Ministry of Higher Education and Scientific Research
University of Carthage
Ecole Polytechnique de Tunisie

وزارة التعليم العالي و البحث العلمي
جامعة قرطاج
المدرسة التونسية للتقنيات

School Project 2.0 Report

EPT'Scope: Multifunction Oscilloscope Based on STM-32

life.augmented

Work prepared by:
Khaireddine BACHHAMBA
Chahine GHANNOUDI
Oussema FARHANI
Aymen LASSOUED
Hedi KSENTINI

Project proposed and supervised by:
Mr. Ahmed BENAHMED (ST)
Mr. Younis LAHBIB (ST)
Mr. Mohamed Fathi KAROUI (EPT)
Mr. Rami BELHADJ KACEM (EPT)
Ms. Dhouha BOUCHAALA (EPT)
Ms. Manel ZOUARI (EPT)

Academic Year : 2024/2025

Rue Elkhawarezmi BP 743 La Marsa 2078
Tel: 71 774 699 Fax: 71 748 843
Site web : www.ept.tn

نهج الخوارزمي ص.ب ٧٤٣ المرسى ٢٠٧٨
الهاتف: 71 774 611 الفاكس: 71 748 843
موقع الواب: www.ept.tn

Contents

List of Figures

List of Tables

Acknowledgements

We would like to express our heartfelt gratitude to all those who contributed to the successful completion of this project, carried out within the framework of the **Projet 2.0** at **École Polytechnique de Tunisie**, in collaboration with **STMicroelectronics**.

We are deeply thankful to **Mr. Youness Lahbib** and **Mr. Ahmed Ben Ahmed**, whose expertise, continuous guidance, and availability were essential throughout the various phases of this project. Their technical insights and strategic orientation greatly enriched the quality of our work, and their dedication to student success has been truly inspiring.

Our sincere thanks also go to our instructors: **Ms. Dhouha Bouchaala**, **Ms. Manel Zouari**, **Mr. Mohamed Fathi Karoui**, and **Mr. Rami Bel Hadj Kacem**. We are grateful for their valuable feedback, encouragement, and constructive critiques, which helped us improve not only the technical depth of our solution but also the rigor and clarity of our presentation. Their unwavering support has played a vital role in our personal and academic growth.

We also wish to thank the entire academic staff of École Polytechnique de Tunisie and the coordination team behind Projet 2.0 for creating an environment that fosters innovation, collaboration, and excellence. This initiative has allowed us to tackle real-world challenges, develop practical skills, and work effectively as a team in a professional setting.

Lastly, we extend our appreciation to our fellow students for their collaboration, exchanges of ideas, and the positive spirit that accompanied us throughout this journey. This experience has been intellectually enriching and personally rewarding, and it marks a significant milestone in our academic path.

GENERAL INTRODUCTION

In today's fast-paced world of technological advancement, embedded systems have become indispensable in numerous domains, ranging from consumer electronics and industrial automation to automotive and medical applications. The increasing performance and decreasing cost of microcontrollers have enabled engineers to design compact, efficient, and multifunctional systems that were once only possible with expensive dedicated equipment.

Among the critical tools in electronics and embedded system design is the **oscilloscope**—an instrument used to observe and analyze electrical signals in real time. Traditional oscilloscopes, while powerful, are often bulky and costly. The need for portable, customizable, and low-cost alternatives has led to a growing interest in implementing oscilloscope functionalities using microcontrollers.

This project, carried out within the framework of the **Projet 2.0** initiative at École Polytechnique de Tunisie, in collaboration with **STMicroelectronics**, focuses on the design and implementation of a **digital oscilloscope based on the STM32 microcontroller platform**. The STM32 family of microcontrollers, known for its powerful processing capabilities and rich set of peripherals, provides an ideal foundation for building such a system.

Through this project, we not only explored the theoretical and practical aspects of signal processing, data acquisition, and embedded system design, but also gained hands-on experience in team collaboration, debugging, and system integration. The work involved multiple stages—ranging from understanding the STM32 architecture and designing the signal conditioning circuit, to implementing data transmission protocols and developing a graphical user interface for real-time waveform display.

This experience, enriched by the guidance of our instructors and industry experts from STMicroelectronics, has significantly contributed to our technical growth and engineering maturity. It reflects the importance of bridging academic knowledge with industrial practice, and highlights the potential of modern microcontroller-based systems in educational and professional contexts.

Project Overview

The core objective of this project was to develop a functional oscilloscope using an **STM32** development board, capable of acquiring, processing, and transmitting analog signals to a computer for visualization. This endeavor was undertaken as part of the **Projet 2.0** initiative at Ecole Polytechnique de Tunisie, in collaboration with **STMicroelectronics**.

To achieve this goal, the development process was divided into several key phases:

- **Understanding the STM32 board**
  This phase involved exploring the architecture, capabilities, and development tools associated with the STM32 microcontroller to ensure an efficient and robust implementation.
- **Basic Testing**
  Initial tests were conducted to validate the fundamental functionalities of the board and peripherals.
- **Signal Acquisition**
  The analog signal was sampled using the ADC (Analog-to-Digital Converter) integrated in the STM32 board.
- **Design of the Level Shifting Circuit**
  A conditioning circuit was developed to adapt input signals to levels suitable for STM32 ADC input.
- **DMA Configuration**
  To ensure efficient data transfer from the ADC to memory, DMA (Direct Memory Access) was configured, allowing high-speed signal acquisition without CPU overhead.
- **Digital Filtering**
  Basic digital filters were implemented to reduce noise and improve the quality of the visualized signal.
- **Data Transmission to PC**
  Communication protocols were established to transfer the processed signal data to a host computer for visualization.

- **Oscilloscope Visualization Software**
  A custom software interface was developed to graphically display the signal in real-time on a computer.

- **3D Implementation**
  A 3D model of the oscilloscope system was created to aid in future integration and enclosure design.

- **Testing and Calibration**
  The final step consisted of extensive testing and calibration to ensure reliable operation and signal accuracy.

This project not only deepened our understanding of embedded systems, signal processing, and hardware-software integration, but also provided a hands-on opportunity to apply theoretical knowledge in a practical and collaborative engineering context.

System Architecture Commentary

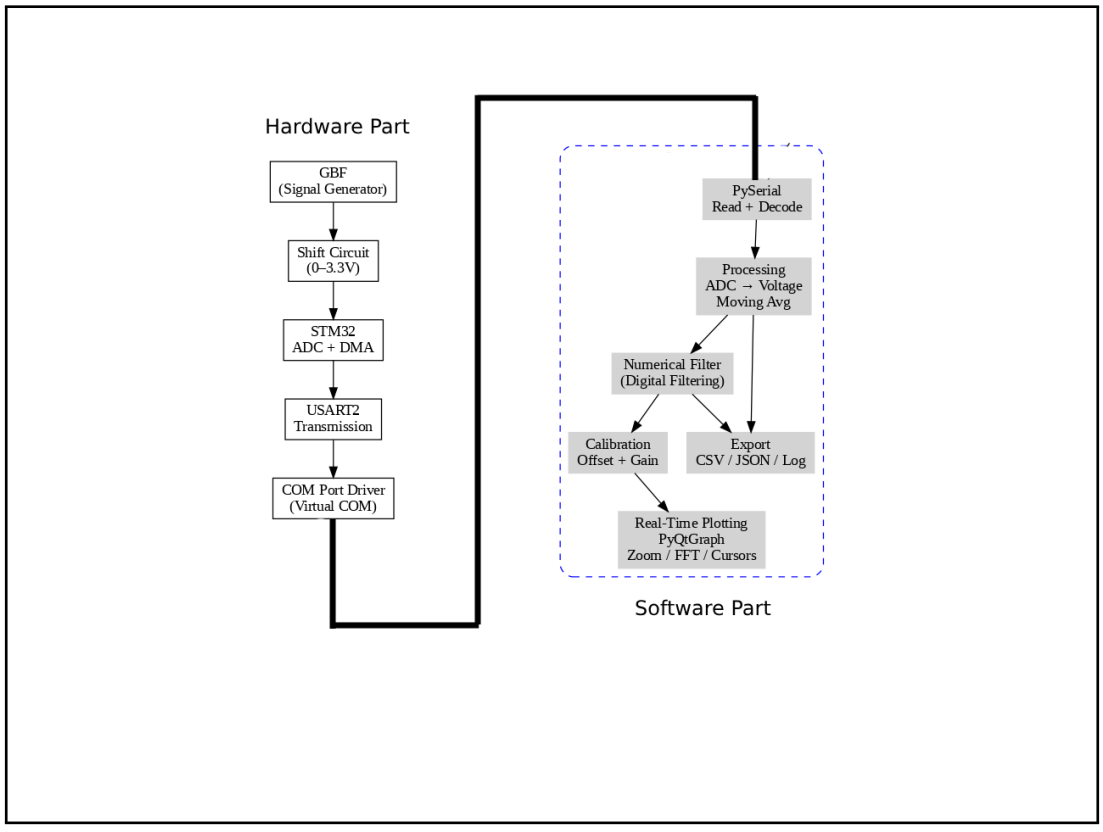


Figure 1: General architecture of the STM32 oscilloscope system

Figure 1 illustrates the complete signal acquisition and visualization pipeline of our oscilloscope system. It is divided into two main domains: the hardware part and the software part, connected via serial communication (USART2 over USB).

a) Hardware Part:

- **Signal Source (GBF)**: The signal generator (GBF) produces an analog waveform used to test the acquisition chain. This signal could be sinusoidal, triangular, or custom-shaped.
- **Shift Circuit (0–3.3V)**: Because the STM32 ADC accepts inputs only between 0V and 3.3V, a level-shifting and conditioning circuit is used to bring external signals into a safe and measurable voltage range.
- **STM32 Microcontroller (ADC + DMA)**: The conditioned analog signal is sampled using the ADC. The DMA peripheral automatically transfers the sampled values to memory, minimizing CPU usage and supporting high-speed acquisition.
- **USART2 Transmission**: The STM32 continuously transmits the acquired digital values over the UART interface using USART2, ensuring minimal latency in data transfer.
- **COM Port (Virtual Serial Driver)**: On the PC, the USB connection is handled by a virtual COM port driver that presents the serial stream to user-level applications in a standard and portable format.

b) Software Part:

- **PySerial – Read and Decode**: A Python script reads the incoming serial data stream and decodes the raw bytes into integer values representing the ADC readings.
- **Processing – ADC to Voltage**: The ADC values are converted to voltage levels using a linear mapping based on the reference voltage (3.3V) and ADC resolution (12-bit $\rightarrow$ 4096 steps).
- **Digital Filtering (Optional)**: A numerical filter, such as a low-pass or moving average filter, can be applied to improve signal clarity by reducing noise.
- **Calibration**: The system supports calibration by applying user-defined offset and gain adjustments. This ensures accurate voltage representation even with non-ideal components or imperfect hardware.
- **Visualization – Real-Time Plotting**: The PyQtGraph library is used to render the signal in real-time with interactive features like zooming, FFT analysis, and cursor-based inspection.

- **Exporting Results**: Users can save the acquired data to files in multiple formats (CSV, JSON, or plain log) for further analysis or archival.

c) Architecture Insights:

This modular architecture offers a clear separation of concerns:

- The **hardware part** handles real-time signal acquisition at a low level, ensuring performance and timing accuracy.
- The **software part** handles interpretation, visualization, and user interaction, offering flexibility and extensibility (e.g., additional analysis tools, GUI improvements).

The architecture is scalable and portable. By simply updating the software filters or extending the GUI features, the system can evolve into a full-featured oscilloscope with minimal hardware changes.

Chapitre

1 STM32 Board Understanding

1.1 Understanding the STM32 NUCLEO-H533RE Board

The STM32 NUCLEO-H533RE is a powerful and versatile development board based on the Arm Cortex-M33 core. It offers a wide range of peripherals and features, making it suitable for signal acquisition and embedded development. Before implementing any functionality, we analyzed its architecture, pin configuration, and onboard peripherals.

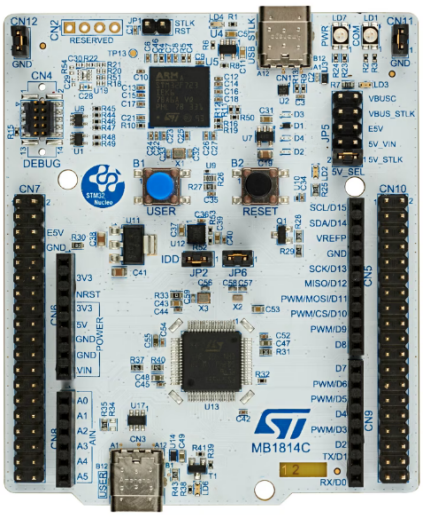


Figure 1.1: STM32 NUCLEO-H533RE Board (Front View)

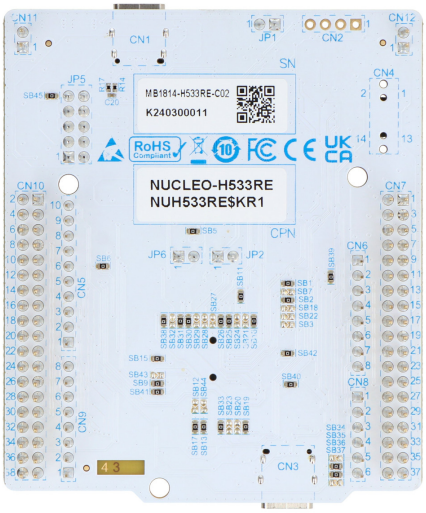


Figure 1.2: STM32 NUCLEO-H533RE Board (Back View)

1.2 Main Technical Specifications

The oscilloscope system designed during this project is built around the STM32 microcontroller and provides a functional and cost-effective solution for real-time signal visualization. While being compact and simple, the system maintains key features expected of a basic digital oscilloscope. Its hardware and software were carefully configured to balance performance, ease of use, and extensibility.

Below are the main technical specifications of the developed prototype:

Table 1.1: Technical specifications of the oscilloscope system

Parameter	Value / Description
ADC resolution	12-bit resolution enabling accurate analog-to-digital conversion of the input signal.
Input voltage range	-3.3V to +3.3V (with conditioning circuit for protection and shifting).
Number of channels	1 channel (extendable to more channels with hardware and software scaling).
Analog bandwidth	DC to 1 MHz, suitable for slow to moderately fast analog signals.
Temporal resolution	Adjustable: from microseconds to seconds per division depending on sampling rate.
Acquisition modes	Continuous (real-time streaming) and triggered (event-based start of acquisition).
Communication interface	UART and USB support for data transmission to the PC.

The system's specifications meet the typical requirements for an educational or diagnostic oscilloscope. Its modular design allows for future upgrades such as multi-channel acquisition, higher sampling frequencies, or additional interface protocols (e.g., SPI or Wi-Fi).

Chapitre

2 EPT'Scope Hardware Development

2.1 Introduction

This chapter outlines the key steps undertaken during the development of our STM32-based oscilloscope. We began by familiarizing ourselves with the STM32 NUCLEO-H533RE development board, followed by the execution of basic functionality tests to validate the essential modules required for our project.

2.2 Basic Tests and Peripheral Validation

Once familiarized with the STM32 NUCLEO-H533RE board, we performed a set of basic tests to validate its core functionalities and ensure proper configuration of essential peripherals. These tests focused on the interaction between two cards: one for signal acquisition and transmission, and the other for signal processing and display, ensuring the board's readiness for advanced tasks like DMA configuration and real-time data processing.

2.2.1 GPIO: LED Blinking Test

As a first test, we configured a digital input and output pin to control an onboard LED using a push-button. This exercise verified the development environment setup, pin initialization, and control logic for both input and output operations. The test procedure included:

- Configuring one GPIO pin as an input for the push-button and another as an output for the LED using STM32CubeMX.
- Writing a firmware routine to toggle the LED state each time the push-button is pressed, generating pulses.
- Verifying the LED's response through visual inspection and, optionally, measuring voltage levels at the output pin with a multimeter.

Expected Outcome: The LED toggles consistently with each push-button press, confirming proper GPIO configuration and control for input and output operations.

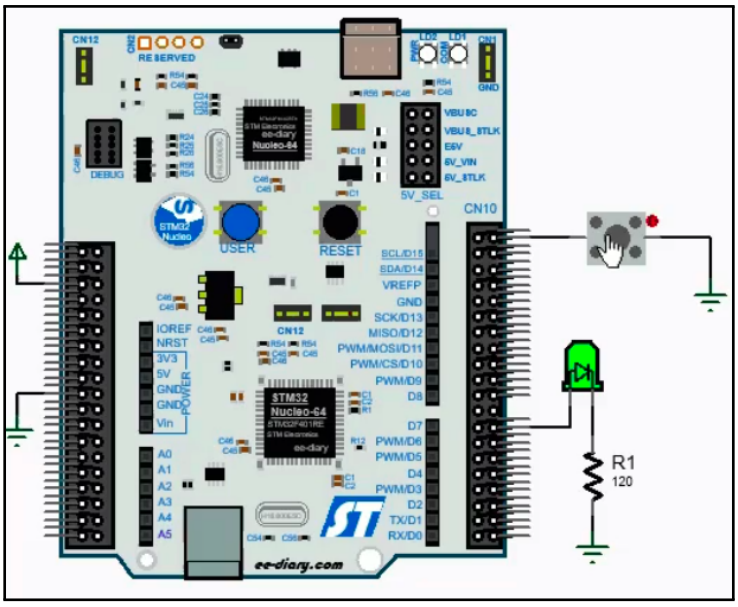


Figure 2.1: GPIO LED and Push-Button Test Setup

2.2.2 USART2: Serial Communication

To validate data transmission, we used the USART2 peripheral to send the string "hello" from the STM32 signal acquisition card to a PC. This functionality is crucial for debugging and future signal transmission between the cards. The test setup included:

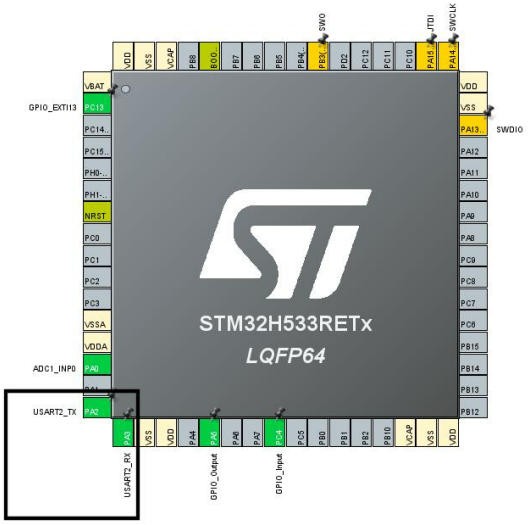


Figure 2.2: Configuration interface

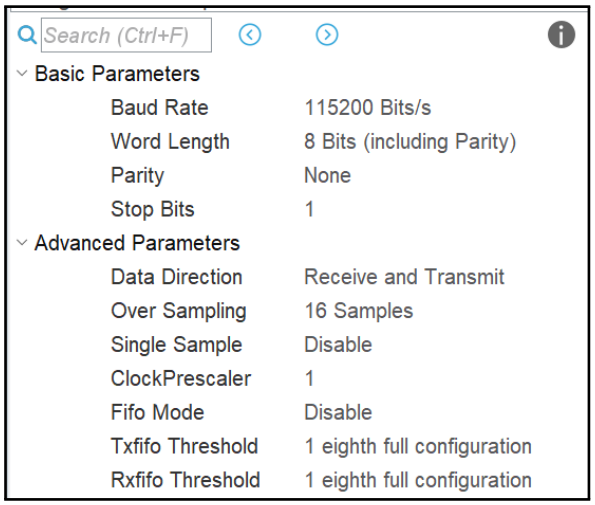


Figure 2.3: STM32CubeMX setup

- Initializing USART2 in asynchronous mode using STM32CubeMX with parameters: baud rate of 115200, 8 data bits, no parity, and 1 stop bit.
- Sending the test string "hello" from the STM32 board to the PC via USB.

- Verifying the output on a terminal emulator (PuTTY) to confirm the string "hello" was displayed correctly.

Expected Outcome: The PC accurately receives and displays the string "hello" without errors, confirming robust USART2 communication.

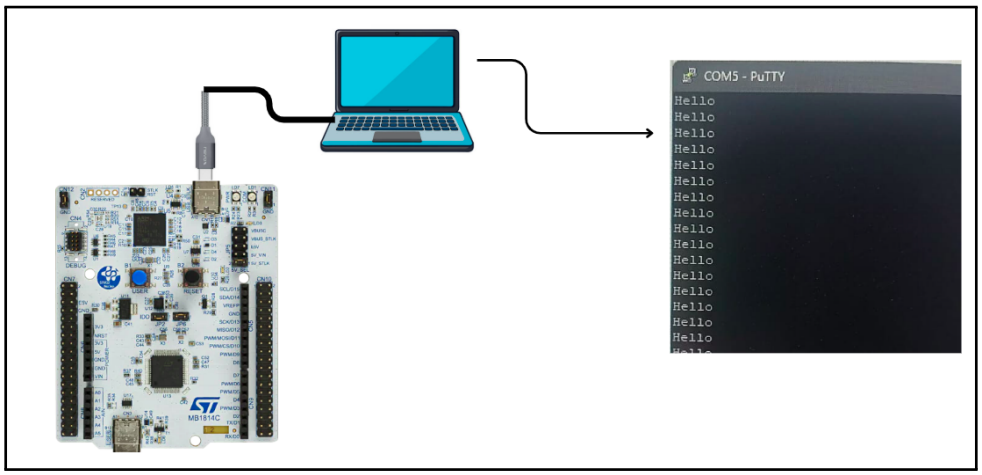


Figure 2.4: USART2 Communication Test Setup

2.3 Signal Acquisition

One of the fundamental components of a digital oscilloscope is the ability to accurately acquire analog signals and convert them into digital data that can be processed and displayed. This task is handled by the **Analog-to-Digital Converter (ADC)** embedded in the STM32 NUCLEO-H533RE microcontroller.

2.3.1 ADC Configuration

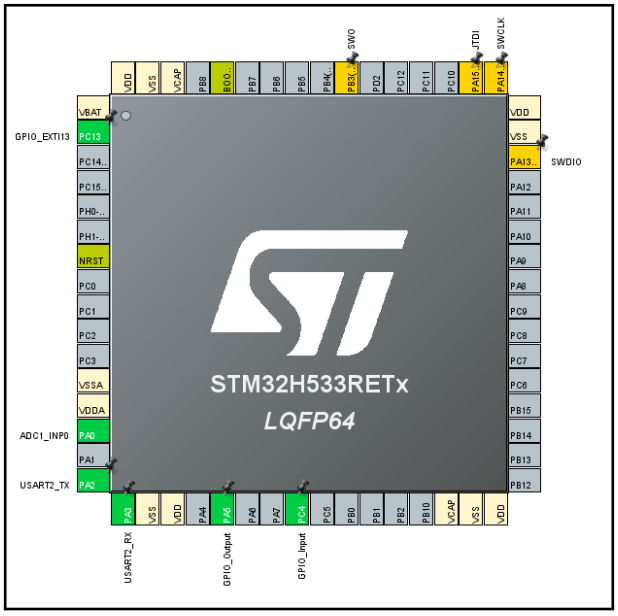


Figure 2.5: ADC Configuration in STM32CubeMX

The STM32 NUCLEO-H533RE board features a high-performance 12-bit ADC capable of precise and efficient sampling. The ADC was configured using STM32CubeMX with the following key parameters:

- **Mode:** Single conversion, triggered by software.
- **Resolution:** 12 bits (4096 levels).
- **Input Channel:** `ADC1_IN0` (connected to the output of the signal conditioning circuit).
- **Continuous Conversion:** Enabled.
- **Data Alignment:** Right-aligned.

These settings offer a balanced trade-off between precision and sampling rate, ideal for capturing analog signals like sine waves or varying sensor outputs.

Below is a snapshot of the embedded code handling ADC acquisition and USART2 transmission. It continuously reads analog values, formats them as strings, and sends them to the PC via USART2:

```
while (1)
{
HAL_ADC_Start(&hadc1);
if (HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY) == HAL_OK)
    {
        uint16_t raw = HAL_ADC_GetValue(&hadc1);
        char msg[20];
        snprintf(msg, sizeof(msg), "ADC: %hu\r\n", raw);
        HAL_USART2_Transmit(&hUSART22, (uint8_t*)msg,
        strlen(msg), HAL_MAX_DELAY);
    }

    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_10);
    HAL_Delay(10); // Small delay to limit USART2 traffic
}
```

Listing 2.1: ADC Reading and USART Transmission

2.3.2 Sampling Frequency and Resolution

The **sampling frequency** determines how often the analog signal is read and converted into a digital value, playing a critical role in the accuracy and fidelity of signal reconstruction. The **resolution** defines how finely the ADC can measure the analog voltage. Key configurations included:

- The ADC was configured to operate at a sampling rate of approximately **100 kHz**, sufficient for observing low- to mid-frequency signals (up to 50 kHz, respecting the Nyquist theorem).
- The sampling rate was controlled via a timer-triggered mechanism, ensuring regular acquisition intervals without CPU overhead.
- A **12-bit resolution** was selected, enabling the ADC to distinguish 4096 discrete voltage levels across its input range, providing a good trade-off between precision and acquisition speed.

With these configurations, the STM32 board is capable of continuously sampling input signals and generating reliable digital data, forming the core of the oscilloscope's acquisition functionality.

2.4 Design of the Level Shifting Circuit

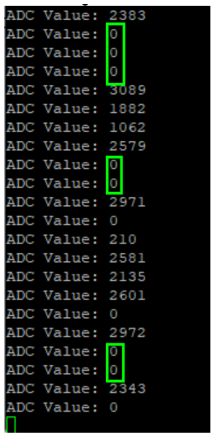


Figure 2.6: Signal clipped due to negative values

As previously discussed, the STM32's ADC cannot read negative voltages—it maps them to zero. This causes waveform clipping and data loss whenever the input signal drops below 0 V, making accurate signal acquisition impossible without prior conditioning.

To address this limitation, we developed a **level shifting circuit** that adds a DC offset to the input signal, effectively shifting the entire waveform into the ADC's valid input range $[0\,\text{V}, 3.3\,\text{V}]$ while preserving its original shape.

To ensure hardware protection, we included a **safety stage** using a Zener diode to limit the input voltage and protect the STM32 from overvoltage. To maximize the ADC's dynamic range, we implemented a **voltage divider** that scales the signal by a factor of 2, keeping the waveform within the ADC's 0–3.3 V window.

In software, this transformation is reversed to reconstruct the original voltage levels during visualization.

The complete level shifting circuit consists of four main blocks:

1. **Protection Circuit:** A Zener diode protects the STM32 by clamping excessive input voltages.
2. **Inverting Summing Amplifier:** This stage shifts the signal by adding a 3.3 V DC component using an op-amp.
3. **Inverting Amplifier:** Restores the correct polarity after the summing operation.
4. **Voltage Divider:** Reduces the amplitude of the shifted signal by half to match the ADC's input range.

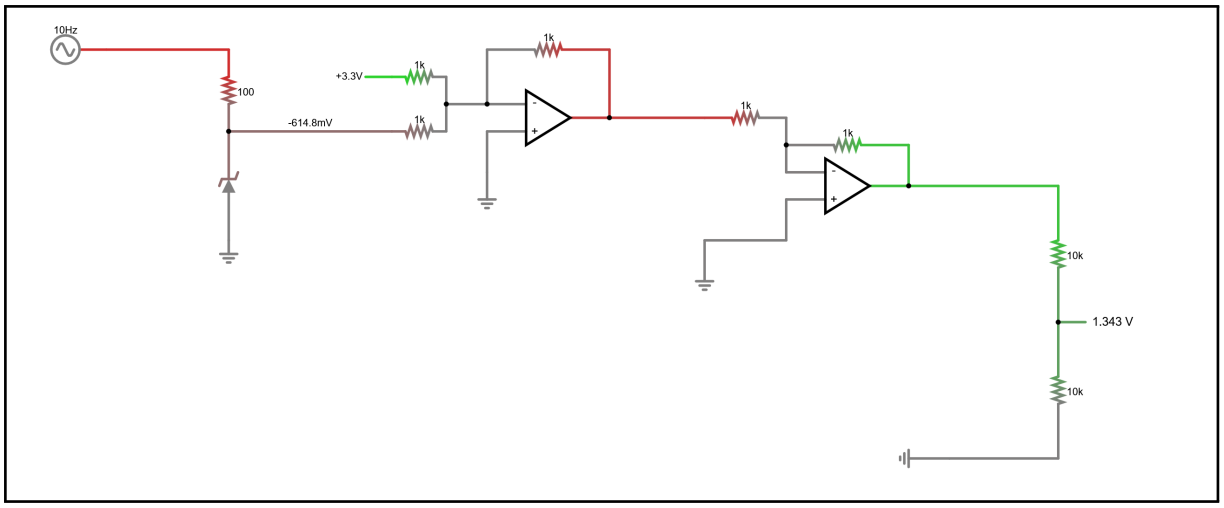


Figure 2.7: Level shifting circuit used to adapt bipolar signals to the STM32 ADC input range

In the following sections, we describe the theoretical design, choice of components, and simulation results that led to the final implementation.

Stage 1: Input Protection Circuit

This stage protects the STM32 and analog processing chain from voltage levels outside the safe range $[0\,\mathrm{V}, 3.3\,\mathrm{V}]$. It consists of a series resistor and a Zener diode connected in reverse bias.

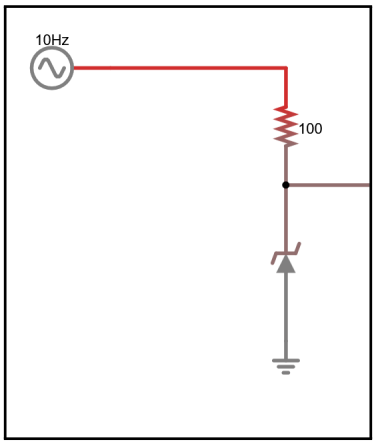


Figure 2.8: safety circuit

- **100 Ω resistor:** Limits current in case of voltage spikes.
- **Zener diode:** Clamps negative voltages below its breakdown value (e.g., $-5.1\,\mathrm{V}$), protecting subsequent stages.

Theoretical Behavior:

- For $|V_{\text{in}}| < V_{\text{Zener}}$: the signal passes through.

- For $V_{\text{in}} < -V_{\text{Zener}}$: diode conducts and clamps the voltage.

This ensures that only safe, bounded voltages are passed to the level-shifting and amplification stages.

Stage 2: Inverting Summing Amplifier

This stage shifts the bipolar input signal into the positive range by adding a DC offset using an inverting summing amplifier configuration built around an op-amp.

- **Inputs:**
  - AC signal (from Stage 1)
  - Constant DC voltage source: $+3.3\,\text{V}$
- **Resistor values:** All equal ($1\,\text{k}\Omega$), for equal weighting.

Output equation:

$$V_{\text{out}} = -\left(\frac{R_f}{R}\right)(V_{\text{in}} + V_{\text{DC}}) = -(V_{\text{in}} + 3.3)$$

Theoretical behavior:

- If $V_{\text{in}} = \pm 1\,\text{V}$, then $V_{\text{out}} = -(\pm 1 + 3.3) = [-4.3, -2.3]\,\text{V}$
- The signal is inverted and shifted downward.

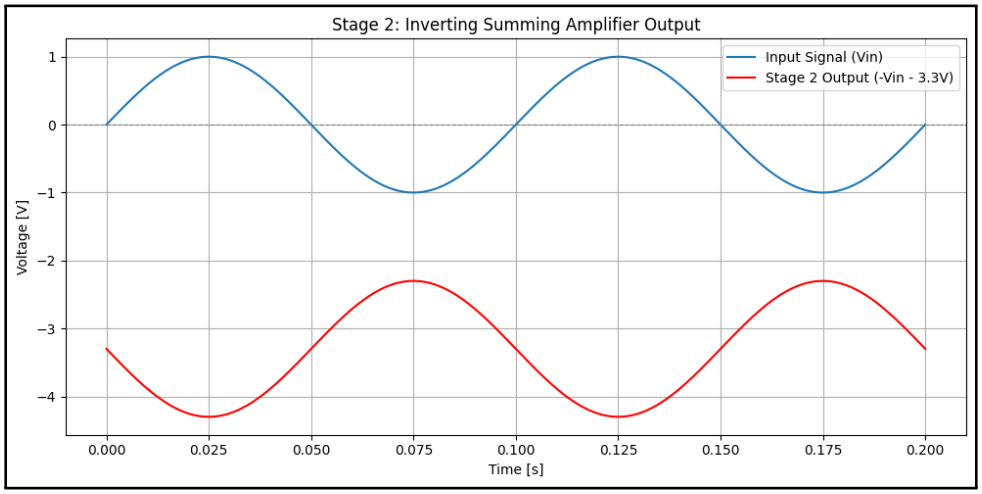


Figure 2.9: Signal Simulation

This sets up the signal for correction in the next stage, which will invert it back into the positive range.

Stage 3: Inverting Amplifier

This stage restores the original polarity of the waveform after the summing amplifier. It uses an op-amp in an inverting configuration, with equal resistors R_3 and R_4 to achieve a gain of -1.

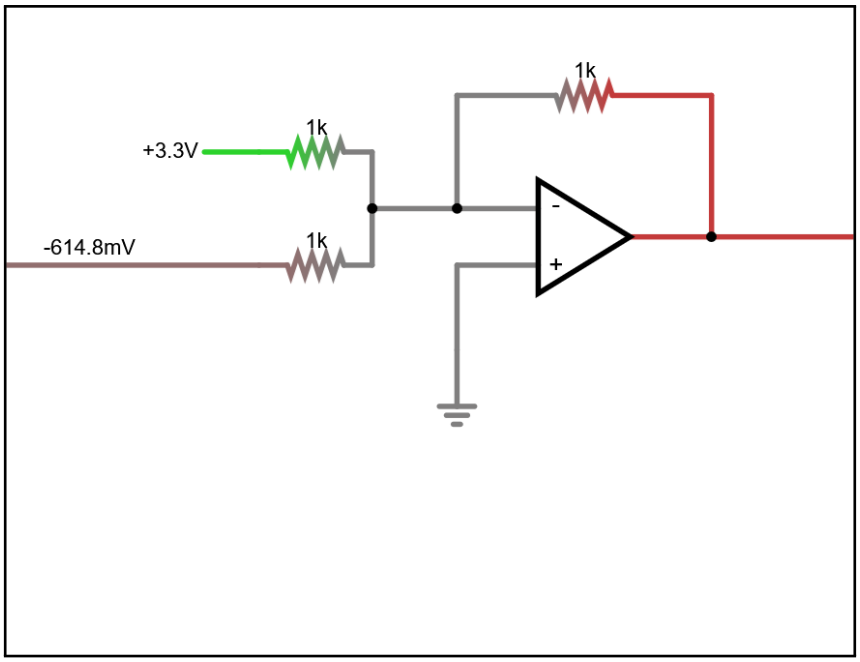


Figure 2.10: summing inverse signal

The output voltage is given by:

$$V_{\text{out}} = -\frac{R_4}{R_3} \cdot V_{\text{in}}$$

Assuming $R_3 = R_4$, this simplifies to:

$$V_{\text{out}} = -V_{\text{in}}$$

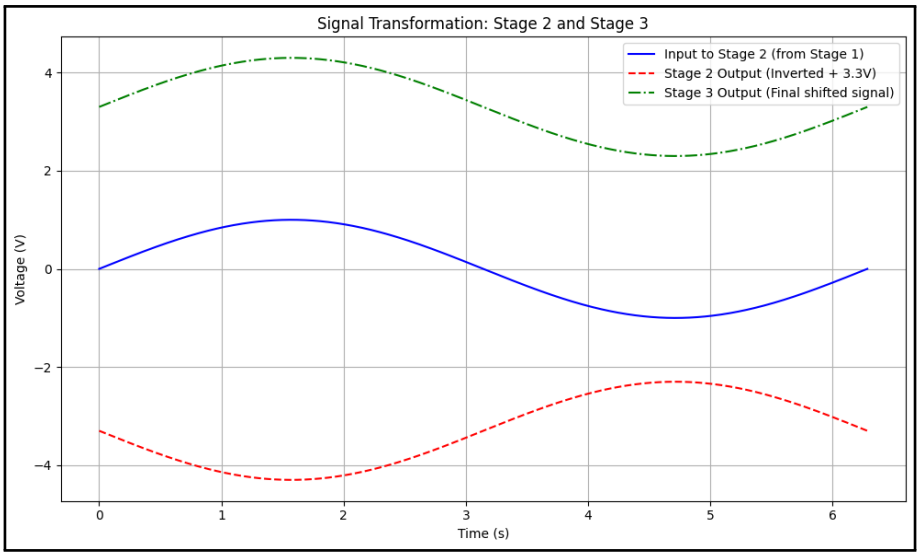


Figure 2.11: signal after stage 3

Thus, the signal is inverted once more, restoring its original polarity while maintaining its amplitude.

Stage 4: Voltage Divider

The final stage ensures the output signal remains within the safe input range of the STM32's ADC $[0\,\mathrm{V}, 3.3\,\mathrm{V}]$. After the previous stage, the signal lies in the range $[2.3\,\mathrm{V}, 4.3\,\mathrm{V}]$, which is too high for direct ADC input.

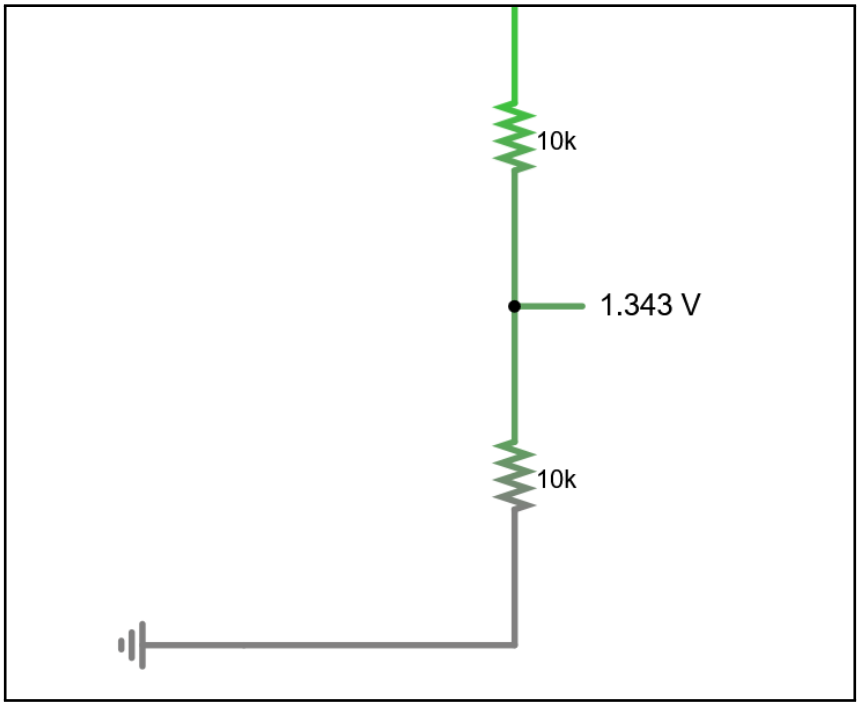


Figure 2.12: Voltage Divider

To scale it down, we use a simple voltage divider consisting of two equal resistors $R_1 = R_2 = 10\,\text{k}\Omega$. The output is given by:

$$V_{\text{out}} = V_{\text{in}} \cdot \frac{R_2}{R_1 + R_2} = \frac{1}{2} \cdot V_{\text{in}}$$

This scales the signal range to:

$$V_{\text{out}} = \frac{1}{2} \cdot [2.3, 4.3] = [1.15, 2.15]\,\text{V}$$

which is fully compatible with the ADC. This stage also helps protect the microcontroller from overvoltage, completing the signal conditioning chain.

2.5 DMA Configuration

The Direct Memory Access (DMA) controller is essential for efficient data handling in the oscilloscope system, allowing peripherals like ADC1 and USART1 to transfer data directly to or from SRAM without continuous CPU involvement. This is critical for the oscilloscope's high-speed data acquisition, where the signal acquisition card uses ADC1 to sample analog signals, and the processing card receives digitized data via USART1. For instance, when ADC1 completes a conversion, DMA1 Stream 0 transfers the 12-bit result directly to a memory buffer, while USART1 can use DMA2 to transmit processed data to a PC or display unit. This setup significantly reduces CPU load, enabling tasks like signal processing or low-power mode entry, and supports continuous real-time sampling at 100 kHz.

2.5.1 DMA Configuration Parameters

The DMA setup, implemented using STM32CubeMX for the STM32 NUCLEO-H533RE board, is optimized for real-time signal capture with the following parameters:

- **DMA Channel**: DMA1 Stream 0 for ADC1, chosen to avoid conflicts with other peripherals.
- **Mode**: Circular mode to enable uninterrupted sampling by automatically restarting buffer filling.
- **Data Direction**: Peripheral-to-memory to transfer ADC conversion data into SRAM.
- **Buffer Size**: 1024 samples, balancing memory use and signal resolution.
- **Data Width**: 16 bits to align 12-bit ADC data properly.
- **Trigger Source**: ADC conversion-complete signal triggers DMA transfer.

- **Interrupts**: Enabled for transfer complete and half-complete to alert CPU for processing or transmission.
- **Memory Addressing**: Memory increment mode enabled; peripheral address fixed.
- **Error Handling**: DMA error interrupts activated for handling transfer issues such as FIFO overruns.

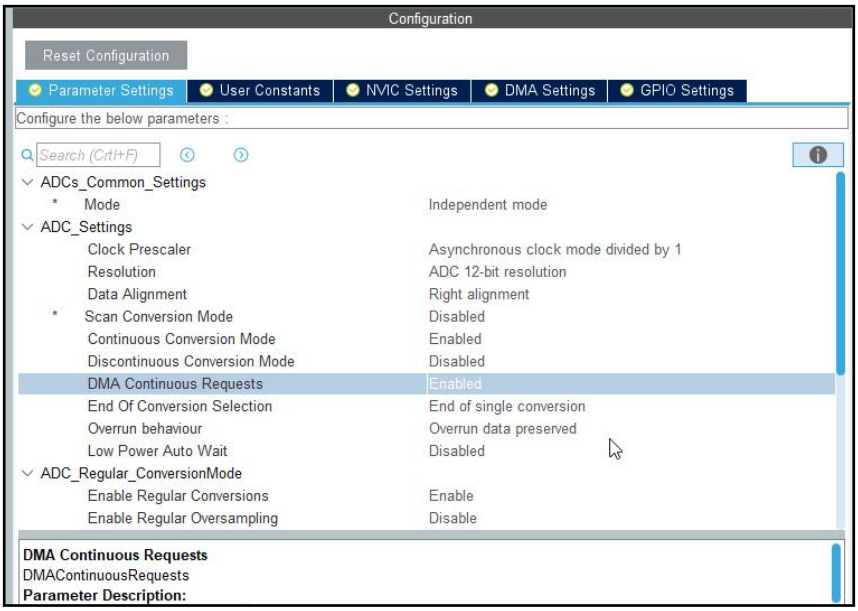


Figure 2.13: DMA Configuration Interface in STM32CubeMX

2.5.2 Buffer Management and System Integration

The DMA's circular buffer mechanism ensures continuous data acquisition by writing ADC1 results into a 1024-sample SRAM buffer. Upon half or full buffer completion, corresponding interrupts allow the CPU to process or forward the data, e.g., via USART2 to a PC or display. This approach supports seamless real-time visualization and waveform reconstruction without data loss, aligning with the ADC's 12-bit resolution and 100 kHz sampling rate. Error interrupts provide robustness by catching transfer faults, ensuring reliable oscilloscope operation.

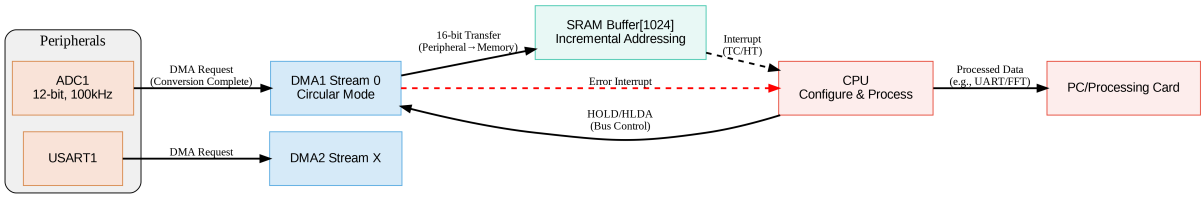


Figure 2.14: Visualization of DMA Buffer Data Flow

2.6 Digital Filtering

To improve the quality of the acquired signal and remove high-frequency noise, we integrated a **digital low-pass filter** into our oscilloscope interface. This filter was implemented in Python using the `scipy.signal` module and can be easily activated from the GUI with a single click on the "Filter" button.

2.6.1 Low-Pass Filter Implementation

The filtering algorithm is based on a classical **Butterworth low-pass filter**, which offers a smooth frequency response and is well-suited for real-time signal processing applications. The function is defined as follows:

```
def low_pass_filter(self,data,cutoff_freq,sampling_rate,order=5):
    """Apply a low-pass filter to the signal."""
    nyquist_freq = 0.5*sampling_rate
    normal_cutoff = cutoff_freq / nyquist_freq
    b, a = butter(order, normal_cutoff, btype='low',analog=False)
    return lfilter(b, a, data)
```

Listing 2.2: Python implementation of the low-pass filter

2.6.2 Explanation

- `cutoff_freq`: The cutoff frequency of the filter, above which higher-frequency components will be attenuated.
- `sampling_rate`: The rate at which the original signal was sampled.
- `nyquist_freq`: Half the sampling rate, as defined by the Nyquist theorem.
- `normal_cutoff`: Normalized cutoff frequency, used for digital filter design.
- `butter`: Designs the Butterworth filter of a given order.
- `lfilter`: Applies the designed filter to the data.

2.6.3 Integration with the GUI

The filter is directly integrated into the graphical interface of the oscilloscope. By clicking the **"Filtering"** button, the user can instantly apply the low-pass filter to the currently displayed signal, smoothing out the waveform and making it easier to analyze.

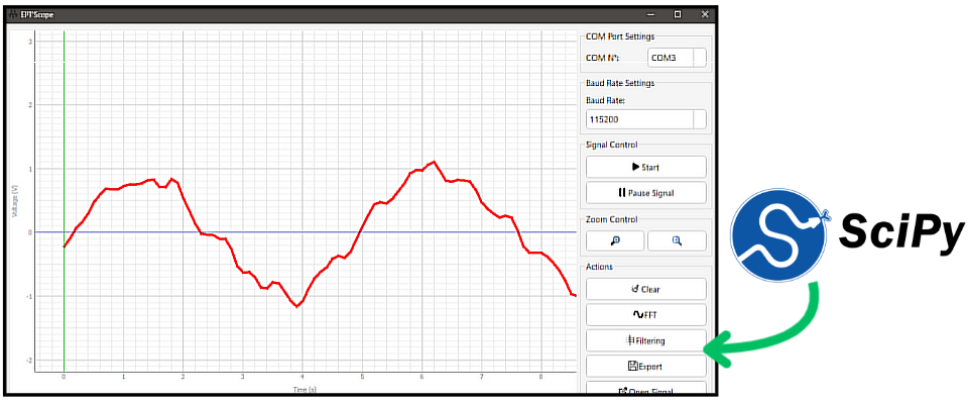
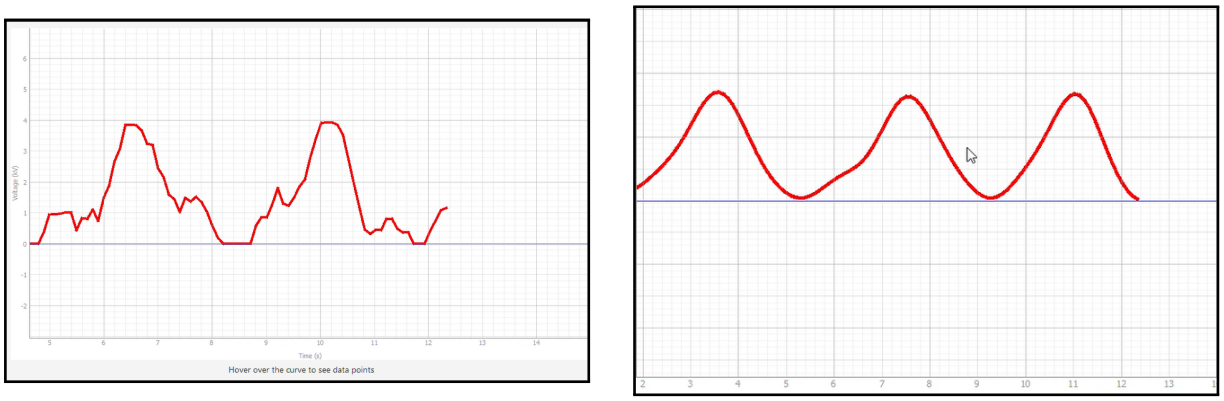


Figure 2.15: Filter button in the GUI for real-time signal smoothing

This approach enhances the readability of the signal by removing undesired noise, providing users with a clearer visualization for diagnostics and analysis.



(a) Raw signal before filtering

(b) Signal after low-pass filtering

Figure 2.16: Comparison of the signal before and after applying the low-pass filter

Interpretation: The comparison between subfigures (a) and (b) clearly illustrates the effect of the low-pass filter. In the raw signal (a), high-frequency noise and irregularities are prominent, which may hinder accurate signal analysis. After filtering (b), the waveform becomes significantly smoother, with unwanted fluctuations attenuated while preserving the overall signal shape. This demonstrates the effectiveness of digital filtering in improving signal clarity for visualization and further analysis.

Chapitre

3 EPT'Scope Software Development

3.1 EPT'Scope Software

3.1.1 Introduction

In this project, we had the opportunity to develop our own oscilloscope software, named **EPTScope**. The development of this software was a crucial component of the project, allowing us to visualize the data acquired by the STM32 in real time. In this section, we detail the implementation process and discuss the challenges we encountered.

Figure 3.1: Software Logo

3.1.2 Tools and Libraries Used

The development of EPTScope was carried out in Python, using several libraries to manage the graphical interface, real-time plotting, data processing, and serial communication. Below is a summary of the key tools and libraries used:

- **PyQt5** – Used to design and implement the graphical user interface (GUI), including widgets like buttons, combo boxes, and dialogs.
- **PyQtGraph** – A fast, interactive plotting library used for displaying real-time waveforms with support for zooming, panning, and exporting plots.
- **PySerial** – Handles serial communication between the PC and the STM32 microcontroller. It allows for reading incoming data through the UART interface.
- **Threading (Python)** – Used to manage asynchronous data acquisition without blocking the GUI. A separate thread reads serial data while the main thread updates the interface.

- **NumPy** – Provides efficient numerical operations, especially for processing and storing waveform data.
- **SciPy** – Used for signal processing, including Fast Fourier Transform (FFT) and digital filtering (Butterworth filter).
- **Datetime & re** – Standard Python libraries used for handling timestamps and parsing formatted data using regular expressions.
- **OS & QFileDialog** – Used for file management operations such as saving images or selecting data paths.
- **Custom Modules:**
  - `math_signal` – Implements mathematical operations on signals.
  - `fft_a` – Manages frequency domain analysis using FFT.

3.1.3 User Interface

The EPTScope software features a graphical user interface (GUI) developed using the PyQt5 framework. It provides an intuitive and responsive environment for users to visualize real-time signals received from the STM32 microcontroller. The main display includes a waveform plot area, control buttons (Start, Pause, Zoom In/Out, Auto-Scale), and configuration panels for user interaction.

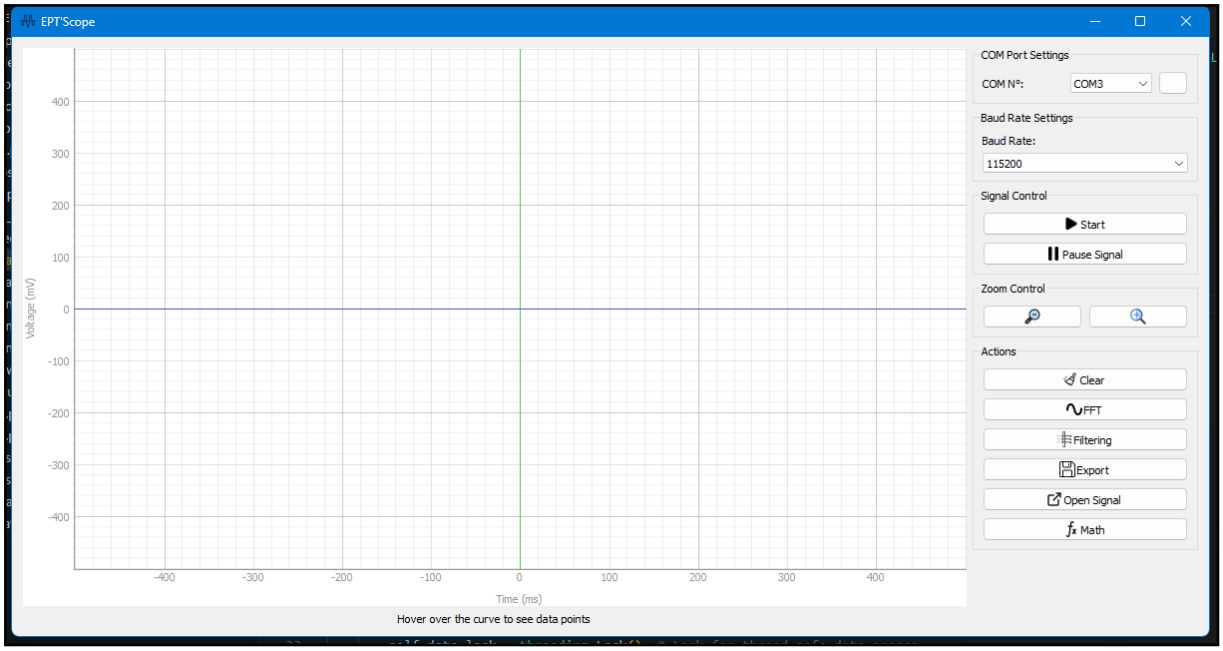


Figure 3.2: Main GUI

3.1.4 Implemented Features

EPTScope includes a wide range of features designed to offer a flexible and user-friendly oscilloscope experience. These features enhance signal visualization, analysis, and data handling:

```
class OscilloscopeUI(QWidget):
    def __init__(self):
        super().__init__()
        self.initUI()
        # Initialize serial connection and data storage
        self.ser = None
        self.data = []
        self.timestamps = []
        self.is_running = False

        self.data_lock = threading.Lock()
        # Initialize timer for plot updates
        self.timer = QTimer()
        self.timer.timeout.connect(self.update_plot)
```

Python Class: OscilloscopeUI

- **Pause/Resume:** Allows users to freeze and resume the signal display in real time.
- **Zoom In/Out:** Enables users to focus on specific portions of the waveform for detailed analysis. This can be done interactively using mouse controls or buttons.
- **Auto-Scaling:** Automatically adjusts the axes to fit the amplitude and time scale of the displayed signal, ensuring optimal visibility.
- **Fast Fourier Transform (FFT):** Offers frequency-domain analysis of the acquired signal. Users can switch to FFT view to observe the frequency components of the signal.
- **Save Signal as Text or Image:** Provides the ability to export the current signal to a text file for further analysis, or to save a snapshot of the waveform as an image (e.g., PNG).
- **Open Previously Saved Signal:** Users can load and visualize previously recorded signals from text files, allowing post-acquisition analysis and comparison.
- **Digital Filtering (Butterworth Filter):** Includes a low-pass Butterworth filter to reduce noise from the acquired signal. Parameters such as cutoff frequency and filter order can be configured.
- **Cursor-Based Measurement:** A movable vertical cursor allows users to inspect specific signal points and read corresponding time and amplitude values directly from the graph.

- **Math Mode:** Include the abality the do math calculation on the signal

3.1.5 How the Software Works

EPTScope operates as a real-time oscilloscope software by integrating serial data acquisition, dynamic plotting, and interactive user control through a modular design. In the next part we will expalin the Software part and how its is working.

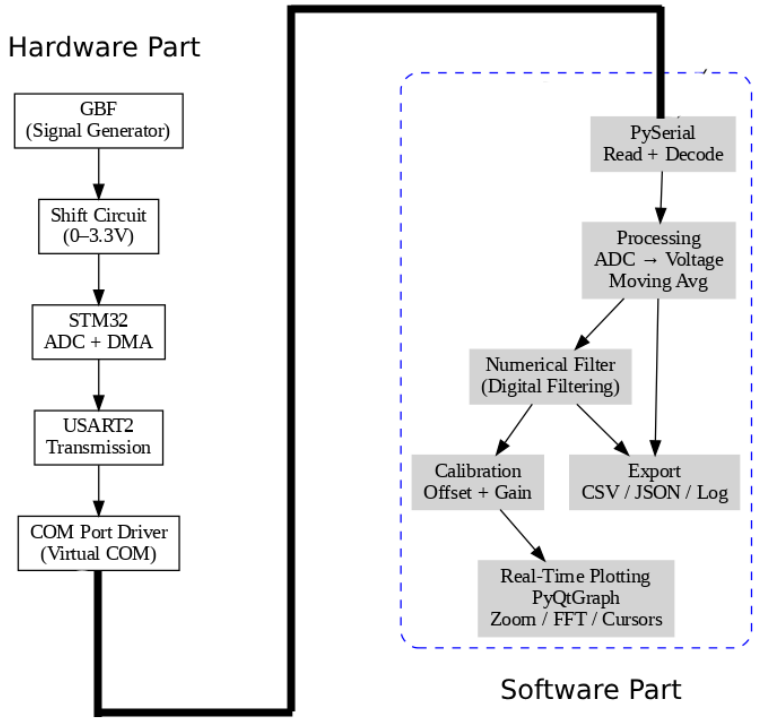


Figure 3.3: EPTScope architecture

Initialization and Configuration

Upon launching the application, the software initializes key components including:

- Serial port manager to detect and list available COM ports.
- Plotting widget configured for real-time signal visualization.

- Timers for refreshing both the waveform display and the list of active COM ports.
- A separate data acquisition thread that listens to the UART stream from the STM32.

The user selects the appropriate COM port and baud rate before pressing the `Start` button to begin acquisition.

Data Acquisition and Threading

Once started, a dedicated background thread reads incoming signal values from the serial port. These values represent analog voltages sampled by the STM32 and sent as ASCII. The software:

- Parses each data point.
- Appends it to a thread-safe data buffer using a locking mechanism to avoid race conditions.
- Associates each value with a timestamp to construct the waveform.

This thread runs independently from the main UI thread, ensuring smooth performance and avoiding interface freezes.

Real-Time Visualization

The main thread uses a `QTimer` to periodically update the PyQtGraph plot. Each update extracts the most recent data from the buffer and redraws the waveform. The graph supports:

- Real-time scrolling and zooming.
- Grid lines and labels for amplitude and time.
- Dynamic auto-scaling.
- Cursor tracking for precise measurement.

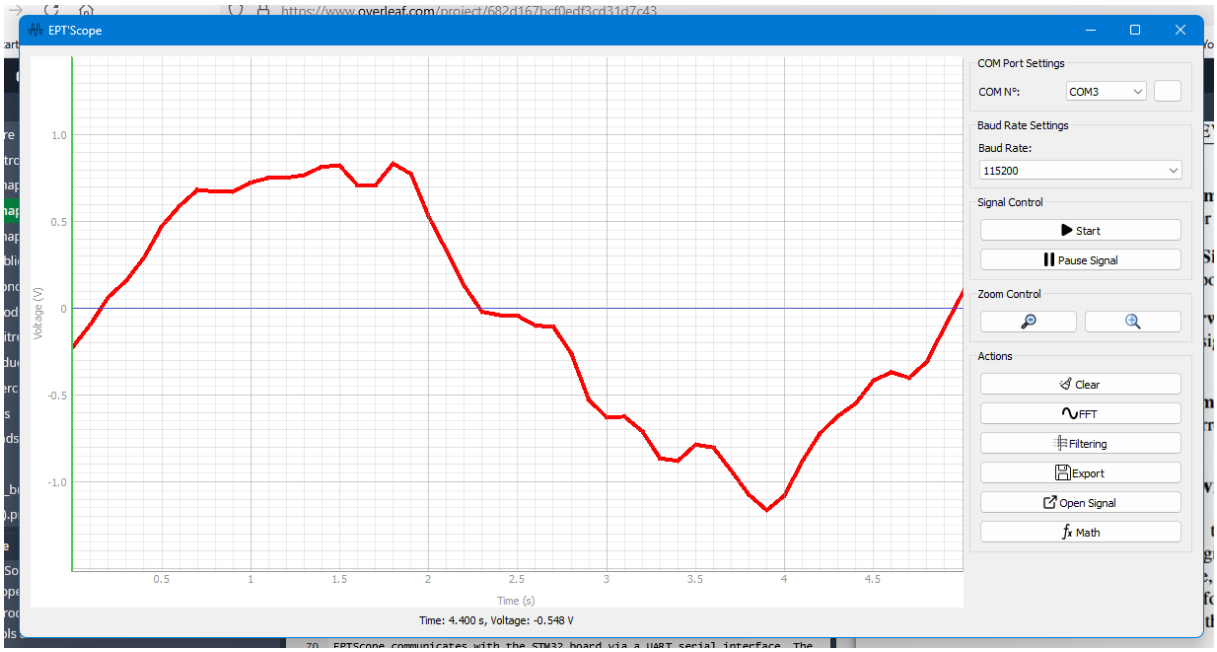


Figure 3.4: Signal test

Signal Processing and Analysis

Several processing operations can be performed on the signal:

- **FFT:** Computes the frequency spectrum of the signal using SciPy's FFT implementation. The result is displayed in a new window.

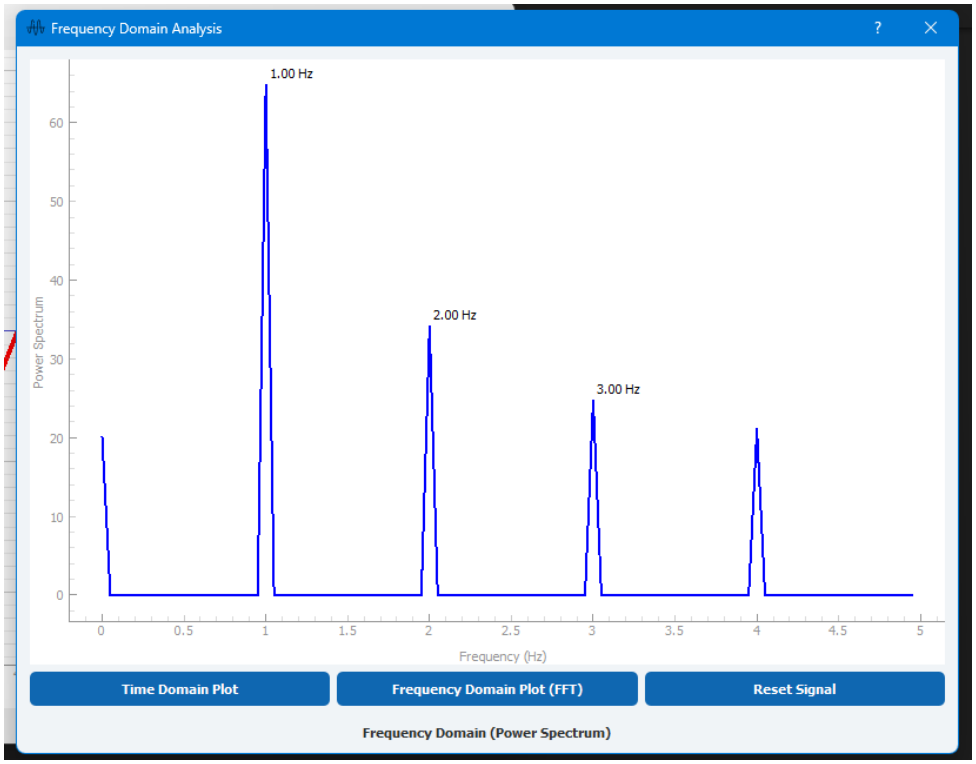
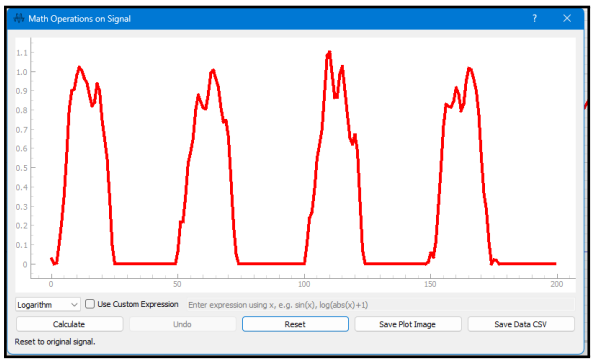
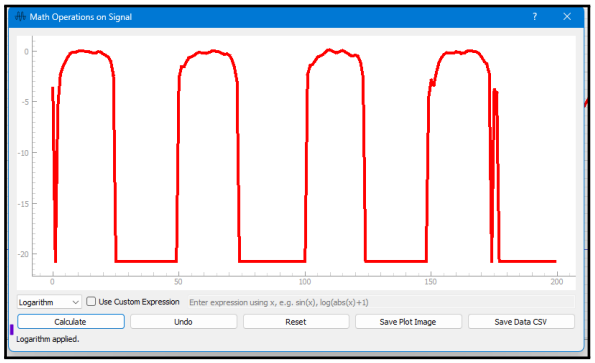


Figure 3.5: Signal in Frequance Domain

- **Digital Filtering:** Applies a Butterworth filter with configurable cutoff frequency and order to remove high-frequency noise.
- **Math Operations:** Allows the application of mathematical functions (e.g., `log(x)`, `sin(x)`) to the signal. Users can also apply custom formulas written in NumPy syntax. .



(a) Normal Signal



(b) Signal After Logarithmique transformation

Figure 3.6: Comparison between original and filtered signals

Data Management

EPTScope offers robust features for signal data handling, ensuring both flexibility and traceability during experiments and analysis:

- **Signal Export:** Acquired signals can be exported in various formats, including:
  - *CSV or plain text*: Ideal for numerical analysis in external tools (e.g., MATLAB, Excel, Python).
  - *PNG images*: Useful for quick sharing or documentation of waveform snapshots.
- **Signal Import:** Previously recorded sessions can be reloaded:
  - Compatible formats can be opened and plotted within the interface.
- **Session Saving:** Entire measurement sessions can be saved:
  - Preserves timestamps, signal values, and calibration settings.
  - Facilitates post-processing and offline review.
  - Promotes reproducibility of experiments.

User Interaction

The GUI layout includes:

- Controls for COM settings, signal acquisition, zoom, and scaling.

- Buttons for exporting, importing, and clearing data.
- Dedicated buttons for FFT, filtering, and math operations.
- Visual feedback via labels and icons for user-friendly interaction.

Thanks to its modular and responsive design, EPTScope provides an efficient and user-oriented environment for signal analysis and visualization.

3.1.6 Communication with the STM32

EPTScope communicates with the STM32 board via **USART2**, configured as a UART interface (e.g., 115200 bps, 8N1). The STM32 sends ADC samples as plain-text lines, each containing one or more integer values.

On the PC side, a background thread uses the PySerial library for non-blocking data reception. Incoming lines are decoded and parsed using regular expressions, and the most recent 12-bit ADC value is converted to voltage as:

$$V = \left(2 \times \frac{\text{ADC}}{4095}\right) \times V_{\text{ref}} - 3.3, \quad V_{\text{ref}} = 3.3\,\text{V}$$

This restores the original bipolar signal range (±3.3 V) after amplification and shifting.

The application maintains a rolling buffer of the last 1000 samples with timestamps for real-time plotting. A mutex lock ensures thread-safe access. Serial errors or malformed data are safely handled by stopping acquisition and closing the port.

This setup ensures responsive and reliable communication between STM32 and the PC.

3.1.7 Conclusion

The development of the EPTScope software successfully integrates real-time data acquisition, signal processing, and user interaction within a robust and modular architecture. Leveraging Python libraries such as PySerial and PyQtGraph, the application ensures smooth communication with the STM32 board, accurate voltage reconstruction, and an interactive user interface for visualization and analysis. Its multithreaded design, combined with efficient data handling and export features, makes EPTScope a reliable tool for oscilloscope-like signal monitoring on a standard PC.

Chapitre

4 EPT'Scope 3D Case

4.1 Introduction

In our project, we are building a compact oscilloscope based on two STM32 Nucleo boards, a custom PCB and a source of alimentation that provides power. To ensure both protection of the electronics and an ergonomic user experience, a thoughtfully designed 3D-printed enclosure is essential. The enclosure must:

- Securely hold the two Nucleo boards and the custom PCB.
- Accommodate wiring harnesses, a battery compartment, and allow easy access to connectors.
- Provide sufficient passive ventilation to prevent overheating during continuous operation.
- Offer comfortable hand grips and rounded edges to facilitate handling—especially when probing or transporting.
- Be simple to manufacture on a desktop FDM printer, using common materials such as PLA.

4.2 Tools and Software

This section details the tools and software used in the design and development of the EPTScope 3D enclosure.

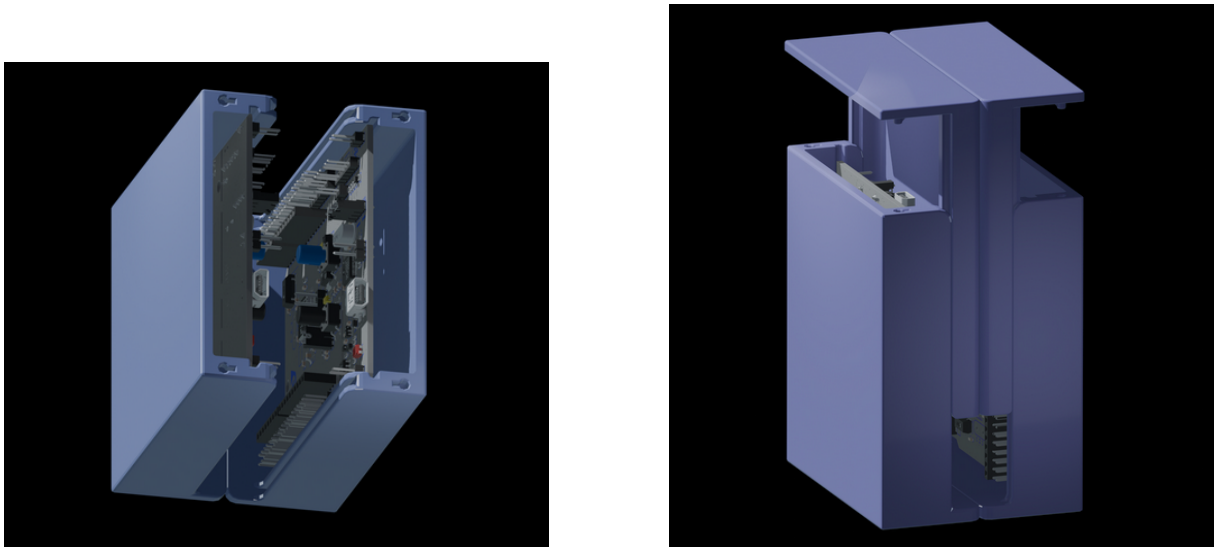
- **CAD Software**:
  - **Prototype 0**: CATIA V5 (for initial conceptual modeling and basic form-factor validation).
  - **Prototypes 1 & 2**: SolidWorks 2024 (for iterative refinement, structural analysis, and detailed feature design).
- **Materials**:

  - PLA (Polylactic Acid) filament, chosen for its low cost, ease of printing, and acceptable mechanical strength for a handheld enclosure.
  - PETG could also be a good choice for higher heat resistance, though final prints remain in PLA.

- **Measurement Tools**: Digital calipers and precision rulers were used throughout prototyping to verify critical dimensions and tolerances.

4.3 Prototype 0

CAD Tool: CATIA V5



(a) Internal view (without lid). **(b) External view (with lid).**

Figure 4.1: Prototype 0 designed in CATIA V5.

Prototype 0 is our first exploratory design. Its main characteristics are:

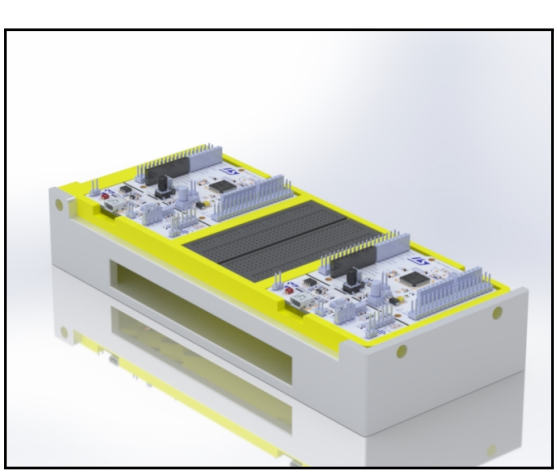
- Modeled in CATIA V5 to quickly validate overall dimensions for two STM32 Nucleo boards, a central breadboard, and the power PCB.
- No ventilation slots or cutouts—used primarily to verify fitment and basic form factor.
- Simple single-shell structure with a removable lid. Intended to confirm that the electronics could be housed without interference.
- Easy to manufacture with minimal overhangs, requiring almost no support material during printing.
- Basic "plug-and-play" arrangement, but lacks specific ergonomics (grips, rounded corners) and thermal management.

Pros and Cons of Prototype 0:

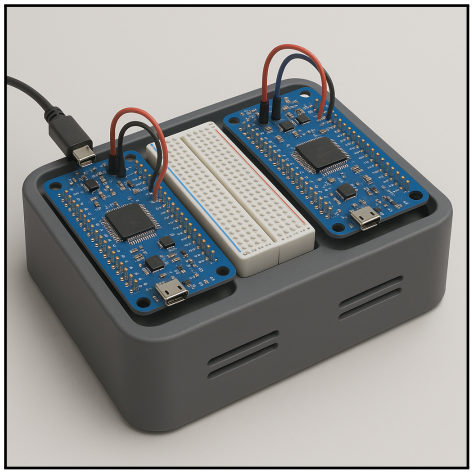
- *Pros*:
  - Rapid CAD iteration in CATIA V5.
  - Confirmed board spacing and dimensions.
  - Simplified print (no supports).
- *Cons*:
  - No ventilation—risk of heat buildup.
  - No integrated battery compartment—battery had to sit externally.
  - Rectangular shape without ergonomic features—awkward to handle.

4.4 Prototype 1

CAD Tool: SolidWorks 2024



(a) 3D view of Prototype 1.



(b) SolidWorks sketch of Prototype 1.

Figure 4.2: Prototype 1 .

Prototype 1 incorporates multiple improvements based on the findings from Prototype 0:

- **Ventilation slots** were introduced on the back side and under the boards (two slots per side, each 5 mm × 20 mm) to allow passive airflow and reduce internal temperature.
- The internal layout was rearranged to add a shallow recessed area for a mini breadboard between the two Nucleo boards, simplifying wiring.
- Top edges and corners were rounded (radius 5 mm) to enhance handling comfort.
- The lid now features snap-fit tabs instead of screws, allowing tool-less removal.

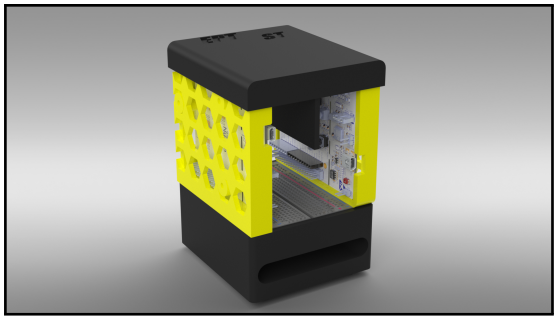
- All features remain compatible with PLA printing—print settings produce minimal overhangs, though some supports are needed for ventilation cutouts.

Pros and Cons of Prototype 1:

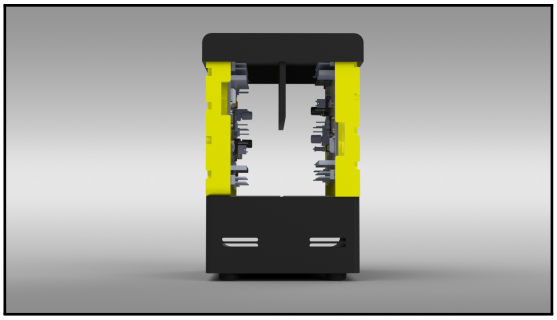
- *Pros*:
  - Added passive ventilation—internal heat distribution improved.
  - Ergonomic rounding and improved lid retention.
  - Breadboard recess simplifies prototyping on-board.
- *Cons*:
  - Slightly larger footprint and increased print time (supports for slots).
  - Weight distribution still top-heavy—lacks dedicated battery housing.
  - Side walls are flat—no integrated grips for handheld use.

4.5 Prototype 2

CAD Tool: SolidWorks 2024



(a) Front view of Prototype 2.



(b) Back view of Prototype 2.

Figure 4.3: Prototype 2 (Final Design).

Prototype 2 is the culmination of iterative improvements, integrating thermal, ergonomic, and modular enhancements:

- **Angled Ventilation**:
  - Strategically placed vents facilitate efficient passive cooling through natural convection. Side vents are angled downward at 15 degrees and bottom vents are parallel to the base plate. This design ensures that the ventilation will work by passive cooling.
  - Slot dimensions: 5 mm × 20 mm (sides), 3 mm × 25 mm (bottom).

- **Modular Mounting System**:
  - The Nucleo boards and the custom PCB are easily accessible through a modular spring-clip mounting system.
- **Ergonomic Design**:
  - Side grips are molded into the enclosure, and rounded edges for comfortable handling.
- **Internal Layout**:
  - Slim profile and optimized wall thickness for weight reduction.
- **Integrated Battery Compartment**:
  - Slide-in compartment on the bottom houses a single 18650 Li-Ion cell.
  - Quick battery swaps are made easy without the need for tools

Since Prototype 2 satisfied all requirements—adequate ventilation, ease of handling, modular board insertion, and reduced weight—it was selected as the final enclosure for the EPTScope project.

4.6 Comparison Between Prototypes

Feature	Prototype 2	Prototype 1	Prototype 0
CAD Software	SolidWorks 2024	SolidWorks 2024	CATIA V5
Ventilation	Angled vents (sides & bottom)	Side ventilation slots (all 4)	None
Profile	Compact and flexible	Flexible	Compact
Safety / Ergonomics	Safe, ergonomic, and stable	Safe	Basic
Ease of Assembly	Medium difficulty (clips)	Easy (snap-fit lid)	Very easy (simple lid)
Material Options	PLA, PETG, ABS	PLA	PLA
Cost	Cost-effective	Cost-effective	Cost-effective
Hand Manipulation	Easy grip (integrated handles)	Very easy (rounded edges)	Moderate (no handles)
Battery Compartment	Integrated slide-in housing	External battery (no compartment)	External battery
Weight (empty)	$\approx$ 120 g	$\approx$ 150 g	$\approx$ 140 g

Table 4.1: Comparison of Prototypes 0, 1, and 2.

From Table 4.1, Prototype 2 clearly outperforms the earlier designs in all critical aspects: it provides superior thermal management, improved ergonomics, ease of assembly, and integrated battery housing—while still remaining cost-effective and straightforward to print.

4.7 Final Selection: Prototype 2 Advantages

We selected Prototype 2 as our final design for the following reasons:

- **Ventilation**: The angled vents on the sides and bottom maximize natural convection, keeping internal components within safe operating temperatures without the need for additional fans.
- **Ergonomics**: Integrated side grips, rounded edges, and a low-profile form factor make it comfortable for handheld use during bench testing or live demonstrations.
- **Modularity**: Spring-clip mounts allow each STM32 Nucleo board and the custom power PCB to be inserted or removed quickly—ideal for rapid hardware iterations and debugging.

- **Weight Reduction**: By optimizing wall thicknesses and internal geometry, the empty enclosure weighs roughly 120 g, making it light for handheld operation without sacrificing rigidity.
- **Integrated Power Compartment**: The slide-in 18650 battery housing ensures a tidy wiring layout and prevents accidental battery disconnection—critical when the oscilloscope is moved or tilted.
- **Manufacturability**: Designed specifically for desktop FDM printing:
  - No support material is needed for most overhangs except minor supports under the angled vents.
  - Print time for the two halves (main body + lid) is under 3 hours total at 50 mm/s.
  - Standard PLA filament (1.75 mm) is used, minimizing cost and offering good dimensional stability.

4.8 Conclusion

Prototype 2 is the definitive 3D-printed case for our STM32-based oscilloscope. Below are the final specifications and additional remarks:

Category	**Details**
External Dimensions	Length: 180 mm Width: 100 mm Height: 60 mm (including side grips)
Material	PLA (Polylactic Acid). PETG is an alternative for higher heat resistance.
Wall Thickness	Main body: 2.5 mm Around clip supports: 3 mm (for added rigidity)
Ventilation Slots	Side vents: 5 mm × 20 mm (angled 15° downward) Bottom vents: 3 mm × 25 mm (horizontal)

Table 4.2: Enclosure Physical and Structural Specifications

Category	Details
Weight (Empty)	120 g
Weight (Assembled)	260 g (including two Nucleo boards, custom PCB, breadboard, wiring, and one 18650 battery)
Post-Processing	Light sanding (220–400 grit) on mating faces Optional matte finish spray for improved tactile comfort
Identification Sticker	Laser-cut label (80 mm × 20 mm) affixed to the back panel with: Model Name: EPTScope Enclosure v2 Dimensions: 180 × 100 × 60 mm Material: PLA (2.5 mm walls) Weight: 120 g (empty) CAD Software: SolidWorks 2024

Table 4.3: Enclosure Weight, Finishing, and Identification

With these design refinements, Prototype 2 delivers a robust, ergonomic, and easy-to-manufacture enclosure that meets all requirements for the EPTScope oscilloscope. The enclosure is now ready for final printing, assembly, and deployment in the laboratory environment.

Chapitre

5 Economic Analysis

Introduction

This chapter presents a comprehensive economic analysis of the EPT'Scope oscilloscope project, examining the financial viability and market potential of this innovative open-source measurement instrument. The analysis encompasses detailed cost breakdowns, market positioning, competitive analysis, and a complete business plan with financial projections. By leveraging modern microcontroller technology and efficient manufacturing processes, EPTScope aims to democratize electronic measurement tools for educational institutions and electronics enthusiasts while maintaining commercial viability.

5.1 Economic Analysis

5.1.1 Project Cost Breakdown

The development of the EPTScope oscilloscope involved various costs across hardware, software development, and manufacturing. This section provides a comprehensive analysis of the economic aspects of the project.

5.1.1.1 Hardware Costs

Component	**Unit Price (TND)**	**Quantity**	**Total (TND)**
STM32 NUCLEO-H533RE	47.28	2	94.56
Custom Level Shifting PCB	60.00	1	60.00
Electronic Components (R, C, Op-Amps)	35.00	1	35.00
18650 Li-Ion Battery	16.00	1	16.00
USB Cables	12.00	2	24.00
Breadboard	12.00	1	12.00
Miscellaneous (Wires, Connectors)	25.00	1	25.00
Total Hardware Cost			**266.56**

Table 5.1: Hardware Cost Breakdown (Tunisian Prices)

5.1.1.2 Development Costs

Development Phase	**Hours**	**Rate (TND/h)**	**Total (TND)**
Hardware Design & Testing	10	10.00	100.00
Firmware Development	15	12.00	180.00
Software Development (EPTScope)	25	10.00	250.00
3D Enclosure Design	8	8.00	64.00
Testing & Calibration	10	10.00	100.00
Documentation	6	8.00	48.00
Total Development Cost	**74**		**742.00**

Table 5.2: Development Cost Breakdown

5.1.1.3 Manufacturing and 3D Printing Costs

Item	Unit Cost (TND)	Quantity	Total (TND)
PLA Filament (1kg for 80 TND)	80.00	0.12	9.60
3D Printer Usage (3 hours)	15.00	3	45.00
Post-Processing Materials	15.00	1	15.00
PCB Manufacturing	60.00	1	60.00
Assembly Labor (2 hours)	35.00	2	70.00
Total Manufacturing Cost			**199.60**

Table 5.3: Manufacturing Cost Breakdown

5.1.1.4 Total Project Cost Summary

Cost Category	Amount (TND)
Hardware Components	266.56
Development	742.00
Manufacturing (per unit)	199.60
Total Project Cost (incl. Development)	**1,208.16**
Cost per Unit (Prototype - Hardware + Manuf.)	**466.16**
Cost per Unit (Hardware for 1 unit + Manuf.)	**476.60**

Table 5.4: Total Project Cost Summary (TND)

5.1.1.5 Value Added Network (VAN) Analysis

Stage	Input Value (TND)	Value Added (TND)	Output Value (TND)
Raw Materials	0.00	266.56	266.56
PCB Manufacturing	266.56	60.00	326.56
Assembly	326.56	70.00	396.56
3D Printing & Enclosure	396.56	69.60	466.16
Testing & Quality Control	466.16	50.00	516.16
Packaging & Distribution	516.16	33.84	550.00
Total Value Added		**550.00**	
Final Product Value			**550.00**

Table 5.5: Value Added Network (VAN) Analysis

5.1.2 Market Analysis

5.1.2.1 Target Market Segments

The EPTScope oscilloscope targets several distinct market segments:

- **Educational Institutions**: Universities, technical schools, and training centers requiring affordable oscilloscopes for student laboratories.
- **Hobbyist Electronics Community**: Makers, DIY enthusiasts, and amateur radio operators seeking cost-effective measurement tools.
- **Small Electronics Businesses**: Startups and small companies needing basic oscilloscope functionality without high capital investment.
- **Developing Markets**: Regions where expensive commercial oscilloscopes are not economically viable.

5.1.2.2 Competitive Analysis

Product	Price (TND)	Bandwidth	Channels	Portability
EPTScope	950	1 MHz	1	Excellent
Hantek DSO5072P	1,450	70 MHz	2	Good
Rigol DS1054Z	1,850	50 MHz	4	Fair
Siglent SDS1052DL+	1,100	50 MHz	2	Fair
DIY Oscilloscope Kits	500-850	0.5-2 MHz	1-2	Variable

Table 5.6: Competitive Analysis

5.1.3 Value Proposition

The EPTScope offers several unique advantages:

- **Cost-Effectiveness**: Significantly lower than commercial alternatives while maintaining essential functionality.
- **Educational Value**: Complete transparency of design allows students to understand oscilloscope principles.
- **Customizability**: Open-source nature enables modifications and enhancements.
- **Portability**: Compact, battery-powered design suitable for field measurements.
- **Modern Interface**: Contemporary software with FFT, filtering, and data export capabilities.

5.2 Business Plan

5.2.1 Executive Summary

EPTScope represents an innovative approach to democratizing electronic measurement tools through open-source hardware and software. By leveraging modern microcontroller technology and efficient manufacturing processes, we aim to provide educational institutions and electronics enthusiasts with an affordable, yet capable oscilloscope solution.

5.2.2 Business Model

5.2.2.1 Revenue Streams

1. **Hardware Sales**: Direct sales of assembled EPTScope units
2. **Kit Sales**: DIY kits for educational and hobbyist markets
3. **Licensing**: Technology licensing to educational institutions
4. **Support Services**: Technical support and training programs
5. **Accessories**: Additional probes, cases, and expansion modules

5.2.2.2 Sales Strategy

Product Variant	**Explanation**
Assembled Unit	Fully built and tested EPTScope device, ready to use out-of-the-box.
DIY Kit	A do-it-yourself version, where the customer assembles the product themselves.
Educational License (10 units pack)	A bundle offering of 10 assembled units sold under a special license, likely with usage rights for institutions.
Bulk Order (50+ units)	Discounted pricing for high-volume purchases (e.g., for resellers or large institutions).

Table 5.7: Product Variants Explanation

Product Variant	**Cost (TND)**	**Price (TND)**	**Margin (%)**
Assembled Unit	476.60	950.00	49.8%
DIY Kit	346.60	650.00	46.7%
Educational License (10 units pack price)	4,766.00	7,500.00	36.5%
Bulk Order (50+ units, price per unit)	428.94	760.00	43.6%

Table 5.8: Pricing Strategy (TND)

5.2.3 Market Entry Strategy

5.2.3.1 Phase 1: Proof of Concept (Months 1-6)

- Complete prototype validation and testing

- Develop initial manufacturing partnerships
- Create comprehensive documentation and tutorials
- Launch crowdfunding campaign to gauge market interest

5.2.3.2 Phase 2: Market Introduction (Months 7-18)

- Begin small-scale production (100-500 units)
- Target educational institutions with pilot programs
- Establish online sales channels and distribution partnerships
- Build community around open-source development

5.2.3.3 Phase 3: Scale and Expand (Months 19-36)

- Increase production capacity to 1000+ units/month
- Develop additional product variants and accessories
- Expand to international markets
- Establish regional service centers

5.2.4 Financial Projections

5.2.4.1 Revenue Projections (3-Year)

Revenue Stream	**Year 1 (TND)**	**Year 2 (TND)**	**Year 3 (TND)**
Hardware Sales (Units)	502,500	1,507,500	2,512,500
Kit Sales	167,500	402,000	603,000
Licensing	83,750	251,250	502,500
Support & Services	33,500	117,250	217,750
Accessories	16,750	67,000	150,750
Total Revenue	**804,000**	**2,345,000**	**3,986,500**

Table 5.9: 3-Year Revenue Projections (TND)

5.2.4.2 Cost Structure

Cost Category	Year 1 (TND)	Year 2 (TND)	Year 3 (TND)
Cost of Goods Sold	402,000	1,055,250	1,675,000
R&D Expenses	167,500	251,250	318,250
Marketing & Sales	100,500	234,500	402,000
Operations	83,750	167,500	284,750
Administration	50,250	117,250	184,250
Total Costs	**804,000**	**1,825,750**	**2,864,250**
Net Profit	**0**	**519,250**	**1,122,250**

Table 5.10: 3-Year Cost Structure and Profitability (TND)

5.2.5 Funding Requirements

5.2.5.1 Initial Investment Needs

Investment Category	Amount (TND)
Initial Inventory	251,250
Manufacturing Equipment	167,500
Software Development	134,000
Marketing & Branding	83,750
Working Capital	117,250
Legal & Regulatory	50,250
Contingency (15%)	120,600
Total Funding Required	**924,600**

Table 5.11: Initial Investment Requirements (TND)

5.2.6 Risk Analysis

5.2.6.1 Technical Risks

- **Performance Limitations**: The 1 MHz bandwidth may limit market appeal
- **Component Availability**: Supply chain disruptions could affect production

- **Quality Control**: Maintaining consistent quality in scaled production

5.2.6.2 Market Risks

- **Competition**: Established players may reduce prices
- **Market Adoption**: Educational institutions may be slow to adopt new technology
- **Economic Conditions**: Budget constraints in educational sector

5.2.6.3 Operational Risks

- **Manufacturing Partners**: Dependence on external manufacturers
- **Regulatory Compliance**: Meeting various international standards
- **Intellectual Property**: Potential patent conflicts

5.2.7 Mitigation Strategies

- **Diversified Supply Chain**: Multiple component suppliers to reduce dependency
- **Quality Assurance**: Comprehensive testing protocols and quality management systems
- **Market Education**: Extensive documentation and training programs
- **Strategic Partnerships**: Collaborations with educational institutions and distributors
- **Continuous Innovation**: Regular product updates and feature enhancements

5.2.8 Success Metrics

5.2.8.1 Key Performance Indicators

Metric	Year 1 Target	Year 2 Target	Year 3 Target
Units Sold	500	1,500	2,500
Market Share (%)	2%	5%	8%
Customer Satisfaction	85%	90%	92%
Return Rate (%)	<5%	<3%	<2%
Revenue Growth (%)	-	192%	70%
Gross Margin (%)	25%	28%	30%

Table 5.12: Key Performance Indicators

5.2.9 Conclusion

The EPTScope project presents a compelling business opportunity in the educational and hobbyist electronics market. With its combination of affordability, functionality, and open-source philosophy, it addresses a clear market need while providing sustainable revenue streams. The financial projections indicate strong potential for profitability by Year 2, with reasonable initial investment requirements and manageable risk factors.

The Value Added Network analysis demonstrates that the product creates significant value through each stage of production, from raw materials to final distribution, with a total value addition of 550.00 TND per unit. This comprehensive value chain ensures competitive positioning while maintaining healthy profit margins across different product variants.

The success of this venture depends on effective execution of the market entry strategy, maintaining product quality, and building a strong community around the open-source platform. With proper funding and strategic partnerships, EPTScope has the potential to capture a significant share of the entry-level oscilloscope market while contributing to electronics education worldwide, particularly in Tunisia and similar markets.

GENERAL CONCLUSION

Throughout this project, we have explored and implemented a comprehensive solution to address a real-world technical challenge. From initial research and design to final implementation and testing, each stage of the project has contributed to deepening our understanding of the subject matter and strengthening our practical skills.

The work allowed us to apply theoretical knowledge in a concrete context, reinforcing our competencies in system design, problem-solving, and project management. It also highlighted the importance of interdisciplinary collaboration, rigorous planning, and adaptability when facing unexpected obstacles or constraints.

Despite a few challenges encountered during the process—such as time constraints and simultaneous commitments to other academic tasks—we managed to meet the core objectives of the project. Moreover, this experience has provided valuable lessons that will certainly be beneficial for our future academic and professional endeavors.

We are confident that the results presented in this work reflect our commitment to quality and our ability to undertake complex technical projects. Lastly, we would like to express our sincere gratitude to all those who supported and guided us throughout this journey.

This project marks not only the culmination of months of effort but also a stepping stone toward future innovation and development in this field.

Bibliography

[1] STMicroelectronics, *STM32F103 Reference Manual*, RM0008, Rev 21, 2021. Available at: https://www.st.com/resource/en/reference_manual/CD00171190.pdf

[2] STMicroelectronics, *STM32Cube HAL and LL Drivers User Manual*, UM1884, 2023. Available at: https://www.st.com/en/embedded-software/stm32cubef1.html

[3] Summerfield, M., *Rapid GUI Programming with Python and Qt*, Prentice Hall, 2007.

[4] Texas Instruments, *Understanding ADCs: Resolution and Accuracy*, Application Report, 2018.

[5] Horowitz, P. and Hill, W., *The Art of Electronics*, 3rd Edition, Cambridge University Press, 2015.

[6] Lipscomb, A., *Embedded Systems: Real-Time Operating Systems for ARM Cortex-M Microcontrollers*, Jonathan Valvano, 2020.

[7] Smith, S. W., *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, 1997.

[8] Axelson, J., *USB Complete: The Developer's Guide*, 5th Edition, Lakeview Research, 2015.