair to send the signal instead of SPI's non-differential. The main advantage of isoSPI is an increased distance between devices and better resistance to interference [29].

2.2.6 CAN

CAN (Controller Area Network) uses a pair of wires called CAN high and CAN low. Communication is using differential signals. CAN message is created from ID, control, data, and CRC. Because CAN is not master/slave communication, every device broadcasts its message to the bus with ID, which identifies what is inside the data. Control is used to define how long the message is in bytes [30], [31].

The main advantage of CAN is its high resistance to interference because of the usage of a differential signal.

2.2.7 Communication for treadmill

For our treadmill, I will select the communication that most closely matches the requirements from section 2.2. For this purpose, I created a table comparing the communication protocols mentioned above. The distance is based on speed so I wrote down the distance for the required minimum speed of 500 kbps.

Communication	Speed	Num of devices	Interference resistance	Distance
UART	5 Mbps	2	Low	15 m
RS-232	1 Mbps	2	Low	15 m
RS-485	10 Mbps	32	High	1200 m
I^2C	5 Mbps	127	Low	10 m
SPI	10 Mbps	127	Low	10 m
isoSPI	1 Mbps	127	High	10 m
CAN	1 Mbps	127	High	100 m

Table 2.1: Communication comparison [21], [24], [26], [29], [32], [33]

For our treadmill, I choose the CAN bus. While RS-485, SPI, and isoSPI are also viable options, I have disregarded RS-485 due to its two-wire communication. Utilizing UART for communication between more than two devices lacks a synchronization signal, making it unreliable. Similarly, SPI would require an additional wire for each device, which could lead to a significant increase in wiring complexity when dealing with multiple devices. Lastly, I have not chosen isoSPI as there is limited information available regarding its applications and usage in projects and hardware layout.

Chapter 3

Hardware design

The sensor platform will contain strain gauges and PCBs with a microprocessor unit to communicate via the CAN bus with the CCU.
I need to determine the number of strain gauges, choose the ideal strain gauge and create an electrical schematic with PCB that will measure voltage from the strain gauge and it will send it via CAN bus to the CCU.

3.1 Load sensors – strain gauges

The number of strain gauges is based on the required accuracy and the available mounting positions on the treadmill. The basic platform must use at least four strain gauges in every corner to measure the CoP of the patient. For my sensor board, I have the opportunity to use six strain gauges because the treadmill has six locations where I can mount them, as you can see in Fig. 3.1 or Appendix A. I have decided to use those mounting places and usage of the six strain gauges to get higher accuracy.

I require that a preson weighting 100 kg can run on the belt. I also need to include impacts when the person is running, which means a strain gauge load of 150 kg must be sufficient. The maximal dimensions of the strain gauge are defined by the dimensions of the platform, where the height is limited to 10 mm and the width is limited to 25 mm. For ease of use, I require full bridge configuration and temperature compensation.

Required parameters for strain gauges:

- Force range of strain gauge from 0N to 1500 N (approximately 150 kg)
- Maximal height 10 mm
- Full bridge configuration
- Temperature compensation

Figure 3.1: Position of strain gauges on the treadmill

3.1.1 VPG 380

Strain gauge made by Vishay, Model 380 (Fig. 3.2). The strain gauge is manufactured in seven different types, where PB-250 kg is the most suitable for my use. It has dimensions of 107.3 mm by 70 mm, 10 mm in height. It has a full bridge construction and optional AMP#103957-4 connector. The temperature is compensated for a range from -10°C to +40°C [34]

Figure 3.2: VPG 380 [34]

3.1.2 FX29

Strain gauge made by TE connectivity, model FX29 (Fig. 3.3). The strain gauge is manufactured in multiple versions. The strain gauge can have a load range from 50 N to 1000 N. Dimensions are 19.10 mm by 19.4 mm and 5.45 mm in height. It sells as a standard strain gauge, amplified output, and strain gauge with integrated I^{2}C communication. Output has a 440146-4 connector. It has compensated temperature range from 0°C to 50°C [35].

Figure 3.3: FX29 [35]

3.1.3 FC23

Strain gauge made by TE connectivity, model FC23 (Fig. 3.4). The strain gauge can have a load range from 250 N to 5000 N. Dimensions are 31.75 mm and 10.2 mm in height. The strain gauge comes as a standard strain gauge or with amplified output. The output does not have a connector. The strain gauge has a full bridge construction. It has a mounting screw hole M3 [36].

Figure 3.4: FC23 [36]

3.1.4 Strain gauge for the sensor platform

I selected the FX29 strain gauge because it has all the required features, which are measuring force, height, full bridge construction, and temperature compensation.

From the offer, I choose FX29X-040B-0200-L. It has a maximal force of 1000 N, which is roughly 100 kg and can be overloaded by 2.5 times, which means it will handle the force of 2500 N which is roughly 250 kg. That means in ideal circumstances on the treadmill could stand a human with a weight over 200 kg. It comes as a standard strain gauge without output amplification (the transfer ratio between input and output voltage is 20 mV/V). The cable length is 40 mm with 440146-4 connector. Because I choose the variant without integrated I^2C I will create a output voltage measurement with ADC. Dimensions of the gauge are in Fig 3.5 [35].

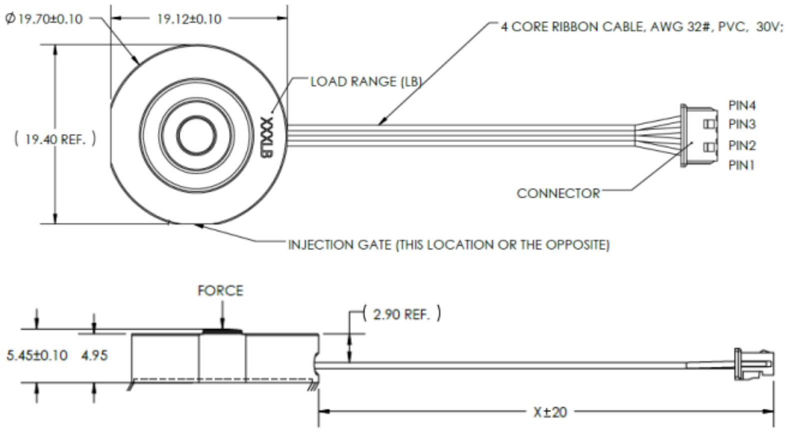

Figure 3.5: FX29 dimensions [35]

3.2 Analog to digital converter

Analog-to-digital converter (ADC) is used for measuring analog data and converting it to digital data that will read, for example, a microprocessor. In my case, I will use ADC to convert the analog voltage value of the strain gauge output to the digital signal for the microprocessor.
ADC must have those parameters:

- 3.3 V
- Minimal sampling speed 1 kS/s
- 24-bit
- Minimal 1 channel with differential input

I selected the 3.3 V version because I want to use a 3.3 V microprocessor so I will have the same supply voltage for the whole PCB. We require to measure the sensor platform data and visualize them at a speed of 100 Hz, which means I need to sample data from the strain gauge at a minimal sampling speed of 100 S/s. I choose 1 kS/s because I want to measure some values from the strain gauge and average them out before sending them to the CCU.

3.2.1 Choosing ADC converter

AD converters I was considering are listed in the Tab. 3.1.

Marking	Voltage [V]	Speed [kS/s]	Built-in reference	Noise [nV]
MCP3561-E/ST	1.8–3.6	153.6	No	90
ADS1259IPW	2.5–5.0	14	Yes	700
ADS1254EG4	1.8–3.6	20	No	50
ADS131M04	2.7–3.6	32	Yes	5350
AD7766BRUZ	1.7–3.6	128	Yes	20000
PCM4201PW	1.8–5.0	108	Yes	50
ADS1254E	0–6.0	20	No	70

Table 3.1: ADC comparison [37]–[42]

I choose ADS131M04. It has a higher sampling speed than ADS1259IPW, ADS1254EG4, and ADS1254E, it has an internal voltage reference of 1.2 V, and it has a programmable gain amplifier of input voltage (external differential amplifier not required). It has a higher noise ratio than MCP3561-E/ST, but in my application, the noise ratio of ADS131M04 is acceptable. It has SPI communication, which is better than I^2C for my application because it is less likely to get interference from the motor, which was mentioned in section 2.2.

3.3 Measurement unit for strain gauge

For the measurement unit, I can choose from different control devices like a single-board computer or a microprocessor unit. Single-board computers, for example, Raspberry Pi is not suitable for my application because they must run an operating system. This causes a slow system loop which would also slow down my measurements. Microprocessors do not have an operating system the only thing that runs on them is the program, which increases the computional power. For my application, I will use MCU.
There are many MCU manufacturers, from which I have chosen STM because I have obtained previous experience. For example, other manufacturers are Athmel and Microchip.

My determination to choose STM is simple testing on the breadboard with the development board called Nucleo. They have many types of processors, from low-power 8-bit to high-end 32-bit. The disadvantage is availability on the market associated due to the chip shortage.

3.3.1 STM microprocessor

STM has two main series of processors - STM8 and STM32. For my application, I will use the 32-bit variant, which offers more computation power than an 8-bit variant.
STM32 has a several series of their microprocessor. They are produced in the following series [43]:

- High performance - STM32F, STM32H
- Mainstream - STM32G, STM32F
- Ultra-low power - STM32L, STM32U
- Wireless - STM32W

For my application, I will consider only STM32F variants with STM32G variants. Because I will use SPI to communicate with ADC and CAN to communicate with the CCU, I require STM, which will have those communication protocols. In the table below, I will mention only STMs with those communication capabilities.

Marking	Clock freq [MHz]	Flash [kB]	RAM [kB]
STM32F2	120	128–1000	up to 128
STM32F4	180	128–1536	up to 320
STM32F7	216	256–2000	256–512
STM32H5	250	128–2048	32–640
STM32H7	240	up to 2 MB	564kB–1,4MB
STM32F1	72	16–1000	4–96
STM32F3	72	16–512	16–80

Table 3.2: STM32 comparison [44]–[50]

From table 3.2, I selected STM32F4, which has enough flash memory and RAM and has a 180 MHz clock CPU. I choose this STM also because I have STM32F446RET6 in my Nucleo-F446 board. I will program and test the board on STM32F446RET6, but I will choose other variants of STM that I can use instead of F446. The main reasons are availability, price, and performance. A weaker STM should do the job, as I don't plan to do any special calculations.

Other STMs must have the same footprint as STM32F446 to have the same PCB design.

Every version of STM32F4 has a compatible footprint with F446. From the datasheets, I read that STM32H573xx, STM32H562xx, STM32H563xx, STM32H503xx, STM32H7B3x, STM32H7A3x, STM32H7B0x have compatible footprints. STM32F1, STM32F3, and STM32H7 have slightly different footprints, which means PCB redesign, but then it should work as F446.
I bought those STM32 for testing:

- STM32F446RET6 - 180 MHz, 512 kB, RAM 128 kB [45]
- STM32F446RCT7 - 180 MHz, 256 kB, RAM 128 kB [45]
- STM32F412RET6 - 100 MHz, 512 kB, RAM 256 kB [45]
- STM32F401RDT6 - 84 MHz, 384 kB, RAM 96 kB [45]
- STM32F334R8T6 - 72 MHz, 64 kB, RAM 12 kB [50]
- STM302RCT6TR - 72 MHz, 256 kB, RAM 40 kB [50]
- STM32F105RBT6 - 72 MHz, 128 kB, RAM 256 kB [49]
- STM32F103RCT7TR - 72 MHz, 256 kB, RAM 48 kB [49]

3.4 Linear voltage regulators

Linear voltage regulators (LDO) are used for regulating the high voltage at the input to the lower voltage at the output. In my case, I will use LDO to get 3.3 V for powering my PCB because the PCB will have 5 V supply from CCU and the architecture of PCB (CPU) is 3.3 V. This voltage will power the MCU, ADC, and CAN transceiver. ADC has two supply voltages called analog and digital supply. It is recommended to use two separate LDOs for each supply. I will use the same LDO for MCU, CAN, and ADC digital supply and extra LDO for ADC analog supply.

For powering MCU, I need approximately 100 mA, and for ADC, I need approximately 10 mA, so 150 mA LDO should be enough.

I choose ADP122AUJZ-3.3, which has 3.3 V and 300 mA at the output. I choose 300 mA because there could be current spikes from MCU and CAN transceiver. This LDO has a 5.5 V input and a low quiescent current [51].

For the analog power of ADC, I choose the same IC – ADP122AUJZ-3.3.

3.5 Reference voltage supply

Those supplies are used to create accurate output voltage with minimum voltage drift. I use this supply for the strain gauge because I want as accurate results as possible.

I choose LM4132EMF-3.3, which has a 5.5 V input voltage, an output voltage of 3.3 V, and an inaccuracy of 0.5 %. The output current is 20 mA. Output voltage drift is 25 ppm/mA [52].

3.6 CAN transceiver

STM has built-in CAN, but the output signal is not differential, which means I need to add IC that will create a differential signal from the STM CAN signal. CAN transceiver must be compatible with the CAN voltage that I will communicate with the CCU. We agreed on bus voltage 3.3 V because RPi and STM can accept 3.3 V input. I choose SN65HVD as my CAN transceiver, which matches my requirements [53].

3.7 Connectors, active and passive parts

As connectors, I will use the same connectors for the whole PCB. I used the same connectors that are used at the strain gauge, which are 440146-4 connectors manufactured by TE connectivity. For debugging, I am using an IDC-10 pin two-series connector.

When programming the STM32, I utilized connectors from Tag-Connect. These connectors offer a significant advantage in terms of saving space on the PCB. Unlike traditional connectors that require physical placement on the PCB, Tag-Connect connectors only require the creation of pads on the PCB where the Tag connector will be used for programming purposes. From their offer, I choose the connector TC2050-IDC-NL-050-STDC14, which can be directly plugged into the ST-Link V3, which is a programmer for STM processors. You can see the 6-pin variant of the Tag connector in Fig. 3.6 [54].

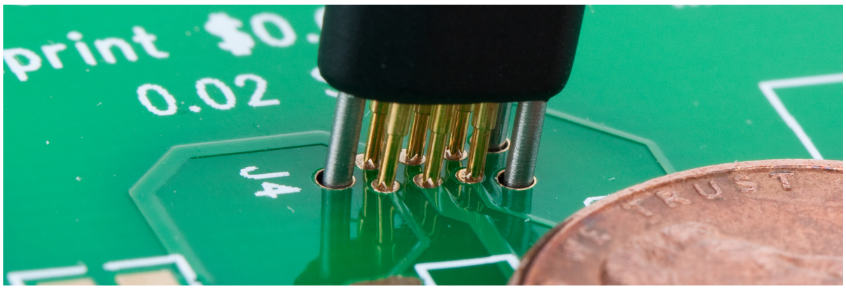

Figure 3.6: 6-pin Tag connector [55]

As active parts, I choose oscillators for STM. I use ECS-80-10-30B-CKM-TR which is an 8 MHz crystal and ECS-.327-9-1210-TR which is a 32.768 kHz crystal [56], [57]. The 8 MHz crystal is mandatory to create a stable MCU clock signal. The 32.768 kHz crystal is used to generate a clock signal for an internal timer.

Then I choose the ESD diode for CAN communication which is used to suppress the transient voltage of the CAN bus. The diode I use is ESDCAN01-2 [58].

As passive parts, I must choose resistors and capacitors. Resistors are SMD 0805 with a 100 mW power rating. As capacitors, I used two types of dielectric based on the wanted accuracy of capacity. Capacitors for oscillators have C0G dielectric, and other capacitors have X7R dielectric. All capacitors are rated for 25 VDC.

Chapter 4

Measurement electronic design

In this chapter, I will set out the requirements for a schema and printed circuit board. I will then create the schema, test it with the development board and create a printed circuit board.

4.1 Schema

Before creating a PCB for the sensor board, I must make a schema. For this purpose, I choose the software Altium Designer, version 23.3.1 (Build 30), a student license in which I will create a schema and PCB as well. The schema will contain all the electrical parts mentioned in previous chapters. The schemas will be divided into sections based on the purpose of each section.

At each supply voltage input, I added one or two more capacitors that manufacturers recommended. The reason is better filtration of input voltage.

I will start with the AD converter, MCU and CAN, then add voltage sources and connectors.

4.1.1 Schema for MCU STM32F446RE

When creating the schema for STM32F446RE, I first chose pins that are connecting together MCU, ADC, CAN, and UART. Then I added pins for programming STM32, where I use SWD programming protocol. The main advantage lies in the capability of real-time debugging for STM32, which could be more beneficial compared to other possible choice, JTAG. After that, I added two push buttons for resetting MCU and booting MCU. I won't use that method for uploading the program, but I should be able to upload the program in case I have no STLink. Lastly, I added capacitors with a capacity of 100 nF between every VDD and VSS, VDDA, and VSSA, which the manufacturer recomended. At VDDA and VSSA, I also included a capacitor with a capacity of 4.7 μF, and at VCAP I added a capacitor with a capacity of 2.2 μF. The manufacturer recommends those values.

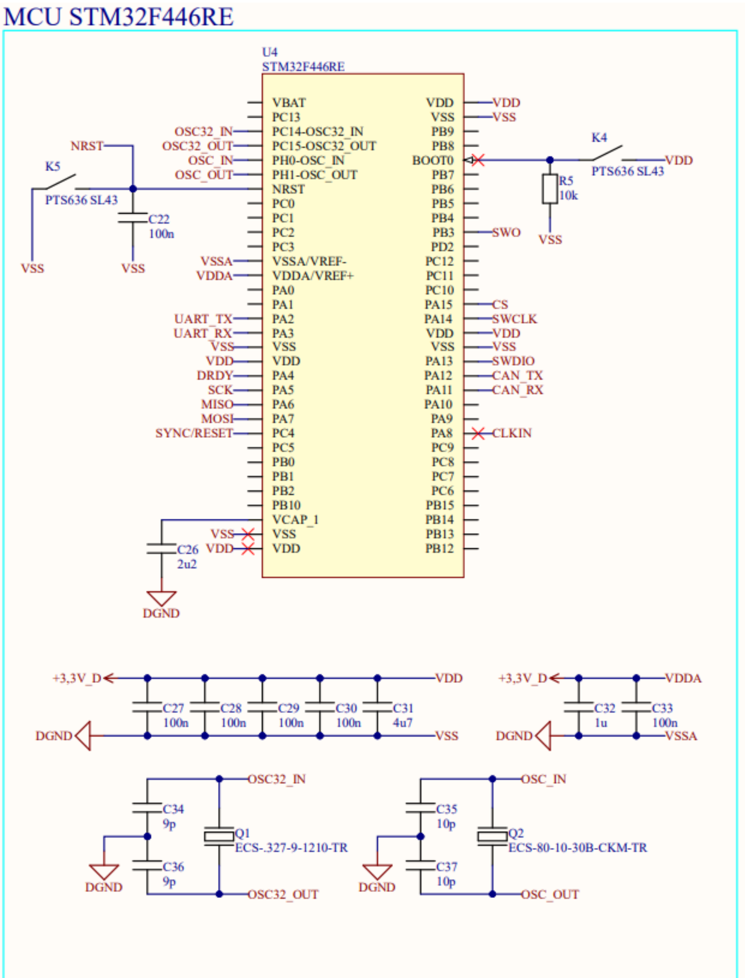

Figure 4.1: Schema for MCU STM32F446RET6

4.1.2 Schema for ADC

ADS131M04IPWT is connected to MCU via CLKIN, MOSI, MISO, SCK, DRDY, CS, and SYNC/RESET pins. For the power supply, I used AVDD, AGND and DVDD, DGND, which require 3.3 V input. Both inputs are isolated and share only the same ground signal. For filtration of the input signal, I used a capacitor with a 1 μF capacity defined by the manufacturer. The manufacturer also requires a 220 nF capacitor at pin CAP (external reference capacitor).

In order to read the voltage from the strain gauge, I used channel 0 labeled as AIN0P (positive input) and AIN0N (negative input). As the manufacturer requires, I added two 1 kΩ resistors and a 10 nF capacitor at those pins.

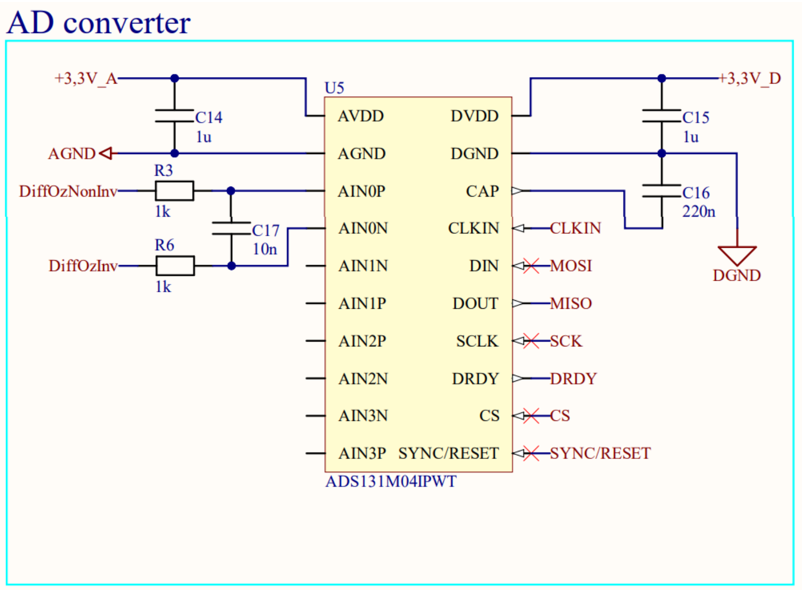

Figure 4.2: Schema for ADC ADS131M04IPWT

4.1.3 Schema for CAN to CAN differential

From MCU CAN, I create a differential CAN signal with SN65HVD230. Output from STM32 is connected to the D and R pins of the SN65HVD230. The supply voltage has three capacitors with capacities of 10 nF, 100 nF, and 1 μF. The main reason for those capacitors are spikes that are generated when communicating through CAN.

CAN must be terminated at the end of the bus with 120 Ω resistors to eliminate crosstalks. For that, I added two 60 Ω resistors and two 0 Ω resistors at the output. In the PCB at the end of a bus, I soldered two 0 Ω to make the termination, otherwise, those resistors won't be soldered. Resistors are connected with 4.7 nF capacitors to the ESDCAN diode to protect the PCB from electrostatic discharge. 100 pF capacitors at CANH and CANL are used to remove spikes from the bus.

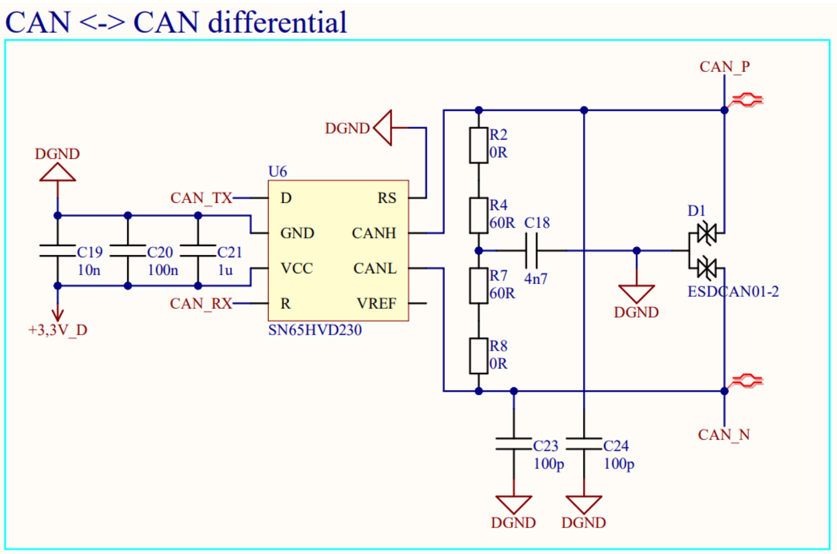

Figure 4.3: Schema for CAN to CAN differential

4.1.4 Voltage supplies

I created three voltage supplies. The first voltage supply is the digital supply which is used for powering ADC, MCU, and CAN. The second voltage supply is the analog supply which is used for powering ADC, and the third voltage source is the referential voltage supply for the strain gauge.

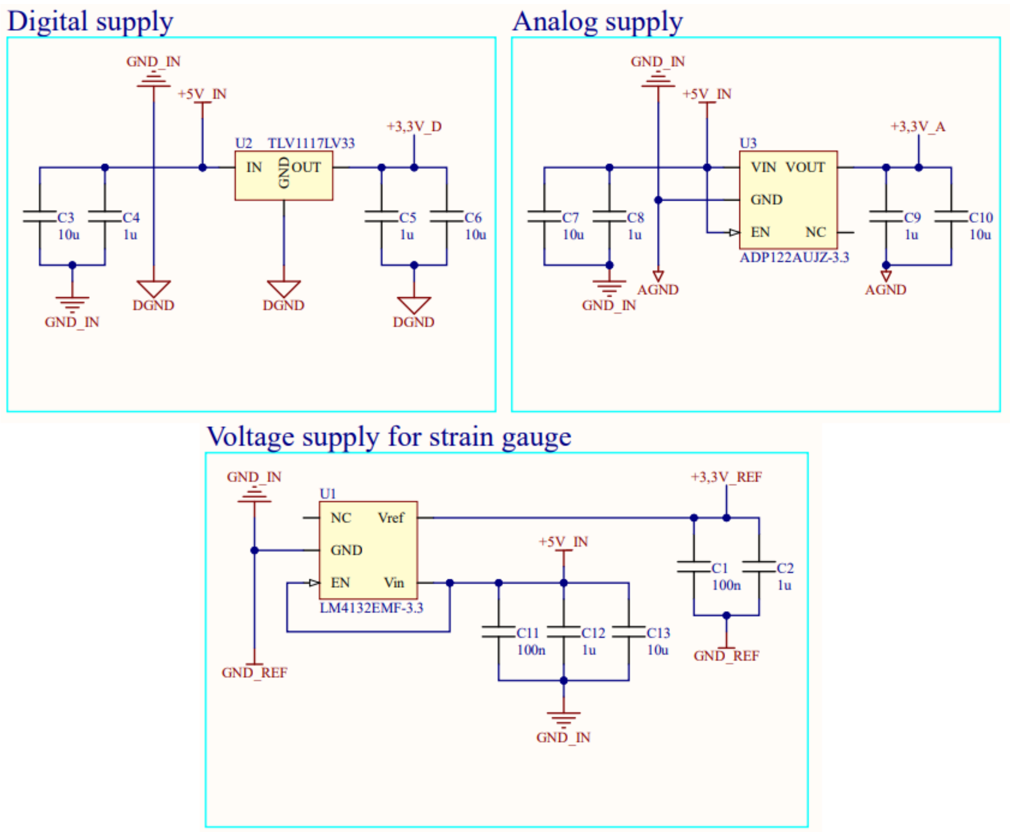

Figure 4.4: Schema for voltage supplies

4.2 Connectors

I use two connectors to connect the PCB to the CCU via CAN and one connector to connect the strain gauge. I also use an IDC-10 pin connector to allow connecting and debugging without STLink. For programming, I created an SMD layout for the Tag-Connector. I also create an SMD layout for the Tag-Connector for debugging.

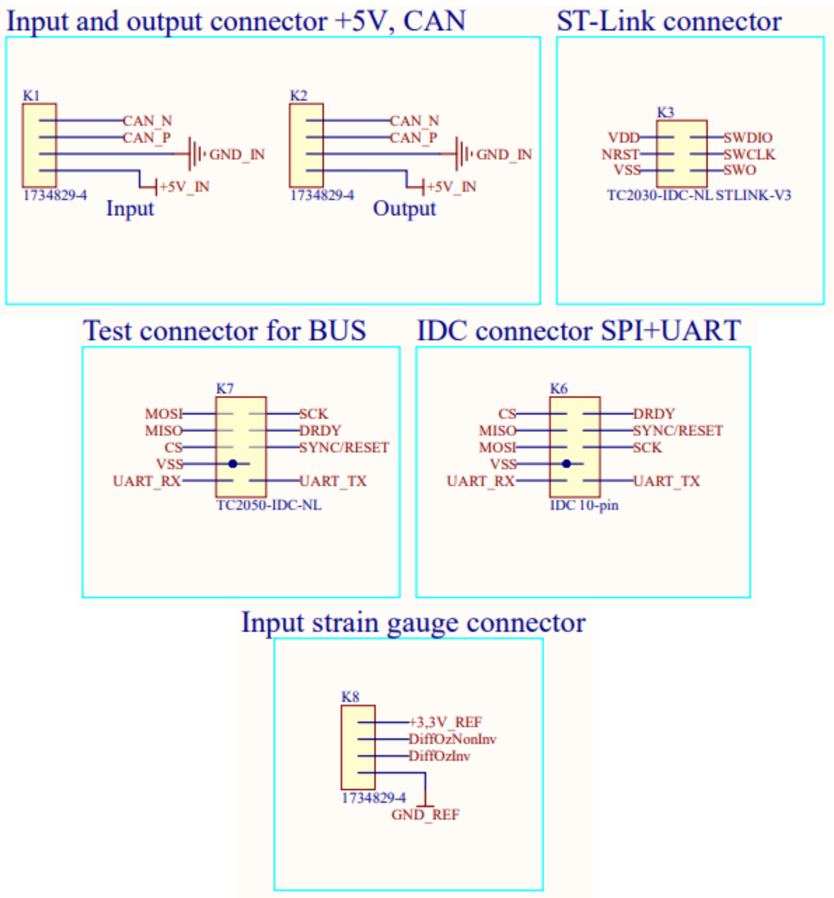

Figure 4.5: Schema for connectors

4.3 Testing on NUCLEO F446 development board

Before creating and manufacturing PCB, I will test my schema with the breadboard and Nucleo development board to find out if everything I designed will work. Firstly I created a program for ADC and tested its response while measuring voltage from the laboratory power supply. After that, I added a voltage reference and strain gauge to test whether the ADC will respond correctly once I load the strain gauge. Lastly, I tested the communication between Nucleo and the CCU via CAN. All tests passed, and I started making the PCB.

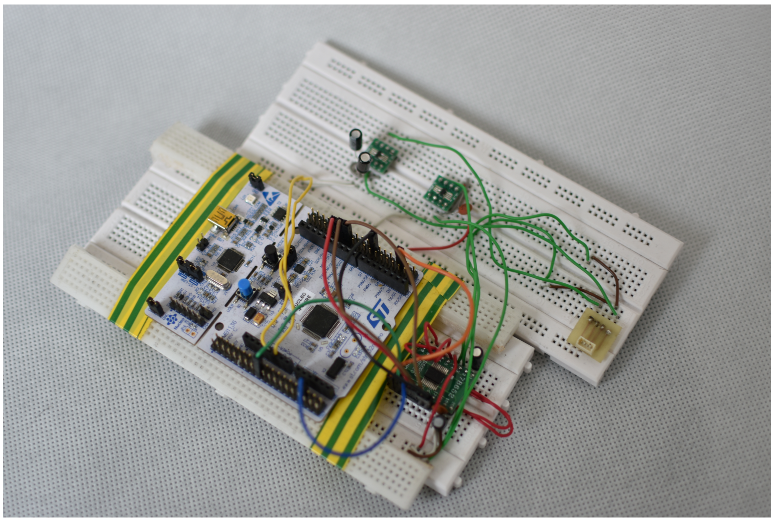

Figure 4.6: Breadboard with Nucleo, testing of ADC and strain gauge

4.4 PCB requirements

The PCB must be able to withstand interference, especially from the motor. It must be able to communicate with SPI, CAN bus, and UART without any errors. PCB's maximal dimensions must be 110 mm x 25 mm, which is limited by the platform's design. The maximal height is limited to the height of the strain gauge, which is 5.35 mm.

The initial step is to choose a manufacturer who will produce my PCBs because it will determine the production options I might follow. From manufacturer, I will choose the PCB material, the number of layers, trace width, and trace distance, and the parts placement on the PCB.

4.4.1 PCB material and number of layers

As a PCB manufacturer, I choose JLCPCB because I have a good personal experience with them, and their production speed is 7 to 14 days (from sending files to receive the package).

As a material for my PCB, I can choose only FR-4 (Flame retardant with glass fiber fabric) material from JLCPCB because the don't offer any other type. The advantages of this material are low price and high dielectric strength [59].

For the number of layers, I am choosing from 2-layer, 4-layer, and 6-layer PCB. 2-layer PCBs are dedicated to the design of slow-speed applications. 4-layer PCBs are dedicated to designs with low supply impedance, and 6-layer PBCs are dedicated to designs with high supply impedance. The main advantage of 4-layer and 6-layer PCBs is better EMC, which is electromagnetic compatibility (the device is compatible with the electromagnetic environment where it will be functioning, and it does not produce high electromagnetic interference) [60]. In a 4-layer PCB, the layers are folded like signal 1, ground, supply, and signal 2, and in 6-layer PCB layers are folded like signal 1, ground, signal 2, signal 3, supply, and signal 4. For my application, I will use a 4-layer PCB, which is better for EMC, and two signal layers should be enough because I won't have so many traces [61].

Other parameters I can choose while manufacturing PCB are outer and inner copper weight and layer stack up. Copper weight means how high is the copper layer on PCB. I choose basic values of 1 oz (approx 35 um) as the outer layer and 0.5 oz (approx 15 um) as the inner layer. The weight of copper doesn't affect PCB's electrical parameters too much, it only matters if I want to run a high current through PCB, where more copper means I can run a higher current on the same track width [62]. As a material, I choose FR-4 with a height of 1.065 mm and the dielectric between the inner and the outer layers with a height of 0.21 mm.

As the last parameter, I can choose the color of the solder mask, which is the cover layer for the PCB that prevents oxidation of the PCB and allows easier soldering. For the first test, I choose a blue solder mask, and for the final PCB, I choose a black solder mask. PCB color is only the design thing and doesn't affect PCB's function.

4.4.2 Trace width and trace distance, design rules

The initial step is to set the trace width and distance between them. I will use the same trace width and distance for all traces except communication and supply. Standards like IPC-2221 and IPC-2152 define those parameters. For my PCB design, I choose IPC-2221. I used a calculator from [63] for trace width. I choose current up to 0.25 A and maximal temperature rise 10°C with ambient temperature 40°C. The required trace width is 0.12 mm for internal traces and 0.05 mm for external traces. For easier possible repairing of traces, I choose higher width of 0.254 mm. The main advantage of wider PCB traces is lower resistance. The minimal distance between traces is 0.05 mm based on the IPC-2221 table from [64].

Vias, which are holes that are used to connect traces through PCB, must be designed to withstand current flowing through them. For vias, I used drills with a diameter of 0.6 mm. Those vias can withstand 2.4 A, which I calculated with [65]. As plating thickness, I used 18 um, which is the thickness that chosen manufacturer created.

Other design rules are defined by the manufacturer. As the important rules, I will mention the drill hole size 0.15 mm - 6.3 mm, minimum annular ring 0.13 mm, minimum hole-to-hole clearance 0.5 mm, minimum pad-to-pad clearance of 0.127 mm, minimum via-to-pad clearance of 0.254 mm, a minimum spacing between traces 0.09 mm (for 4-6 layers), solder bridge 0.254 mm, trace to outline distance 0.2 mm.

4.4.3 Communication design requirements

While creating high-speed communication, I will do specific traces to reduce the risk of miscommunication which is based on the speed of the communication. For SPI, I will use a speed of 1 Mb/s or higher, and for CAN, I will use a speed of 500 kb/s or higher.

Especially for SPI, it is recommended to keep SPI traces as short as possible with the same length as all SPI traces. If SPI traces won't have the same length, there could be a risk of crosstalks because the signal propagates equally fast, reaching the receiver faster on a shorter trace than on a longer trace [66]. Then I need to create traces with 50 Ω impedance, which is used to deliver maximal power over the communication lines [66]. I will use an automatic trace width calculator built into the Altium Designer for impedance matching. The high-speed lines shouldn't have sharp right-angle corners because they cause more reflection from traces [67].

For CAN and UART, I will use the same rules as for SPI. It is not necessary, but it eliminates the risk of miscommunication.

4.4.4 PCB waterproofness

I plan to make waterproof electronics. Therefore, I must ensure that water will not destroy my PCB. I would pour my PCBs into resin or other material to ensure waterproofness. Communication with PCBs would be fine because I will first connect all connectors and test PCBs before pouring. When pouring, PCB connectors would be connected, which means I cannot disconnect them after this process, but the PCBs will be waterproof. One main disadvantage is that I would need to change all PCBs if PCB were destroyed. For that, I will use cables with enough reserve, and when PCB would be destroyed, I would only cut off the destroyed PCB and crimp new connectors to the cable.

4.4.5 Components positions

When placing components, I choose input and output connectors on one end (right) and the strain gauge connector on the other (left). The reason is that I need my PCB as narrow as possible. Then I place ADC on the left side as close to the strain gauge as possible to reduce voltage drop over the PCB. I placed MCU in the middle of the PCB because it is close to ADC and the input and output connectors. On the left side, I placed voltage regulators; on the right side, I placed the CAN transceiver because it is close to the input and output connectors. Lastly, I placed connectors for programming on the right side, where there was a free space.

You can see the model of the PCB in Appendix B.

4.5 PCB assembling and testing

After sending an order to JLCPCB and receiving the PCBs and components, I will solder those components of the PCB and test if the PCB works as I wanted.

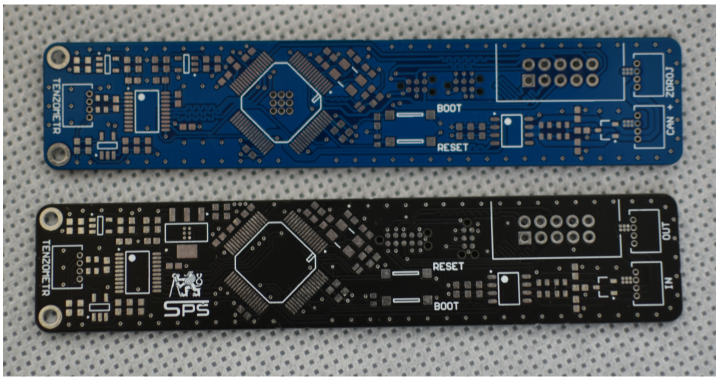

Figure 4.7: First (blue) and second (black) version of PCBs

4.5.1 PCB assembly and stencil

I dispensed the solder paste on the PCB with a stencil, which is a template for an ideal solder paste application. After that, I placed all components on the solder paste and heated the PCB with the solder paste on a heated plate, which melted the paste and solders all components.

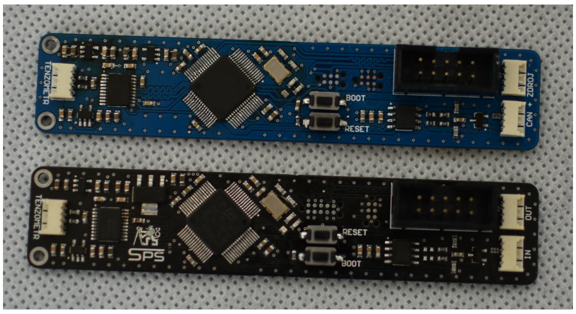

Figure 4.8: First and second version of PCBs, assembled

4.5.2 PCB testing

Firstly, I tested whether all voltages were as required. Then I tested the uploading program via SWD with STLink v3. I created a program similar to Nucleo, uploaded it to the MCU, and tested if ADC responded correctly to voltage from the laboratory power supply and loaded strain gauge.

All tests passed, but only two problems appeared on version one. The first problem was resetting of MCU without pressing the reset button, which was caused by a capacitor that was too far from the MCU pin. The second problem was shutting down the voltage regulator for MCU, CAN, and other components due to overcurrent. Both issues are fixed in the second version, where the capacitor is as close to the pin as possible, and the voltage regulator is changed to a 1 A regulator from a 500 mA regulator. I later found out that I did not make a good solder on the first version, and under the components was a low resistance path through not melted solder paste which caused a high current on the regulator. Now, single PCB current consumption is maximal at about 100 mA.

Chapter 5

Algorithms

In the development of my PCB, I have the task of creating three separate algorithms that will later be interconnected. Firstly, I will design an algorithm for communication between a computer and STM using UART. Then I will design an algorithm for setting up and reading data from ADC. Lastly, I will create an algorithm for sending and receiving data over the CAN bus.

5.1 The programming environment and language

As a programming language for STM, I can choose programming in C, C++, microPython, and more. I chose to program in C because I need a fast reading from ADC and sending via CAN to the CCU, where microPython would be too slow.

5.2 Algorithms

First, I will create a UART algorithm and test communication between PCB and the computer. Then I will create an ADC algorithm and test the output of ADC while loading the strain gauge. Lastly, I will create a CAN bus algorithm and test sending voltage from ADC to the CCU. As a programming environment, I am using a STM32CubeIDE.

5.2.1 UART communication algorithm

UART algorithm is used for easier debugging of STM. I created a function uart_print() which input is a string that will be sent to the UART and can be read for example, at PC with UART to USB converter and software like Putty or Realterm. I primarily used UART to debug an ADC and to test the CAN communication between the CCU and my PCB. For clearing a UART buffer I created the function clear_char_buffer(), which is used to clear the buffer to which I am saving the data to send via UART.

5.2.2 ADC algorithm

For communication with ADC, I will be using SPI. Firstly, I set the speed of the SPI to 11.25 Mb/s. Then I read from the datasheet which registers I must set up in the ADC and which I will need to read data (voltage) from ADC. I will use those registers and commands:

- MODE register - set word length to 32 bits
- CLOCK register - disable channels 1, 2, 3, set oversampling
- GAIN register - set PGA (programmable amplifier) to amplify voltage from strain gauge
- NULL commands - sends word of zeros to the ADC, the response from ADC are measured data

When the STM32 is connected to the power, it will wait to receive a logical high on the slave select pin, which indicates that ADC is ready. Then it sends data to the MODE register. Then it will set up the CLOCK register and GAIN register. Then in the while loop, it will start every loop read_adc_data() function, which sends a NULL command to ADC and then reads the response with data from ADC.

5.2.3 Communication algorithm

CAN bus is CSMA/CD (Carrier Sense Multiple Access with Collision Detection) protocol, which means that every device can start sending data anytime, and when there is a collision detection on the bus, the CAN protocol will solve it. When the CAN sends data, it simultaneously receives the data from the bus, which is used to detect those collisions.

If I am sending messages from multiple devices, CAN will send the message with the highest priority, which is determined by the ID. For example, if I have two devices with IDs 101 and 111, and both will start to send the message at the same time, both will send 1, and then the device with the 0 will pull down the device which is now sending 1. The second device will detect this pull-down (because when the device is transmitting, it is receiving as well) and will stop sending data until the given time stamp, after which it will try to send the data again.

For the CAN bus, I need to create CAN bus IDs for every PCB and CCU. I will have a few messages, so I will use a standard 11-bit ID. In table 5.1 I mention only some IDs. In the table, if there is, for example, data from the first STM32, then the ID for message data from the second STM32 is the ID from the previous message plus 1, which means data from the second STM32 is 0111 1111 1001.

All CAN messages I defined are in Appendix C.

Message	Bin address	Hex address
Data from first STM32	0111 1111 1000	7F8
Response after first STM32 configuration	0111 1111 0000	7F0
Requirement for first STM32 configuration	0111 1110 1000	7E8
Error message for all STM32	0111 1110 0000	7E0
Error message for first STM32	0111 1110 0001	7E1
Error message from first STM32	0111 1101 1000	7D8
Change delay between CAN messages	0111 1011 1111	7BF

Table 5.1: CAN communication - message IDs

In the program, I must implement a timer that is used to send CAN messages after a given time because otherwise, the CAN would be too fast, and the CCU would not be able to keep up with the bus. For that, I also added mentioned message Change delay between CAN messages.

When the CCU first communicates with the PCBs, it will send requirements for every STM32 to configuration. When the STM32 receives this message, it will measure 1000 values from ADC, do the arithmetical mean, and set the result as zero. As a response, the STM32 will send the message with a set offset value.

After that, in the while loop (after ADC configuration), every time after ADC reads data I am checking if enough time has passed since the last send of the CAN message. If so, I will either send the message to the bus, or I will wait and check this condition in the next loop.

The flowchart of program is in Fig. 5.1.

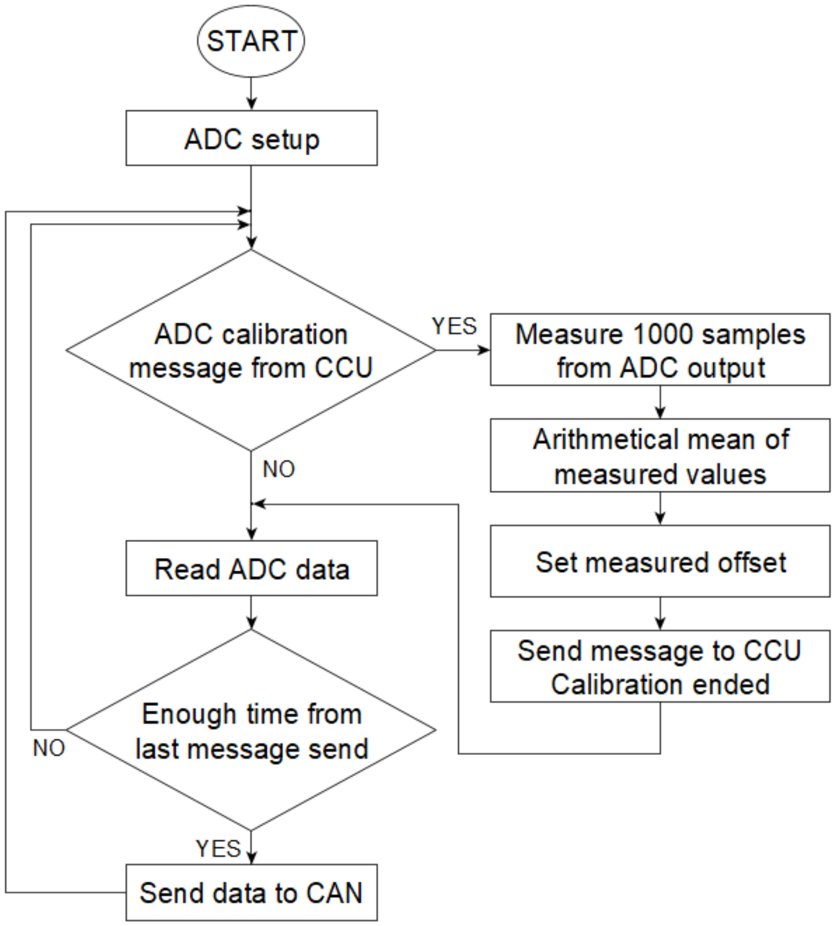

Figure 5.1: Algorithm flow chart

Chapter 6

Experiments

In this chapter, I will present the results of the experiments I did. I will present measurements of interference and inaccuracy of measuring voltage from strain gauge at PCB and measurements of walking and running over the sensor platform.

For measuring data, I used the CCU with a program created by Josef Vágner. The program reads and saves the data that are comming from the CAN bus. I use the saved data to generate plots.

6.1 Strain gauge measurements on platform

Firstly I measured the output values (voltage) of strain gauges without any external load on the platform, the only load was the platform itself, which is screwed to the chassis of the treadmill (strain gauges were preloaded).

After that, I measured the data from strain gauges while walking on the sensor platform.

6.1.1 Measurement inaccuracy with the motor turned off

When I was measuring PCB's output, I found that two of the six PCBs were outputting different signals. As you can see in Fig. 6.1 the output value of the strain gauge is jumping between 0 V and 0.1 V without any load, which is an inaccuracy of ± 0.05 V. Other PCBs were outputting values of inaccuracy ± 0.004 V as you can see in Fig. 6.2. I found out that the ADC caused this. After replacing it with a new one, the measuring results were as expected.

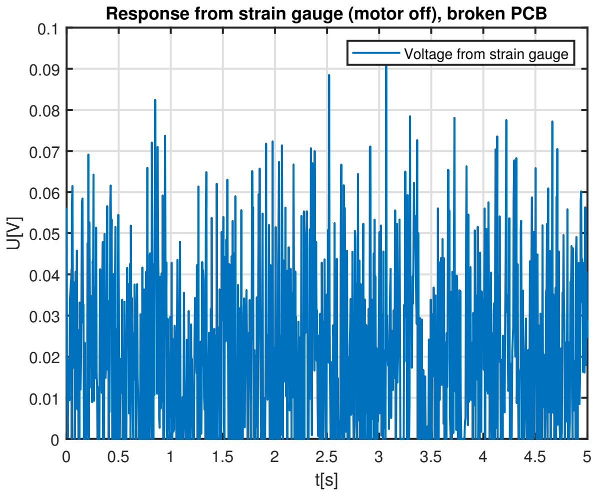

Figure 6.1: Response of one strain gauge with the motor switched off, broken PCB

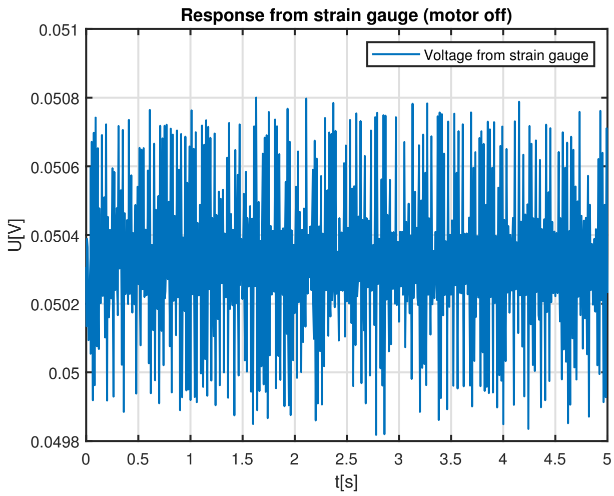

Figure 6.2: Response of one strain gauge with the motor switched off, good PCB

6.1.2 Measurement inaccuracy with the motor turned on

When measuring data while the motor was running, there was interference caused by the motor. The inaccuracy of measurement was based on the distance from the motor. The nearest PCB to the motor was measuring voltage with an inaccuracy of 0.005 V, as you can see in Fig. 6.3. I connected the PCB to the chassis of the treadmill, which is connected to the ground, and the interference was significantly reduced, as you can see in Fig. 6.4.

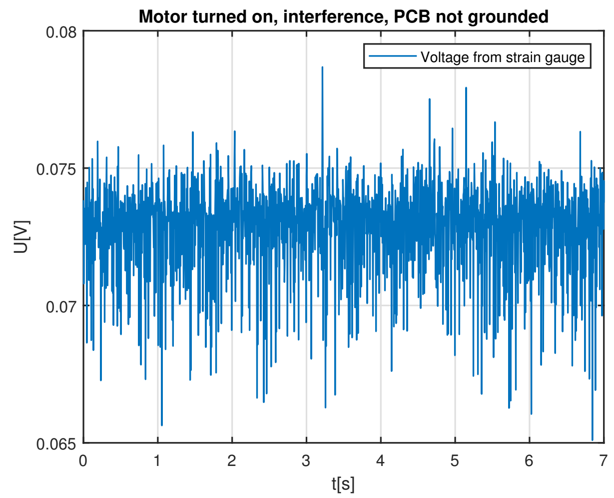

Figure 6.3: Interference of the motor

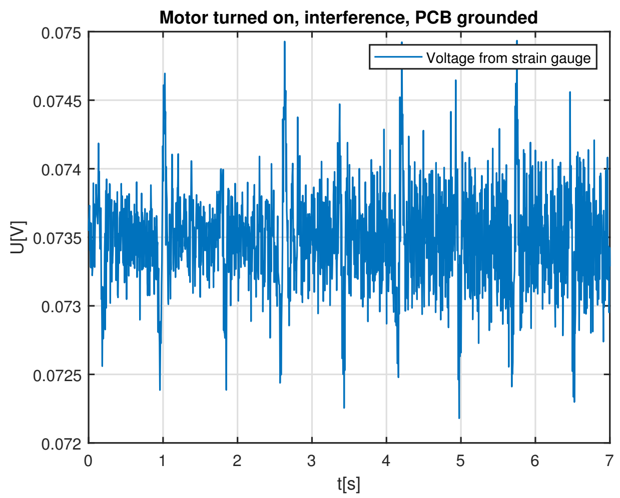

Figure 6.4: Interference of the motor

6.1.3 Interference from SPI

I measured the SPI communication speed's effect on the voltage from strain gauges. As you can see in Fig 6.5, I measured the output voltage for the speed of SPI from 351.562 kBps to 45 MBps. The speed of SPI does not cause any significant interference with the measurement.

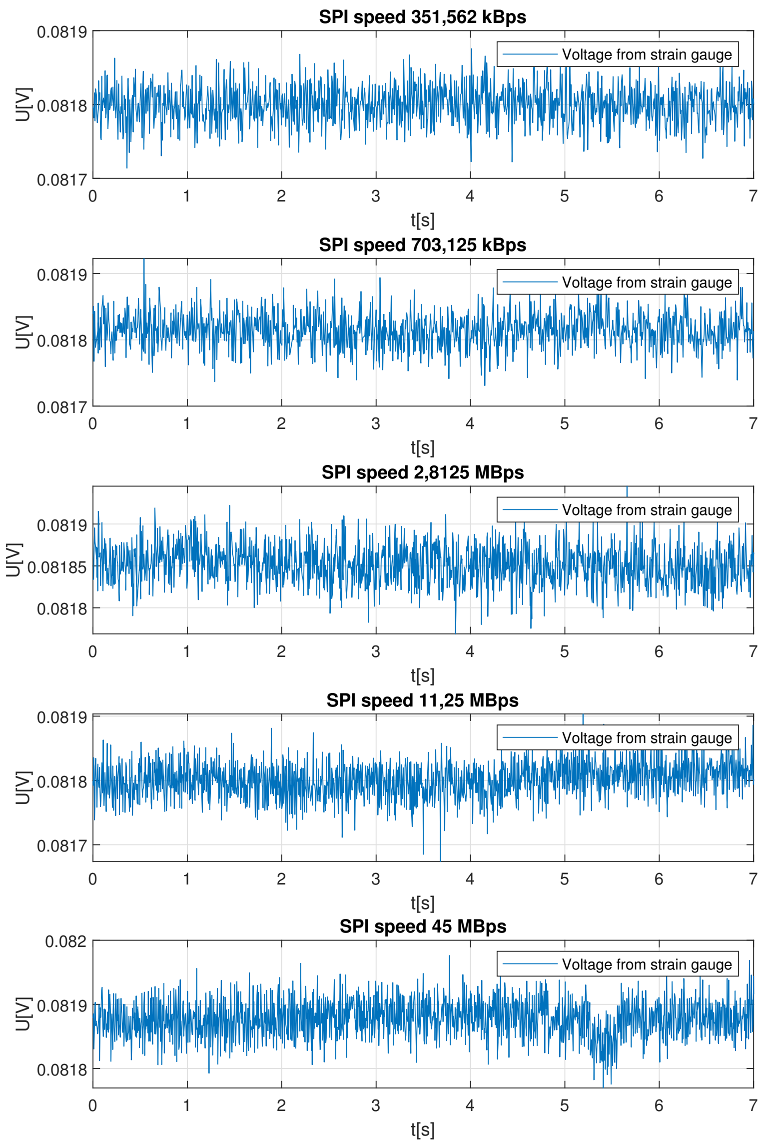

Figure 6.5: Interference of the motor

6.1.4 Measurement while walking

I measured data from all strain gauges with the motor turned on and with the load that I generated by walking on the platform. As you can see in the Fig. 6.6 data from strain gauges are clear without any significant interference. In the graph, you can also see the spikes that indicate every footfall (higher spikes) or bounce (lower spikes) of the person walking.

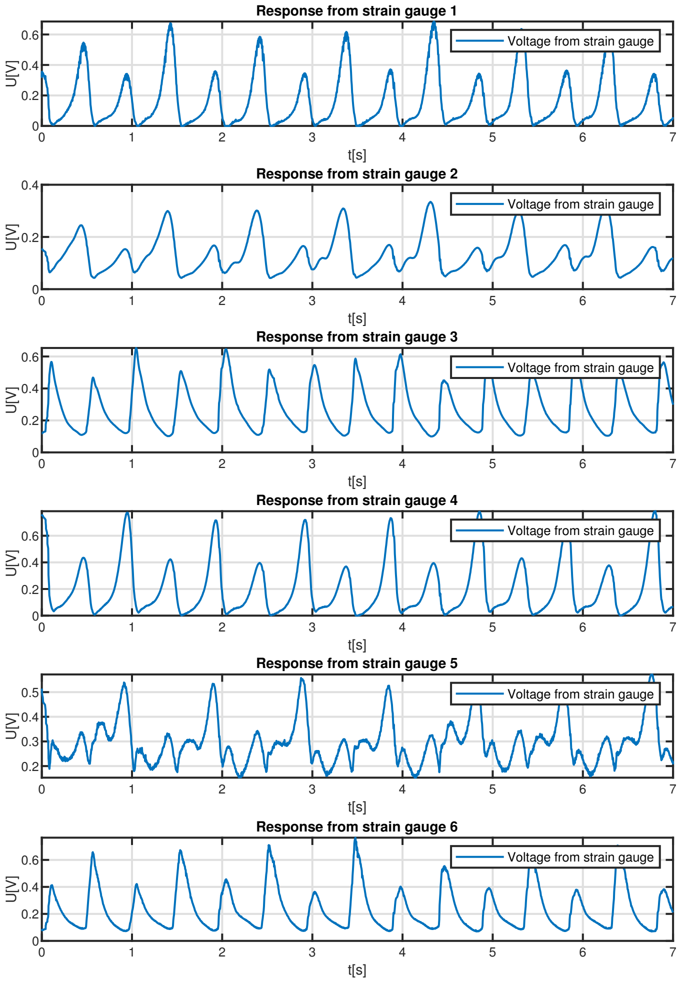

Figure 6.6: Response of all strain gauges with the motor turned on, walking

From these graphs, it can be seen that strain gauges with PCBs work as required. It measures the load on the platform in 6 different positions and sends correct voltages from strain gauges to the CCU.

Chapter 7

Conclusion and future work

Our goal was to make a sensor platform and evaluation equipment (CCU) for a treadmill that will measure the CoP of the patient that is walking or running on the treadmill. The CoP must be displayed on the CCU with important data such as width and length of step, number of steps, and time of exercise. We also wanted to measure data over time and create statistics for patients' treatment procedures.

My goal was to design the sensor platform. This involved selecting appropriate sensors and creating custom PCBs to capture data from these sensors. Additionally, I focused on establishing a reliable communication protocol that would seamlessly transmit the collected data to the CCU for further processing and analysis.

7.1 Conclusion

I have successfully made the sensor selection for the platform, opting for strain gauges. I have designed PCBs capable of measuring voltage from the strain gauges and transmitting it to the CCU via the CAN bus. I have also addressed challenges related to interference from the motor and rectified issues with a faulty ADC. Furthermore, I prepared the PCBs for the subsequent phase of the project, which is water resistivity.

Other parts of the project are also done. David Laušman created a PCB and box for the CCU, and Josef Vágner did the program that reads data from the sensor board, displays it, and saves it. David Laušman also created basic statistics on patients' treatment.

7.2 Future work

One of the main objectives to do in the future is to use a new sensor board instead of the basic one from the treadmill. The current board bends when running on it, which devalues the measured data.

In the future, I may enhance the PCB design by reducing its dimensions. This can be achieved by using smaller components, particularly opting for a compact MCU, which occupies the majority of space on the board. Additionally, I might eliminate unnecessary connectors that are not utilized when mounted on the treadmill. For instance, the 10-pin Tag connector and 10-pin IDC connector can be removed.

Another aspect is to test the usage of more strain gauges and their impact on the measurement. By increasing the number of strain gauges, the distribution of mass can be better captured, potentially resulting in more precise CoP measurements.

Bibliography

[1] Y. F. Luckyarno, P. Cherntanomwong, and R. Wijaya, “Posturometry data transmission using visible light communication”, in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Jun. 2016, pp. 1–4. DOI: 10.1109/ECTICon.2016.7561275. [Online]. Available: https://ieeexplore.ieee.org/document/7561275 (visited on Apr. 10, 2023).

[2] J. Bruštíková, “USE OF TENZOMETRIC PLATFORM IN AMBULATORY THERAPY”, Bachaleor thesis, Czech technical university in Prague, Prague, 2016. [Online]. Available: http://hdl.handle.net/10467/67612 (visited on Apr. 11, 2023).

[3] B. Helštýnová, “Analysis of Posturometric Data”, Diplomová práce, Technical univesity of Ostrava, University Banska, 2012. [Online]. Available: http://hdl.handle.net/10084/92986 (visited on Apr. 10, 2023).

[4] S. Das and D. Tibarewala, “STABILOMETRIC POSTURAL STEADINESS ANALYSIS OF POST- STROKE HEMIPLEGIC PATIENTS”, *International Journal of Engineering Science and Technology*, vol. 3, pp. 4626–4637, Jun. 6, 2011. [Online]. Available: https://www.researchgate.net/publication/294259105_STABILOMETRIC_POSTURAL_STEADINESS_ANALYSIS_OF_POST-_STROKE_HEMIPLEGIC_PATIENTS (visited on Apr. 10, 2023).

[5] E. M. Monsell, J. M. Furman, S. J. Herdman, H. R. Konrad, and N. T. Shepard, “Computerized Dynamic Platform Posturography”, *Otolaryngology–Head and Neck Surgery*, vol. 117, no. 4, pp. 394–398, 1997, ISSN: 1097-6817. DOI: 10.1016/S0194-5998(97)70132-3. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1016/S0194-5998%2897%2970132-3 (visited on Apr. 11, 2023).

[6] "BER_CDP In use forest", Bertec. (2023), [Online]. Available: https://images.squarespace-cdn.com/content/v1/5b3256317e3c3a8e8e029991/1599059648191-MADEHZWXG8A8SWZFHVI3/BER_CDP_InUse_Forest%2BCameron.jpg?format=1000w (visited on Apr. 11, 2023).

[7] "Mill arm". (2023), [Online]. Available: https://cdn.shopify.com/s/files/1/1333/0475/products/MillArm_1024x1024@2x.jpg?v=1593104201 (visited on Apr. 10, 2023).

[8] *Meden-inmed_axelero_gait_and_balance_brochure.pdf*, 2022. [Online]. Available: https://en.meden.com.pl/wp-content/uploads/2022/12/meden-inmed_axelero_gait_and_balance_brochure.pdf (visited on Apr. 10, 2023).

[9] *Focus Trainer*, 2022. [Online]. Available: https://hydrophysio.com/wp-content/uploads/2022/09/Focus-Trainer-2022-Final-16-09-2022.pdf (visited on Apr. 10, 2023).

[10] "Ewacmedical Underwater treadmill integrated", Ewacmedical Underwater treadmill. (2023), [Online]. Available: https://www.ewacmedical.com/wp-content/uploads/Products/Underwater_Treadmill/Integrated/integrated4.jpg (visited on Apr. 10, 2023).

[11] "Drop-in3.jpg (540×394)", Ewacmedical. (2023), [Online]. Available: https://www.ewacmedical.com/wp-content/uploads/Products/Underwater_Treadmill/Drop-In/drop-in3.jpg (visited on May 23, 2023).

[12] P. Tutak, "APPLICATION OF STRAIN GAUGES IN MEASUREMENTS OF STRAIN DISTRIBUTION IN COMPLEX OBJECTS", *Walter de Gruyter GmbH*, Dec 2014, vol. 2014, no. 2, p. 6, 2014, ISSN: 1689-9636. DOI: 10.1515/jacsm-2015-0004. [Online]. Available: https://www.researchgate.net/publication/276511938_Application_Of_Strain_Gauges_In_Measurements_Of_Strain_Distribution_In_Complex_Objects (visited on Apr. 11, 2023).

[13] "Strain gauge diagram", michsi. (2023), [Online]. Available: https://www.michsci.com/wp-content/gallery/illustrations/strain-gauge-diagram-example.png (visited on Apr. 11, 2023).

[14] S. Mondal, V. M. Datar, G. Majumder, N. K. Mondal, K. C. Ravindran, and B. Satyanarayana, "Leak test of Resistive Plate Chamber gap by monitoring absolute pressure", *Journal of Instrumentation*, vol. 14, no. 04, P04009, Apr. 2019, ISSN: 1748-0221. DOI: 10.1088/1748-0221/14/04/P04009. [Online]. Available: https://dx.doi.org/10.1088/1748-0221/14/04/P04009 (visited on Apr. 11, 2023).

[15] *AR1000 Series Resistive Touch Screen Controller Data Sheet*, 2016. [Online]. Available: https://cz.mouser.com/datasheet/2/268/40001393C-1512021.pdf (visited on Apr. 21, 2023).

[16] “Capacitive vs Resistive Touch”, Newhaven Display. (2023), [Online]. Available: https://newhavendisplay.com/blog/capacitive-vs-resistive-touch/ (visited on Apr. 11, 2023).

[17] “Capacitive Touchscreens”. (2023), [Online]. Available: https://www.gtk.co.uk/capacitive-touchscreens (visited on Apr. 11, 2023).

[18] “Engeneering projects capacitive display”. (2023), [Online]. Available: https://engineeringprojectideas.files.wordpress.com/2015/05/capacitive-touch-screen.png (visited on Apr. 11, 2023).

[19] “Nintendo wii”, Nintendoshop. (2023), [Online]. Available: https://www.nintendoshop.cz/galerie/8_5129/name-original.jpg (visited on Apr. 11, 2023).

[20] Y.-y. Fang and X.-j. Chen, “Design and Simulation of UART Serial Communication Module Based on VHDL”, in *2011 3rd International Workshop on Intelligent Systems and Applications*, May 2011, pp. 1–4. DOI: 10.1109/ISA.2011.5873448. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5873448 (visited on Apr. 11, 2023).

[21] A. Amelia, Julham, B. V. Sundawa, M. Pardede, W. Sutrisno, and M. Rusdi, “Implementation of the RS232 communication trainer using computers and the ATMEGA microcontroller for interface engineering Courses”, *Journal of Physics: Conference Series*, vol. 890, no. 1, p. 012 095, Sep. 2017, ISSN: 1742-6596. DOI: 10.1088/1742-6596/890/1/012095. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/890/1/012095 (visited on Apr. 11, 2023).

[22] “Fundamentals of RS-232 Serial Communications | Analog Devices”. (2023), [Online]. Available: https://www.analog.com/en/technical-articles/fundamentals-of-rs232-serial-communications.html (visited on Apr. 11, 2023).

[23] X. Su, S. Wang, D. Zhu, and Q. Zhu, “RS-485 serial port pseudo-full-duplex communication research and application”, in *2010 Prognostics and System Health Management Conference*, Jan. 2010, pp. 1–5. DOI: 10.1109/PHM.2010.5413343. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5413343 (visited on Apr. 11, 2023).

[24] *RS485 Quick Guide*, 2020. [Online]. Available: https://www.analog.com/media/en/technical-documentation/product-selector-card/rs485fe.pdf (visited on Apr. 11, 2023).

[25] J. Lázaro, A. Astarloa, A. Zuloaga, U. Bidarte, and J. Jiménez, “I2CSec: A secure serial Chip-to-Chip communication protocol”, *Journal of Systems Architecture*, vol. 57, no. 2, pp. 206–213, Feb. 1, 2011, ISSN: 1383-7621. DOI: 10.1016/j.sysarc.2010.12.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762110001633 (visited on Apr. 11, 2023).

[26] "I2C Bus Range and Electrical Specifications, Freescale 9S12 HCS12 MC9S12 I2C Hardware", Mosaic documentation web. (2023), [Online]. Available: `http://www.mosaic-industries.com/embedded-systems/sbc-single-board-computers/freescale-hcs12-9s12-c-language/instrument-control/i2c-bus-specifications` (visited on Apr. 11, 2023).

[27] J. Qiang, Y. Gu, and G. Chen, "FPGA Implementation of SPI Bus Communication Based on State Machine Method", *Journal of Physics: Conference Series*, vol. 1449, no. 1, p. 012 027, Jan. 2020, ISSN: 1742-6596. DOI: `10.1088/1742-6596/1449/1/012027`. [Online]. Available: `https://dx.doi.org/10.1088/1742-6596/1449/1/012027` (visited on Apr. 11, 2023).

[28] S. Campbell. "Basics of the SPI Communication Protocol", Circuit Basics. (Feb. 13, 2016), [Online]. Available: `https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/` (visited on Apr. 11, 2023).

[29] "Isolated SPI Communication Made Easy | Analog Devices". (2023), [Online]. Available: `https://www.analog.com/en/technical-articles/isolated-spi-communication-made-easy.html` (visited on Apr. 11, 2023).

[30] M. Kubis, P. Beno, M. Gutten, and M. Danko, "CAN Bus Comunication using LabViewSystem", in *2019 20th International Scientific Conference on Electric Power Engineering (EPE)*, May 2019, pp. 1–4. DOI: `10.1109/EPE.2019.8777930`. [Online]. Available: `https://www.researchgate.net/publication/334768982_CAN_Bus_Comunication_using_LabViewSystem` (visited on Apr. 11, 2023).

[31] "Introduction to CAN (Controller Area Network) - Technical Articles". (2019), [Online]. Available: `https://www.allaboutcircuits.com/technical-articles/introduction-to-can-controller-area-network/` (visited on Apr. 11, 2023).

[32] "Understanding the Difference Between UART vs SPI", Total Phase Blog. (Jun. 29, 2016), [Online]. Available: `https://www.totalphase.com/blog/2016/06/spi-vs-uart-similarities-differences/` (visited on Apr. 11, 2023).

[33] Administrator. "Maximum Cable Length For a CAN Bus". (), [Online]. Available: `https://www.buenoptic.net/encyclopedia/item/537-maximum-cable-length-for-a-can-bus.html` (visited on Apr. 11, 2023).

[34] *Co-Planar Beam Load Cell Model 380*, 2021. [Online]. Available: `https://www.manat.cz/wp-content/uploads/2021/01/380.pdf` (visited on Apr. 11, 2023).

[35] *FX29*, 2019. [Online]. Available: `https://cz.mouser.com/datasheet/2/418/8/ENG_DS_FX29_A5-2497876.pdf` (visited on Apr. 11, 2023).

[36] *FC23*, 2020. [Online]. Available: https://cz.mouser.com/datasheet/2/418/6/ENG_DS_FC23_A9-1130240.pdf (visited on Apr. 11, 2023).

[37] *Two/Four/Eight-Channel, 153.6 ksps, Low-Noise 24-Bit Delta-Sigma ADCs*, 2019. [Online]. Available: https://cz.mouser.com/datasheet/2/268/MCP3561_2_4_Family_Data_Sheet_DS20006181C-2257924.pdf (visited on Apr. 11, 2023).

[38] *Industrial, 14kSPS, 24-Bit Analog-to-Digital Converter with Low-Drift Reference*, 2011. [Online]. Available: https://www.ti.com/lit/ds/symlink/ads1259.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1681240339454&ref_url=https%253A%252F%252Fcz.mouser.com%252F (visited on Apr. 11, 2023).

[39] *24-Bit, 20kHz, Low Power ANALOG-TO-DIGITAL CONVERTER*, 2001. [Online]. Available: https://www.ti.com/lit/ds/symlink/ads1254.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1681240424086&ref_url=https%253A%252F%252Fcz.mouser.com%252F (visited on Apr. 11, 2023).

[40] *ADS131M04 4-Channel, Simultaneously-Sampling, 24-Bit, Delta-Sigma ADC*, 2021. [Online]. Available: https://www.ti.com/lit/ds/symlink/ads131m04.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1681240510417&ref_url=https%253A%252F%252Fcz.mouser.com%252F (visited on Apr. 11, 2023).

[41] *AD7766BRUZ*, 2007. [Online]. Available: https://cz.mouser.com/datasheet/2/609/AD7766-3119004.pdf (visited on Apr. 11, 2023).

[42] *Low Power, 24-Bit, Single Channel Audio Analog-to-Digital Converter*, 2006. [Online]. Available: https://www.ti.com/lit/ds/symlink/pcm4201.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1681240665252&ref_url=https%253A%252F%252Fcz.mouser.com%252F (visited on Apr. 11, 2023).

[43] "STM32 MCUs 32-bit Arm Cortex-M", ST. (2023), [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html (visited on Apr. 11, 2023).

[44] "STM32 F2 series", ST. (2023), [Online]. Available: https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32f2-series.html (visited on Apr. 11, 2023).

[45] "STM32 F4 series", ST. (2023), [Online]. Available: https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32f4-series.html (visited on Apr. 11, 2023).

[46] "STM32 F7 series", ST. (2023), [Online]. Available: https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32f7-series.html (visited on Apr. 11, 2023).

[47] "STM32 H5 series", ST. (2023), [Online]. Available: https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32h5-series.html (visited on Apr. 11, 2023).

[48] "STM32 H7 series", ST. (2023), [Online]. Available: https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32h7-series.html (visited on Apr. 11, 2023).

[49] "STM32 F1 series", ST. (2023), [Online]. Available: https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-mainstream-mcus/stm32f1-series.html (visited on Apr. 11, 2023).

[50] "STM32 F3 series", ST. (2023), [Online]. Available: https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-mainstream-mcus/stm32f3-series.html (visited on Apr. 11, 2023).

[51] *ADP122/ADP123 (Rev. E)*, 2009. [Online]. Available: https://cz.mouser.com/datasheet/2/609/ADP122_123-3121625.pdf (visited on Apr. 11, 2023).

[52] *LM4132, LM4132-Q1 SOT-23 Precision Low Dropout Voltage Reference datasheet (Rev. G)*, 2016. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm4132-q1.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1681216142649&ref_url=https%253A%252F%252Fwww.mouser.pl%252F (visited on Apr. 11, 2023).

[53] *3.3-V CAN Transceivers datasheet (Rev. A)*, 2008. [Online]. Available: https://www.ti.com/lit/ds/symlink/sn65hvd231q.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1681243349989&ref_url=https%253A%252F%252Fcz.mouser.com%252F (visited on Apr. 11, 2023).

[54] "Tag-Connect TC2050", ST. (2023), [Online]. Available: https://www.tag-connect.com/product/tc2050-idc-nl-050-stdc14 (visited on Apr. 11, 2023).

[55] "Tag-Connect", Tag-Connect. (2023), [Online]. Available: https://www.tag-connect.com/ (visited on Apr. 11, 2023).

[56] *Ecx_53b-3074848.pdf*, 2017. [Online]. Available: https://cz.mouser.com/datasheet/2/122/ecx_53b-3074848.pdf (visited on Apr. 11, 2023).

[57] *ECX_1210-1099240.pdf*, 2021. [Online]. Available: https://cz.mouser.com/datasheet/2/122/ECX_1210-1099240.pdf (visited on Apr. 11, 2023).

[58] *Datasheet - ESDCANxx-2BLY - Automotive dual-line TVS in SOT23-3L for CAN bus*, 2019. [Online]. Available: https://cz.mouser.com/datasheet/2/389/esdcan01_2bly-1551146.pdf (visited on Apr. 11, 2023).

[59] "All You Need to Know About FR-4 PCB", pcbgogo. (2022), [Online]. Available: https://www.pcbgogo.com/Article/All_You_Need_to_Know_About_FR_4_PCB.html (visited on Apr. 13, 2023).

[60] C. f. D. a. R. Health. "Electromagnetic Compatibility (EMC)", FDA. (Thu, 06/02/2022 - 13:20), [Online]. Available: https://www.fda.gov/radiation-emitting-products/radiation-safety/electromagnetic-compatibility-emc (visited on Apr. 13, 2023).

[61] J. Šandera, *Návrh plošných spojů pro povrchovou montáž*, Praha, 1 vols. Praha: BEN - technická literatura, 2006, vol. 1, ISBN: 80-7300-181-0.

[62] "What PCB Copper Thickness Should You Use? | Blog | CircuitMaker", Altium. (2021), [Online]. Available: https://resources.altium.com/p/what-pcb-copper-thickness-should-you-use#05-ozsq-ft-or-10-ozsq-ft-copper-thickness (visited on Apr. 13, 2023).

[63] "Printed Circuit Board Trace Width Tool | Advanced Circuits". (2023), [Online]. Available: https://www.4pcb.com/trace-width-calculator.html (visited on May 17, 2023).

[64] "IPC-2221B PCB Trace Spacing / Clearance by Voltage". (2023), [Online]. Available: https://www.smpspowersupply.com/ipc2221pcbclearance.html (visited on Apr. 13, 2023).

[65] "The CircuitCalculator.com Blog » PCB Via Calculator". (2007), [Online]. Available: http://circuitcalculator.com/wordpress/2006/03/12/pcb-via-calculator/ (visited on Apr. 13, 2023).

[66] "Tips for Optimal High Speed SPI Layout Routing". (Aug. 14, 2019), [Online]. Available: https://resources.pcb.cadence.com/blog/2019-tips-for-optimal-high-speed-spi-layout-routing (visited on Apr. 13, 2023).

[67] *AN201383 - S25FL-P SPI Flash Family PCB Layout Guide*, 2017. [Online]. Available: https://www.infineon.com/dgdl/Infineon-AN201383_S25FL_P_SPI_Flash_Family_PCB_Layout_Guide-ApplicationNotes-v03_00-EN.pdf?fileId=8ac78c8c7cdc391c017d074178f9649a (visited on Apr. 13, 2023).

Appendix A

Treadmill platform screw hole location

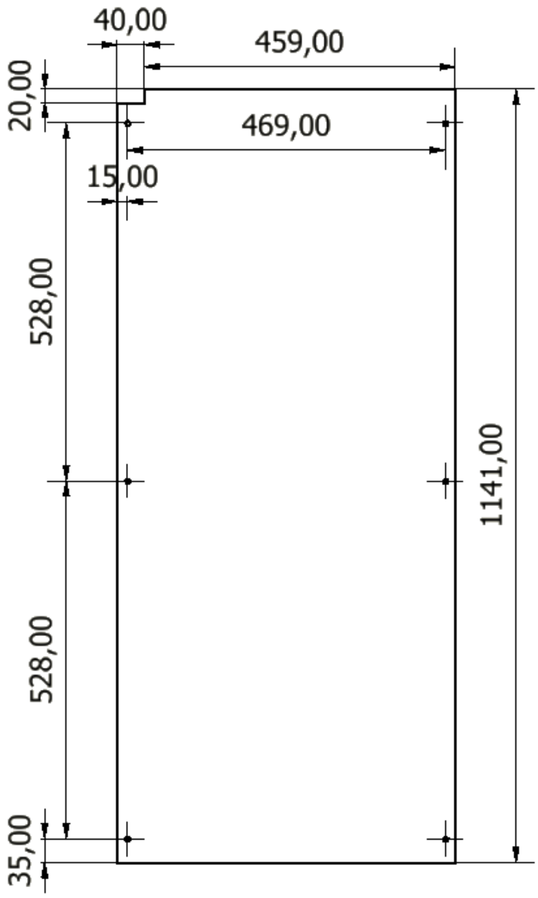

Figure A.1: Treadmill platform screw hole location

Appendix B

PCB 3D model

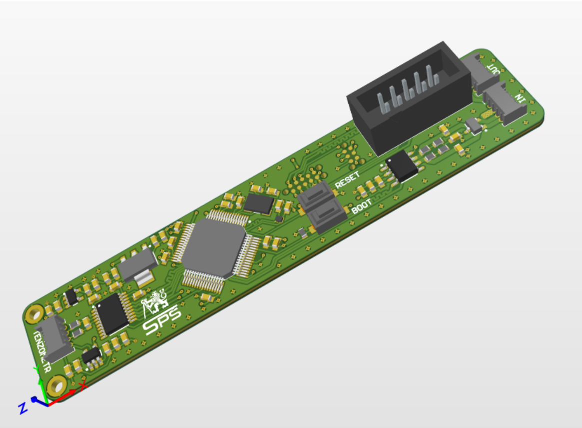

Figure B.1: PCB 3D model

Appendix C

CAN messages

Message	Binary	Hexadecimal
Data from ADC 1	0111 1111 1000	7F8
Data from ADC 2	0111 1111 1001	7F9
Data from ADC 3	0111 1111 1010	7FA
Data from ADC 4	0111 1111 1011	7FB
Data from ADC 5	0111 1111 1100	7FC
Data from ADC 6	0111 1111 1101	7FD

Table C.1: CAN - DATA from ADC

Message	Binary	Hexadecimal
Response after config STM 1	0111 1111 0000	7F0
Response after config STM 2	0111 1111 0001	7F1
Response after config STM 3	0111 1111 0010	7F2
Response after config STM 4	0111 1111 0011	7F3
Response after config STM 5	0111 1111 0100	7F4
Response after config STM 6	0111 1111 0101	7F5

Table C.2: CAN - configuration response

Message	Binary	Hexadecimal
Config request to STM 1	0111 1110 1000	7E8
Config request to STM 2	0111 1110 1001	7E9
Config request to STM 3	0111 1110 1010	7EA
Config request to STM 4	0111 1110 1011	7EB
Config request to STM 5	0111 1110 1100	7EC
Config request to STM 6	0111 1110 1101	7ED

Table C.3: CAN - configuration request

Message	Binary	Hexadecimal
Error from RPi to every STM	0111 1110 0000	7E0
Error from RPi to STM 1	0111 1110 0001	7E1
Error from RPi to STM 2	0111 1110 0010	7E2
Error from RPi to STM 3	0111 1110 0011	7E3
Error from RPi to STM 4	0111 1110 0100	7E4
Error from RPi to STM 5	0111 1110 0101	7E5
Error from RPi to STM 6	0111 1110 0110	7E6

Table C.4: CAN - error message from RPi

Message	Binary	Hexadecimal
Error from STM 1 to RPi	0111 1101 1000	7D8
Error from STM 2 to RPi	0111 1101 1001	7D9
Error from STM 3 to RPi	0111 1101 1010	7DA
Error from STM 4 to RPi	0111 1101 1011	7DB
Error from STM 5 to RPi	0111 1101 1100	7DC
Error from STM 6 to RPi	0111 1101 1101	7DD

Table C.5: CAN - error message from STM