TECHNISCHE UNIVERSITÄT
CHEMNITZ

Department of Electrical Engineering and Information Technology

Chair of Measurement and Sensor Technology

Project Documentation

„Project Lab Embedded Systems "

Guided by:	Dr. Ahmed Yahia Kallel
Group:	8
Members:	Taha Yassine Akermi
	Fabian Schuh
	Ibrahim El Khalil Chaida
Project:	**Real-Time Embedded System for Signal Optimization in Battery Analysis**
Date:	**02.07.2025**

Table of Content

1. Abstract

Accurate, real-time monitoring of battery impedance is essential for applications ranging from electric vehicles to portable electronics. This work presents the design and implementation of a modular embedded platform that combines a Howland voltage-controlled current source (VCCS), precision differential amplifiers, and an STM32 microcontroller with dual synchronised ADC channels to acquire bidirectional current and voltage signals. On-chip digital filtering and offset correction are performed in real time, and processed data are streamed via UART to a Raspberry Pi for visualisation, logging, and impedance computation. AC sweep measurements (10 mHz–10 MHz) on shunt resistances of 10 - 50 mΩ demonstrate a flat impedance response around the target R_1 values, with ±1 % magnitude and phase accuracy maintained up to ~158 Hz and a −3 dB bandwidth spanning 290 Hz to 82 MHz. Bode and Nyquist analyses confirm the system's fidelity for electrochemical impedance spectroscopy. The highly integrated, scalable architecture provides a foundation for advanced battery characterisation techniques and can be adapted to diverse chemistries and capacities in future iterations.

2. Introduction :

Modern battery-powered systems from electric vehicles to portable electronics require accurate, real-time monitoring of key battery parameters such as current, voltage, internal resistance, and state-of-charge (SoC)(1,13). Traditional battery monitoring circuits often suffer from limited bandwidth, insufficient resolution, or slow response to dynamic load conditions. In applications like rapid charging or diagnostic testing, these limitations can significantly affect system reliability and safety.

This project aims to develop a real-time embedded system that demonstrates optimised signal processing techniques for either bioimpedance analysis or battery characterisation. The system will acquire signals, perform real-time optimisation and analysis, and display results through an interactive demonstration platform(12).

The aim is to develop a real-time embedded system for accurate signal acquisition and optimisation in battery analysis. The goal is to sense bidirectional current flowing through a battery with high precision, filter and calibrate the acquired signals in real time, and make the data available for further processing, visualisation, and impedance profiling(12).

The system is built around a Voltage Controlled Current Source (VCCS) based on a Howland current pump, which injects precise current signals into the battery. A precision low-ohm shunt resistor and differential amplifier condition the measured current for digitisation. An STM32 microcontroller handles signal acquisition using dual synchronised ADCs, with digital filtering and offset correction performed in real time. The processed data are streamed to a Raspberry Pi dashboard for visualisation and logging(12).

This embedded platform provides the foundation for advanced battery characterisation techniques, including impedance spectroscopy, internal resistance profiling, and cycle performance evaluation. The modularity of the system allows it to be extended to various battery chemistries and capacities in future iterations.

3. System architecture

This section outlines the high-level system design used to measure battery impedance and optimise signal processing through a real-time embedded setup. The architecture leverages a hybrid analog - digital signal chain that combines precision analog front-end circuits with STM32-based real-time acquisition and Raspberry Pi-based post-processing and visualisation.

3.1 Block diagram

The system architecture follows a modular design approach, with each block representing a key subsystem in the signal chain. The overall signal path is as follows:

- **DAC (STM32)**: Generates programmable excitation waveforms (e.g., sine waves or multisine) via DMA-driven LUTs.
- **Howland VCCS**: Converts the DAC voltage into a precise ac current for excitation, independent of battery impedance.

- **Battery/Shunt**: The battery is the device under test; a precision shunt resistor is used to measure injected current.
- **Differential amplifiers**: Used to sense both the battery voltage and the shunt voltage with high CMRR and precision (ADA4522).
- **STM32 ADC**: Samples both signals in synchrony via dual ADC channels using DMA.
- **SPI interface**: Transfers captured waveform data to a Raspberry Pi for post-processing, visualisation, and logging.

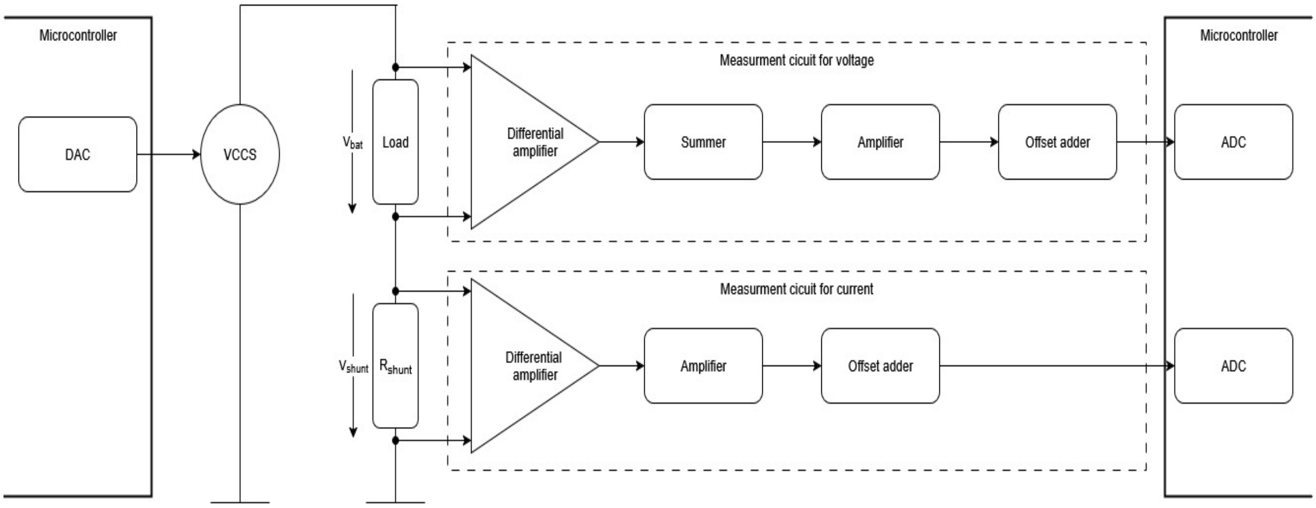

Figure 1: diagram

3.2 Signal Flow

STM32 DAC → VCCS (Howland) → Battery + Shunt → Differential amplifiers → STM32 ADC → SPI → Raspberry Pi

The signal flow describes the timing and communication between system elements:

1. **Excitation generation**
   - A DAC output is triggered periodically by a timer on the STM32 microcontroller.
   - The waveform (e.g., sine, multisine) is stored in memory and streamed to the DAC via DMA.
   - Output frequency is typically in the range of 1 Hz to 10 kHz for impedance analysis.
2. **Current injection via VCCS**
   - The DAC signal is converted into a current using a Howland Current Source circuit.
   - The current is injected into the battery terminal through a known shunt resistor.
3. **Signal sensing**
   - Two differential amplifier stages are used:
     - One measures voltage across the battery.
     - The other measures the voltage across the shunt.
   - Both signals are filtered and level-shifted to match the ADC input range.
4. **Acquisition and buffering**
   - The STM32 ADC samples the two analog channels simultaneously.
   - DMA is configured to transfer data to memory without CPU interruption.
   - Once a full buffer is ready, the STM32 sends a signal to the Raspberry Pi.
5. **Data transmission & Processing**
   - The STM32 sends the buffer over SPI.
   - The Raspberry Pi receives and stores the data..

3.3 Hardware/Software partitioning

Table 1 – Hardware/Software Partitioning

Function	Hardware component	Software role
Waveform Generation	STM32 DAC	Timer-triggered LUT via DMA for continuous excitation
Current Injection	Howland VCCS	Analog current source driven by DAC
Voltage Sensing	Differential Amplifier (ADA4522)	Precision voltage sensing with high CMRR and filtering
Current Sensing	Shunt + Differential Amplifier	Measures voltage drop across shunt, filters & level shifts
Signal Acquisition	STM32 Dual ADCs	Simultaneous sampling, real-time DMA buffering
Communication	STM32 UART Interface	High-speed data transfer to Raspberry Pi
Processing & Visualization	Raspberry Pi (Python scripts)	Analysis and plotting

This partition ensures that real-time acquisition and waveform generation are handled by STM32, while heavy data processing and visualisation are delegated to the Raspberry Pi, enabling a scalable and responsive architecture.

4. System structure and components

In real-life applications, the LM324 operational amplifier is used. However, for LTspice simulations, the Texas Instruments model of the LM324 is known to be faulty, as it introduces an unwanted DC offset. Therefore, we use the ADA4522 model instead for more accurate simulation results.

4.1 Howland voltage-Controlled current source (VCCS)

4.1.1 Purpose and functionality

The Voltage-Controlled Current Source (VCCS) forms the core of the excitation stage in our battery analysis system. Its purpose is to convert a control voltage (generated from a DAC or waveform source) into a bidirectional current that can be injected through the battery under test. This current allows for real-time measurement of battery impedance and dynamic response.

To achieve this, we implemented a **Howland current pump**, a classic op-amp configuration known for its ability to maintain constant current regardless of load variations, provided proper resistor matching is maintained.

4.1.2 LTspice simulation

Before physical construction, the circuit was modelled and tested in LTspice. The simulated design uses four matched resistors (R10–R13 = 10 kΩ) and a small-signal feedback resistor (R14 = 4 Ω), with additional loading via R16 and a virtual battery circuit (V1 + Rshunt).

- **Input signal**: Sine wave centered at 1.65 V, generated by V5
- **Offset/reference**: 1.65 V DC from V4
- **Power rails**: ±12 V (V3, V2)
- **Load**: 3.7 V battery + 10 mΩ resistor + 100 mΩ shunt

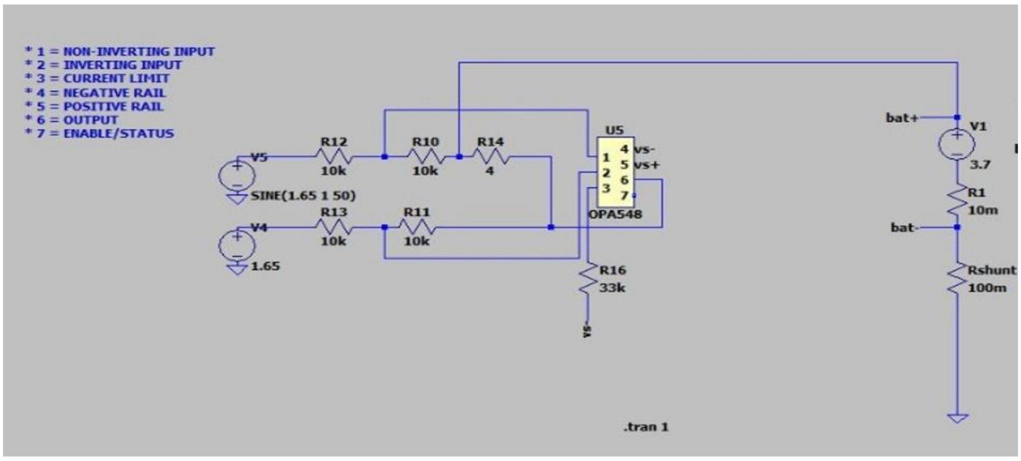

Figure 2 : Howland voltage-Controlled current source

The simulation confirmed linear current control, proper bandwidth (up to several MHz), and full control over current amplitude via the differential input.

4.1.3 Hardware implementation

The simulated circuit was then constructed on a breadboard using an OPA 548 (for initial low-current testing) and 1% tolerance resistors. The power rails were provided by a ±6V lab supply, and the differential input was sourced from a function generator simulating the STM32 DAC output.

4.1.4 Oscilloscope validation

To validate the circuit's dynamic performance, the current waveform across the shunt resistor was captured using a digital oscilloscope. The test input was a sine wave, and the measured frequency response confirmed high-speed capability.

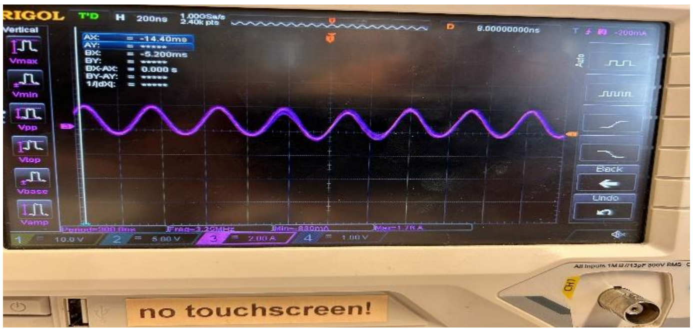

Figure 2: Validation of the Howland circuit

- **Frequency**: ~3.25 MHz
- **Amplitude**: Ranged from –0.88 A to +1.76 A
- **Period**: ~308 ns
- **Shape**: Clean sine wave with minor offset error (to be corrected digitally)

These results confirm that the analog front-end can deliver high-speed, accurate current suitable for real-time battery impedance analysis.

4.2 Voltage measurement

4.2.1 Stage 1: Inverting differential amplifier

To measure the differential voltage across the battery terminals with high precision and reject common-mode noise, the first stage employs an **inverting differential amplifier** configuration using the precision op-amp **ADA4522**.

Circuit description:

- The battery's **positive terminal (bat⁺)** is connected through **R3 = 100 kΩ** to the **inverting input** of the op-amp U6.
- The **negative terminal (bat⁻)** is connected through **R5 = 100 kΩ** to the **non-inverting input**, which is biased to ground via **R2 = 100 kΩ**.
- A feedback resistor **R4 = 100 kΩ** connects the output to the inverting input.

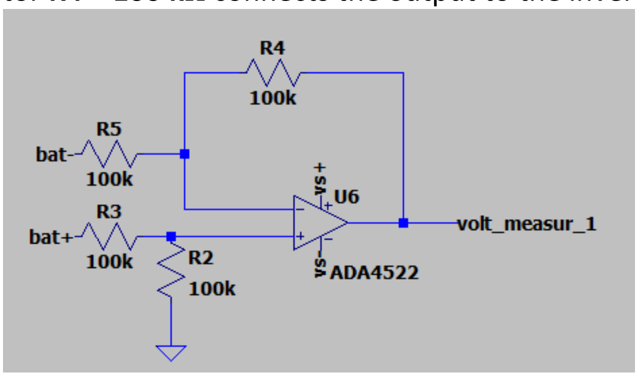

Figure 3 : First stage – Differential voltage measurement circuit

Functionality:

- The amplifier subtracts V(bat–) from V(bat+), amplifying the true voltage across the battery.
- Since all resistors are equal, the gain is 1, and the output volt_measur_1 equals:

$$V_{\text{out}} = V_{\text{bat+}} - V_{\text{bat-}}$$

The gain is unity (–1) due to matched resistors:

$$\text{Gain} = -\frac{R_3}{R_4} = -1$$

4.2.2 Stage 2: Offset subtraction (inverting differential subtractor)

Circuit description

- **Op-amp**: ADA4522, configured as a unity-gain differential amplifier.
- **Resistors**: R15, R17, R18, R16 each = 100 kΩ, forming a symmetrical bridge.
- **Inputs**:
  - **Positive (+) input** receives the signal volt_measur_1 through R15.
  - **Negative (–) input** receives the 3.7 V reference (V_6) through R17 and feedback from the output through R18.
  - The fourth resistor (R16) ties the unused bridge node to ground, ensuring balance.

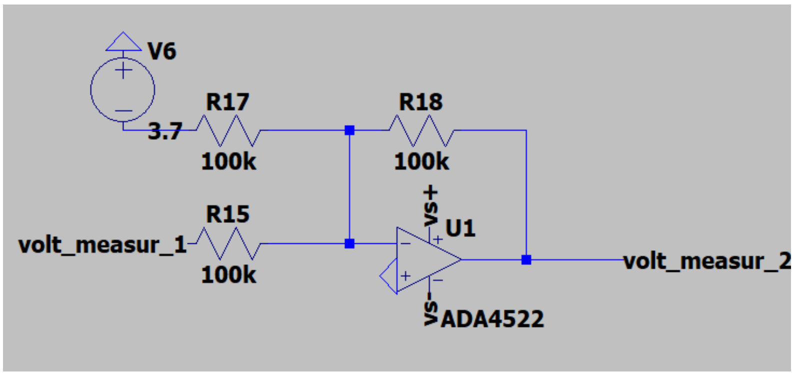

Figure 4 : Offset Subtraction

Functionality

This stage subtracts the fixed reference voltage from the incoming signal without gain:

$$V_{\text{measur\_2}} = -(v_{\text{measur\_1}} - V_{\text{offset}}) = -(\text{volt\_measur\_1} - 3.7\ \text{V})$$

4.2.3 Stage 3: Inverting gain amplifier

Circuit description

- Op-amp: ADA4522 configured in an inverting amplifier topology.

- Resistors:
  - R19 = 1 kΩ (input resistor)
  - R20 = 600 kΩ (feedback resistor)

Functionality

This stage amplifies and inverts the signal volt_measur_2 by a factor of:

$$\text{Gain} = -\frac{R_{20}}{R_{19}} = -\frac{600\ \text{k}\Omega}{1\ \text{k}\Omega} = -600$$

$$V_{\text{measur\_3}} = -600 \times V_{\text{measur\_2}}$$

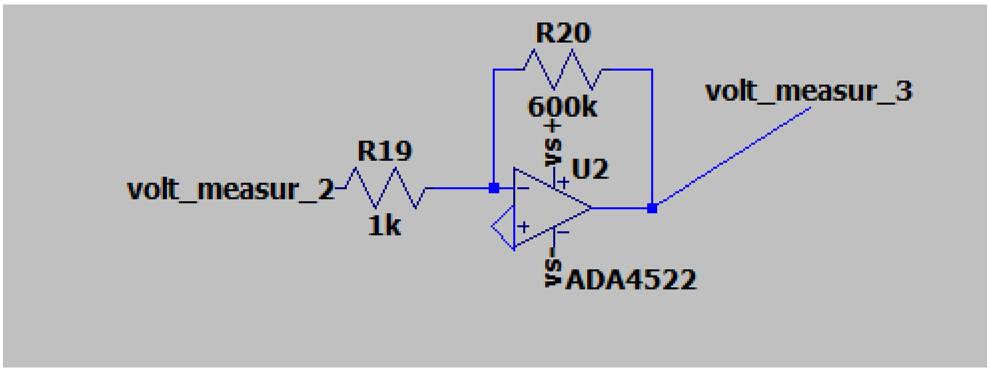

Figure 5 : Inverting gain amplifier

4.2.4 Stage 4: AC coupling & Offset injection

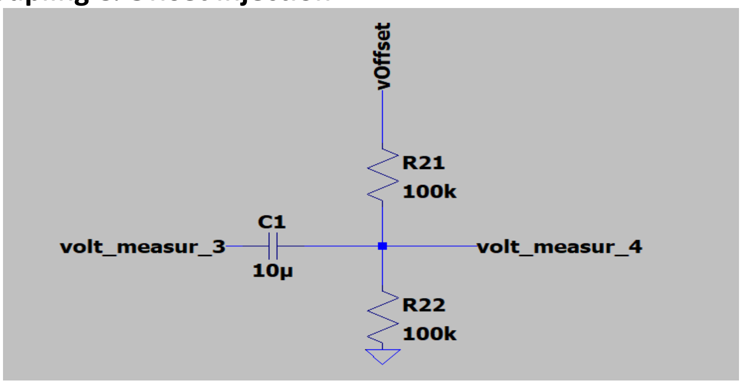

Figure 6: AC coupling & Offset injection

Circuit description

- **C1 (10 µF):** Series coupling capacitor that blocks DC components from the preceding stage.
- **R21, R22 (100 kΩ each):** Form a mid-point voltage divider tied to **Voffset** = 3.3 V and ground. In effect, this injects an offset of 1.65V
- The junction of C1, R21, and R22 provides the final output **volt_measur_4**.

Functionality

It removes any residual DC and then rebases the waveform into the ADC's required input range via offset injection.

Offset Injection:

The divider sets a new DC bias:

$$V_{\text{measur\_4}}(t) = V_{\text{AC}}(t) + 1/2 * V_{\text{offset}}$$

With $(V_{\text{offset}} = 3.3v)$ the entire waveform now resides within **0 - 3.3 V**, suitable for the STM32 ADC.

4.3 Current sensing

4.3.1 Stage 1: Differential shunt sensing

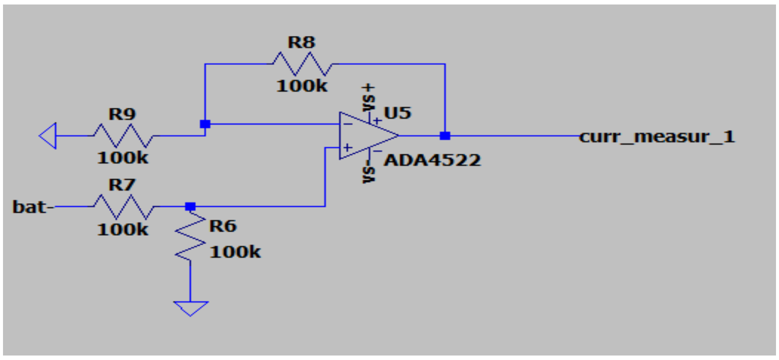

Figure 7: Differential amplifier stage for current measurement (curr_measur_1)

Circuit description:

- **Input terminals:**
  - The inputs are connected across the **shunt resistor** placed in the battery current path (bat+ and bat−).
- **Op-Amp:**
  - ADA4522 – a precision, zero-drift, low-noise op-amp ideal for low-offset current measurements.
- **Resistor network:**
  - R6 = R7 = R8 = R9 = 100 kΩ
  - This matched resistor network ensures **unity gain** and **high common-mode rejection**, ideal for differential sensing.

Functionality:

The circuit in **Figure 7** represents the first stage of current measurement using a differential amplifier based on the ADA4522 op-amp. This configuration is used to sense the voltage drop across a shunt resistor placed in the current path between the battery and the ground, enabling indirect current measurement:

$$V_{\mathrm{out}} = V_{bat-}$$

With equal resistor values, the gain is 1. Thus, the output curr_measur_1 reflects the **exact voltage across the shunt**, which correlates linearly with current via Ohm's law:

$$I_{\mathrm{bat}} = \frac{V_{\text{bat-}}}{R_{\mathrm{shunt}}}$$

This stage is crucial for accurate real-time current sensing in battery analysis systems.

4.3.2 Stage 2: Inverting gain amplifier

Figure 8 shows the second stage in the current measurement chain, which applies a precise gain to the small shunt-sense voltage.

Circuit description

- **Op-amp:** ADA4522 configured in an inverting topology.
- **Resistors:**
  - **R23 = 1 kΩ** (input resistor, from curr_measur_1)
  - **R24 = 60 kΩ** (feedback resistor, from output curr_measur_2 back to the inverting input)

Functionality

This stage multiplies and inverts the incoming shunt voltage by the ratio of the feedback and input resistors:

$$\mathrm{Gain} = -\frac{R_{24}}{R_{23}} = -\frac{60\ \mathrm{k\Omega}}{1\ \mathrm{k\Omega}} = -60$$

Hence, the output signal is:

$$V_{\mathrm{curr\_measur\_2}} = -60 \times V_{\mathrm{curr\_measur\_1}}$$

This 60× amplification brings the millivolt-level shunt signal into the volts, making it suitable for the ADC's dynamic range while preserving polarity information via the inversion.

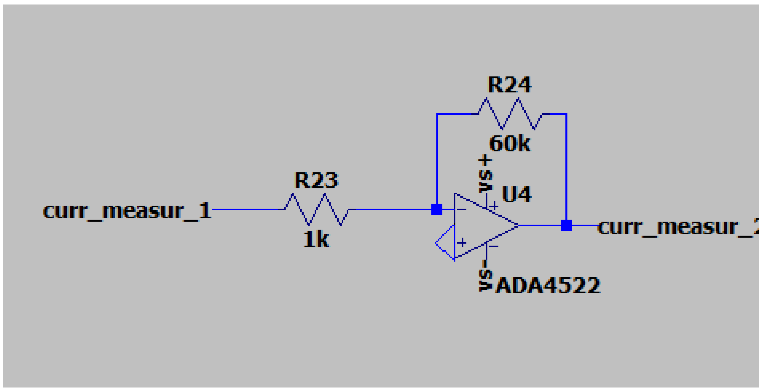

Figure 8: Current measurement stage using a differential amplifier

4.3.3 Stage 3 - AC coupling and offset biasing for current signal

The final stage in the current measurement chain applies **AC coupling** and introduces a **DC offset** to center the signal within the ADC input range of the STM32.

- Capacitor **C3 (10 µF)** blocks the DC component from the previous stage and passes only the AC component of the signal (curr_measur_2).
- The **voltage divider** formed by **R25 = R26 = 100kΩ** biases the AC signal around $V_{\text{offset}} = 3.3\ \text{V}$, the mid-supply reference.
- The resulting signal at curr_measur_3 becomes:

$$V_{\text{curr\_measur\_3}}(t) = V_{\text{AC}}(t) + 1/2 * V_{\text{offset}}$$

This ensures the amplified current waveform remains within the ADC's acceptable input range, avoiding negative voltages or clipping in a single-supply system.

5. Signal analysis & Impedance characterisation

5.1 Measurement setup

To analyse the battery's impedance behaviour across a wide frequency spectrum, an **AC sweep test** was performed in LTSpice. A sinusoidal waveform was generated simulating the **STM32 DAC** with a voltage source, amplified through the **Howland Current Source (VCCS)**, and injected into the simulated battery and then to ground via a precision shunt resistor. The **voltage across the battery** and the **current through the shunt** were simultaneously sampled using the measurement circuit.

The frequency sweep was logarithmic, ranging from **10 mHz to 10 MHz**, allowing resolution of both low-frequency and high-frequency impedance components.

5.2 Bode plot - Magnitude

Figure 9 illustrates the Bode magnitude plot of the measured battery impedance.

- The midband region displays a flat response of around **–38 dB**, corresponding to a resistance of approximately **12.6 mΩ**.
- At frequencies above ~10 kHz, the impedance decreases due to capacitive parasitic
- effects and analog bandwidth limits.

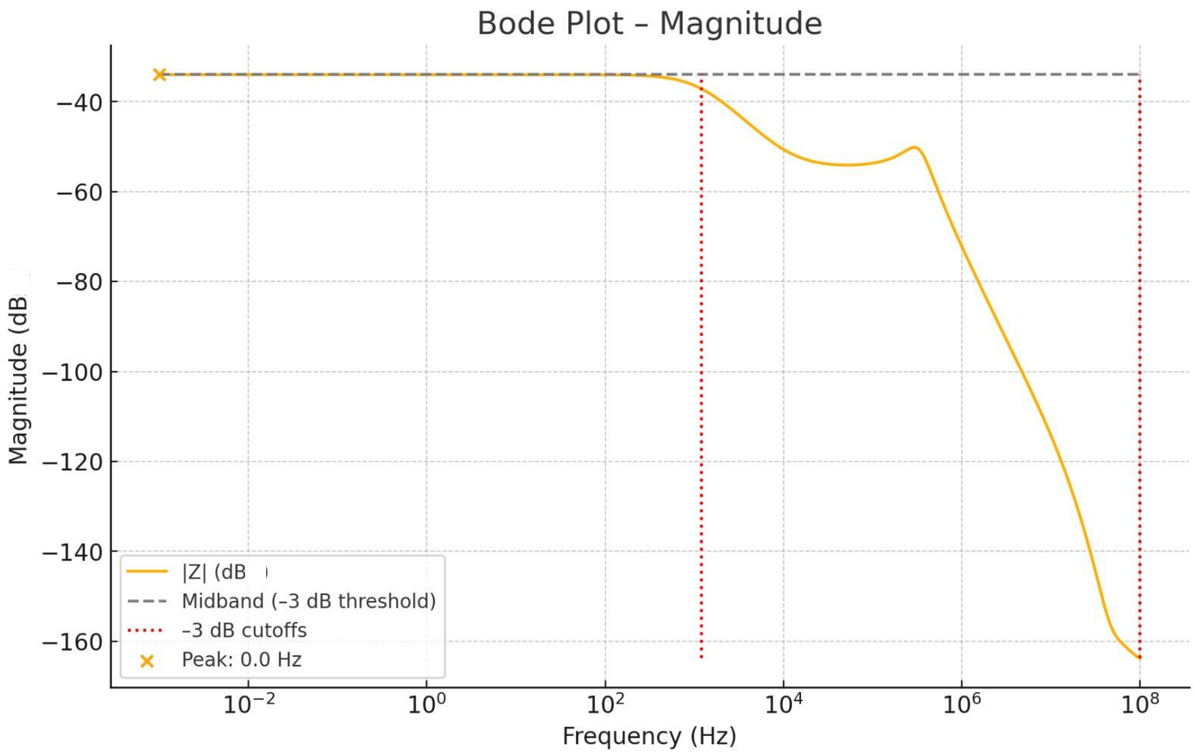

Figure 9: Bode plot - Magnitude

5.3 Bode plot - Phase

Figure 10 presents the phase response of the impedance.

- Initially, the phase remains steady, but it starts to shift negatively at higher frequencies, indicating inductive and capacitive behaviour within the system.
- Rapid phase transitions at around the MHz range suggest reactive dynamics and signal chain limitations.

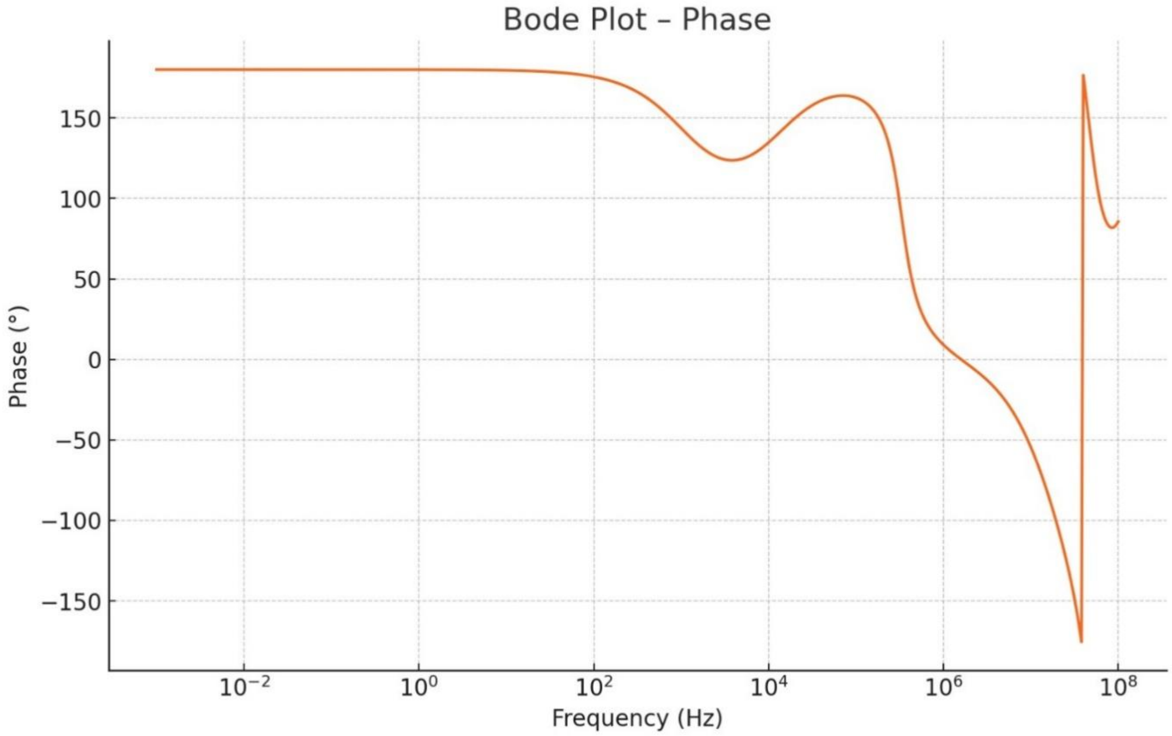

Figure 10: Bode plot - Phase

5.4 Bandwidth determination

To determine the bandwidth in the Spice simulation, the impedance spectrum should be compared to the expected value, which is R1. The maximum error was set as 1%. Thus, the voltages at the nodes 'volt_measur_4' and 'curr_measur_3' were used to calculate the impedance spectrum via an ac sweep. For the calculation itself, the formula

$$R = \frac{V_{bat}}{I} = \frac{V_{bat}}{\frac{V_{shunt}}{R_{shunt}}} = \frac{V_{bat}}{\frac{V_{shunt}}{0.1}} \qquad (1)$$

was used.

An additional factor of 0.1 had to be added to account for the different gain in the amplifiers for the voltage and the current measurement for a final formula of

$$R = \frac{V_{bat}}{V_{shunt}*100} \qquad (2)$$

.

The circuit was then simulated with different values for R1, which represents the targeted battery impedance. Then, a linear bode plot for the calculation of R1 using (2) was generated.

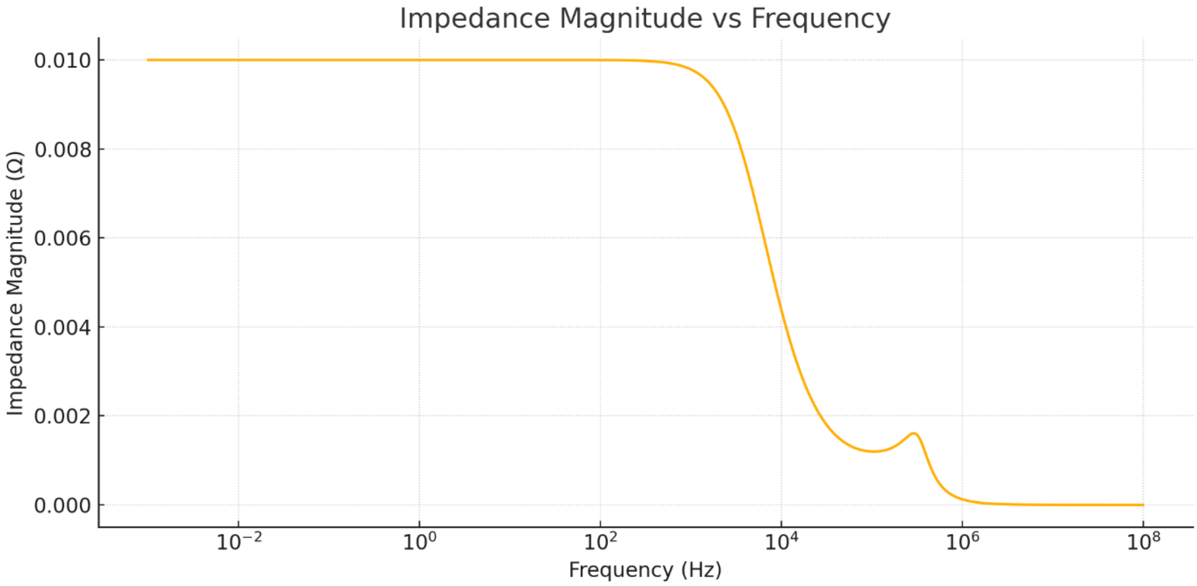

Figure 11: Linear bodeplot(Magnitude) for R1 = 10 mΩ

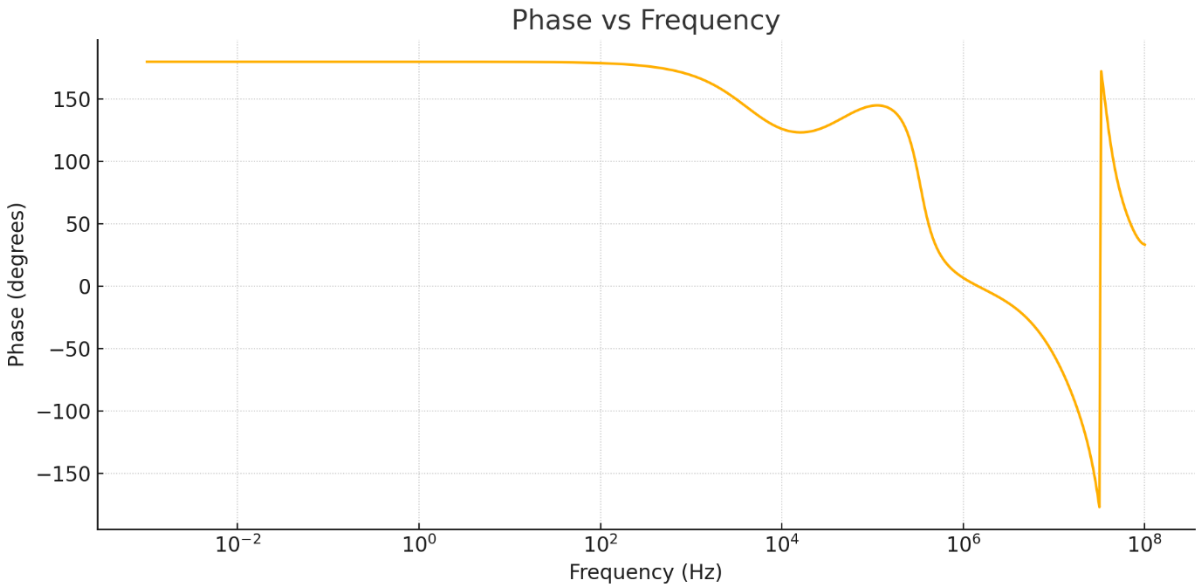

Figure 12: Linear bodeplot(Phase) for R1 = 10mΩ

This was done for simulated impedances of R1 = 10mΩ, = 20mΩ, = 30mΩ, = 40mΩ, = 50mΩ. The associated data was then analysed to find the cutoff frequency of the circuit, which is the point where either the phase or the magnitude develops an error bigger than 1%. The expected Phase is 180 °.

Table 2 : Based on these results, we can define the bandwidth of the system as 158 Hz, ranging from ~0Hz to 158 Hz

Value of R1 (Ω)	Magnitude Error (Ω)	Frequency Before 1st Magnitude Error Exceedance (Hz)	Phase Error (°)	Frequency Before 1st Phase Error Exceedance (Hz)
10m	0.1m	660	1.8	158
20m	0.2m	660	1.8	158
30m	0.3m	660	1.8	158
40m	0.4m	660	1.8	158
50m	0.5m	660	1.8	158

5.6 Linearity verification

To prove the linearity of a system, several conditions must be met. These are the principles of additivity
S[x1(t) + x2(t)] = S[x1(t)] + S[x2(t)]
and homogeneity
S[a · x(t)] = a · S[x(t)].
To determine the linearity of the system of this project, several simulations were done in LTSpice.
Proving homogeneity was achieved by simulating the circuit, once with an input amplitude of 1 V and once with an input amplitude of 3V. The resulting Voltages at the nodes 'curr_measur_3' and 'volt_measur_4' for the 1V simulation were then multiplied by 3 and compared to the simulation results for 3V.

Table 3: homogeneity of the circuit

Output Channel	Sweep Voltage	Slope *m*	R^2
V_1 (volt_measur)	1 V	1.49853	0.99901
	3 V	1.49701	0.99807
V_2 (curr_measur)	1 V	-1.49859	0.99888
	3 V	-1.49696	0.99782

The linearity simulations demonstrate that both voltage outputs scale almost perfectly with the source across the 1 V and 3 V ranges. In the direct-voltage channel (‘volt_measur_4’), the gain remains at ≈ 1.498 with only a 0.10 % shift between sweeps, while the shunt-derived channel (‘curr_measur_3’) shows a negative slope of ≈ - 1.498, reflecting the inversion of the current-sense measurement, with a 0.11 % difference. Coefficients of determination (R^2) between

0.9978 and 0.9990 indicate that over 99.7 % of the output variation is explained by the linear model, leaving negligible non-linear error. These metrics confirm that the system is both highly homogeneous - tripling the input produces a 3× change in each output - and that its residual non-linearity is minimal, ensuring reliable performance across the operating range.

To demonstrate the principle of additivity, an additional measurement was conducted; this time with a voltage source of 4V. When compared to the simulation results of the voltage sources with 1V and 3V, a very marginal error of just 0.02% could be observed when applying the principle of additivity. This comparison can be seen in Table 4.

Table 4 : Additivity of the circuit

Channel	Amp. @1 V (V)	Amp. @3 V (V)	Predicted @4 V (V)	Measured @4 V (V)	Δ (V)	Δ (%)
volt_measur	1.50722	4.52064	6.02786	6.02691	0.000157	0.0157
curr_measur	1.50721	4.52088	6.02810	6.02690	0.000199	0.0199

The measurement front-end demonstrates both homogeneity - output scaling proportional to input within 0.11 % - and additivity - the 4 V response matching the sum of the 1 V and 3 V responses within 0.02 %. These results confirm that the circuit operates as a linear system over the tested range.

6. STM32 microcontroller integration

The STM32G474RET6 microcontroller plays a central role in our real-time data acquisition and signal forwarding system. This high-performance MCU is part of STMicroelectronics' STM32G4 series, which is optimised for mixed-signal applications and advanced analog features.

6.1 Core functionality

The firmware running on the STM32 is designed to transform it into a robust front-end for analog signal acquisition. It supports the following functions:

- **Real-Time Signal Acquisition**: The microcontroller samples analog inputs from ADC1 and ADC2 with precise timing. It uses Direct Memory Access (DMA) to ensure continuous sampling with minimal CPU load.
- **Data Forwarding over UART**: Sampled data is packaged into buffers and transmitted via the USART1 interface to a connected Raspberry Pi. This offloads processing to the Pi and allows the STM32 to focus on acquisition.
- **Performance Optimisation**: DMA and interrupt-driven logic minimise the use of CPU cycles, enabling high-throughput and deterministic behaviour.
- **User Interaction and Telemetry**: A push-button and onboard LED are used for basic user interaction. The firmware also includes support for heartbeat/ status messages via UART.
- **Robust Error Handling**: HAL-based error callbacks and assert functions ensure that the system halts safely in case of configuration or runtime faults.

6.2 Libraries and middleware used

- **CMSIS Startup and Core Files**: startup_stm32g474retx.s, system_stm32g4xx.c
- **STM32 HAL Drivers**:
  - ADC1 & ADC2 (independent mode, DMA-driven)
  - DAC1 (timer-triggered, DMA-driven)
  - DMA (Channels 1, 2, and 3)
  - TIM2 (used as trigger source)
  - USART1 (UART communication in FIFO mode)
- **BSP (Board Support Package)** for the NUCLEO-G474RE board:
  - BSP_LED_Init, BSP_LED_Toggle – for the user LED
  - BSP_PB_Init, BSP_PB_Callback – for the user push button
  - BSP_COM_Init – simplified UART configuration
- **C Math Library**: Used to generate sine waveform lookup tables.
- Other GPIO ports (Pxx) are initialised for the clock but not used directly in the signal chain

Table 5: STM32 Pin Configuration and Peripheral Functions

Pin	Peripheral	Function
PA0	ADC1_IN1	Analog input (Channel 1)
PA1	ADC2_IN2	Analog input (Channel 2)
PA4	DAC1_OUT1	Analog output (e.g., sine waveform)
PC4	USART1_TX	UART Transmit
PC5	USART1_RX	UART Receive

.

6.3 System configuration in STM32CubeIDE

- System Core: DMA, GPIO, NVIC, RCC, SYS
- Analog: ADC1, ADC2, DAC1
- Timers: TIM2
- Connectivity: USART1
- Board Support: NUCLEO-G474RE BSP

6.4 Summary

In this system, the STM32G474RET6 acts as a dedicated data acquisition module. It continuously samples analog channels, buffers data using DMA, and sends it over UART to a Raspberry Pi. The microcontroller remains focused on time-critical acquisition tasks, while the Raspberry Pi handles high-level signal processing, visualisation, and control. This division of labor ensures a robust and scalable architecture.

7. Web-Based visualisation using Raspberry Pi

7.1 Overview

To enable real-time visualisation and data interaction, a **Raspberry Pi** was integrated into the system as a lightweight web server and data dashboard platform. The Raspberry Pi processes incoming ADC samples from the STM32 microcontroller, computes the impedance, and renders interactive visualisations in the browser. The web interface is built using **Flask (v3.1.1)** and **pandas (v2.3.0)**, along with front-end technologies like **HTML5**, **CSS3**, and **JavaScript (Chart.js)**. This provides a portable and low-cost solution for embedded signal monitoring.

7.2 Project structure

The software running on the Raspberry Pi includes the following key files:

- **App.py**: The main Python Flask application. It loads the simulated measurement data, cleans and parses it using pandas, calculates impedance as:

$$Z = \frac{V_{\text{volt4}}}{V_{\text{curr3}}} \div 100$$

- It serves two routes:
  - / for rendering the dashboard/splash page using index.html.

- /data for delivering the parsed data as JSON to the frontend.

- **Measurement.txt**: A tab-delimited data export from LTSpice (used due to limited hardware access), including voltage and current measurements. It forms the backend source for all impedance calculations and chart rendering.
- **Templates/index.html**: A single Jinja2 template that defines the user interface. It initially displays a splash screen with the project title, supervisor, and team names. Clicking the "VIEW MEASUREMENTS" button transitions the user to the dashboard page.
- **Static/styles.css**: Manages visual appearance — background video scaling, font styles, button layout, and grid layout for the charts.
- **Static/video.mp4**: A muted, looping full-screen video background to enhance the visual aesthetics of the dashboard.
- **Env/**: The local Python virtual environment containing installed packages (Flask, pandas, etc.).

7.3 Front-End features

Splash screen (first page)

- Displays project name: *"Real-Time Embedded System for Signal Optimization in Battery Analysis"*
- Supervisor and student names are overlayed on top of a muted looping video.
- A button labelled "VIEW MEASUREMENTS" is at the bottom.

Dashboard (second page)

- A **2×4 grid** containing **eight animated Chart.js line plots**:
  - Seven channels for internal measurements.
  - One computed impedance plot.
- Data points are drawn with a 100 ms interval between samples, creating a real-time animation effect.

7.4 Data pipeline summary

1. STM32 streams ADC data via SPI.
2. Raspberry Pi receives and stores the data in memory.
3. Impedance is calculated using:

$$Z = \frac{V_{\text{volt4}}}{V_{\text{curr3}}} \div 100$$

4. Flask backend sends this data as JSON.
5. JavaScript front-end fetches it and plots live graphs.

This visualisation layer provides engineers with a responsive, low-latency interface to monitor signal quality and validate system performance, especially useful during testing and calibration stages.

8. Conclusion and future works

The objective of this project was to develop a real-time embedded system for signal optimisation in battery analysis based on the STM32 Nucleo-G474RE platform.

During the project, a system was designed that injects a current into a battery and measures the resulting voltage and current through current sensing. To achieve this, the system was divided into several subsystems.

The first subsystem developed was a voltage-controlled current source (VCCS), implemented using a Howland circuit. This circuit generates the current injected into the battery. The second and third subsystems are the measuring circuits that enable the acquisition of the battery's voltage and current responses to the injected current. These were implemented using several operational amplifier circuits that condition the battery voltage and current signals into levels suitable for the STM32's ADC input range, while maintaining sufficient amplitude for accurate measurement.

These three subsystems were developed and tested with the help of simulations in LTspice. During this process, several issues were detected and resolved. The most significant problem was a faulty simulation model of the LM324 operational amplifier provided on the official Texas Instruments website. To address this, a high-precision operational amplifier, the ADA4522, was used in simulations. With this change, the circuit was confirmed to work as intended, achieving a bandwidth of 158 Hz. The circuit's linearity was also evaluated and verified.

The STM32 microcontroller was programmed to function as a data collection and distribution point. Additionally, the STM32's DAC serves as the AC voltage source controlling the VCCS. Its main function, however, is to record the battery's response to the injected current using its ADCs. The collected data is then transmitted to a Raspberry Pi via UART for further analysis. The Raspberry Pi analyses the data and visualises it on a display for easier user access.

On the hardware side, all components were assembled and the circuits were built. The VCCS functionality was tested successfully, and all STM32 functionalities were implemented and verified. However, the measuring circuits could not be tested or deployed due to time constraints.

In the future, the measuring circuits should be activated and commissioned. Subsequent measurements can then be performed to confirm or refute the simulation results. For example, real-world electrical noise may negatively impact system performance. If any performance or accuracy degradation is observed between simulation and actual hardware, these issues will need to be investigated further and appropriate solutions developed.

Finally, the system's intended application - battery impedance measurement and characterisation analysis - can be performed on test samples. The results should be validated against an established technique to conclusively demonstrate the system's functionality as a real-time embedded system for signal optimisation in battery analysis.

9. References

1 Olfa Kanoun, Ahmed Yahia Kallel (2024, April). *High-performance efficient embedded systems for impedance spectroscopy: Challenges and potentials*. Researchgate. https://www.researchgate.net/publication/380232368_High-performance_efficient_embedded_systems_for_impedance_spectroscopy_Challenges_and_potentials

2 Ahmen Yahia Kallel (2024, October). *Design of Optimized Broadband Excitation Signals for Consistent Impedance Spectroscopy Measurements*. Researchgate. https://www.researchgate.net/publication/389261508_Design_of_Optimized_Broadband_Excitation_Signals_for_Consistent_Impedance_Spectroscopy_Measurements

3 Electronics Tutorials. (n.d.). *Introduction to the amplifier*. Electronics Tutorials. https://www.electronics-tutorials.ws/amplifier/amp_1.html

4 Random Rick (2021, Mai 16). Set up multiple ADCs on STM32 microcontrollers using DMA. Youtube https://www.youtube.com/watch?v=AloHXBk6Bfk

5 Texas Instruments (2023). *An engineer's guide to current sensing* (Rev. B). https://www.ti.com/lit/eb/slyy154b/slyy154b.pdf?ts=1748978714169&ref_url=https%253A%252F%252Fwww.bing.com%252F

6 ControllersTech (2019, November 13). DAC in STM32 || Sine wave || HAL || CubeIDE. Youtube. https://www.youtube.com/watch?v=6Z1L6ox63j0

7 Analog Devices (2025, February). Data Sheet ADA4522-1/ADA4522-2/ADA4522-4. https://www.analog.com/media/en/technical-documentation/data-sheets/ada4522-1_4522-2_4522-4.pdf

8 Neil Dey (2023, February 15). LTspice error: timestep too small. Stackexchange. https://electronics.stackexchange.com/questions/405979/ltspice-error-timestep-too-small

9 Shawn Hymel (2019, September 1). Getting Started with STM32 - Working with ADC and DMA. Digikey. https://www.digikey.de/en/maker/projects/getting-started-with-stm32-working-with-adc-and-dma/f5009db3a3ed4370acaf545a3370c30c

10 Texas Instruments (n.d.). LM324-N. TI. https://www.ti.com/product/de-de/LM324-N#design-tools-simulation

11 ST Microelectronics (n.d.) STM32 Nucleo-64 development board with STM32G474RE MCU, supports Arduino and ST morpho connectivity. ST. https://www.st.com/en/evaluation-tools/nucleo-g474re.html

12 Ahmed Yahia Kallel (n.d.). Real-Time Embedded System for Signal Optimization in Bioimpedance/Battery Analysis. TU Chemnitz. https://www.tu-chemnitz.de/etit/messtech/OPAL/Embedded%20System/PLES2025/PLES25_AYK_Demo2.pdf

13 InnovationNewsNetwork (2024, October 23). Electrochemical impedance spectroscopy method boosts EV battery monitoring. InnovationNewsNetwork. https://www.innovationnewsnetwork.com/electrochemical-impedance-spectroscopy-boosts-ev-battery-safety/52159/