CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Measurement

Bachelor's Thesis

Acoustic Detection System Based on Two Microphones

Martin Maxa
Cybernetics and Robotics
martin@mmaxa.cz

May 2023
Supervisor: Ing. Jakub Svatoš, Ph.D.

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Maxa** Jméno: **Martin** Osobní číslo: **491938**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra měření**

Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Systém pro akustickou detekci založený na dvou mikrofonech

Název bakalářské práce anglicky:

Acoustic Detection System Based on Two Microphones

Pokyny pro vypracování:

Navrhněte a realizujte systém umožňující detekci impulzních akustických událostí.
Systém bude obsahovat mikrokontroler, komunikační jendotku a GPS a bude umožňovat simultální zpracovanáni ze dvou MEMS mikrofonů.
Data z MEMS mikrofonů budou zpracována mikroprocesorem a odesílána na vzdálený server.

Seznam doporučené literatury:

[1] Sensor Fusion, Sensitivity Analysis and Calibration in Shooter Localization Systems, Akman, C., at al., Sensors and Actuators A: Physical Volume 271, 2018, Pages 66-75.
[2] Evaluation of Gunshot Detection Algorithms, Chacón-Rodríguez, A., at al., IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, vol. 58, No. 2, 2011
[3] System for Acoustic Detection, Svatoš, J.; Belák, J.; Holub, J., 24th IMEKO TC4 International Symposium and 22nd International Workshop on ADC and DAC Modelling and Testing. Palermo: IMEKO, 2020. p. 347-352. ISSN 0237-028X.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jakub Svatoš, Ph.D. katedra měření FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **02.02.2023** Termín odevzdání bakalářské práce: _____________

Platnost zadání bakalářské práce:
do konce letního semestru 2023/2024

Ing. Jakub Svatoš, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

.

Datum převzetí zadání	Podpis studenta

Acknowledgement / Declaration

I would like to thank my supervisor Jakub Svatoš for his support, mentorship, expert guidance, and valuable feedback throughout my thesis.

I am also grateful to my partner for her unwavering support and to my family for their constant support throughout my studies.

Special thanks go out to my friend Martin Adamec for his proofreading and correction.

I declare that the presented work was developed independently and that 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 25. 5. 2023

. . . .

Abstrakt / Abstract

Tato bakalářská práce se zabývá vytvořením akustického detekčního systému založeného na principu dvou mikrofonů. Cílem systému je monitorovat okolí a po detekci akustické impulzní události tuto událost zaznamenat a výsledná data odeslat na vzdálený server.

Systém se skládá z hlavní DPS obsahující MCU, modul Wi-FI, GPS modul a dvou menších desek pro přesné umístění MEMS mikrofonů.

Po zaznamenání akustické události na obou mikrofonech a určení časového zpoždění mezi dopadem akustické vlny na první a druhý mikrofon jsou data odeslána pomocí Wi-Fi a TCP protokolu na vzdálený server. Odeslaná data obsahují nahraná akustická data a metadata obsahující časové zpoždění, polohu jednotky a časovou značku získanou pomocí GPS.

Systém je založen na dvou mikrofonech pro minimalizaci počtu jednotek potřebných pro 2D lokalizaci akustikckých událostí. Jelikož je každá jednotka osazena dvěma mikrofony stačí pro lokalizaci teoretiky jen dvě jednotky.

Klíčová slova: detekce akustických událostí se dvěma mikrofony, detekce impulzních událostí, detekce střelby, MCU, Wi-Fi, GPS, MEMS mikrofony

This bachelor's thesis focuses on developing an acoustic detection system based on the principle of two microphones. The system aims to monitor the surroundings and, upon detecting an acoustic impulse event, record the event and send the resulting data to a remote server.

The system consists of a main board containing an MCU, Wi-Fi module, GPS module, and two smaller boards for precise placement of the MEMS microphones.

After capturing the acoustic event on both microphones and determining the time delay between the impact of the acoustic wave on the first and second microphones, the data is transmitted via Wi-Fi and TCP protocol to a remote server. The transmitted data includes recorded acoustic data and metadata containing time delay, unit location, and timestamp obtained through GPS.

The system is based on two microphones to minimize the units required for acoustic event localization. Theoretically, with each unit equipped with two microphones, only two units are sufficient for localization.

Keywords: acoustic event detection with two microphones, impulse event detection, gunshot detection, MCU, Wi-Fi, GPS, MEMS microphones

Contents

/ Figures

Chapter 1
Introduction

Significant progress in sensor and processor technologies has been achieved in recent years. Current sensors and processors are notably smaller, more efficient, and have lower energy consumption. This progress allows easier integrating of acoustic detection units into public spaces, enabling more efficient systems for acoustic signal detection and environmental monitoring.

Historically, these systems were primarily used in the military sector. However, with technological advancements and the growing need for public space security and monitoring, their utilization has also significantly expanded into civilian spheres. Today, they can provide valuable information not only for military operations but also for urban monitoring, crime detection, traffic management, and other applications.

This bachelor's thesis focuses on designing and implementing a system capable of detecting impulsive acoustic events, with particular attention given to gunshot detection. The proposed system integrates a microcontroller, a communication unit enabling internet connectivity, and a GPS unit for localization purposes. The system allows simultaneous data processing from two MEMS microphones, enabling real-time monitoring of incoming acoustic impulses. The data is transmitted to a remote server upon detecting an acoustic impulse signal.

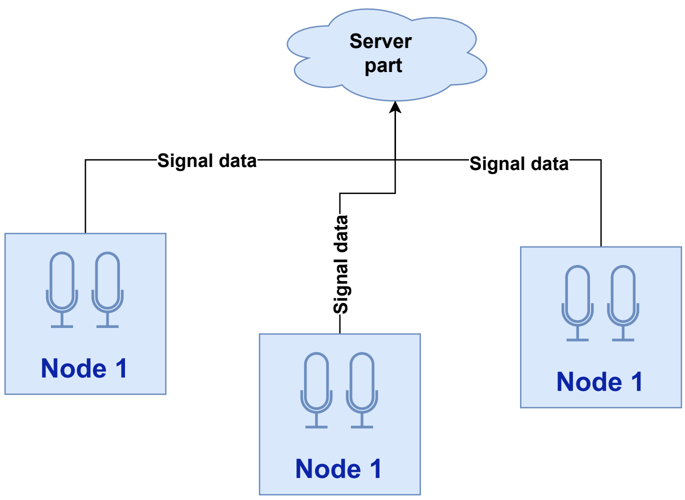


Figure 1.1. Expected layout of the acoustic detection system

The structure of this thesis is divided into three main chapters: Acoustic impulse event theory background, Hardware implementation of acoustic impulse event detection system, and Firmware.

The second chapter provides a detailed overview of the theoretical concepts and fundamentals essential for understanding the functional principles of the proposed system for acoustic impulse detection. It includes a description of the acoustic signal and explores methods for detecting impulsive signals. The theoretical chapter provides the necessary depth of knowledge to comprehend the functionality of this acoustic detection system.

The main topic of the third chapter is the design and implementation of the printed circuit board (PCB). This element is designed and created with regard to the functional requirements of the system. The components integrated into the board include key parts such as the processor, power circuits, Wi-Fi module for communication, and GPS module for localization purposes. Additionally, MEMS microphones are integrated as sensors for acoustic signal detection.

Afterward, the fourth chapter focuses on firmware development, which is necessary to ensure the desired functionality of the entire system. This part of the work concentrates on implementing software that enables the operation and control of individual system components.

The final chapter of this thesis is devoted to the result presentation. It includes evaluating achieved results in the context of set goals and system requirements. It also provides a comprehensive overview of the designed system.

Chapter 2
Acoustic impulse event theory background

This chapter focuses on the theoretical background of impulsive acoustic events. The basics of describing acoustic signals from firearm discharges are presented here, including the description of individual components such as muzzle blast, shock wave, and mechanical action. Furthermore, this chapter discusses the peak detection in the acoustic signal based on robust peak detection using Z-score. Additionally, localization based on the Angle of Arrival (AoA) method is explained in this section.

2.1 Signal theory

In order to correctly detect and describe the acoustic signal of a gunshot, it is necessary to understand and comprehend the individual events that happen during the shot. The resulting waveform is influenced by many factors, like the caliber of the gun, the type of ammunition and the surrounding environment. Hence, detecting a gunshot acoustic signal is a complex task, and this section will focus on the various aspects describing or influencing the resulting signal shape.

2.1.1 Muzzle blast

At the moment of firing from a gun, an acoustic wave called muzzle blast is generated and propagates through space in all directions. However, the majority of the acoustic energy is expelled in the direction the gun barrel is pointing. [1]

The muzzle blast is one of the primary components of a gunshot signal. It refers to the sound and pressure wave generated by the expansion of gases from the gun. In Figure 2.1 it is possible to see that a muzzle blast typically lasts 3 ms, and it spreads at the speed of sound. [2] In the figure, it can also be seen that it spreads through space along with its reflection.

However, a system based on muzzle blast detection only is not necessarily robust. The unreliability is basically due to the fact that most of the acoustic energy propagates in the direction of the gunshot and decreases with increasing off-axis angle. In addition, weapons equipped with a silencer have significantly reduced level of the acoustic signal - muzzle blast.[3]

2.1.2 Shock wave

If a bullet is fired at supersonic speed (more than 343 m/s at 20 °C), a shock wave is formed in addition to the muzzle blast. The shock wave propagates conically and travels to the projectile's trajectory with a $\frac{\Theta_M}{2}$ angle. The angle of this cone, denoted as Θ_M, depends on the speed of the bullet, denoted as v, and is related to the Mach number M, which is defined as the ratio of the bullet's velocity to the speed of sound c. Specifically, the cone propagating angle can be expressed as

$$\Theta_M = 2 \cdot arcsin(\frac{1}{M}). \tag{1}$$

Therefore, the faster the bullet travels, the smaller the angle of the cone, and vice versa.

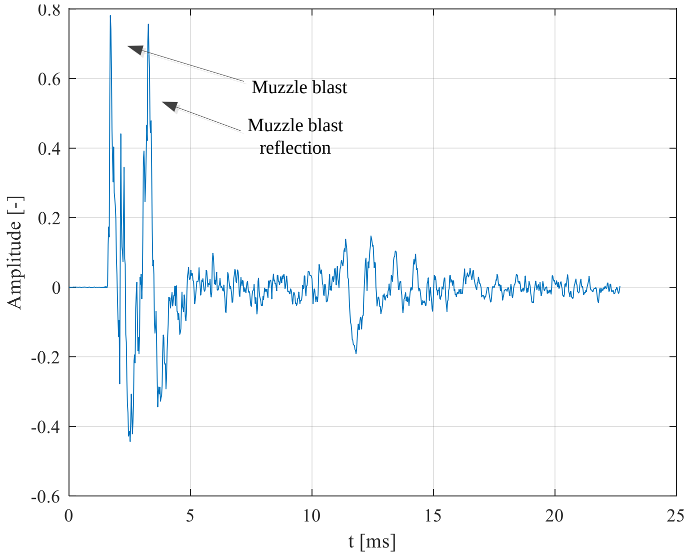


Figure 2.1. Recorded acoustic signal corresponding to a 7.65 mm subsonic short gun gunshot with reflection (Source: [4])

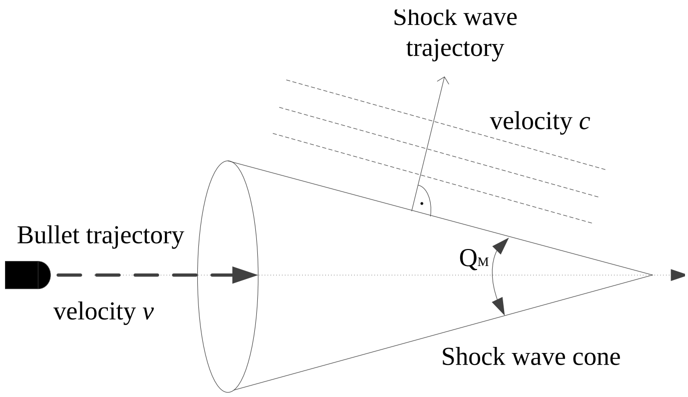


Figure 2.2. Acoustic wave of a supersonic bullet (Source: [4]).

2.1.3 Mechanical action

There are certain firearms that make audible mechanical sounds, such as the trigger and hammer mechanism, the ejection of spent cartridges and the loading of new ammunition by the weapon's manual or automatic loading system. However, the noise generated by these operations is generally much quieter than the sound of the shot and the shock wave of the bullet for supersonic projectiles. Therefore, these sounds are only significant if a microphone is placed close enough to the firearm to record them. This type of acoustic signal may be present in recordings in certain cases, but is generally not suitable for detection. [1]

2.2 Signal processing

As previously described, the shape of the acoustic signal generated by a gunshot could vary significantly, making it difficult to detect and classify. The system would have to use advanced signal processing to achieve accurate detection and classification, which would place significant demands on each detection unit.

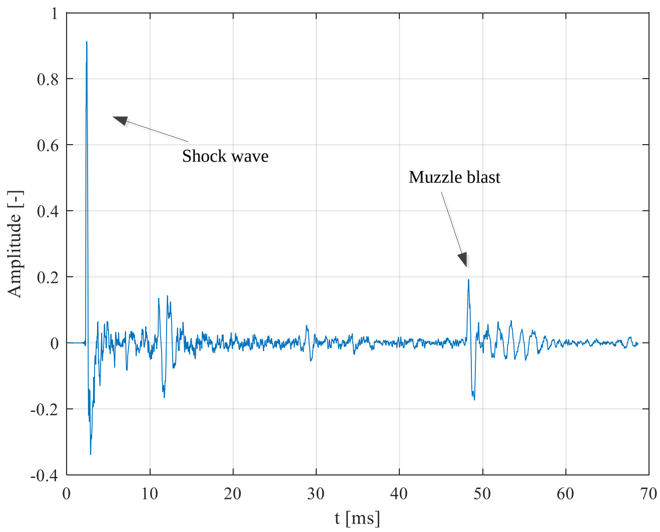


Figure 2.3. Recorded acoustic signal corresponding to a 9 mm supersonic short gun gunshot (Source: [4]).

To implement this acoustic event detection system, an approach was chosen in which the server will handle the advanced signal processing. On the other hand, the local units will be responsible for monitoring the surroundings and detecting acoustic pulses reliably.

If a high-energy acoustic signal is detected, the local unit will record the event and transmit it to the server. This work focuses specifically on the local part of the system, which monitors and sends recorded acoustic events that could indicate a gunshot.

2.2.1 Peak detection

The algorithm for acoustic event detection is a crucial element in this work. Its purpose is to reliably detect signal peaks, even when the signal contains a large amount of noise (SNR - signal-to-noise ratio around 20 dB). A suitable combination of methods and algorithms must be used to achieve this goal. In addition to the robustness of the detection, it is necessary to ensure the real-time applicability of the algorithm with the lowest possible computation time.

Various methods were tested and considered for the accurate detection of peaks in acoustic signals to achieve optimal results.

Among the naive approaches is a method based on pure signal thresholding. However, it is unable to adapt to signal variability. Nevertheless, if the signal is pre-processed, the functionality of this method can be significantly improved.

To enhance detection, pre-processing of the signal by accentuating rapid frequencies using the power of two on signal difference $diff(signal)^2$ was applied before thresholding. This technique enables the detection of higher frequencies, but unfortunately, it has several disadvantages. One of these disadvantages is that this method is sensitive to false detections at higher noise levels. Another drawback is the limited ability to adapt to various conditions and types of signals.

Another used improvement was use Teager's energy [5] of the signal instead of the raw signal, which improved method. Although better results were achieved, the desired detection accuracy was not reached. Additionally, the computational demand for detection significantly increased.

Several other methods for peak detection, including the Median Filter, Correlation Against a Template, and Discrete Wavelet Transform, were also considered. However, these methods were deemed excessively complex for the specific task of peak detection. While they have shown efficacy and applicability in gunshot detection, as demonstrated in [6], they were overly sophisticated for peak detection.

In the end, the Z-scores method was chosen, which proved to be the most suitable choice for peak detection. Z-scores are characterized by sufficient adaptability to the surrounding environment and their ability to adapt to various signals. Correct peak detection was achieved up to a SNR value 15. This method is considered suitable for this work as it allows accurate detection of peaks in time signals with sufficient reliability and adaptability.

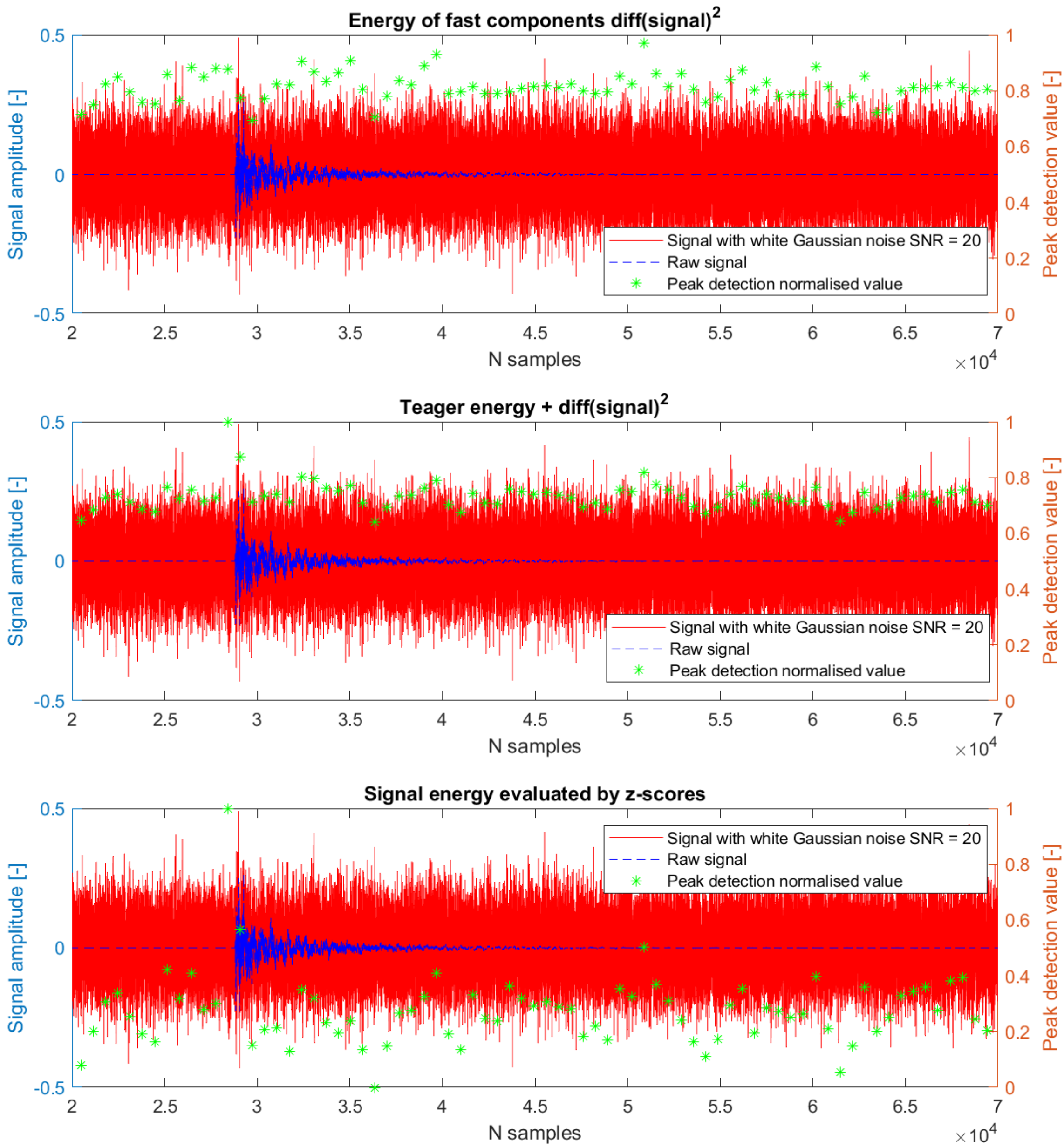


Figure 2.4. Comparasion between acoustic pulse detection at SNR = 20 dB

2.2.2 Z-score robust peak detection

The Z-score, also known as the standard score, is a statistical indicator that shows how far a given standard deviation value is from the mean in a given data set. It is the ratio of the difference between the value and the mean to the standard deviation

$$Z = \frac{X - \mu}{\sigma}, \tag{2}$$

where X is the observed value, μ is the mean, and σ is the standard deviation of the data set.

Z-score is often used to detect outliers and to identify extreme values in a data set. The method is also suitable for real-time data processing because it is sufficient to store a certain amount of historical data of the mean and standard deviation for calculating the Z-score. Therefore calculating the current Z-score for the newly arrived value at any point is possible.

For acoustic peak detection, Z-score proves to be a suitable method because at the moment of the shot, there is usually a significant standard deviation in the time series that indicates a peak.

A threshold is used to determine which values are classified as peaks. This threshold is based on a multiple of the mean of the energy value in last few samples, e.g., 3 or 5.

2.3 Localization based on signal theory

Localization of the shot is an essential aspect of this system, in addition to detecting and transmitting data to the server. There are various methods for determining the location using a network of units that monitor the surrounding environment, namely Angle of Arrival (AoA), Time of Arrival (ToA), Time Difference of Arrival (TDoA), and many others.

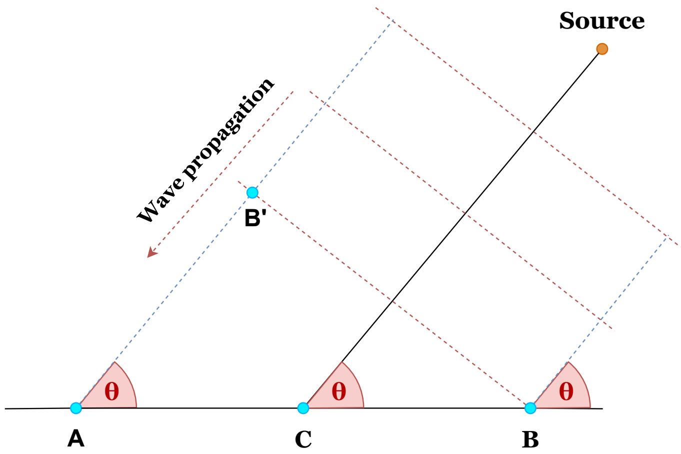


Figure 2.5. Angle of Arrival - wave aproximation by plane

The Angle of Arrival (AoA) algorithm was chosen for shot localization. The handling of this algorithm will be performed on the server side and thus is not a part of this work. However, it imposes hardware requirements on the detection unit.

2.3.1 Angle of Arrival

The Angle of Arrival (AoA) localization algorithm is a widespread technique employed for identifying the position of a wireless device. AoA is predicated on triangulation algorithms that necessitate knowledge of the locations of several reference points in combination with the angle at which the signal arrives at these reference points. This method involves determining the angles between the signal source and each reference point, subsequently identifying the intersection point of these angular lines within a 2D or 3D space.

2.3.2 Angle of Arrival derivation of the formula

To utilize the Angle of Arrival (AoA) method, it is necessary to derive the formula for the acustic detection system based on two microphones. This formula is derived in this section.

Assuming the distance of the acoustic signal source from the microphone is given by the equation

$$r > \frac{2d^2 f_{max}}{c}, \tag{3}$$

where d represents the distance between the microphones, f is the frequency, and c denotes the speed of sound. [7]

Under these conditions, the acoustic wave can be approximated as planar, which significantly simplifies the computation required to determine the angle of arrival of the acoustic signal at the unit.

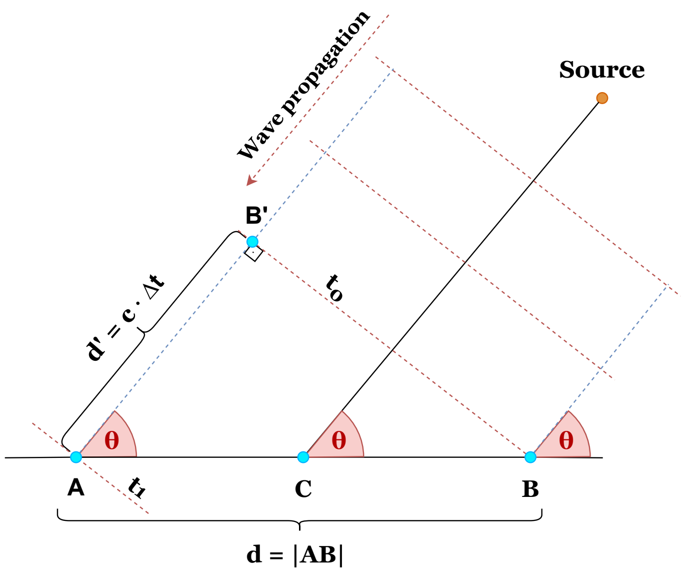


Figure 2.6. Angle of Arrival - computation visualization

Referring to Figure 2.6, two microphones, A and B, are assumed to be separated by a distance $d = |AB|$. Let the planar acoustic signal arrive at microphone B at time t_0. The distance $|AB'|$ is denoted by d'. This distance is equal to $d' = c \cdot \Delta t$, where c is the speed of sound and Δt equals the difference in times, t_1 and t_2, when the signal reaches microphone A and B, respectively. At this point, the angle of arrival can be expressed by the following equation

$$cos(\theta) = \frac{d'}{d}. \tag{4}$$

This formula can be rewritten to express θ in the following form

$$\theta = arccos(\frac{d'}{d}) = arccos(\frac{c \cdot \Delta t}{d}). \tag{5}$$

2.3.3 Huygens' principle

Huygens' principle is a statement formulated by Dutch physicist Christiaan Huygens in the 17^{th} century that explains how waves propagate through space. According to this principle, each point on the wavefront acts as a source of new waves that propagate in all directions. As these secondary waves interact, they interfere with each other, creating a new waveplate that propagates forward in a straight line.

This phenomenon is significant for the detection of acoustic signals because it allows the signal to propagate even if there is an object between the microphone and the source of the acoustic signal.

Hence, it also causes a variation in the distance traveled due to reflections and refractions of the signal. This causes deviations in the calculation of the localization of the acoustic signal source. A dense network of detection systems is needed to minimize this calculation error. [4]

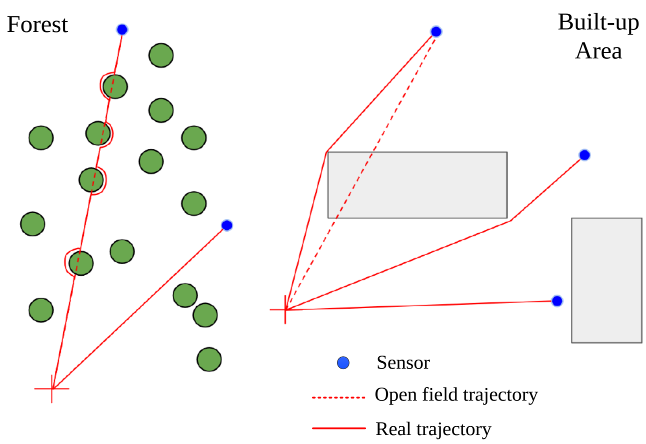


Figure 2.7. Sound propagation through different environments (Source: [4]).

Chapter 3
Hardware implementation of acoustic impulse event detection system

The preceding section described the theoretical foundations of acoustic signal processing, whereas this section is focuses on the hardware aspects of the system.

It is essential to ensure that each unit of the acoustic signal detection system satisfies the required specifications to achieve the intended functionality. Several requirements have been identified, namely device localization, precise timestamping, high-quality acoustic signal processing, and reliable communication with the server. A block diagram satisfying these requirements is shown in Figure 3.1. Each requirement is essential to the system's overall performance and was carefully considered during the design phase.

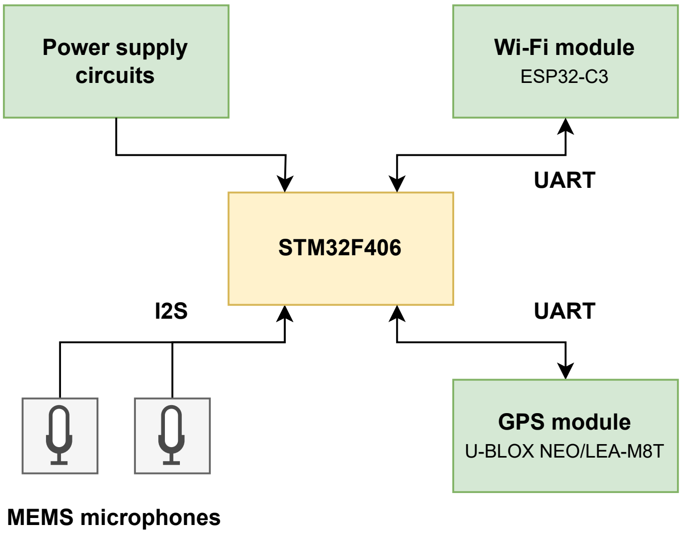


Figure 3.1. Block diagram of the system

3.1 MCU

The microcontroller unit (MCU) is the main component for signal acquisition, basic signal processing, peripheral operation, and communication with the server. It was crucial to select an appropriate MCU that meets the application requirements. Due to the global chip shortage caused by the pandemic, finding a suitable processor for this work was challenging. Almost all usable processors were bought out many weeks in advance. The finally chosen processor was one of the few suitable for this work, which was available on our faculty thanks to doc. Fisher.

The processor must possess adequate SRAM memory size to store data from two microphones and support communication protocols compatible with GPS, Wi-Fi, and MEMS microphones. The selected processor, STM32F407VGT6, provides 192 kB of

SRAM memory size, which is sufficient for storing microphone data for further processing. Additionally, the processor supports UART and I2S communication protocols, facilitating efficient communication with GPS, Wi-Fi, and MEMS microphones.

In summary, the STM32F407VGT6 processor fulfills the requirements for SRAM memory size to store data from two microphones and supports both UART and I2S communication protocols, enabling seamless and rapid communication with GPS, Wi-Fi, and MEMS microphones.

3.2 GPS localization

For accurate localization of each unit, it is necessary to work with precise time (maximal deviation of hundreds of ns) and position (with an accuracy of maximum units of meters). The GPS module is the ideal solution for both requirements. It is possible to choose from a wide range of available modules.

The most widely used are those from the Swiss company u-blox. The **NEO/LEA-M8T** module was selected with a time accuracy of up to 20 ns outside and up to 500 ns inside the building. Localization accuracy is quoted to be 2.5 meters. This module is also claimed to be a low powered with a maximum current of 67 mA. The average consumption is 28 mA. [8] The module is connected to the processor via a UART interface.

During the development process, prototype boards were created for testing purposes first, which helped with writing the firmware. The prototype boards allowed testing and debugging of the GPS module and its integration with the processor. Finally, this design was used for the final PCB. The board could then incorporate the improvements made during the testing phase.

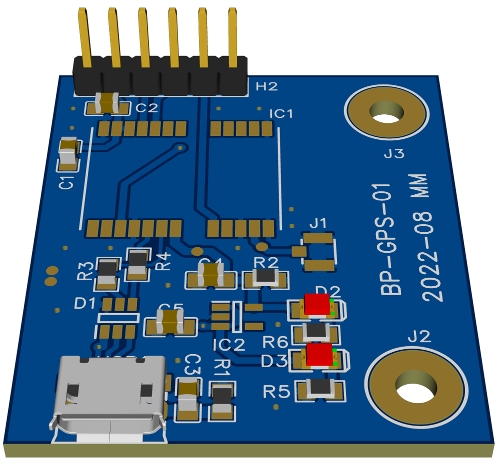


Figure 3.2. PCB prototype with NEO/LEA-M8T

3.3 Wi-Fi connection

Each acoustic event detection unit must be connected to the internet to send information about individual events. When an acoustic event is detected, the unit must send all data, including the entire time record of the event and additional data, such as location and recording timestamp, via Wi-Fi to remote servers for further processing and classification. LoRa technology was rejected due to low transmission speed and capacity, and the ZigBee protocol was also considered, but it has a lower range and less developed infrastructure.

There are many Wi-Fi modules. The **ESP32-C3** processor has been selected as the Wi-Fi module from the examined alternatives, as it offers superior availability, extensive support, represents a recent development, and remains cost-effective.

The ESP32-C3 processor, developed by Espressif Systems, is designed for low-power applications with Wi-Fi and Bluetooth support. This module is equipped with a range of peripheral devices and a rich set of features that make it an ideal solution for a wide range of applications. In this work, the ESP32-C3 processor is used for system communication via Wi-Fi with a TCP server as a Wi-Fi module.

This module communicates with the MCU through the UART interface. The utilization of AT commands for this purpose facilitates efficient management of the module. Employing these commands contributes to effective interaction between the module and the MCU. A more comprehensive description of the communication between the MCU and the module can be found in section 4.4.

A prototype board for the Wi-Fi module was also created during the development process, similar to the GPS module. The original plan was to use the ESP8266 processor, which was outdated and had a small amount of flash memory (1 MB) available for uploading the AT commands firmware. A new processor was chosen after an unsuccessful attempt to upgrade the flash memory to 4 MB. Insights from the development of the test board were used in the design of the final board.

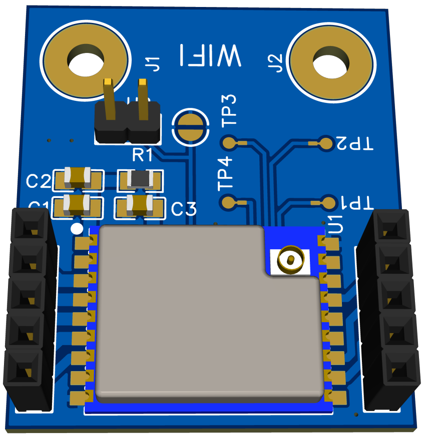


Figure 3.3. PCB prototype with ESP8266

3.4 MEMS microphone

It is required to select proper microphones to record acoustic events in high quality. The requirements for high SNR (greater than 60 dB), high sensitivity (greater than - 30 dBFS), frequency range 0 - 20 kHz, and directivity were key factors in the selection. There are many types of microphones, such as dynamic, condenser, piezoelectric, and MEMS microphones. Among those considered were electret and MEMS microphones.

The MEMS microphone was selected for the purpose of this work because this type of microphone already has a digital output, eliminating the need for an external analog-to-digital converter. As a result, the use of a MEMS microphone simplifies the circuit design. It reduces the device's power consumption by providing a simple connection to the processor without any other component. Compared to other types of microphones, MEMS microphones provide greater flexibility and digital output that simplifies data processing.

MEMS microphones use a microelectromechanical system to record audio signals. This type of microphone consists of miniature mechanical structures integrated into a silicon wafer. These structures are capable of generating electrical signals in response to acoustic waves, thus enabling the conversion of audio signals into digital data. MEMS microphones are characterized by their high sensitivity and low noise level, allowing even very delicate sounds to be captured. Their characteristics make them suitable for this application.

The digital MEMS microphone from STMicroelectronics **MP23DB01HP** was selected, which has a very low power consumption (285 μA - normal mode, 800 μA - performance mode), digital PDM output, SNR ratio of 65 dB, and sensitivity of -41 dBFS. Its characteristics make it ideal for small `Always-on` systems due to its small size, low power consumption, and good acoustic properties. [9]

3.4.1 PDM signal

Pulse Density Modulation (PDM) is a form of modulation used for analog signal transmission. A PDM signal is a binary data stream where a high pulse density in the signal means a high signal amplitude, and a low pulse density means a low signal amplitude. In this sense, a PDM signal is similar to $\Sigma - \Delta$ modulation. It is pulse width modulation (PWM), where the width of the individual pulses represents the signal's amplitude.

The PDM signal allows digital transmission from the microphone with minimal signal distortion without using an AD converter. The processing of the PDM signal on the MCU is performed using digital signal processing. The PDM signal is transferred to the MCU using the I2S audio bus, but other serial communications, such as SPI, DFSDM, and SAI, can also be used. [10]

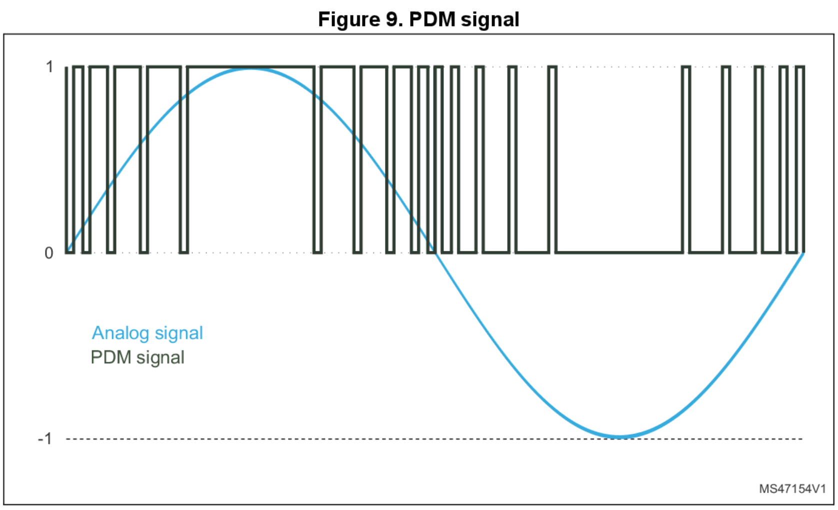


Figure 3.4. PDM signal waveform from MEMS microphone converted to PCM signal (Source: [10])

3.5 Power

In order to ensure the proper functioning of the system, it is important to address the power supply requirements effectively. Acoustic event detection systems need to be placed in various locations, which can present challenges in providing a consistent and reliable power source. As a result, Power over Ethernet (PoE) has been chosen as the preferred power supply method, allowing the system to be powered by a single cable, even over long distances (100 m for IEEE 802.3at [11]). The integration of additional circuits with the Power over Ethernet (PoE) setup is necessary to ensure the reliable operation of the device. A more detailed explanation of this process can be found in the following sections of this chapter.

3.5.1 Power over Ethernet

PoE is a technology that allows the transmission of electrical energy and data through standard Ethernet cables. This technology simplifies powering devices, even in places where it would typically be challenging to provide a source of energy.

In this work, PoE is used solely as a means of power transmission and does not serve for communication. Considering the future development of this system, a power supply based on PoE has already been implemented. In the future, communication via Wi-Fi could be complemented by Ethernet cables.

The PoE standard is defined according to IEEE standards, specifically IEEE 802.3af, IEEE 802.3at, and IEEE 802.3bt.

It was selected to use the IEEE 802.3at (also PoE+) standard, which is newer and offers several protective mechanisms that ensure the safety and reliability of power transmission between PSE (Power Sourcing Equipment) and PD (Powered Device).

One of these is the detection mechanism determining whether the PD is compatible with PoE. Furthermore, classification allows the PD to inform the PSE of its energy consumption, which enables the PSE to manage power allocation and ensures higher reliability for PD.

In addition to detection and classification, this standard also provides protection against overloading and short circuits, which disconnects the device from power if an overloading or short circuit is detected. This protection safeguards both PSE and PD from damage and ensures safe operation.

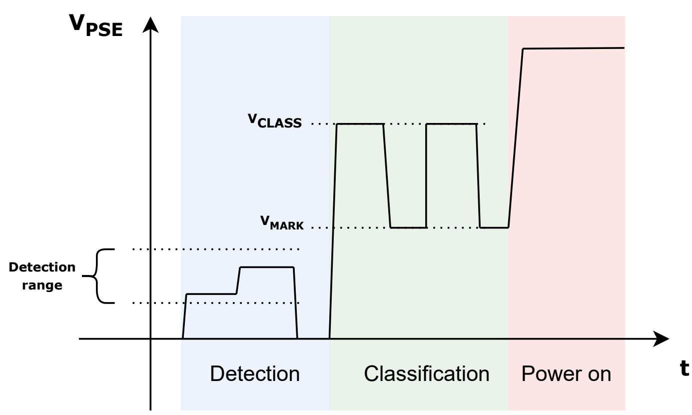


Figure 3.5. PoE Detection and Clasification start-up for TPS2379 [12]

Another advantage is the guaranteed backward compatibility with the 802.3af standard, making it possible to connect devices to these PSEs as well. [11]

To control the active PoE in this system, the integrated circuit TPS2379DDAR was selected. This IC provides efficient and reliable power transmission, detection, and classification mechanisms between the PSE and PD. Its selection ensures the stable and safe operation of the PoE system, and its compatibility with the IEEE 802.3at standards allows for flexibility in future upgrades and expansions, including the possibility of communication over Ethernet.

3.6 PCB

The Printed Circuit Board (PCB) is this acoustic detection system's primary hardware component, containing the necessary components and modules for its operation. This section provides explanation of the design of the PCB, which is split into two subsections.

The first subsection focuses on the main board containing the system's core components. The second subsection is dedicated to a smaller board that holds the MEMS microphone.

3.6.1 Main PCB

The main board is designed to accommodate all the modules mentioned earlier, which are necessary for the system. Based on the experience earned from designing prototype boards for the Wi-Fi and GPS modules, a similar configuration was adopted with minor alterations to enhance functionality.

A significant feature of the main board is the power management section, which contains a Power over Ethernet (PoE) and a DC/DC converter. The converter transforms the supplied power to a 5 V level. A voltage regulator further ensures a stable 3.3V for MEMS microphones, ESP32-C3, and U-BLOX module.

Furthermore, the main board is designed with connectors for two MEMS microphones. These connectors serve as the interface between the microphones and the rest of the system, ensuring proper signal transmission between the two microphones and the main board.

3.6.2 Microphone board

The microphone board is a component of the system, primarily responsible for the precise positioning of the MEMS microphone. It features a connector for interfacing with the main board, ensuring communication and signal transmission.

The placement of the microphone is essential for determining the angle of signal propagation of acoustic impulse events. By measuring the delay between the arrival times of these events at each microphone, the system can accurately calculate the direction from which the acoustic impulse wavefront is propagating. In this specific application, the microphones are positioned 150 mm apart. This distance has been selected to provide an optimal balance between spatial resolution and the size of the system, ensuring accurate detection and localization of the sound sources in the environment.

3.7 Case for Board

The enclosure is divided into four distinct parts. The primary section is designed to contain the main PCB, which includes the processor, WiFi module, GPS module, and

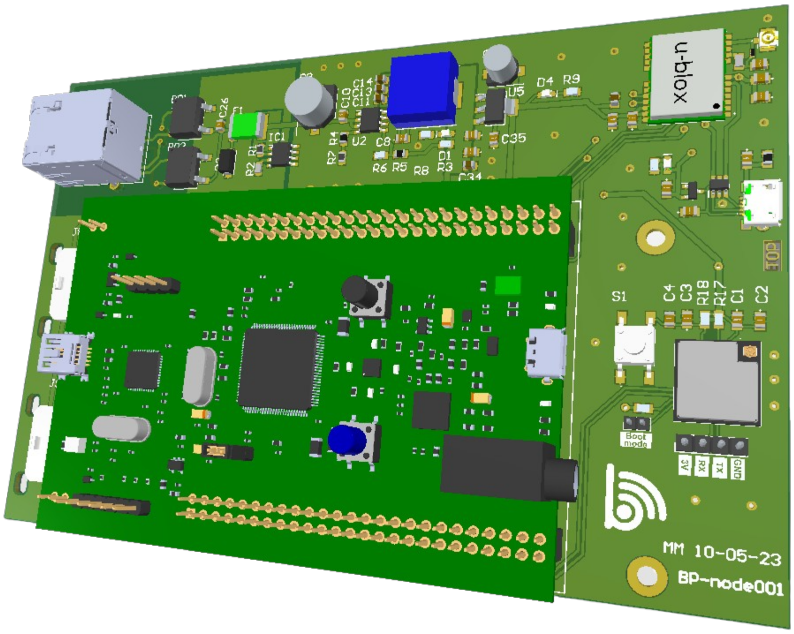


Figure 3.6. Main PCB for acoustic pulse detection system

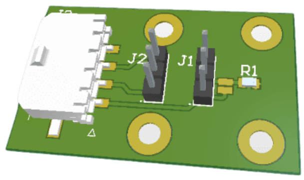


Figure 3.7. PCB holder for MEMS microphone

power circuits. This design ensures that the core elements of the PCB are protected. Additionally, the central part of the model is prepared to accommodate a sealing mechanism, providing an additional layer of protection against environmental factors such as dust and moisture. Threaded inserts are used to guarantee that the PCB remains stable and secure during operation.

A second module is designed specifically for PCB with microphones. This module, connected to the bottom of the central part, positions the microphones optimally to monitor their surroundings without barriers or unnecessary sound reflections.

The model of the enclosure is illustrated in Figure 3.8. It was designed using Autodesk Inventor. The design has been prepared for 3D printing, ensuring ease of production and assembly. Detailed drawings of each part of the model can be found in the appendices, providing further insight into the design.

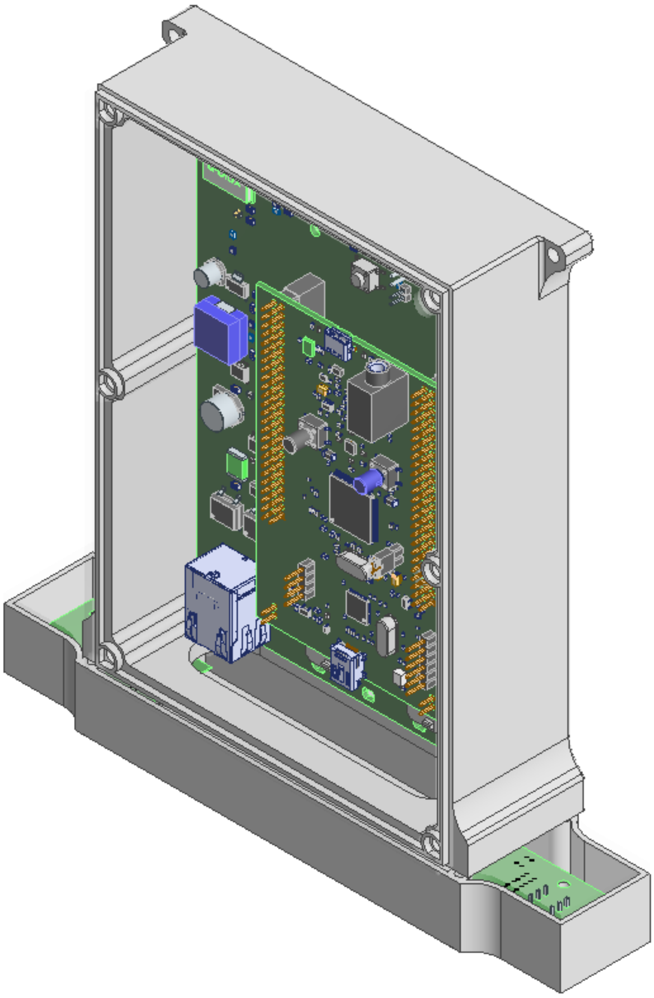


Figure 3.8. Enclosure

Chapter 4
Firmware

The aim of this chapter is to provide a detailed overview of the firmware architecture designed for the proper functioning of the gunshot detection system. Key functions and techniques used for processing the acoustic signal, such as signal processing, robust peak detection, and transmission to a remote server, are described.

Additionally, an overview of the firmware architecture, its components, and their descriptions, including the interaction between the firmware and individual peripherals such as the MEMS microphone, Wi-Fi module ESP32-C3, GPS module NEO/LEA-M8T, and MCU is provided.

Furthermore, the algorithms used for gunshot detection and sound signal processing are described in this section.

Finally, the section discusses testing and validation methods used to verify the functionality of the firmware. The methods used to test the firmware's functionality are described, and the results of these tests are presented.

4.1 Acustic signal processing

Acoustic signal processing is crucial for accurate and reliable gunshot recognition in various environments and situations. It is necessary to ensure proper acquisition, analysis, and processing of acoustic signals using several techniques and algorithms to achieve this goal.

This section describes the process of acquiring data from MEMS microphones and converting this data from PDM to PCM format, which is an essential step for further analysis and processing of the acoustic signal.

Following, peak detection in the signal using Z-score is discussed, a statistical method that helps identify critical parts of the signal associated with gunshots. This method requires the windowing technique, which is also described here. It is widely used for analyzing acoustic signals because it allows dividing the signal into smaller parts, which are then analyzed separately, significantly improving the system's ability to detect gunshots in various environments and under different conditions.

4.1.1 Acustic signal data acquisition

For the system's proper functioning, it is necessary to obtain accurate acoustic signal data. For this purpose, a MEMS microphone was chosen, described in subsection 3.4. Its output is a PDM (Pulse Density Modulation) signal, which needs to be converted to PCM (Pulse Code Modulation) format for further analysis and processing. Acoustic data are acquired using the I2S (Inter-IC Sound) interface, which allows reading PDM data from the MEMS microphone. The Direct Memory Access (DMA) method is used for efficient data reading, which minimizes the processor load during data transfer. Specifically, the *HAL_I2S_Receive_DMA* function from the HAL library is used.

Frstly one half of the buffer *PDM_buff* is filled with data, while the other half remains empty and ready for further filling. Once the first half of the buffer is filled, a callback function is called, which processes this data and stores it in a larger circular buffer, where the data is stored for following processing. In the meantime, more data starts to be stored in the second half of the buffer. After processing the first half, it starts to be filled again, and the second half is processed. This process alternates, thereby achieving efficient data processing.

PDM data is converted to PCM format using the *pdm2pcm.h* package provided by STMicroelectronics. Specifically, the *PDM_Filter* function is used in the callback. This function processes the data stored in the buffer *PDM_buff* and converts it to PCM format, which is suitable for further acoustic signal analysis.

```
void HAL_I2S_RxHalfCpltCallback(I2S_HandleTypeDef *hi2s) {

   // Define temporary arrays for PDM and PCM data
   uint16_t AppPDM[128 / 2];
   uint16_t AppPCM[16];

   // Change of endianness and store it in AppPDM array
   for (int index = 0; index < 128 / 2; index++) {
      AppPDM[index] = HTONS(PDM_buff[index]);
   }

   // Apply PDM filtering to convert PDM data to PCM data
   PDM_Filter(AppPDM, (uint16_t*) (AppPCM), &PDM1_filter_handler);

   // Push the processed PCM data into the PCM_data structure
   push_to_pcm_data(PCM_data, AppPCM);
}
```

The acquired PCM data are stored in the PCM_data structure, the definition of which is shown below. The structure includes the necessary variables for the circular buffer, pointers, and counters used for windowing and the calculation of z-scores.

```
typedef struct
{
   uint16_t *pcm_data;
   int16_t head;
   int16_t tail;
   bool recording;

   // window function
   int16_t win_idx;
   int16_t win_tail;

   // z-score
   float *energy_data;
   float *z_score_lap;
   int16_t zscr_head;
} pcm_data_t;
```

The *pdm2pcm.h* library package contains functions that convert a PDM data stream from MEMS microphones to PCM. The PDM2PCM library has the function to decimate and filter a PDM data stream from a digital microphone, which is converted into an output PCM signal. The output PCM stream is implemented with 16-bit resolution. Various decimation factors can be configured to adapt to different PDM clocks. The library also offers a configurable high-pass filter and digital volume. [13]

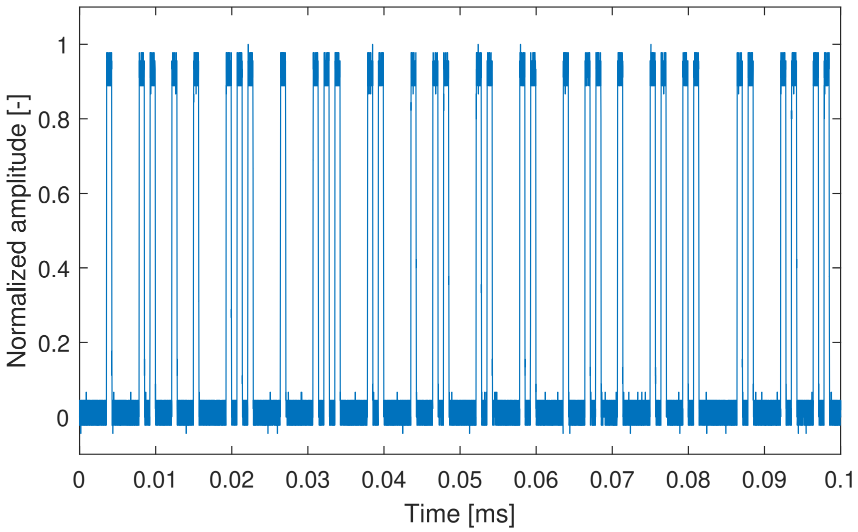


Figure 4.1. Acoustic data acquired from MEMS microphone

4.1.2 Windowing function

The windowing method is used for peak detection, which allows for detecting peaks in individual segments of the acoustic signal in real time. In this work, this method is applied with a strong emphasis on efficiency.

When storing PCM data in the buffer, a variable that tracks the amount of data needed to fill one window is incremented.

When applying the windowing method, only a pointer to the start of the window in the circular buffer data structure is retained, from which the data is then read. After processing, this pointer is incremented by the window length W - O, where W is the window length in N samples, and O is the overlap in N samples. The system then waits for the next window to be filled.

This method is essential for subsequent processing, as described in the following chapter. Once this length is reached, a window is prepared for processing.

The window length was chosen to be 20 ms with a 25 % overlap (5 ms). For a sampling frequency of 16 kHz, the window length in samples is given by N = fs * t and amounts to W = 320 samples. The overlap length in samples is equal to O = 80.

The window length was determined based on the typical duration of the acoustic impulse signal of a gunshot. This signal typically has a peak duration of a few milliseconds [2]; thus, the window should contain the entire peak recording.

4.1.3 Z-score peak detection

The *z_score* function is used for peak detection in the acoustic signal using the Z-score calculation, which has already been mentioned in subsection 2.2.2. For the purpose of the calculation, the signal energy value is computed using the *energy(pcm_data)* function. Subsequently, the *zscr_head* pointer in *pcm_data* is updated, and the computed energy value is stored in the circular buffer *zscr_energy*. The average value and standard deviation of the energy in the *zscr_energy* buffer need to be calculated. Then, the

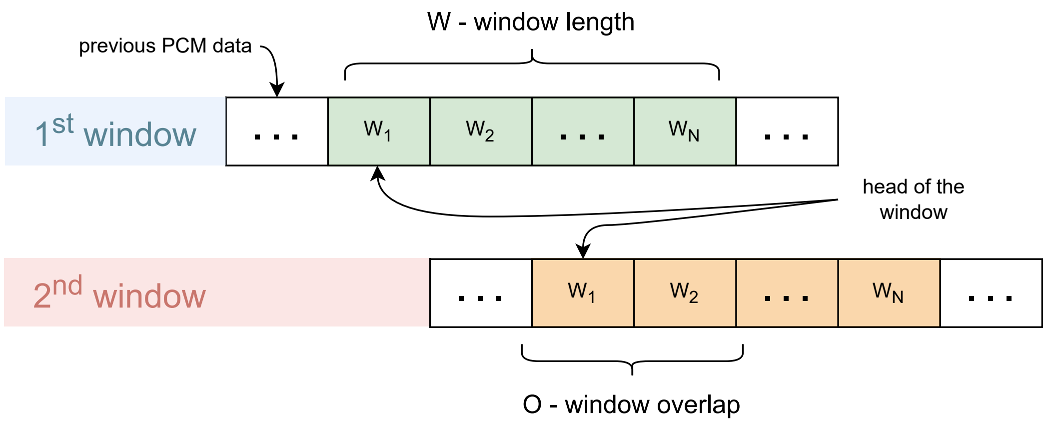


Figure 4.2. Acoustic signal processing using the windowing method in a ring data buffer

Z-score for the current energy value is calculated as *(energy_val - mean_val) / std_val*, and this result is stored in the *zscr_lap* circular buffer.

The average Z-score value for the *zscr_lap* buffer is then computed. The function checks whether the current Z-score is greater than twice the average Z-score value. If this condition is met, a peak is detected.

Therefore, the *z_score* function represents a crucial part of the gunshot detection algorithm, focusing on the Z-score calculation and identification of significant signal parts associated with gunshots.

```
FUNCTION z_score(pcm_data)
   // Update the head of the circular buffer
   UPDATE zscr_head in pcm_data

   // Calculate the energy of the acoustic signal
   energy_val <- COMPUTE energy(pcm_data)
   zscr_energy[zscr_head] <- energy_val

   // Calculate the mean and standard deviation of the energy data
   mean_val <- COMPUTE mean of zscr_energy buffer
   std_val <- COMPUTE std of zscr_energy buffer and mean_val

   // Compute and store the Z-score value
   z-score <- (energy_val - mean_val) / std_val;
   zscr_lap[zscr_head] <- z-score

   // Compute mean value of z-score
   zscr_mean <- COMPUTE mean of zscr_lap buffer

   // Peak detection
   IF z-score > 2 * zscr_mean THEN
      DETECT peak
   END IF

END FUNCTION
```

4.1.4 Time difference of signals

After capturing peaks on both microphones, it is possible to calculate the time delay, which is determined by the distance expressed in the number of samples between individual peaks. The program evaluates this distance and, assuming a sampling frequency of 16 kHz subsequently determines the delay. This procedure ensures the precise determination of the delay between the two microphones. Accurately established delay enables the server-side evaluation of the angle of incidence on the unit and subsequently performs the localization of the acoustic impulse event.

4.2 Comunication protocols

This section aims to describe the communication protocols utilized with the periphery, complementing the information provided in the previous section and contributing to the completeness of this thesis. The communication protocols discussed include the Universal Asynchronous Receiver-Transmitter (UART) and the Inter-IC Sound (I2S). These protocols are employed to effectively communicate between components within the system, ensuring its overall reliability.

4.2.1 UART

The UART is an asynchronous serial protocol that enables data transfer between devices. This protocol consists of two signals, TX (transmit) and RX (receive), to carry out data transfer. The microcontroller sends data using the TX signal and receives data using the RX signal. Each device handles the timing of the signal and therefore the baud rate must be selected at the beginning. Furthermore, a synchronous USART variant is also exist.

The transmitting device sends a data frame which is sent serially bit by bit. Each data frame contains a start bit, followed by 5-9 data bits and one or two stop bits. A parity bit can also be added before the stop bits. The start bit signals to the receiving device that new data is forthcoming and is succeeded by the data transmitted with the least significant bit first.

The UART communication protocol used for communication between the MCU and the GPS and Wi-Fi modules.

4.2.2 I2S

The I2S communication protocol is widely used for communication between MCU and peripherals. I2S was initially developed by Philips Semiconductors (NXP Semiconductors now) and is a synchronous, serial communication protocol designed primarily for transferring digital audio data between integrated circuits in an electronic device [14]. Unlike the previously discussed UART protocol, which is designed for asynchronous, bidirectional communication, I2S is optimized explicitly for the needs of digital audio systems.

Within the scope of this thesis, I2S serves as the communication medium between the processor and the MEMS microphone. The data transmitted using I2S are in Pulse Density Modulation (PDM) format. In the section 3.4 and 3.4.1 the MEMS microphones and the PDM signal are described in more detail.

The I2S protocol employs three separate lines for communication: a clock line (SCK), a word select line (WS), and a data line (SD) [14]. The clock line synchronizes the data transfer, while the word select line indicates the beginning of a new data word and distinguishes between left and right channels in a stereo audio system. The data line carries the actual digital audio data. With support for various data word lengths and sampling rates, the I2S protocol is a flexible choice for various audio applications.

Furthermore, in the implementation of the I2S protocol within the MCU of this project, DMA is utilized to efficiently transfer data without overloading the MCU. The system utilizes a 16-bit data word length within a 16-bit frame, which allows for the effective transmission of high-quality digital audio data. The chosen audio frequency for this application is 32 kHz, providing a suitable sampling rate that balances audio quality and resource consumption.

4.3 GPS module

The GPS module communicates with the processor through the UART (Universal Asynchronous Receiver-Transmitter) interface. The communication is performed using two protocols, which are described below. Additionally, the EXTINT pin is used for precise timing, which allows requesting a timestamp. This is discussed in subsection 4.3.3.

4.3.1 NMEA

The communication protocol used by the GPS module to communicate with the processor is NMEA 0183 or UBX. NMEA 0183 is a standardized protocol for transmitting position data. This protocol defines the message format and rules for their exchange. The message format is defined in [ublox-protocol]. The attached log of messages demonstrates examples of messages in the NMEA 0183 format that the processor continuously receives. These messages contain information about the geographical position, time, velocity, and other relevant data.

```
$GNGSA,A,3,01,04,31,17,19,28,,,,,,,15.70,4.38,15.081F
$GPGSV,3,1,09,01,47,149,41,03,81,024,14,04,53,202,37,17,37,264,1673
$GPGSV,3,2,09,19,34,293,25,21,26,149,21,25,00,019,,28,24,051,1976
$GNGLL,5005.01244,N,01426.79060,E,180732.00,A,A72
$GNRMC,180733.00,A,5005.01238,N,01426.79072,E,0.441,,200523,,,A65
$GNVTG,,T,,M,0.441,N,0.816,K,A33
$GNGGA,180733.00,5005.01238,N,01426.79072,E,1,06,4.38,347.0,M,44.3,M,,44
```

These messages provide information about:

- GNGSA: Satellite constellation and fix precision.
- GPGSV: Information about visible satellites and signal quality.
- GNGLL: Geographic position in latitude and longitude.
- GNRMC: Recommended minimum GNSS data, including position and ground speed.
- GNVTG: Course over ground and ground speed.
- GNGGA: Position data, including latitude, longitude, altitude, and time.

In this work, only the GNGLL message is decoded to obtain information about the unit's position.

4.3.2 UBX Protocol

The UBX (u-blox Extended) protocol is a proprietary serial communication protocol developed by u-blox for interacting with their GPS/GNSS modules. The UBX protocol is designed to be efficient and easy to use. It utilizes a binary message format optimized for efficient data transmission between the GPS module and an external device. Each message in the UBX protocol consists of a header, data, and a checksum for data integrity verification.

The TIM-TIM2 message, used for precise timestamping, is part of the UBX protocol. This message provides a synchronized time reference for accurate time measurements.

4.3.3 Precise Timing

The GPS module features the capability of precise timing with an accuracy of tens of nanoseconds using an external interrupt. The reference time can be selected by setting the time source parameter to UTC, GPS, GLONASS, BeiDou, and Galileo using the UBX-CFG-TP5 message. The processor receives the precise timestamp in the UBX-TIM-TIM2 message when the UBX-TIM-TIM2 message is enabled, or a falling or rising edge is triggered on the EXTINT pin. It is through handling the EXTINT that the system generates an interrupt and requests the precise timestamp. [15]

4.4 Wi-Fi module

As section 3.3 mentions, the ESP32-C3 processor can operate as a Wi-Fi module controlled by AT commands. For this purpose, firmware named AT Command Set has been loaded into the processor. The firmware developed by the manufacturer supports a set of text commands used for communication with the module via a serial interface.

AT commands (AT standing for attention) were originally developed by Hayes Microcomputer Products for analog modems and have since become the standard for communication with modems and other devices. Nowadays, the Hayes AT command set includes commands for data, fax, voice, and SMS communications. [16]

With the firmware provided by Espressif Systems, the ESP32-C3 can be configured to operate as a Wi-Fi module controlled via the UART interface. In the following section, the steps involved in configuring the module to connect to a Wi-Fi AP and a TCP server will be outlined. These example will cover the necessary commands for setting up the Wi-Fi and TCP connectivity, while functionality checks and other details will be omitted. A complete overview of the AT Command Set is available on the Espressif Systems website [17].

4.4.1 Module settings

To establish a Wi-Fi and TCP connection using the AT Command Set for ESP32-C3, follow these steps.

Firstly, enable the configuration saving to the NVS area using the command

```
AT+SYSSTORE=1.
```

Next, set the Wi-Fi mode of the ESP32-C3 using the command

```
AT+CWMODE=<mode>[,<auto_connect>].
```

Set the <mode> parameter to `station mode` with an automatic connection.

Connect to the desired AP using the command

```
AT+CWJAP=[<ssid>],[<pwd>][,<bssid>][,<pci_en>][,<reconn_interval>]
          [,<listen_interval>][,<scan_mode>][,<jap_timeout>][,<pmf>].
```

Once connected to the Wi-Fi network, set up the TCP server using the Single Connection mode (AT+CIPMUX=0) command, followed by

```
AT+CIPSTART=<"type">,<"remote host">,<remote port>[,<keep_alive>]
            [,<"local IP">].
```

The <`type`> parameter specifies the type of connection (TCP/v6, UDP/v6, SSL), while <`remote host`> and <remote port> define the remote IP address and port, respectively. Additional parameters, such as the keep-alive interval and local IP, can also be specified.

Note that these settings can be configured before connecting to the system, allowing the module to connect automatically to the specified AP and TCP server. This passive setup enables seamless connectivity without the need for manual intervention.

4.5 TCP

The TCP (Transmission Control Protocol) protocol is used for communication between the MCU and the TCP server. TCP is a transport protocol within the Internet protocol family that enables reliable communication between network devices.

The first subsection focus on data transmission from the MCU. It describes the MCU's principles and procedures to create data frames transmitted via the TCP protocol. The frames contain sound data, metadata, and other necessary information for an accurate evaluation on the server side.

The second subsection briefly describes the functionality of the TCP server, which receives data and metadata from the MCU.

4.5.1 Sendig data

As mentioned before, this work uses a Wi-Fi module for data transmission from the MCU. The MCU communicates with the Wi-Fi module through the UART interface. To facilitate communication over UART, transmitting data in the form of ASCII characters is necessary. Therefore, the measured values are converted as noted below.

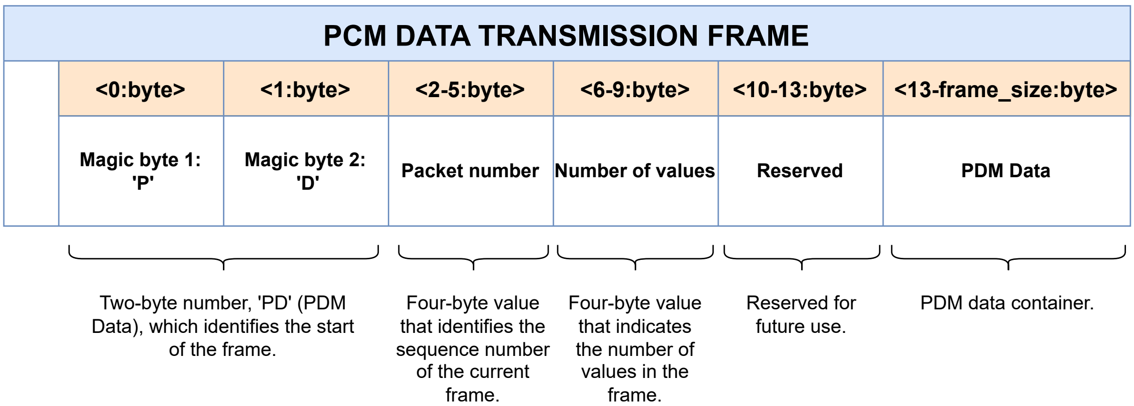


Figure 4.3. Structure of data transmission frame containing acoustic signal data after peak detection

After detecting peaks and calculating the time delay between microphones, the data is gradually sent within data frames. The schema of such a frame is illustrated in Figure 4.3. The frame consists of a header and data. The header for transmitting PDM data includes the identifier `PD`, followed by the frame number and the number of values in a single frame. Each item in the header is encoded into 4 bytes. The header also contains four reserved bytes for future use. Currently, these reserved bytes hold the value 'XXXX'.

Following the header, the measured values are stored in hexadecimal format in ASCII representation within 4 bytes. The measured data, which is of type uint16_t and ranges from 0 to 65535, is stored in this format.

After successful data transmission, metadata labeled as `MD` is sent. These metadata include information about the angle and time delay between microphones.

4.5.2 TCP server

The TCP server is implemented in Python using the standard socket module. It serves as the receiving part for data transmission and utilizes socket communication to establish client connections.

The server is configured to listen on a specific address and port. It starts listening for incoming connections once the socket is successfully bound to the designated address and port.

When a client connects to the server, a connection is established. The server then receives data from the client using the recv function. The received data is stored in a file for further processing and analysis. The server continuously receives data until all the data is received.

Upon completion, the server closes the connection with the client and properly terminates the connection.

The received data is further processed and stored in a file, where the data is formatted for easy processing. Specifically, the hexadecimal values are converted to decimal format, and commas separate individual values.

This server implementation covers the essential aspects of TCP communication, including socket initialization, connection establishment, data reception, and control termination. It provides a foundation for processing data sent from a microcontroller.

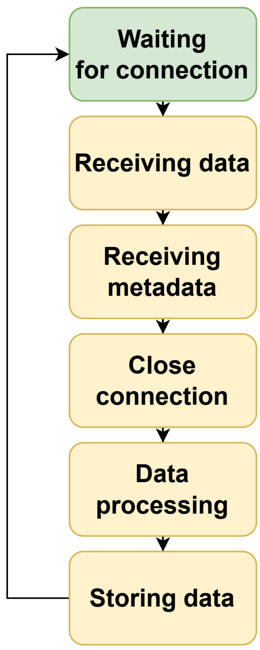


Figure 4.4. TCP server architecture diagram for receiving data from an acoustic detection unit

4.6 Program flow overview

In previous sections, individual parts of the firmware, which controls the system for acoustic detection, were described. This section describes the whole principle without going into deeper details. Those are described in the previous sections. This section combines individual information and connects them into a comprehensive overview of how the program works.

The unit constantly monitors its surroundings using MEMS microphones and records acoustic signal data. These data are stored in a structure used for proper storage and processing. Every time data is saved a check if the window used for peak detection is filled.

Peak detection is based on the z-score calculation, which is used to identify significant signals. If a peak is detected, a time stamp is recorded. After detecting a peak on both microphones, a time difference between these microphones is calculated, which allows for calculating the angle of the acoustic signal's incidence on the serves side. This information is necessary to determine the direction of the sound source.

After acquirement the timestamp, the system waits for the data buffer to be filled with PCM acoustic data to the required record length. Once this state is reached, the data acquisition from the MEMS microphone is terminated, and the process moves to the next phase. The buffer is filled to 3/4 of its size. 1/4 remains for data preceding the acoustic event. The processed data is gradually sent in predefined frames. These frame structures ensure the correct arrangement and formatting of the data for transmission. Each frame contains specific information, such as the frame number, the number of values, and the actual data in ASCII/hex format.

After successfully transmitting all data, the reception of new data is restarted, and the entire program runs in a continuous loop, which allows for the monitoring and processing of sound signals in real-time. In this way, it is ensured that the unit operates continuously and provides relevant information about the sound environment. The entire process is designed to be reliable, efficient, and capable of correctly processing and transmitting the necessary data.

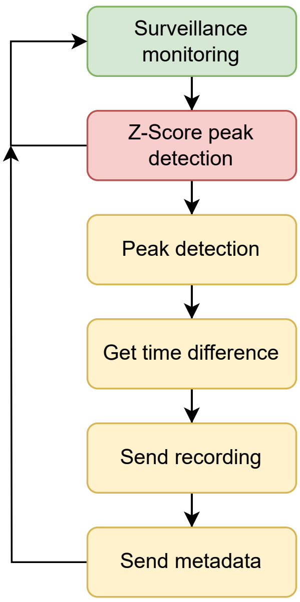


Figure 4.5. Acoustic detection unit architecture diagram

Chapter 5
Results

This chapter is dedicated to the presentation of results achieved throughout the realization of this bachelor's thesis. The forthcoming subsections elucidate the development of a system capable of detecting impulsive acoustic events, the technical challenges met and subsequently overcame, and the proposed solution. Examining these results is necessary to evaluate the overall accomplishment of this thesis, and it enables the identification of areas that could be subjected to future research and development.

5.1 Hardware

Throughout this study, the components necessary for the development of this system were carefully selected. These individual components were tested on specially designed prototype boards, which expedited the development and helped verify the functionality of the design. The insights collected from prototype boards facilitated the creation of the main PCB, which handled previous limitations. The main PCB houses the processor, a Wi-Fi module, a GPS module, connectors for MEMS microphones, and power circuits. Regrettably, due to the late delivery of the main PCB, the system was tested exclusively on prototype boards.

A small board and a casing were designed to ensure the accurate positioning of microphones at precise distances from each other. The case was developed as part of prototype development to be easily 3D printed while meeting the requirements for airtightness and resistance to water and dust in outdoor environments.

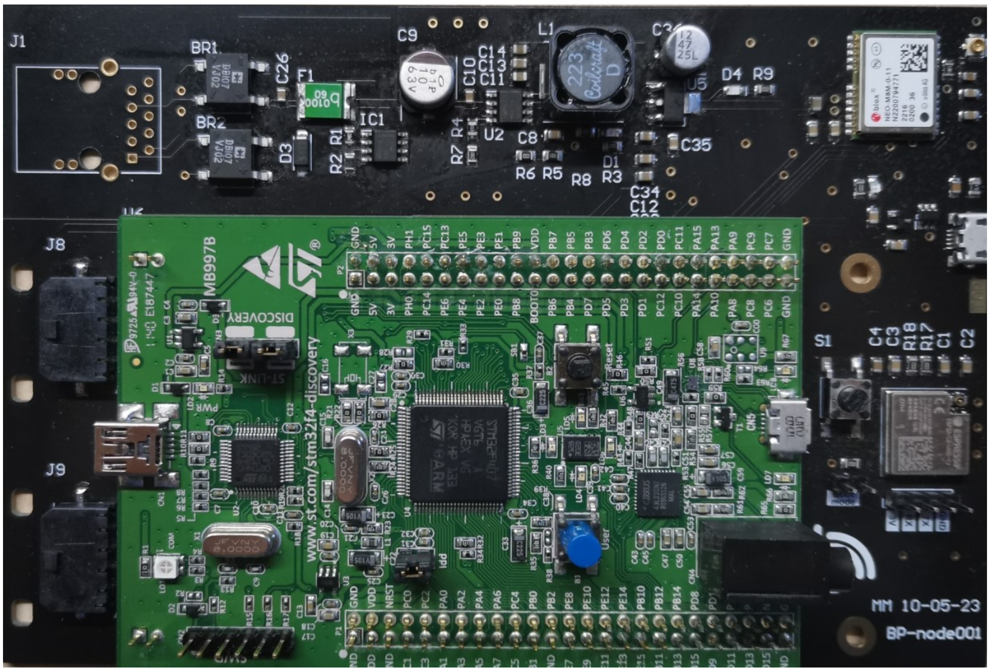


Figure 5.1. Assembled main PCB for the acoustic detection unit

5.2 Firmware

The firmware was developed considering the desired characteristics and the components utilized. During testing, particular attention was given to the peak detection algorithm based on the z-score method that assesses the energy of individual windows of the acoustic signal. This technique proved reliable during testing in Matlab and implementation on the device. This algorithm demonstrates robustness and yields reliable results with correct settings. Furthermore, the necessary program elements for operating the MEMS microphones through the I2S bus using DMA were written.

Consequently, data are continually stored in a structure where they are processed and evaluated. After the peak detection and acquisition of the acoustic signal, methods were established to acquire a timestamp from the GPS module, determining the delay between the microphones. This information is subsequently used to determine the angle of arrival of the acoustic signal. The final segment of this chapter saw the development of a mechanism for transmitting data to the server. Figure 5.2 illustrates the record of an acoustic event captured by the device, which was then sent to a TCP server. The server received and processed the data into the final form visible in the figure.

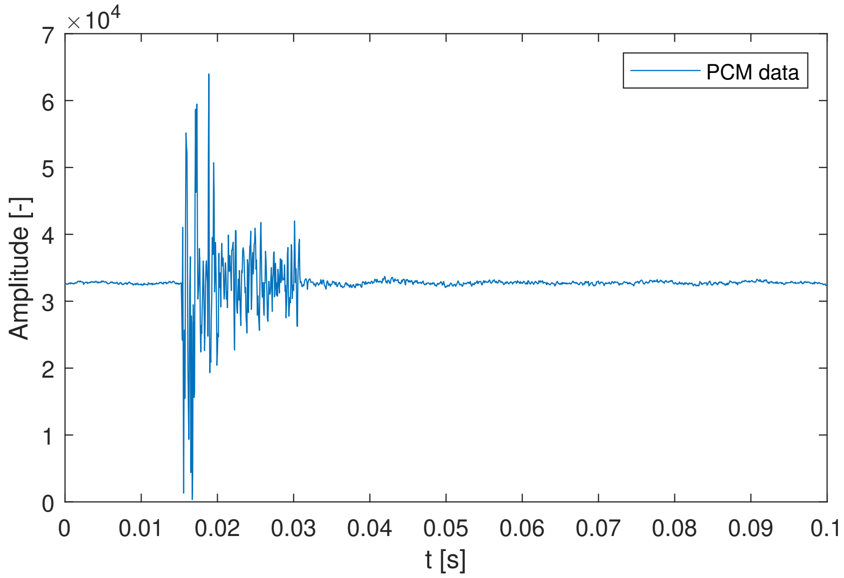


Figure 5.2. Record of an acoustic event captured by the device

Chapter 6
Conclusion

This bachelor's thesis focused on the design of a system capable of detecting acoustic impulse events. The development process can be partitioned into three distinct stages.

Firstly, it was necessary to understand the given problem and comprehend the topic of gunshot detection. This is discussed in the first chapter of this thesis.

In the subsequent chapter, the task was to design the hardware component of the system. The assignment required a device containing a communication unit, a GPS, and a microcontroller capable of simultaneously processing data from two MEMS microphones. This device was designed and tested on prototype boards devised at the beginning of the development process. Unfortunately, due to a delay caused by the supplier, the final PCB board, which combined individual boards and improved their functionality, was not tested. Therefore, the functionality was tested only on prototype boards with a similar design to the final PCB. As per the requirements, the hardware allows MEMS microphones through the MCU, connects the board to a remote server via Wi-Fi, and locates the unit using GPS, including the ability to record an accurate timestamp.

In the last chapter, the firmware was developed, which served as an essential part of the system's functionality. According to the requirements, it should be able to detect acoustic impulses and subsequently transmit the recorded data to a server. This data also include information about the time difference between the microphones for determining the angle of arrival of acoustic signal according to AoA. Although the system was not tested on gunshots, it was tested on impulse signals (clapping, hitting an object, popping a balloon) and accurately detected acoustic peaks.

Thus, all the parts of the assignment were fulfilled. A system that detects impulsive acoustic events was designed and implemented, incorporating prescribed components and functionalities.

References

[1] Maher, Robert C.. Acoustical Characterization of Gunshots. In: *2007 IEEE Workshop on Signal Processing Applications for Public Security and Forensics*. 2007. pp. 1-5. Available from DOI 10.1109/IEEECONF12259.2007.4218954.

[2] Akman, Çağlar, Tolga Sönmez, Özgür Özuğur, Abdil Burak Başli, and M. Kemal Leblebicioğlu. Sensor fusion, sensitivity analysis and calibration in shooter localization systems. *Sensors and Actuators A: Physical*. 2018, Vol. 271, pp. 66-75. ISSN 0924-4247. Available from DOI https://doi.org/10.1016/j.sna.2017.12.042. Available from `https://www.sciencedirect.com/science/article/pii/S0924424716306458`.

[3] Maher, Robert C.. Modeling and Signal Processing of Acoustic Gunshot Recordings. In: *2006 IEEE 12th Digital Signal Processing Workshop 4th IEEE Signal Processing Education Workshop*. 2006. pp. 257-261. Available from DOI 10.1109/DSPWS.2006.265386.

[4] Svatos, Jakub, and Jan Holub. Impulse Acoustic Event Detection, Classification, and Localization System. *IEEE Transactions on Instrumentation and Measurement*. 2023, Vol. 72, pp. 1-15. Available from DOI 10.1109/TIM.2023.3252631.

[5] El Bouny, Lahcen, Mohammed Khalil, and Abdellah Adib. A Wavelet Denoising and Teager Energy Operator-Based Method for Automatic QRS Complex Detection in ECG Signal. *Circuits, Systems, and Signal Processing*. Oct, 2020, Vol. 39, No. 10, pp. 4943-4979. ISSN 1531-5878. Available from DOI 10.1007/s00034-020-01397-8. Available from `https://doi.org/10.1007/s00034-020-01397-8`.

[6] Chacon-Rodriguez, Alfonso, and Pedro Julian. Evaluation of gunshot detection algorithms. In: *2008 Argentine School of Micro-Nanoelectronics, Technology and Applications*. 2008. pp. 49-54.

[7] Parsayan, Ali, and Seyed Mohammad Ahadi. Real Time High Accuracy 3-D PHAT-Based Sound Source Localization Using a Simple 4-Microphone Arrangement. *IEEE Systems Journal*. 2012, Vol. 6, No. 3, pp. 455-468. Available from DOI 10.1109/JSYST.2011.2176766.

[8] U-BLOX. *Product Summary - NEO/LEA-M8T series*. Available from `https://content.u-blox.com/sites/default/files/products/documents/NEO-LEA-M8T_ProductSummary_%28UBX-16000801%29.pdf`.

[9] STMicroelectronics. *MEMS audio sensor: digital microphone with multiple performance modes*. Available from `http://www.st.com/en/product/mp23db01hp`.

[10] STMicroelectronics. *Interfacing PDM digital microphones using STM32 MCUs and MPUs*. Available from `https://www.st.com/resource/en/application_note/an5027-interfacing-pdm-digital-microphones-using-stm32-mcus-and-mpus-stmicroelectronics.pdf`. Application note.

[11] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 3: CSMA/CD Access Method and Physical Layer Specifications Amendment 3: Data Terminal Equipment (DTE) Power via the Media Dependent Interface (MDI) Enhancements. *IEEE Std 802.3at-2009 (Amendment to IEEE Std 802.3-2008).* 2009, pp. 1-137. Available from DOI 10.1109/IEEESTD.2009.5306743.

[12] Instrument, Texas. *TPS2379 IEEE 802.3at PoE High-Power PD Interface With External Gate Driver*. Available from `https://www.ti.com/lit/ds/symlink/tps2379.pdf`. Datasheet.

[13] STMicroelectronics. *STM32Cube PDM2PCM software library for the STM32F4/F7/H7 Series*. Available from `https://www.st.com/resource/en/user_manual/dm00482421-stm32cube-pdm2pcm-software-library-for-the-stm32f4f7h7-series-stmicroelectronics.pdf`. User manual.

[14] NXP Semiconductors. *UM11732 I2S bus specification*. Available from `https://www.nxp.com/docs/en/user-manual/UM11732.pdf`. User manual.

[15] U-BLOX. *u-blox 8 / u-blox M8 Receiver description*. Available from `https://content.u-blox.com/sites/default/files/products/documents/u-blox8-M8_ReceiverDescrProtSpec_UBX-13003221.pdf`.

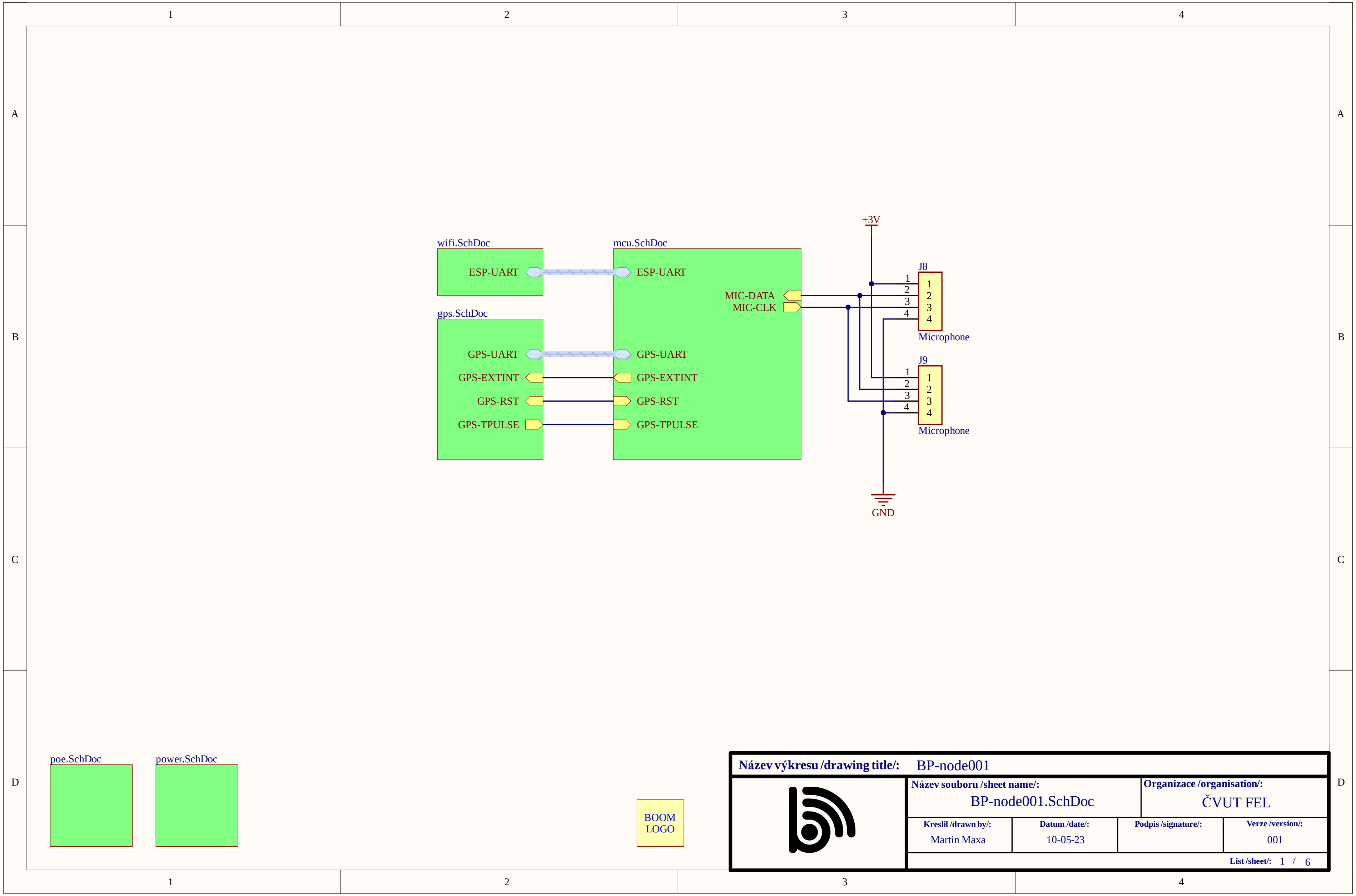
[16] Bies, Lammert. *Hayes modem AT command set*. Available from `https://www.lammertbies.nl/comm/info/hayes-at-commands`.

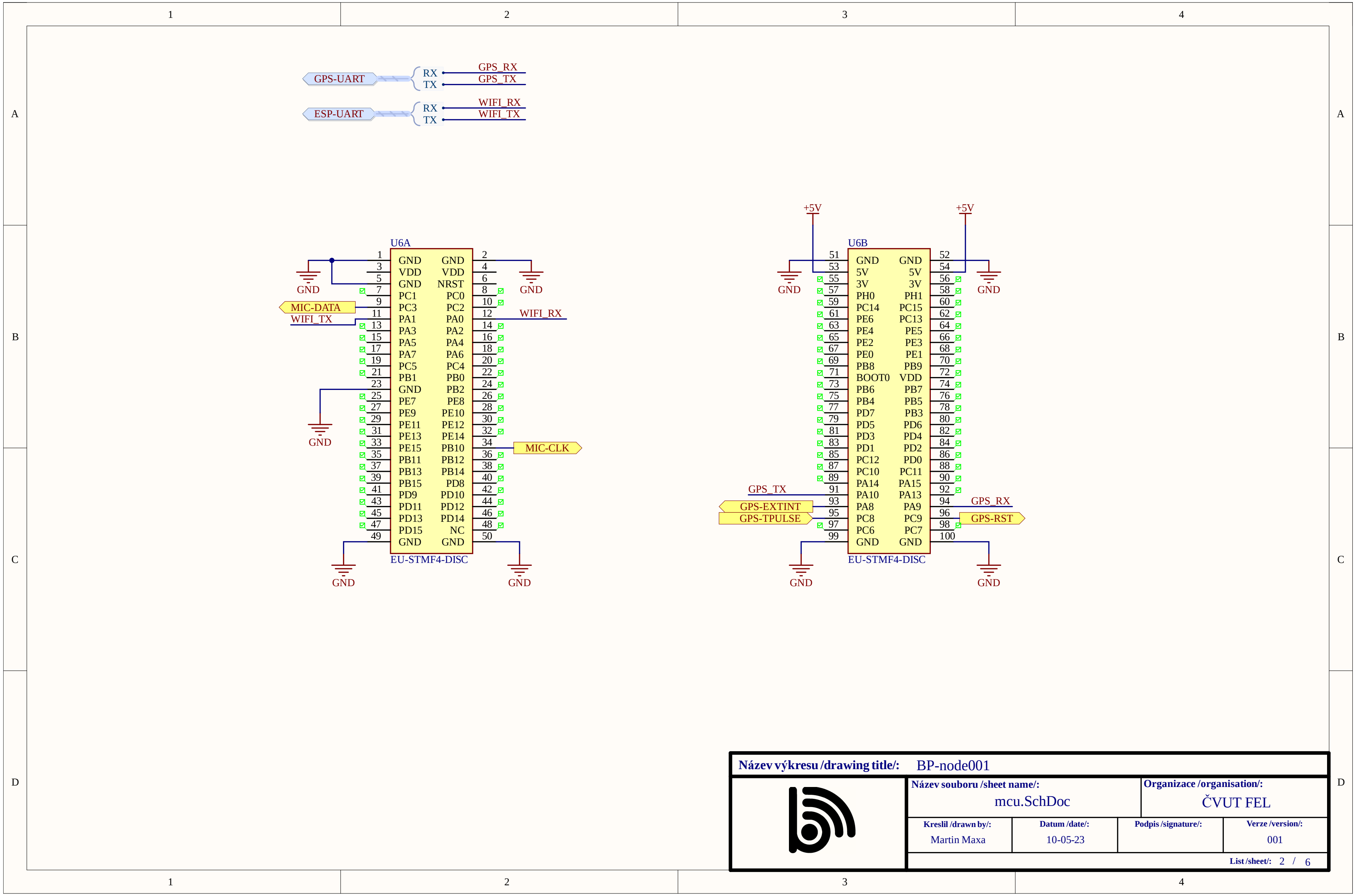
[17] *AT Command Set*. Available from `https://docs.espressif.com/projects/esp-at/en/latest/esp32c3/`.

Appendix A
Glosary

ADC	•	Analog-to-Digital Converter
AoA	•	Angle of Arrival
DMA	•	Direct Memory Access
FEE	•	Faculty of Electrical Engineering
GPS	•	Global Positioning System
HAL	•	Hardware Abstraction Library
I2S	•	Inter-IC Sound
MCU	•	Microcontroller
MEMS	•	Micro Electro Mechanical Systems
PCB	•	Printed Circuit Board
PCM	•	Pulse-Code Modulation
PD	•	Powered Device
PDM	•	Pulse Density Modulation
PoE	•	Power over Ethernet
PSE	•	Power Sourcing Equipment
PWM	•	Pulse Width Modulation
RX	•	Receiver
SCK	•	Serial Clock
SNR	•	Signal-to-Noise Ratio
TCP	•	Transmission Control Protocol
TX	•	Transmitter
UART	•	Universal Asynchronous Receiver-Transmitter
UBX	•	u-blox Extended

Appendix B
Schematic


wifi.SchDoc
ESP-UART
mcu.SchDoc
ESP-UART
MIC-DATA
MIC-CLK
gps.SchDoc
GPS-UART
GPS-EXTINT
GPS-RST
GPS-TPULSE
GPS-UART
GPS-EXTINT
GPS-RST
GPS-TPULSE
+3V
J8
Microphone
J9
Microphone
GND
poe.SchDoc
power.SchDoc
BOOM LOGO
Název výkresu /drawing title/: BP-node001
Název souboru /sheet name/: BP-node001.SchDoc
Organizace /organisation/: ČVUT FEL
Kreslil /drawn by/: Martin Maxa
Datum /date/: 10-05-23
Podpis /signature/:
Verze /version/: 001
List /sheet/: 1 / 6


GPS-UART
RX
TX
GPS_RX
GPS_TX
ESP-UART
RX
TX
WIFI_RX
WIFI_TX
U6A
EU-STMF4-DISC
MIC-DATA
WIFI_TX
WIFI_RX
MIC-CLK
GND
U6B
EU-STMF4-DISC
+5V
GPS_TX
GPS-EXTINT
GPS-TPULSE
GPS_RX
GPS-RST
Název výkresu /drawing title/: BP-node001
Název souboru /sheet name/: mcu.SchDoc
Organizace /organisation/: ČVUT FEL
Kreslil /drawn by/: Martin Maxa
Datum /date/: 10-05-23
Podpis /signature/:
Verze /version/: 001
List /sheet/: 2 / 6

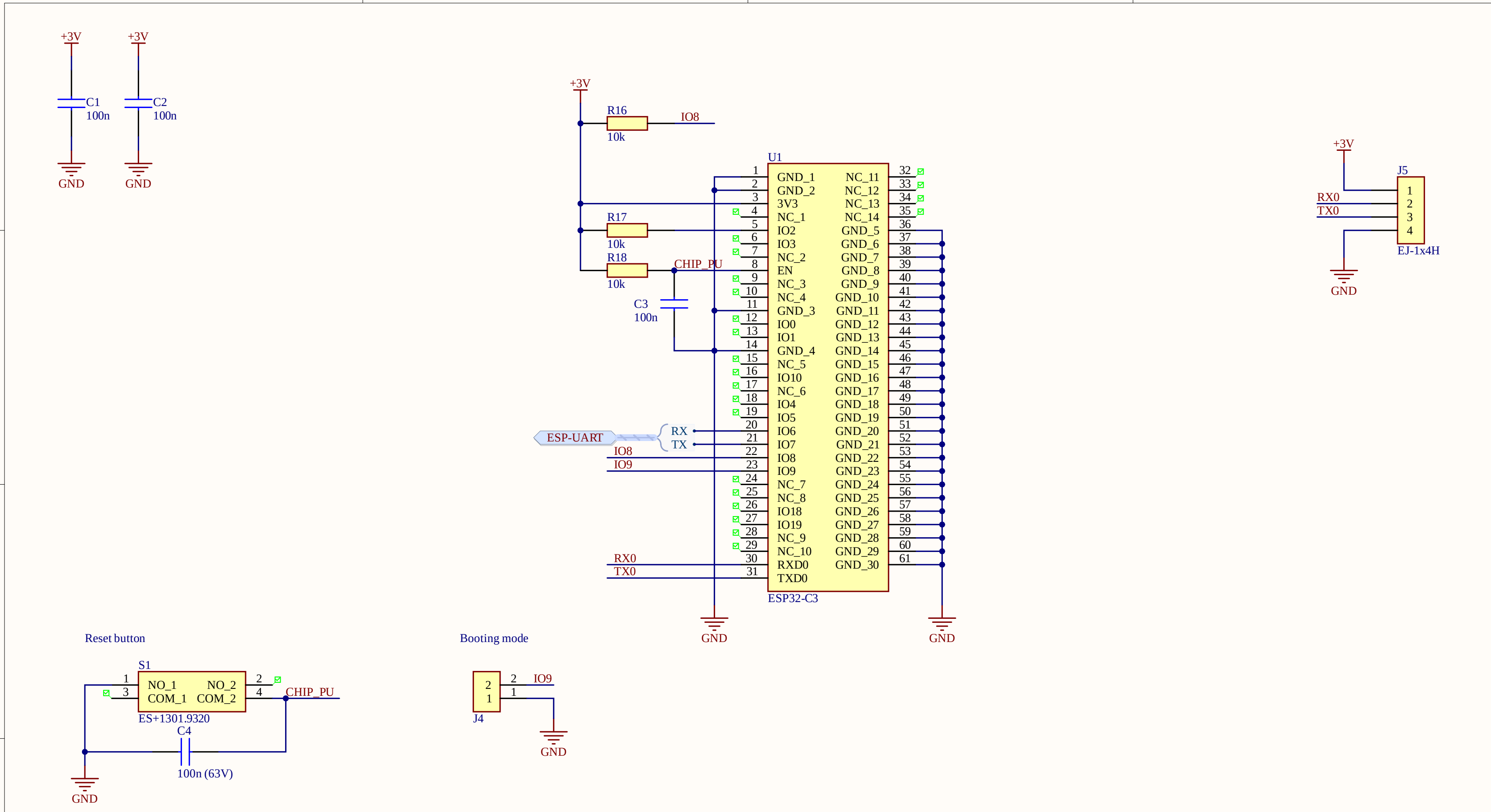
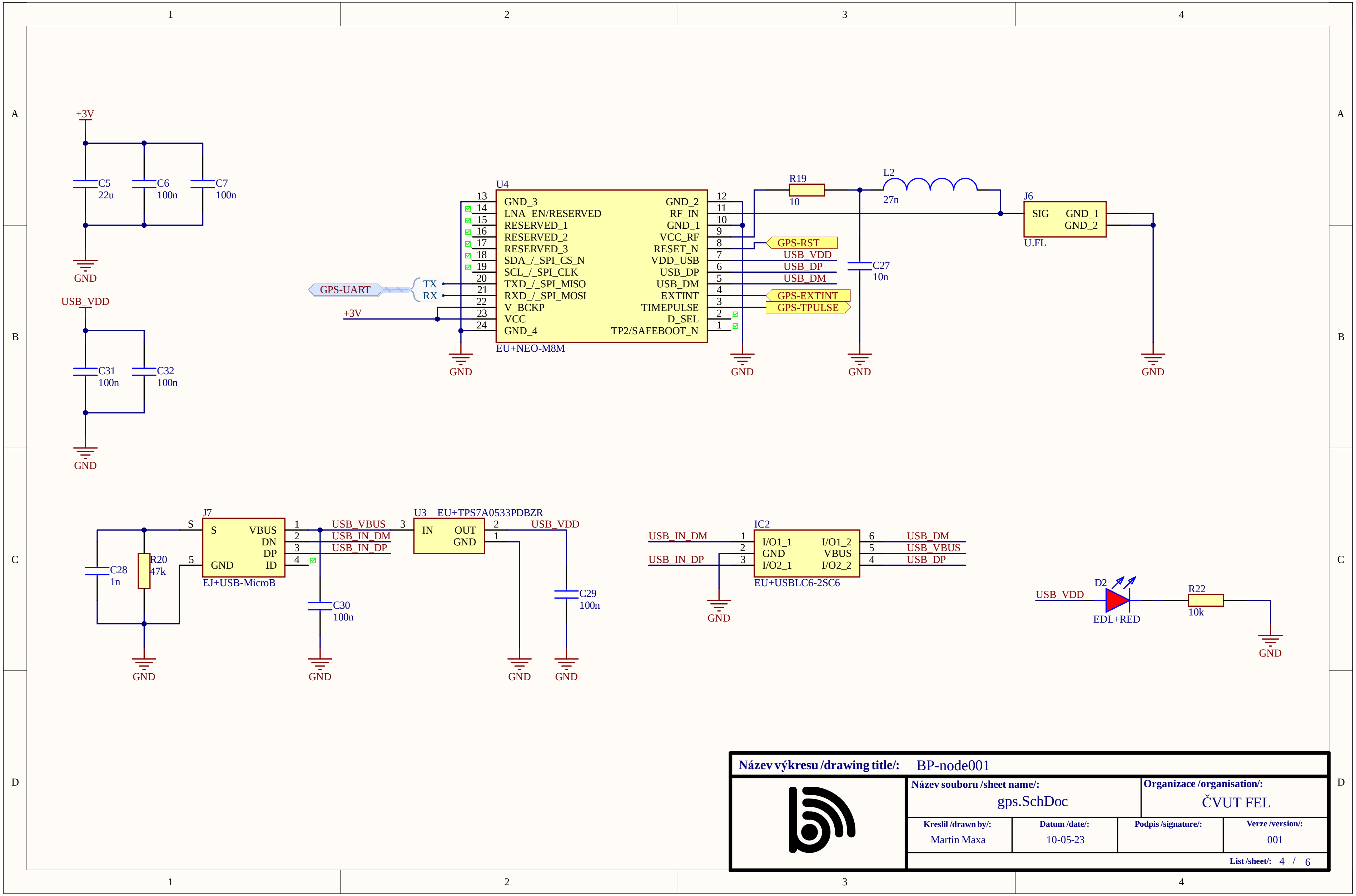
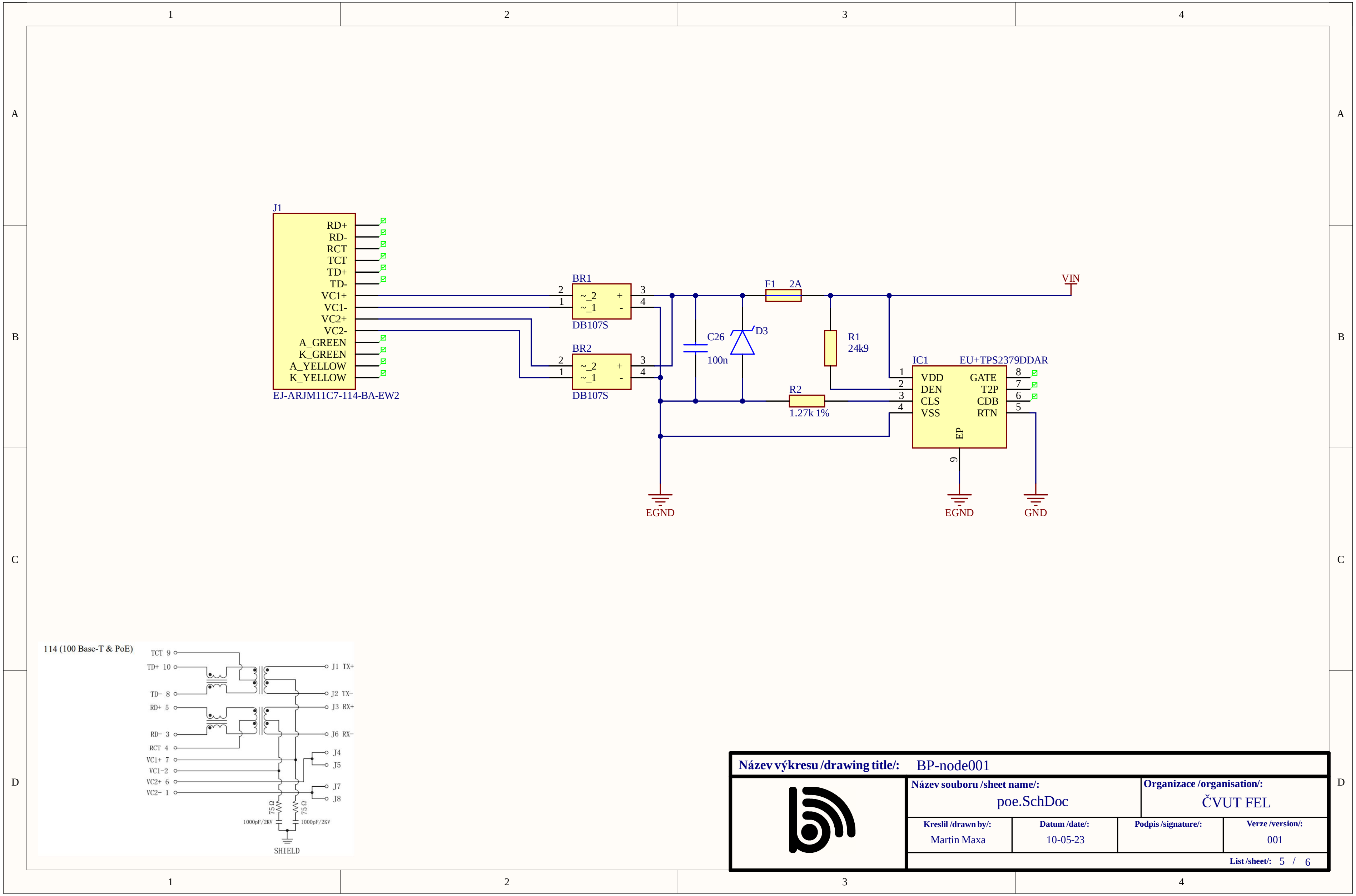


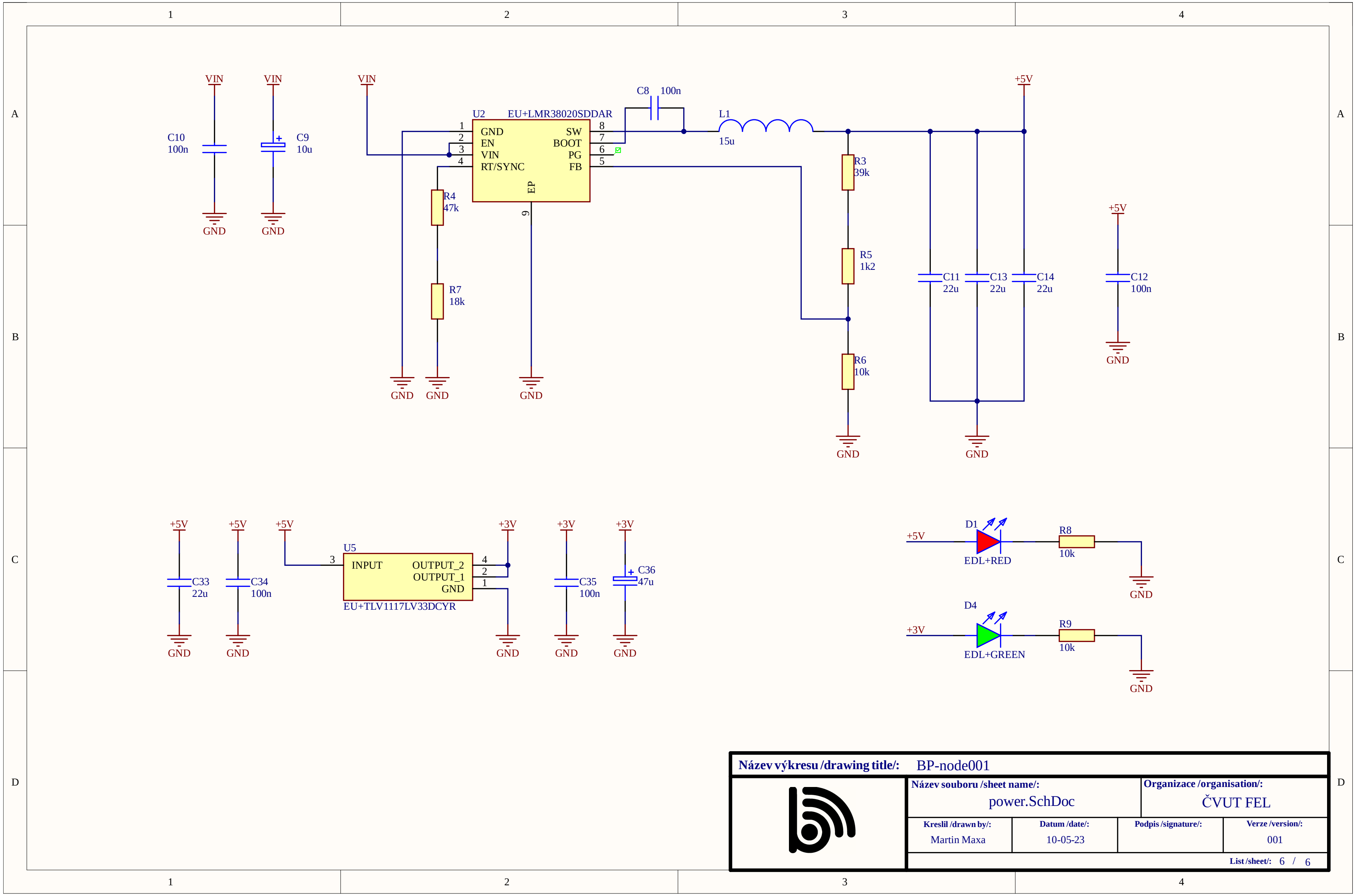
Table 4: Strapping Pins

Booting Mode [1]			
Pin	Default	SPI Boot	Download Boot
GPIO2	N/A	1	1
GPIO8	N/A	Don't care	1
GPIO9	Internal pull-up	1	0

Název výkresu /drawing title/: BP-node001				
Název souboru /sheet name/: wifi.SchDoc		**Organizace /organisation/:** ČVUT FEL		
Kreslil /drawn by/: Martin Maxa	**Datum /date/:** 10-05-23	**Podpis /signature/:**	**Verze /version/:** 001	
List /sheet/: 3 / 6				

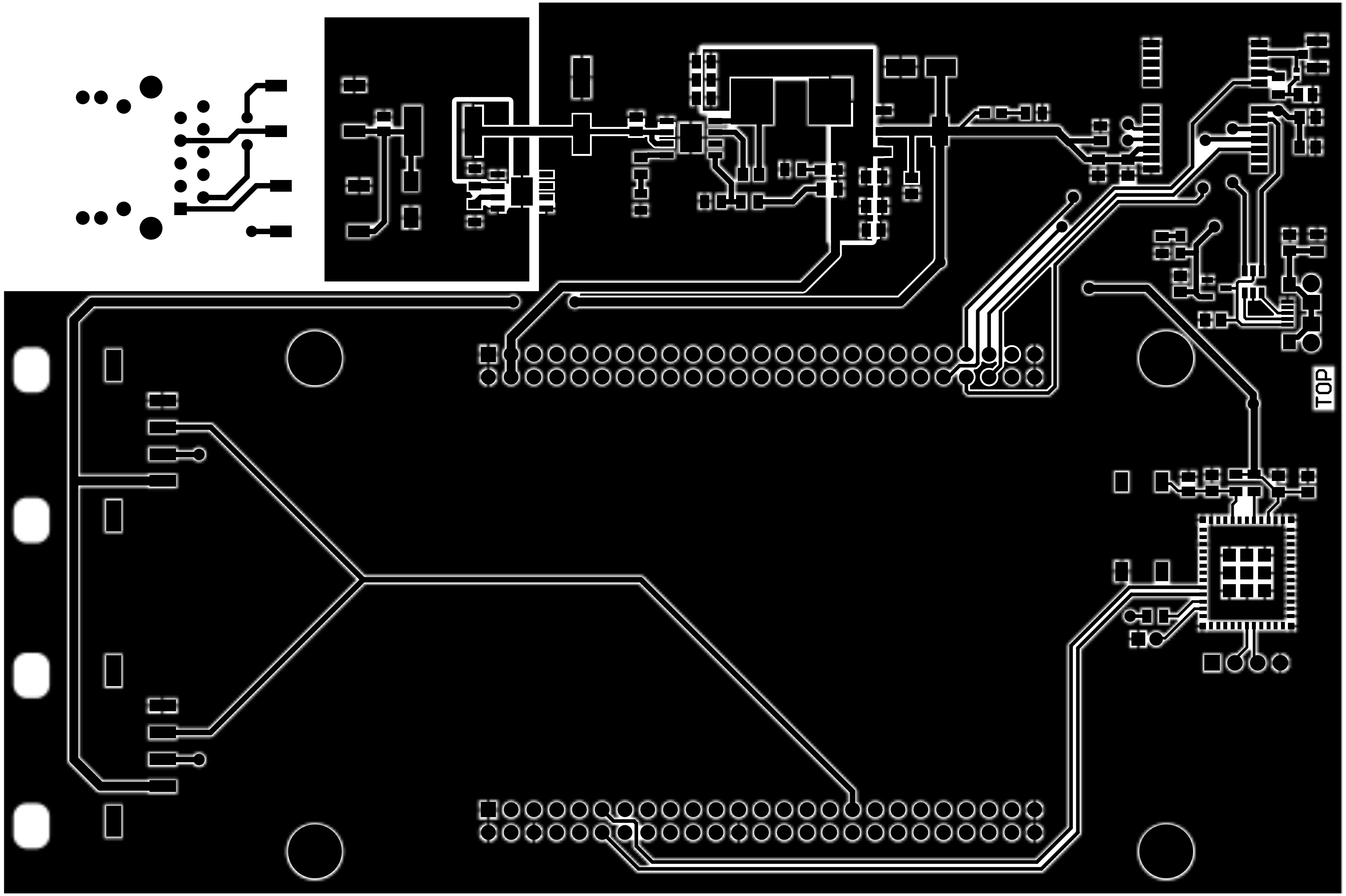

+3V
C5 22u
C6 100n
C7 100n
GND
USB_VDD
C31 100n
C32 100n
GND
U4
EU+NEO-M8M
GND_3
LNA_EN/RESERVED
RESERVED_1
RESERVED_2
RESERVED_3
SDA_/_SPI_CS_N
SCL_/_SPI_CLK
TXD_/_SPI_MISO
RXD_/_SPI_MOSI
V_BCKP
VCC
GND_4
GND_2
RF_IN
GND_1
VCC_RF
RESET_N
VDD_USB
USB_DP
USB_DM
EXTINT
TIMEPULSE
D_SEL
TP2/SAFEBOOT_N
GPS-UART
TX
RX
+3V
GPS-RST
USB_VDD
USB_DP
USB_DM
GPS-EXTINT
GPS-TPULSE
R19 10
L2 27n
C27 10n
J6
SIG
GND_1
GND_2
U.FL
GND
J7
EJ+USB-MicroB
S
VBUS
DN
DP
GND
ID
C28 1n
R20 47k
USB_VBUS
USB_IN_DM
USB_IN_DP
C30 100n
U3 EU+TPS7A0533PDBZR
IN
OUT
GND
USB_VDD
C29 100n
IC2
EU+USBLC6-2SC6
USB_IN_DM
USB_IN_DP
I/O1_1
GND
I/O2_1
I/O1_2
VBUS
I/O2_2
USB_DM
USB_VBUS
USB_DP
USB_VDD
D2
EDL+RED
R22 10k
GND
Název výkresu /drawing title/: BP-node001
Název souboru /sheet name/: gps.SchDoc
Organizace /organisation/: ČVUT FEL
Kreslil /drawn by/: Martin Maxa
Datum /date/: 10-05-23
Podpis /signature/:
Verze /version/: 001
List /sheet/: 4 / 6


J1
RD+
RD-
RCT
TCT
TD+
TD-
VC1+
VC1-
VC2+
VC2-
A_GREEN
K_GREEN
A_YELLOW
K_YELLOW
EJ-ARJM11C7-114-BA-EW2
BR1
DB107S
BR2
DB107S
C26
100n
D3
F1 2A
R1
24k9
R2
1.27k 1%
VIN
IC1 EU+TPS2379DDAR
VDD
DEN
CLS
VSS
GATE
T2P
CDB
RTN
EP
EGND
EGND
GND
114 (100 Base-T & PoE)
TCT 9
TD+ 10
TD- 8
RD+ 5
RD- 3
RCT 4
VC1+ 7
VC1- 2
VC2+ 6
VC2- 1
J1 TX+
J2 TX-
J3 RX+
J6 RX-
J4
J5
J7
J8
75 Ω
75 Ω
1000pF/2KV
1000pF/2KV
SHIELD
Název výkresu /drawing title/: BP-node001
Název souboru /sheet name/: poe.SchDoc
Organizace /organisation/: ČVUT FEL
Kreslil /drawn by/: Martin Maxa
Datum /date/: 10-05-23
Podpis /signature/:
Verze /version/: 001
List /sheet/: 5 / 6

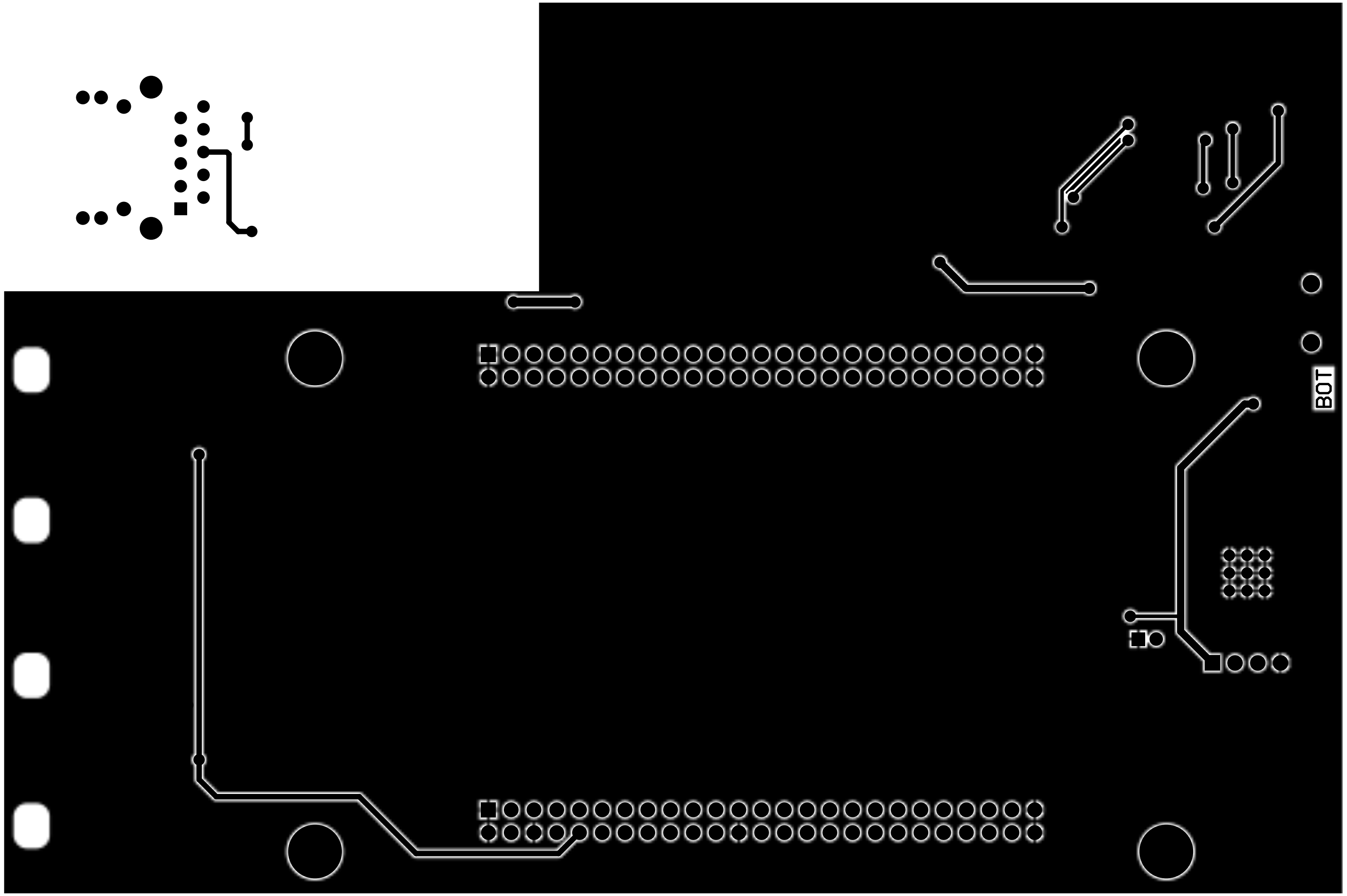


VIN
C10
100n
C9
10u
GND
U2
EU+LMR38020SDDAR
GND
EN
VIN
RT/SYNC
SW
BOOT
PG
FB
EP
C8
100n
L1
15u
+5V
R3
39k
R4
47k
R5
1k2
R7
18k
R6
10k
C11
22u
C13
22u
C14
22u
C12
100n
U5
INPUT
OUTPUT_2
OUTPUT_1
EU+TLV1117LV33DCYR
+3V
C33
22u
C34
100n
C35
100n
C36
47u
D1
EDL+RED
R8
10k
D4
EDL+GREEN
R9
10k
Název výkresu /drawing title/: BP-node001
Název souboru /sheet name/: power.SchDoc
Organizace /organisation/: ČVUT FEL
Kreslil /drawn by/: Martin Maxa
Datum /date/: 10-05-23
Podpis /signature/:
Verze /version/: 001
List /sheet/: 6 / 6

Appendix C
PCB Gerbers

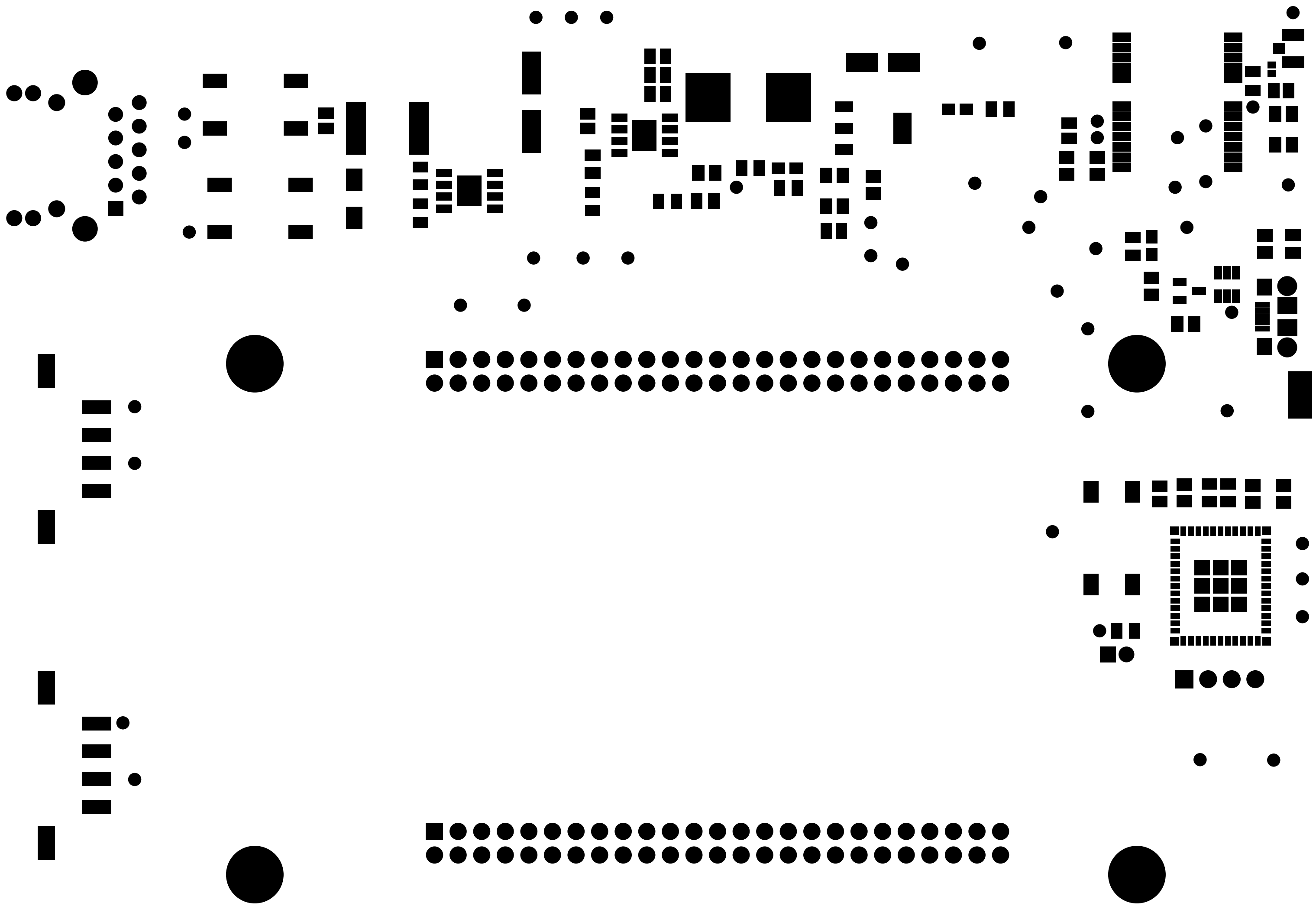


Top Layer

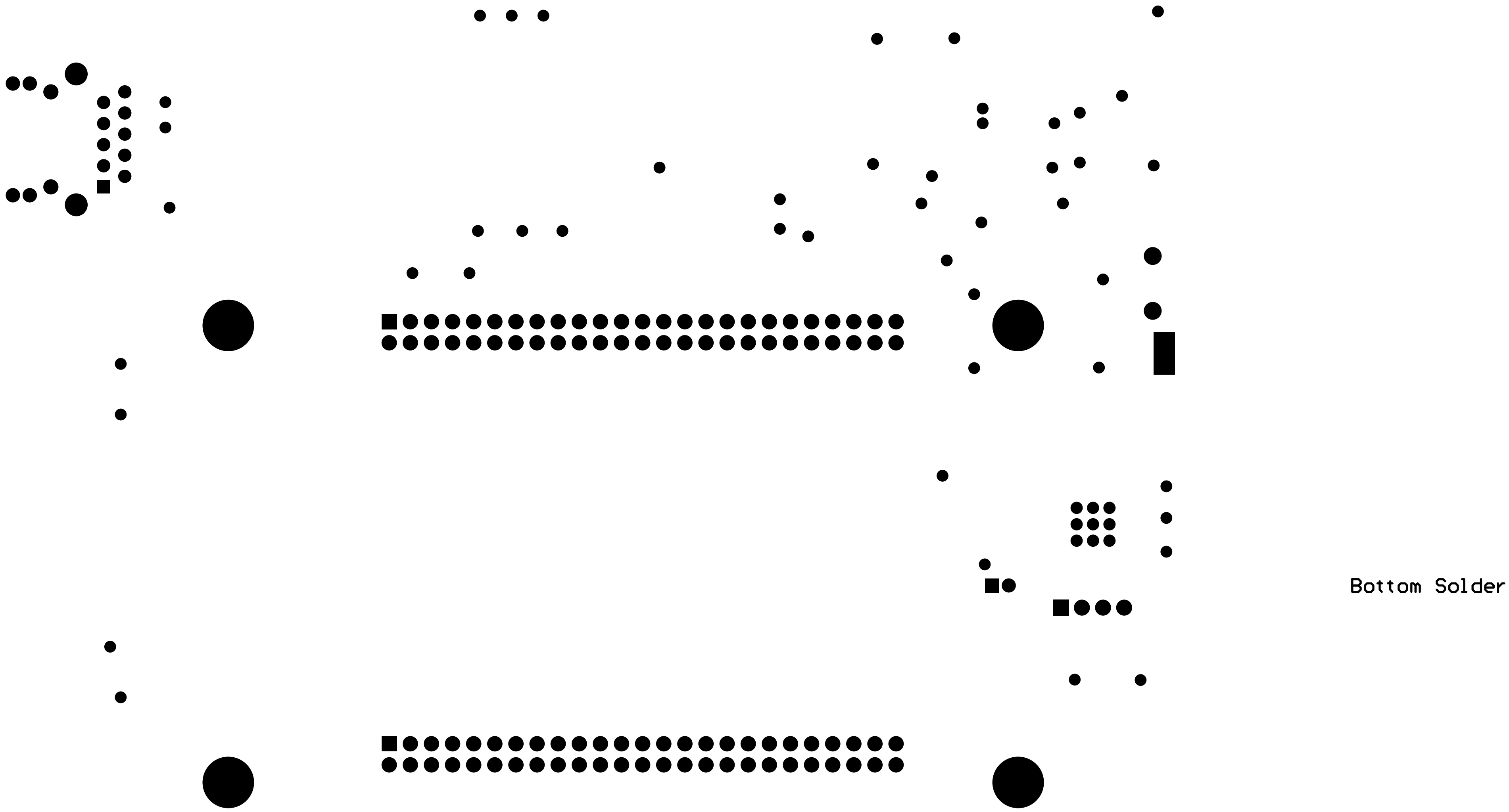


Bottom Layer

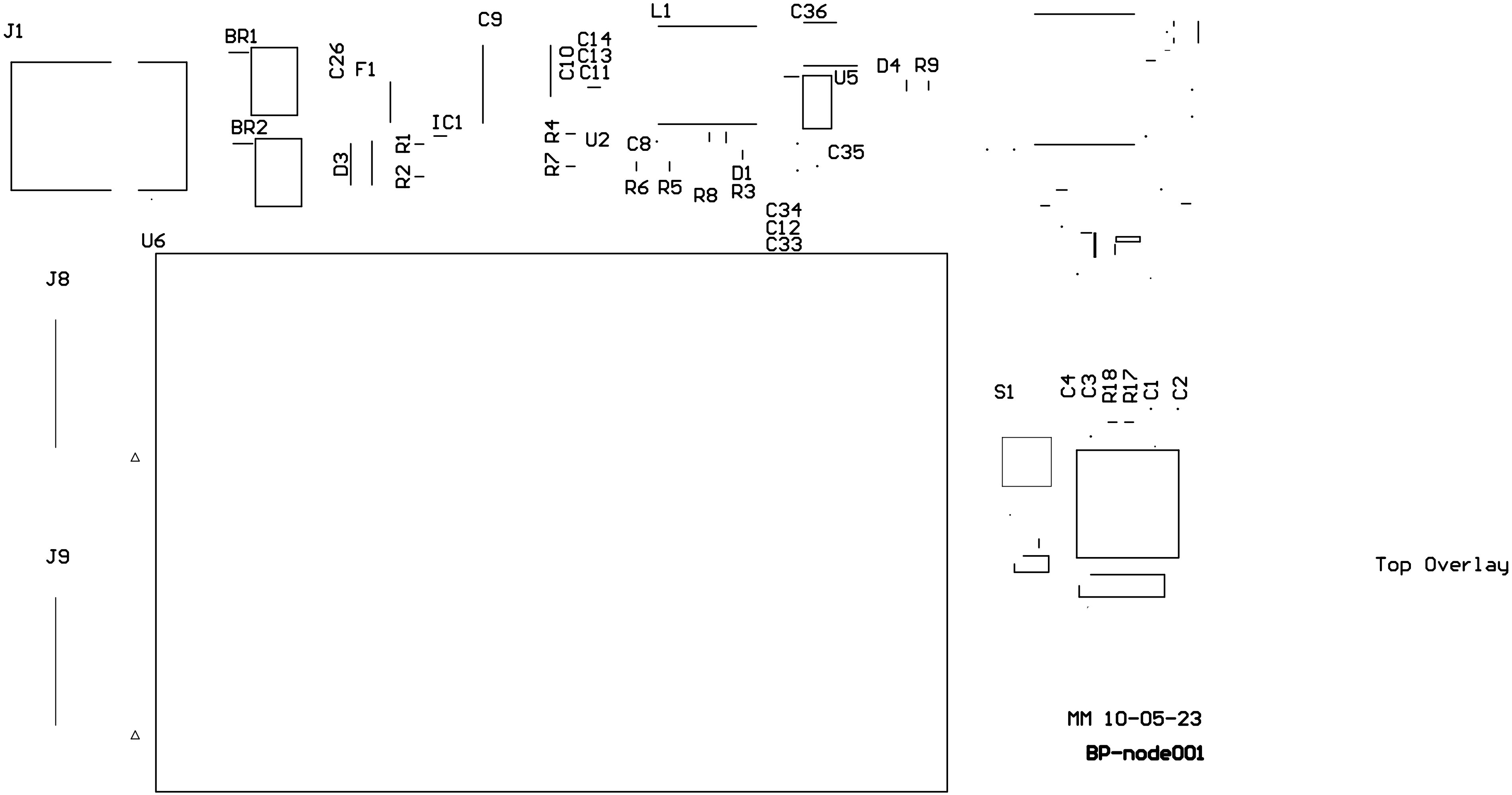
Panel outline



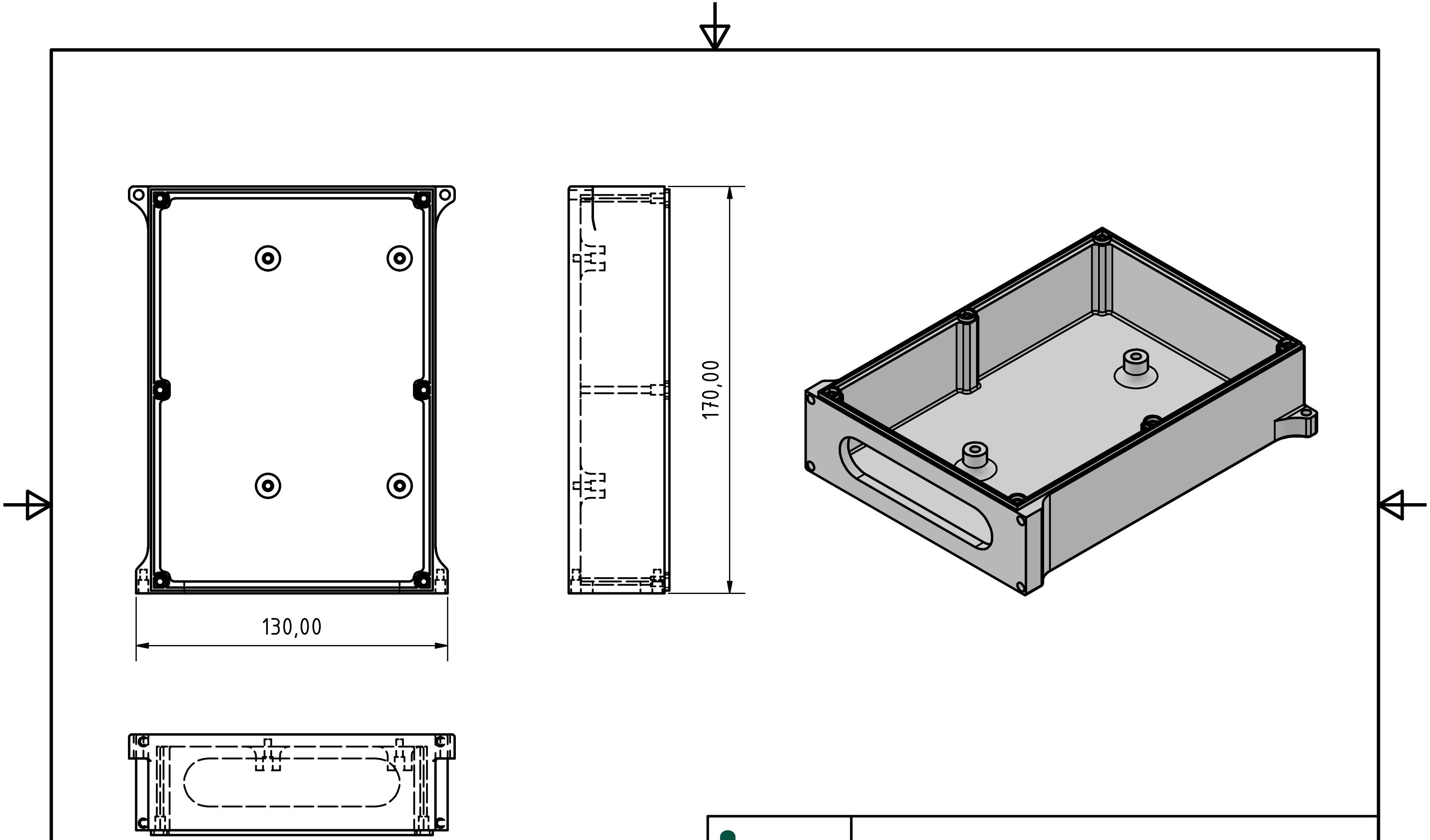
Top Solder


Bottom Solder


Top Pasete

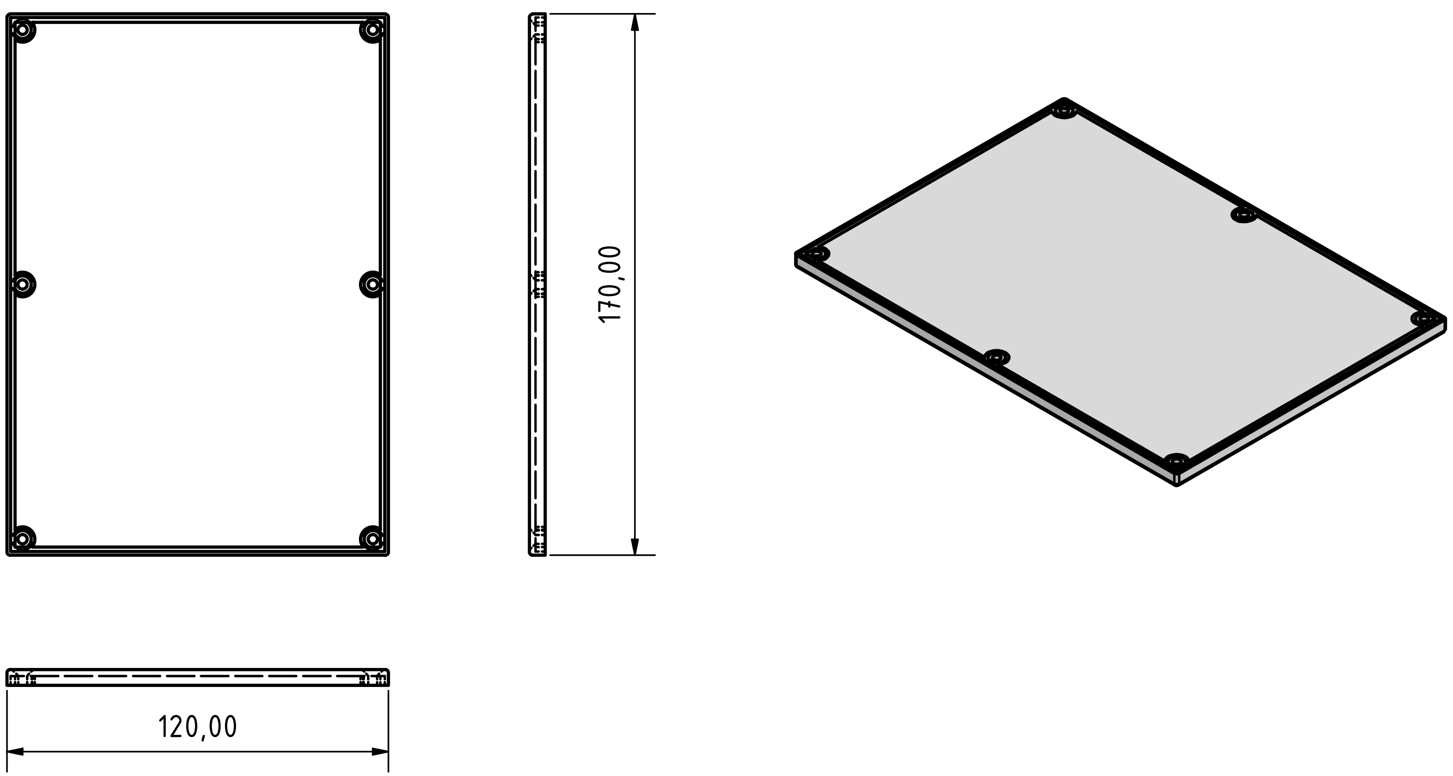

J1
BR1
BR2
C26
F1
D3
R2
R1
IC1
C9
R7
R4
C10
C14
C13
C11
U2
L1
C8
R6
R5
R8
D1
R3
C36
U5
C35
D4
R9
C34
C12
C33
U6
J8
J9
S1
C4
C3
R18
R17
C1
C2
Top Overlay
MM 10-05-23
BP-node001

Appendix D
Enclosure



Drawing title: BP-main_bot

Autor: Martin Maxa	Version: 001
Date: 13.05.2023	Organisation: ČVUT FEL



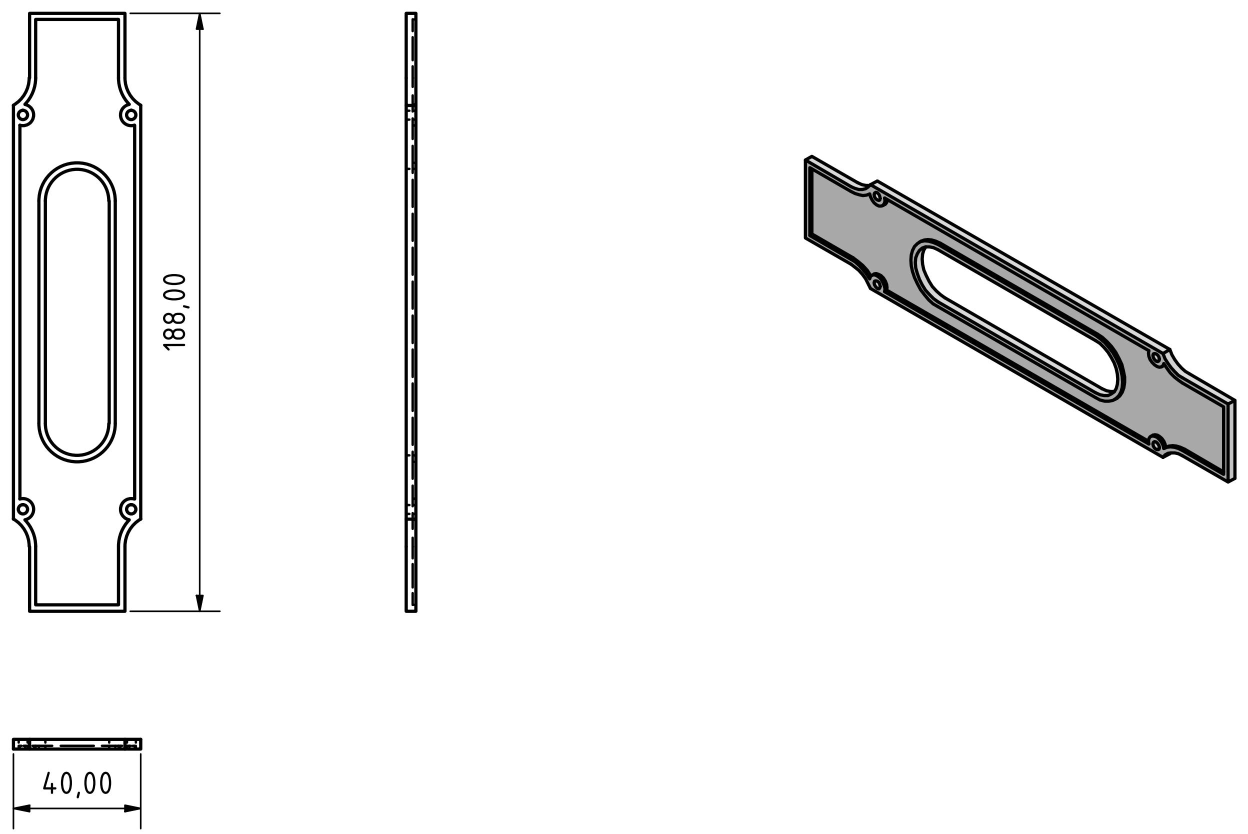
Drawing title: BP-main_top

Autor: Martin Maxa

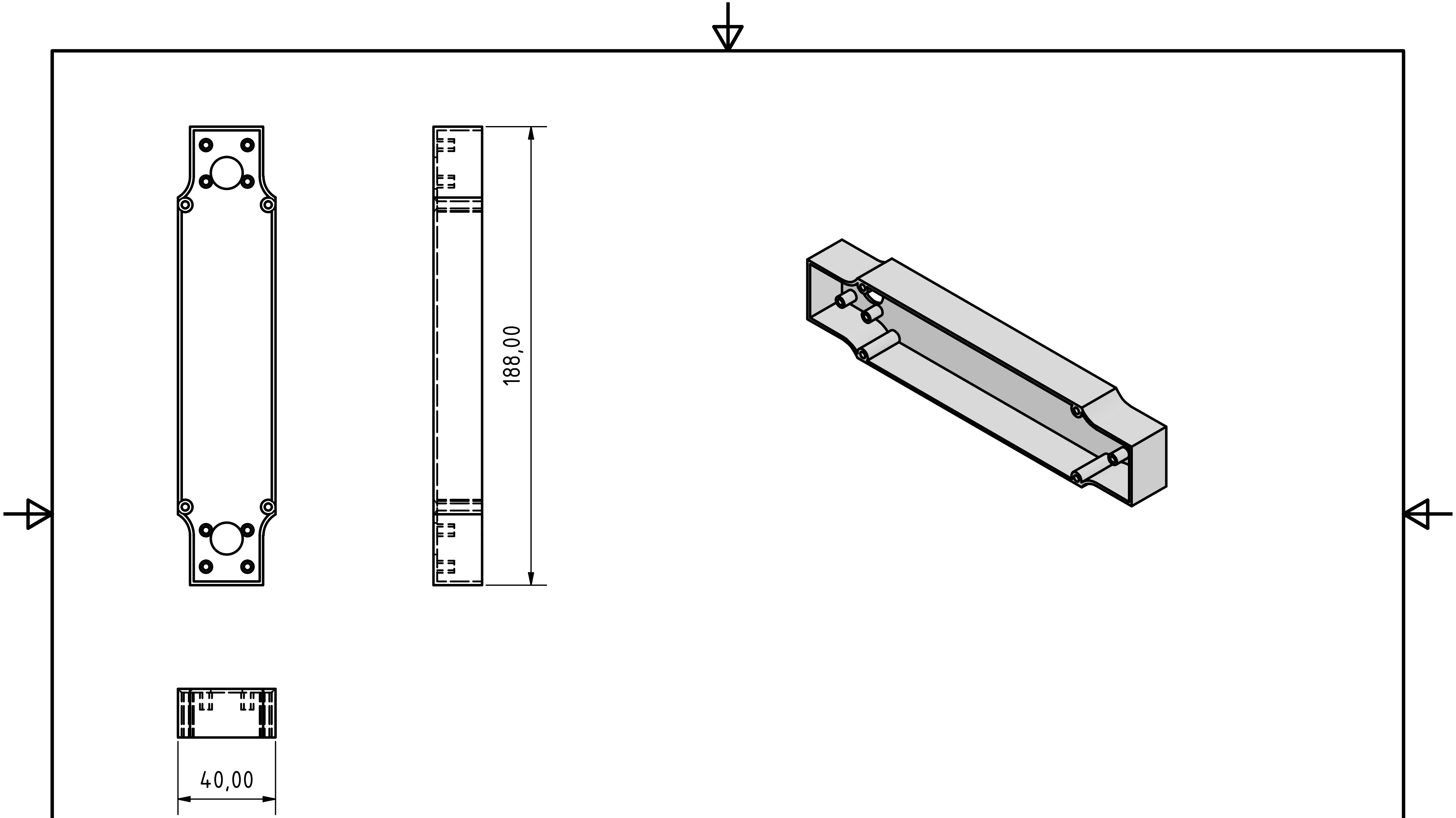
Version: 001

Date: 13.05.2023

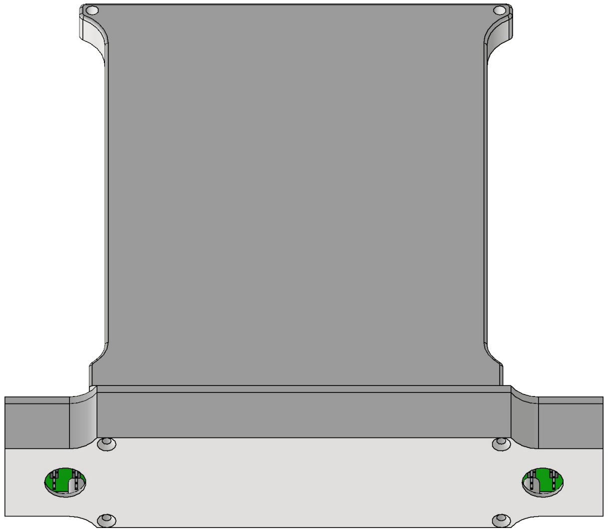
Organisation: ČVUT FEL

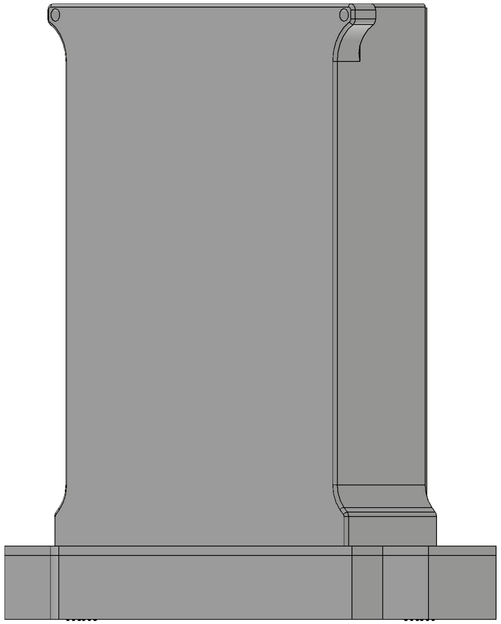


Drawing title: BP-holder	
Autor: Martin Maxa	Version: 001
Date: 13.05.2023	Organisation: ČVUT FEL



	Drawing title: BP-holder-case	
	Autor: Martin Maxa	Version: 001
	Date: 13.05.2023	Organisation: ČVUT FEL





	Drawing title: BP-enclosure_assembly	
	Autor: Martin Maxa	Version: 001
	Date: 13.05.2023	Organisation: ČVUT FEL