

|  |
| --- |
| Yen Dang |
| Design and Implementation  of Digital Current Loop Device  for Frequency Converter |
|  |
|  |

|  |
| --- |
| Information Technology  2024 |
|  |

VAASAN AMMATTIKORKEAKOULU

UNIVERSITY OF APPLIED SCIENCES

Information Technology

**ABSTRACT**

Author Yen Dang

Title Design and Implementation of Digital Current Loop

Device for Frequency Converter

Year 2024

Language English

Pages 47 + 2 Appendices

Name of Supervisor Jani Ahvonen

In the field of motor technology, frequency converters are used to modify and regulate the motor’s speed. However, these devices are often located far from the motor, requiring signal transfer through an intermediate device. Therefore, this thesis focuses on the design and implementation of a digital current loop device for a frequency converter to serve as this intermediary. In addition, this current loop device has an advanced feature of fast and slow increments or decrements, which is very convenient and user-friendly for users.

The work methods for this project were split into three main stages: designing the circuit schematic and the prototype, developing an Arduino software program and manufacturing the physical Printed Circuit Boards (PCB) for the digital current loop device. The expected inputs such as torque (Nm) and speed (rpm) values were fed into a rotary encoder and were displayed on an LCD I2C screen simultaneously. Then, the digital inputs ranged of 0-50Nm for the torque and 0-3000rpm for the speed were converted into the 0-20mA current loops through DACs DFR0552 connected to the Arduino Nano board. For testing purposes, a 250 Ω resistor was used to simulate the receiver acting as the frequency converter.

In this research, the prototype’s measurement was divided into four main parts: Filter, Digital Signal Processing, Digital Analog Converter (DAC) and Analog to Digital Converter (ADC). These individual circuits on breadboards were verified with the software program simultaneously to achieve the accurate outcomes before integrating them together. However, the poor quality of the breadboards generated some noise when these individual circuits were combined, exceeding the expected current and voltage outputs. Hence, to eliminate the breadboard noise, we proceeded to manufacture the physical PCBs for testing the integrated circuits.

The results from the integrated PCBs were consistent with the individual circuit initially tested on the breadboards. Consequently, the desired torque and speed values showed positive correlations with the corresponding current values ranging from 0–20 mA. Then, the PCBs were manually assembled into two metal boxes connected with each other by using the CAT6 cable to generate the finished digital current loop device. Due to the CAT6 cable contained four twisted pair wires, so we should connect the signal wires in pairs to avoid the capacitive cross-talk issues. Furthermore, to eliminate frequency interference, the proper grounding conductors and shielded cables were very important in high-precision electronic designs, particularly for sensitive analog signals in potentially noisy environments. This assembly process minimized the external interferences and increased the 0-20mA current loop outputs with more accuracy.

Keywords Encoder, LCD I2C, Arduino Nano, DAC, current loop, ADC, digital current loop device, frequency converter

**CONTENTS**

[1 Introduction 8](#_Toc178770959)

[1.1 Function 8](#_Toc178770960)

[1.2 Block Hardware Diagram 8](#_Toc178770961)

[1.3 Flowchart Software Diagram 9](#_Toc178770962)

[2 Requirement Specification 11](#_Toc178770963)

[2.1 Functional Requirements 11](#_Toc178770964)

[2.1.1 Current Control 11](#_Toc178770965)

[2.1.2 LCD Screen 11](#_Toc178770966)

[2.1.3 Response Time 11](#_Toc178770967)

[2.2 Requirements for Interfaces 11](#_Toc178770968)

[2.2.1 Requirements for User Interfaces 11](#_Toc178770969)

[2.2.2 Hardware Interface Requirements 11](#_Toc178770970)

[2.2.3 Software Interface Requirements 11](#_Toc178770971)

[2.2.4 Communication Requirements 12](#_Toc178770972)

[2.3 Other Requirements 12](#_Toc178770973)

[2.3.1 Performance Requirements 12](#_Toc178770974)

[2.3.2 Usability Requirements 12](#_Toc178770975)

[2.3.3 Safety Requirements 12](#_Toc178770976)

[2.3.4 Maintenance Requirements 12](#_Toc178770977)

[2.3.5 Portability and Compatibility Requirements 12](#_Toc178770978)

[2.3.6 Environmental Requirements 13](#_Toc178770979)

[2.4 Restrictions of Design 13](#_Toc178770980)

[2.4.1 Restrictions caused by previously used technologies 13](#_Toc178770981)

[2.4.2 Hardware Limitations 13](#_Toc178770982)

[2.4.3 Software Restrictions 14](#_Toc178770983)

[2.4.4 Other Limitations 14](#_Toc178770984)

[3 Overview of Frequency Converters 15](#_Toc178770985)

[4 Background Theory 16](#_Toc178770986)

[4.1 Digital Signal Processing 16](#_Toc178770987)

[4.1.1 Rotary Encoder 16](#_Toc178770988)

[4.1.2 I2C LCD1602 17](#_Toc178770989)

[4.2 Digital-Analog Converter 18](#_Toc178770990)

[4.3 Current Loop 19](#_Toc178770991)

[4.3.1 Transmitter Circuit 19](#_Toc178770992)

[4.3.2 Feedback Loop Circuit 21](#_Toc178770993)

[5 Schematics and Implementation 24](#_Toc178770994)

[5.1 Schematics 24](#_Toc178770995)

[5.1.1 Filter Circuit 24](#_Toc178770996)

[5.1.2 Main Board Circuit 24](#_Toc178770997)

[5.1.3 EncoderLCD Circuit 26](#_Toc178770998)

[5.2 Software Implementation 26](#_Toc178770999)

[5.3 I2C Communication Protocol 31](#_Toc178771000)

[6 Testing and Troubleshooting The Prototype on Breadboards 32](#_Toc178771001)

[7 Design and Manufacture of Printed Circuit Boards 34](#_Toc178771002)

[7.1 Design PCB layouts 34](#_Toc178771003)

[7.1.1 The Filter’s PCB 34](#_Toc178771004)

[7.1.2 The Main Board’s PCB 34](#_Toc178771005)

[7.1.3 The EncoderLCD’s PCB 35](#_Toc178771006)

[7.2 PCBs Fabrication and Assembly 36](#_Toc178771007)

[8 Testing and Comprative Analysis 39](#_Toc178771008)

[8.1 Testing Finished Current Loop Device with A Real Motor 39](#_Toc178771009)

[8.2 Comparative Analysis of Output Results of Ten Devices and Main Prototype 42](#_Toc178771010)

[9 Conclusion 45](#_Toc178771011)

[List of References 46](#_Toc178771012)

[List of Appendices 48](#_Toc178771013)

**LIST OF TABLES**

[Table 1. Encoder logic state. 17](#_Toc178770900)

[Table 2. Lookup table of the encoder changes. 29](#_Toc178770901)

[Table 3. Error measurement range for actual current values. 42](#_Toc178770902)

**LIST OF FIGURES**

[Figure 1. The hardware block diagram 8](#_Toc178770864)

[Figure 2. The flowchart software diagram 10](#_Toc178770865)

[Figure 3. Quadrature output table. 17](#_Toc178770866)

[Figure 4. The external voltage circuit drawn in LTspice. 19](#_Toc178770867)

[Figure 5. Simulation of the external voltage from the TL431 chip. 19](#_Toc178770868)

[Figure 6. The simplified transmitter circuit drawn in LTspice. 20](#_Toc178770869)

[Figure 7. Simulation of the simplified transmitter circuit. 21](#_Toc178770870)

[Figure 8. The simulating receiver and feedback loop circuits drawn in LTspice. 22](#_Toc178770871)

[Figure 9. Simulation of the simulating receiver and feedback loop circuits. 23](#_Toc178770872)

[Figure 10. The filter circuit’s schematic. 24](#_Toc178770873)

[Figure 11. The main board circuit’s schematic. 25](#_Toc178770874)

[Figure 12. The encoderLCD circuit’s schematic. 26](#_Toc178770875)

[Figure 13. The appropriate libraries. 26](#_Toc178770876)

[Figure 14. The calibrations for ADC and DAC for both channels. 27](#_Toc178770877)

[Figure 15. Updating the torque and speed changes through the function of void write\_dac(float torque, int rpm). 27](#_Toc178770878)

[Figure 16. The state of a push-button. 28](#_Toc178770879)

[Figure 17. The state of the encoder changes as an external interrupt. 28](#_Toc178770880)

[Figure 18. Fast or slow increments featuring for four steps forward. 29](#_Toc178770881)

[Figure 19. Safety checks within defined limits. 30](#_Toc178770882)

[Figure 20. Open loop error check featuring. 30](#_Toc178770883)

[Figure 21. I2C network. 31](#_Toc178770884)

[Figure 21. The main components of the prototype testing on breadboards. 33](#_Toc178770885)

[Figure 23. The filter’s PCB. 34](#_Toc178770886)

[Figure 24. The mainboard’s PCB. 35](#_Toc178770887)

[Figure 25. The encoderLCD’s PCB. 36](#_Toc178770888)

[Figure 26. The assembled physical PCBs. 36](#_Toc178770889)

[Figure 27. The finished current loop device. 37](#_Toc178770890)

[Figure 28. Testing results of the finished current loop device. 38](#_Toc178770891)

[Figure 29. Setup for testing the finished device with the frequency converter. 39](#_Toc178770892)

[Figure 30. Real-time outputs displayed in the NCDrive software. 40](#_Toc178770893)

[Figure 31. EMI shielding tapes applied to the CAT6 cable and EMC conductors grounded. 41](#_Toc178770894)

[Figure 32. Current and voltage diagram in the Oscilloscope. 41](#_Toc178770895)

[Figure 33. Desired torque and real current diagram. 43](#_Toc178770896)

[Figure 34. Desired speed and real current diagram. 43](#_Toc178770897)

[Figure 35. Desired torque and actual current output across all devices. 44](#_Toc178770898)

[Figure 36. Desired speeds and actual current output across all devices. 44](#_Toc178770899)

# Introduction

The project’s objective is to design and implement a digital 0-20mA device for frequency converter. Users could adjust the expected values such as torque and speed values on an LCD by rotating an encoder.

## Function

The digital device regulates the current output to achieve a desire setpoint presenting on the LCD screen. This would be accomplished by using the rotary encoder.

## Block Hardware Diagram

The hardware block diagram of the device is designed as below:

24V DC

5V DC

0-20mA

0-20mA

0-20mA

0-20mA

**MCU**

**Trans-mitter**

**Encoder**

**DC-DC Converter**

**24V/5V**

**Isolation**

**DC Converter**

**24V/24V**

**DAC**

**DAC**

**LCD**

**Trans-mitter**

**Receiver**

**Receiver**

’

**+24V DC**

’

**+24V DC**

**+24V DC**

**+24V DC isolated**

Figure 1. The hardware block diagram

According the diagram, the encoder would be used to set the desired current value by users. The desired digital value would be fed into MCU. Then, MCU would process the data by displaying it on the LCD screen. MCU would send the desired digital value to a DAC. The DAC would convert the digital data back into the analog format. The analog output from DAC would control the current levels in the current loops (transmitter and receiver).

## Flowchart Software Diagram

The device’s program is presented, as follows:

Start

Update LCD

decrease

rpm

increase

rpm

encoder\_value > 3

DAC\_1

Yes

Inititalize variables

No

 decrease torque

Check DAC

DAC\_select = = 0

Reset WDT

increase

torque

Global variables

DAC\_select

torque

rpm

DAC\_2

encoder\_state

DAC\_select = !DAC\_select

External interrupt from Encoder

Return

Initialize variables

encoder\_value

encoder\_state

Yes

No

**Main loop**

**Interrupt**

check error

check error

error = = 0

or

error = = 2

error = = 1

or

error = = 3

error = = 0

or

error = = 1

error = = 1

or

error = = 3

torque

OL

rpm

OL

DAC\_select = = 0

encoder\_value < -3

Yes

No

DAC\_select = = 0

SW\_PIN = = 0 and

millis > timex+200

Yes

No

SW\_PIN = = 0 and

millis > timex+200

DAC\_select = !DAC\_select

Yes

No

Figure 2. The flowchart software diagram

# Requirement Specification

## Functional Requirements

### Current Control

The desired current level is adjusted within the range of 0-20mA by turning the knob of the encoder.

### LCD Screen

The LCD screen should display the desired current level adjusted by users.

### Response Time

The device should have a fast response time to quickly and accurately adjust the current level.

## Requirements for Interfaces

### Requirements for User Interfaces

The device should contain the LCD display and the rotary encoder, which can be used to display and turn a knob to achieve the expected digital values, respectively.

### Hardware Interface Requirements

The device could be rotated the encoder manually to achieve the desired values displaying on the LCD display.

Additionally, the device contains two current loop outputs that can be directly connected to the analog inputs of a frequency converter. The use of current loop outputs enables for precise control and communication between the device and the converter, allowing real-time data transmission for adjusting accurate parameters such as torque and speed.

### Software Interface Requirements

The device should allow users to control torque and speed value setpoints. Moreover, the device should have a function to detect errors when any open loops occur.

### Communication Requirements

The device could be programmed via serial communication through MCU’s USB port. This allows the MCU to transfer data to the computer. The serial communication port is fixed, so there is no need to modify the setting.

## Other Requirements

### Performance Requirements

There are some features of the device’s performance, as follows:

* Resolution of measurement: 0.02mA
* Range of measurement: 0-20mA

### Usability Requirements

The device should have a user-friendly design, without the requirements for any specialized skills.

### Safety Requirements

The device must be protected with a fuse to prevent damage from overcurrent or electrical faults.

### Maintenance Requirements

The maintenance requirements could result in a more robust and user-friendly digital 0-20mA device by monitoring remotely and facilitating efficient troubleshooting and repairs if needed. This approach would help users to ensure its long-term system performance and reliability, reduce the risk of unexpected outputs as well.

### Portability and Compatibility Requirements

Frequency converters play a crucial role in a variety of many fields in daily life such as control systems, power electronics, robotics, and automation, so the digital current loop device would be used with various types of frequency converters. Therefore, both portability and compatibility characteristics should be considered.

* For the portability factor, the designed device should be lightweight and small for ease of installation.
* For the compatibility element, the digital current loop device could be able to communicate with different types of communication interfaces when integrated with a wide range of frequency converters.

### Environmental Requirements

The device must operate within the temperature range from −30°C to +30°C to ensure the desired functionality.

Furthermore, the device must operate its own environment as designed. Moreover, the device must comply with proper EMC regulations to minimize any potential interferences and noise.

## Restrictions of Design

### Restrictions caused by previously used technologies

The device must be very simple, so it can use MCU Arduino Nano A000005 board schematic as a basic of design.

The digital 0-20mA device requires the existing industrial systems and control devices to have the 0-20mA input terminals. This is because in the real world, the 4-20mA current loop[[1]](#footnote-1) has commonly used as the standard for signal transmission and electronic control in many industries.

### Hardware Limitations

The physical device should be as small as possible.

With a 0 mA current signal, it might be difficult to detect the system failures[[2]](#footnote-2).

Moreover, the wire loop distance between the transmitter and the receiver should be within 500 meters.

### Software Restrictions

The software for this project would be written in the C language, so only the GCC C-compilers are permitted for developing this program.

### Other Limitations

The power consumption should be minimized as much as possible while ensuring the whole system has sufficient power to operate smoothly.

In addition, ground loops[[3]](#footnote-3) might be an unpredictable problem to crash the system. This is because when many grounded devices placed in different locations are integrated with each other, which might have different voltage potentials, the noise injection generated from their ground connection might affect grounded devices. Therefore, we assumed all grounded components would be placed the same ground point to prevent the ground loops.

# Overview of Frequency Converters[[4]](#footnote-4)

A frequency converter is known as a variable frequency drive or adjustable speed drive, which is used for modifying and controlling the speed and torque of electric motors. Initially, this device converts the alternating current (AC) voltage into direct current (DC) voltage. Then, it reconverts this DC back into AC with adjustable frequency and the levels of voltage. This process allows for precise control over the motor’s performance, enabling it to operate at the desired torque and rotational speed required for the motor.

In addition, frequency converters can reduce motor power consumption by up to 50%, leading to significant energy savings. Moreover, they can collect motor data such as temperature and vibrations via sensors, enabling condition monitoring and preventive maintenance. This allows for remote operation monitoring and troubleshooting through cloud solutions. Therefore, these advantages make frequency converters essential devices in modern industrial applications, offering improved performance, efficiency and control across a wide range of sectors.

However, frequency converters can create different kinds of interference when assembled directly to motors or connected via cables. This includes acoustic noise, harmonic reactions on the network, and electromagnetic compatibility (EMC) noise, which can affect radio devices and data transfers. Therefore, the installation of frequency converters and motors should address and manage noise and interference by complying with EMC requirements.

# Background Theory

## Digital Signal Processing

In the implementation of the digital current loop device, we manipulate the digital desired inputs from the rotary encoder and then convert them to analog signals for transmission to the frequency converter. To ensure accurate tracking and facilitate real-time adjustments, we have integrated an I2C LCD 1602 display in this project. This display serves as a visual interface, presenting the desired digital input signals in a clear and easily readable format, allowing for quick identification of any discrepancies or issues that may arise during operation. Therefore, this feature is useful for users to monitor the system's performance continuously when there are any changes in the wanted digital inputs from the encoder.

### Rotary Encoder

In the project, the rotary encoder is an incremental encoder, which is PEC11R-4220F-S0024. This encoder converts angular motion or position of a shaft into an analog or digital code to determine position or motion[[5]](#footnote-5). This encoder creates a specified number of pulses per rotation. There are two output channels of pulses such as Channel A and Channel B, which are offset from each other to indicate the direction of rotation. The phase relationship between these channels is known as quadrature. When the encoder is rotated in a clockwise direction (CW), Channel A is leading Channel B in the output signal sequence. In contrast, during counterclockwise (CCW) rotation, the signal sequence reverses, meaning Channel B is leading Channel A. The quadrature output table[[6]](#footnote-6) is represented as below.

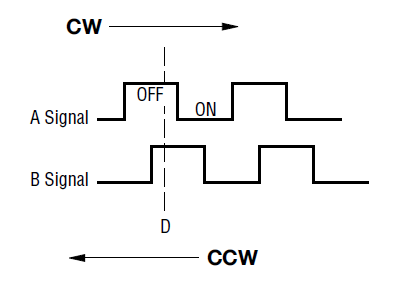


Figure 3. Quadrature output table.

Based on the quadrature output table, we could derive a encoder logic state table, which is illustrated as Table 1.

|  |  |  |
| --- | --- | --- |
| **Clockwise Direction** | | |
| **Position** | **Channel A** | **Channel B** |
| 1 | 0 | 0 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |
| 4 | 0 | 1 |

Table 1. Encoder logic state.

In addition, this encoder is integrated with an additional feature, which is a push switch option. The push button is often used to trigger of specific actions or toggle between different operational modes.

### I2C LCD1602[[7]](#footnote-7)

An I2C LCD1602 is a display device that combines a standard LCD (Liquid Crystal Display) and an I2C (Inter-Integrated Circuit) module attached to its backside. This device could show text and characters on 2 rows with 16 columns. One of the key advantages of the I2C LCD1602 lies in its simplified connection interface. Unlike traditional LCDs that often require multiple pins for operation, this device utilizes only four pins: GND (ground), VCC (power supply), SDA (Serial Data), and SCL (Serial Clock). This configuration simplifies the wiring process.

## Digital-Analog Converter

The Digital-Analog Converter (DAC) is a device that converts a digital input signal into an analog output signal. Specifically, the DAC transforms the binary signal in the form of bits into an analog voltage or current. The DACs find widespread application in various digital signal processing fields, including audio amplifier, video encoder, display electronics, data acquisition systems and motor control.[[8]](#footnote-8)

In this thesis, we utilized two DFRobot Gravity 12-Bit I2C DAC modules equipped with EEPROM (Electrically Erasable Programmable Read-Only Memory) to facilitate the conversion of digital input signals from the encoder into voltage or current signals suitable for the frequency converter. These DAC modules incorporate a 12-bit DAC MCP4725 chip, support a wide input voltage range of 3.3V-5V and include an I2C address selection switch. The integration of EEPROM in these DAC modules significantly enhances data retention and system reliability. This non-volatile memory component could retain the DAC input values while the power is turned-off and resume the DAC output upon power resumption.[[9]](#footnote-9)

Moreover, the input voltage for DAC modules was chosen at 5V. Instead of using the internal voltage of Arduino Nano, we chose for an external voltage circuit, aiming for more stable and precise values for the whole system. Therefore, we used the reference chip of TL431, namely Adjustable precision Zener shunt regulator, to generate a constant output voltage at 5V a reference voltage for both DACs and Arduino Nano. The circuit of the external voltage was designed in LTspice, as follows:

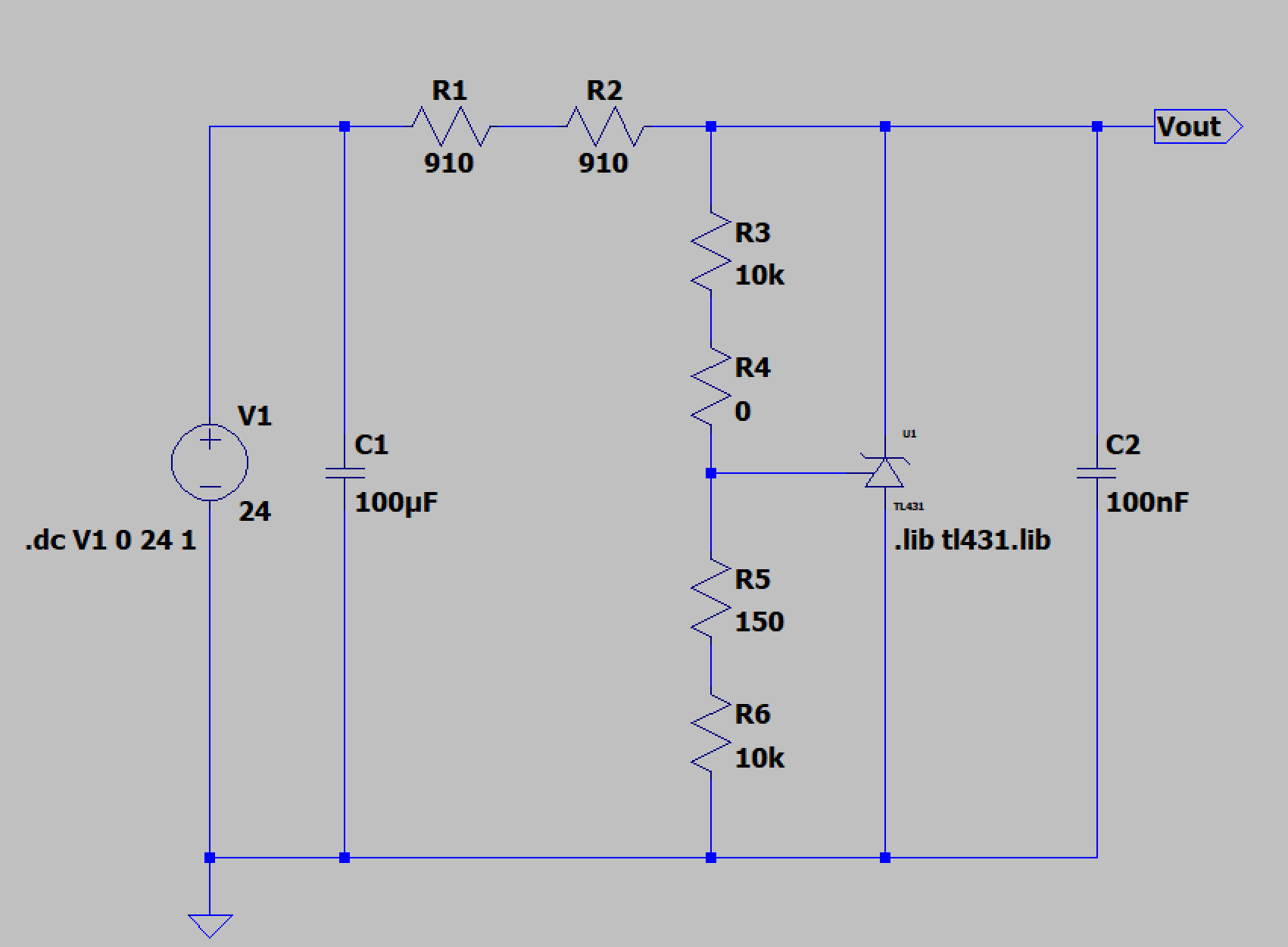


Figure 4. The external voltage circuit drawn in LTspice.

The circuit was then simulated as shown in the picture below. The results indicated that the input voltage of 24V, serving as the voltage source, was successfully converted to the constant output voltage of 5V.



Figure 5. Simulation of the external voltage from the TL431 chip.

## Current Loop

### Transmitter Circuit

After converting the digital input signals to the analog signals, we connected the analog output to a transmitter circuit for converting the input voltage into an output current. The transmitter circuit incorporates two key components: an operational amplifier (LM324) and a transistor (BC547).

The LM324 operational amplifier was utilized for maintaining signal integrity and ensuring that even small voltage changes are accurately represented in the output current. The LM324 series feature cost-effective quad operational amplifiers with true differential inputs, designed for single supply operation within a 3.0–32V range. The amplifier offers low quiescent currents and numerous advantages such as short-circuit protection, true differential input stages. Moreover, the LM324 series also provides low input bias currents, internal compensation, and a common mode range extending to the negative supply. With industry-standard pinouts and ESD (Electrostatic Discharge) protection, these versatile amplifiers suit various applications requiring reliable operation.[[10]](#footnote-10)

Following the amplifying stage, we used the BC547 Bipolar Junction Transistor (BJT), which is used for amplifying current or switching. The transistor has three regions: emitter, base, and collector. By controlling the base current, the transistor can be turned on or off, allowing a larger current to flow between the collector and emitter. This property makes it useful for amplifying small signals or acting as a switch. The BC547 is popular due to its affordability, availability, and ease of use.[[11]](#footnote-11) Hence, this BC547 was used to generate the 0-20mA output current in our device.

Here is the simplified transmitter circuit applied in this project.

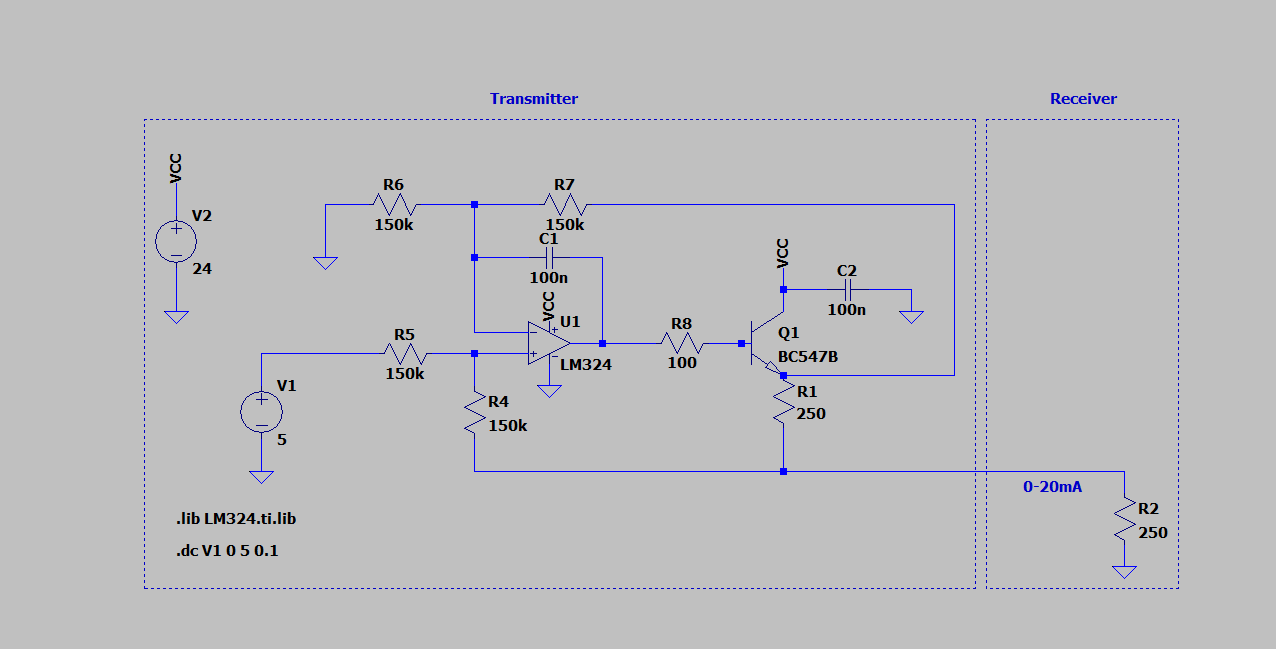


Figure 6. The simplified transmitter circuit drawn in LTspice.

In addition, we also simulated the transmitter circuit above to measure the output current of transmitter and the output voltage of receiver. After simulating, within the input voltage range of 0-5V, the transmitter output current was from 0mA to 20mA and the receiver output voltage was from 0V to 5V. These results were shown in Figure 7 below.

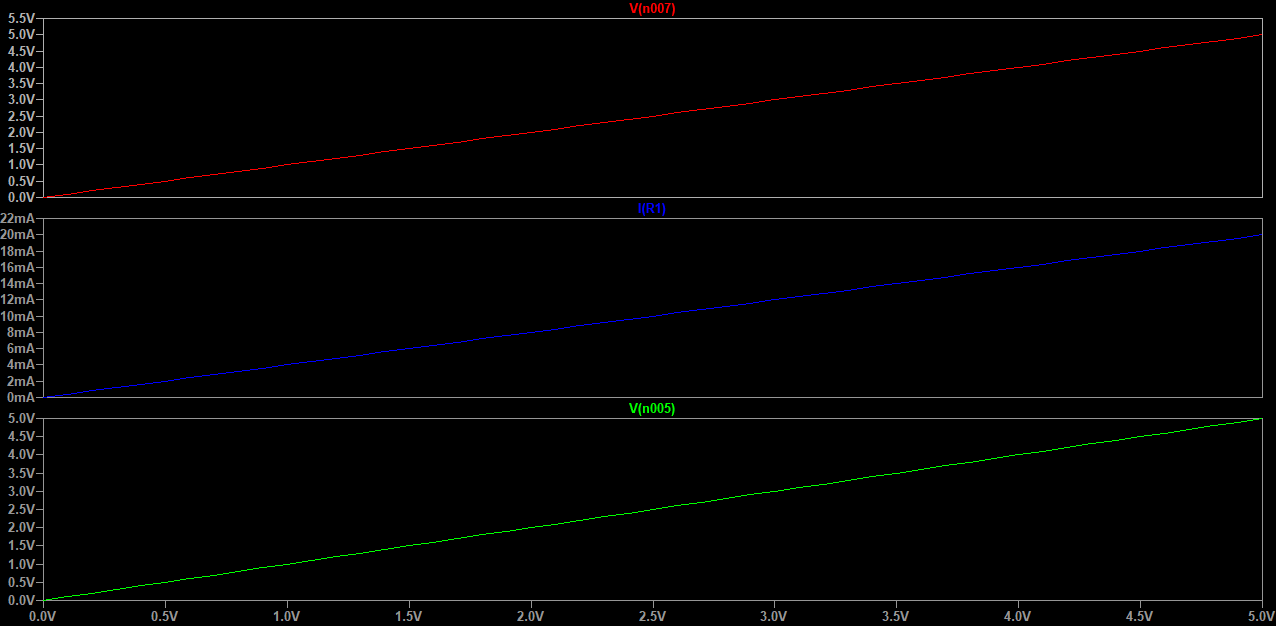


Figure 7. Simulation of the simplified transmitter circuit.

### Feedback Loop Circuit

To complete the current loop, we implemented a feedback loop circuit designed to capture and process the transmitted current signal. This circuit performs the essential function of converting the received current back into its original voltage form, enabling further processing or utilization of the signal. This circuit consisted of two main parts such as a resistor of 250 Ohm simulating frequency converter or receivers and the feedback loop. To be specific, the feedback loop incorporated three main parts: the low pass filter cut-off frequency, two diodes of 1N4148 for ESD protection and a resistor of 250 Ohm.

Firstly, the low pass filter part consisted of a resistor of 10 kOhm, and a capacitor of 100nF for filtering out frequencies above the cut-off point. The cut-off frequency was calculated following the below formula:

This calculated cut-off frequency of approximately 0.16 kHz ensured that the circuit effectively attenuates signals above this threshold, resulting in a more stable output signal.

Secondly, these diodes of 1N4148 acted as a safeguard, protecting the sensitive components of the circuit from potentially damaging voltage spikes that might occur due to static electricity or other sources.

Lastly, the receiver circuit included the resistor of 250 Ohm to control and maintain the targeted current output signals, preserving the 0-20mA current flow transmitted from the transmitter circuit and the receiver.

Here is the simulating receiver (the resistor of 250 Ohm) and feedback loop circuit shown as below.

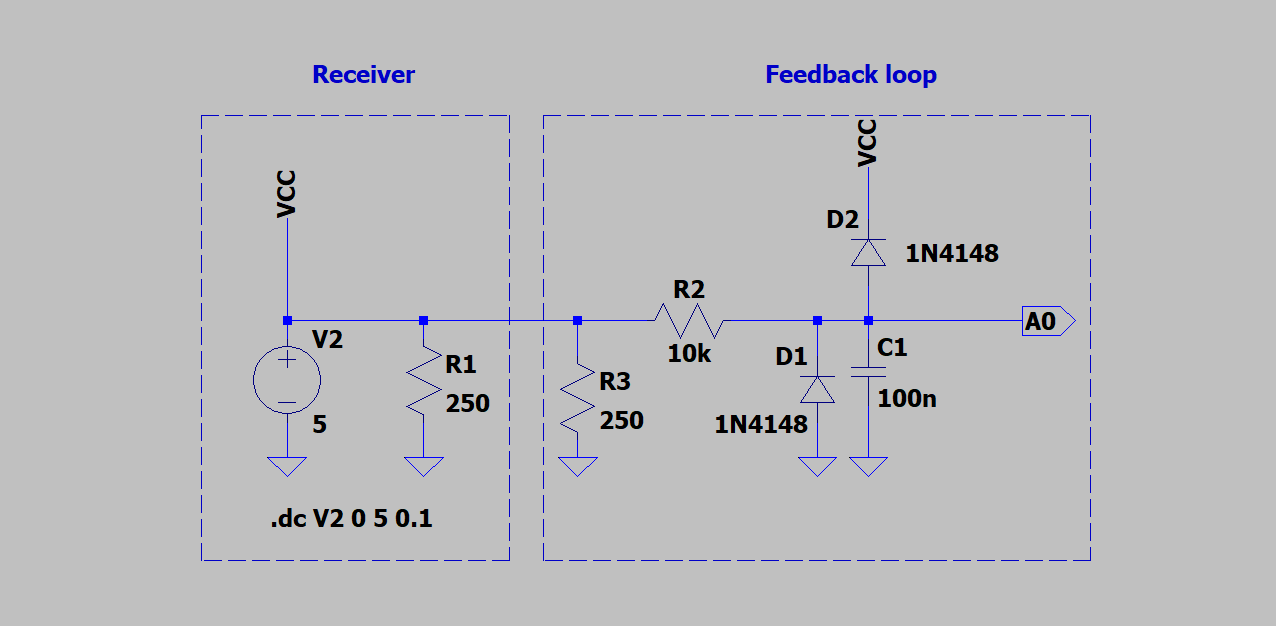


Figure 8. The simulating receiver and feedback loop circuits drawn in LTspice.

The simulation results from this simulating receiver and feedback loop circuits were shown as Figure 9 below. The figure illustrated that the current value going through R1 represented as the receiver or the frequency converter was from 0-20mA. In addition, the 0-20mA current flow was transmitted to the feedback loop. Then, this current flow was converted into the output voltage of 5V.

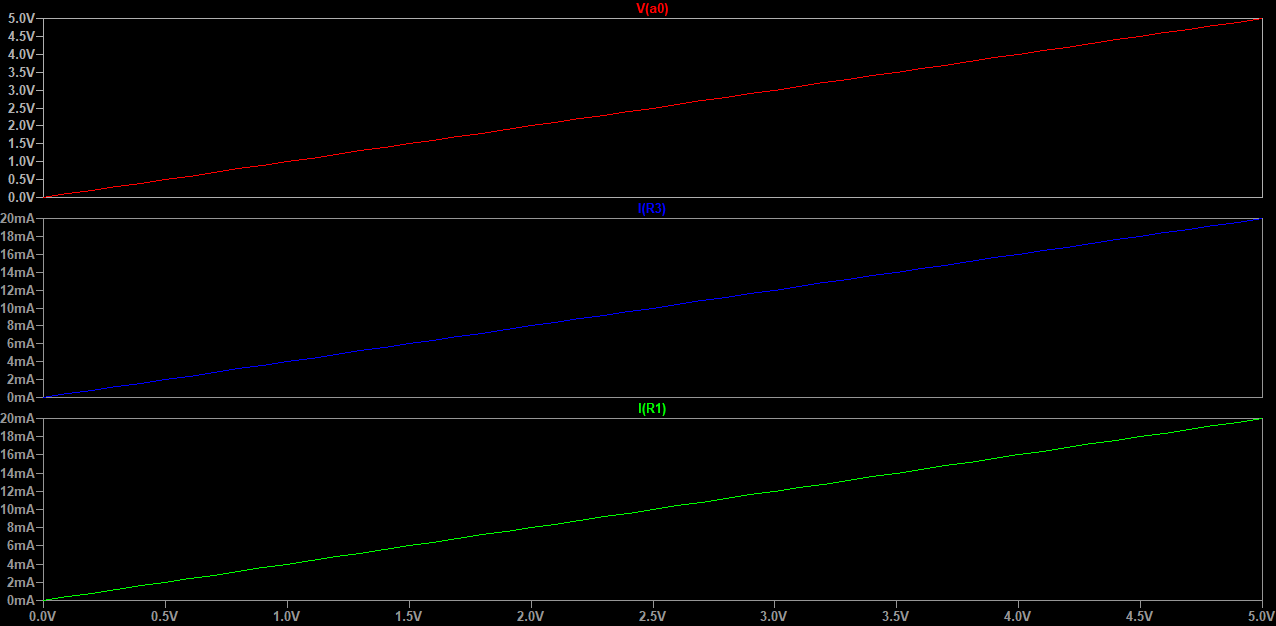


Figure 9. Simulation of the simulating receiver and feedback loop circuits.

# Schematics and Implementation

The project’s schematics were divided into two main parts, including a main board schematic and a digital signal processing schematic. The digital signal processing circuit included the encoder and LCD circuits. The remaining circuits including a filter, external reference voltage, DACs, transmitter and feedback loop circuits connected to Arduino Nano belonged to the main board schematic.

## Schematics

### Filter Circuit

The filter circuit’s schematic was designed as below:

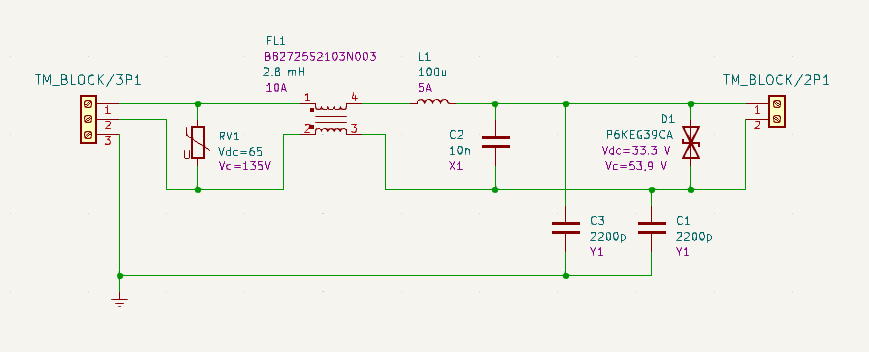


Figure 10. The filter circuit’s schematic.

### Main Board Circuit

The main board circuit’s schematic was designed as below:

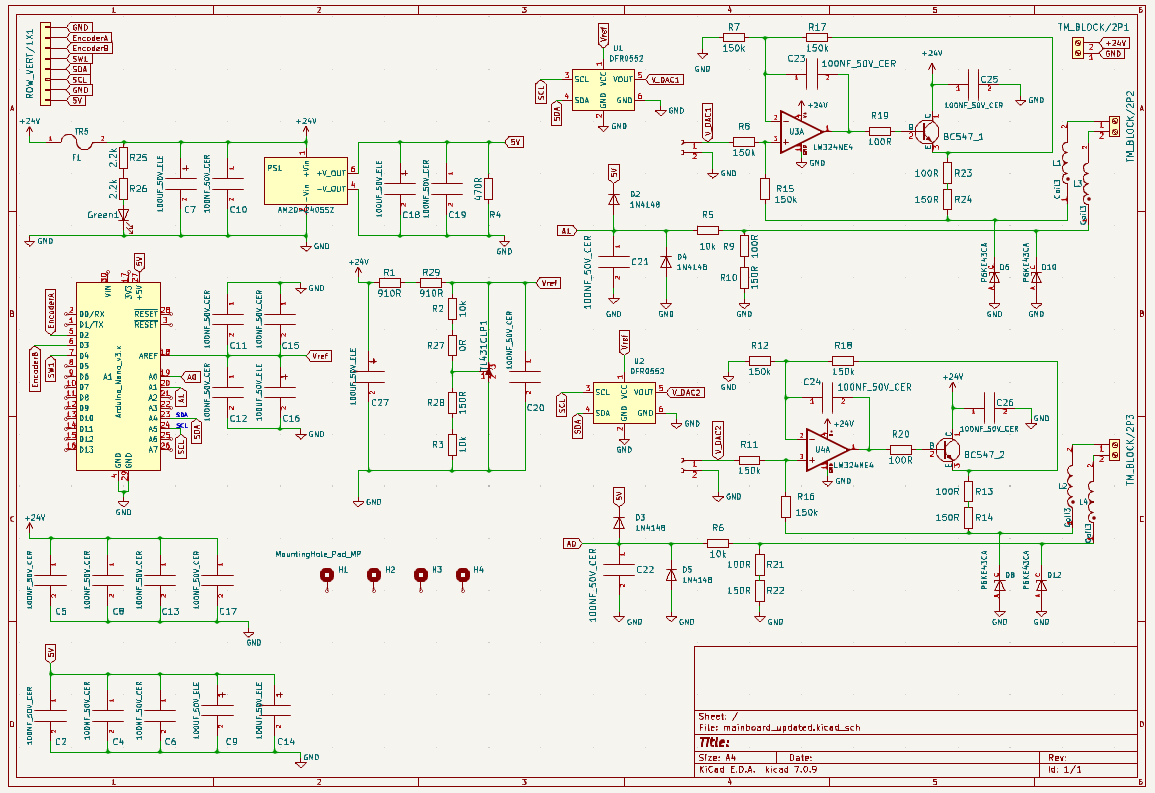


Figure 11. The main board circuit’s schematic.

### EncoderLCD Circuit

The encoderLCD circuit’s schematic was designed as below:

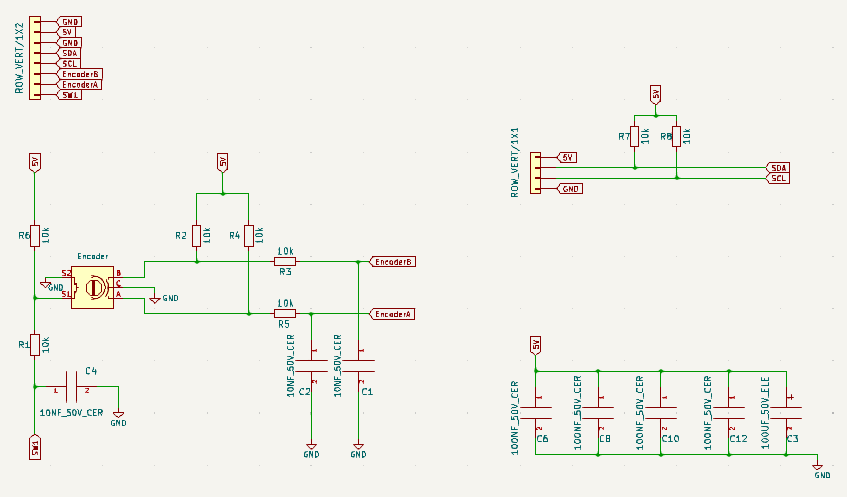


Figure 12. The encoderLCD circuit’s schematic.

## Software Implementation

This project built up the device with two channels of 0-20mA current loop using Arduino software program. The implementation involved setting up and controlling a system with the LCD display, two DACs, the rotary encoder including two main external interrupts, feedback loops and ADC. Additionally, a watchdog timer is implemented to reset the system if it becomes unresponsive. Initially, we needed to install the appropriate libraries used in this project.

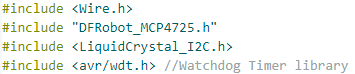


Figure 13. The appropriate libraries.

To address inaccuracies in the DAC's reference voltage, we set a calibration value to achieve a more precise output voltage. Our target was 5V for each DAC when achieving the current value at 20mA, resulting in a calibration value of 5000 millivolts:

**Calibration of DAC = 5\*1000 = 5000**

In addition, the calibration ensures precise conversion between analog and digital values throughout the system. Due to Arduino Nano offers a 10-bit resolution ADC for analog pins, the calibration of ADC could be calculated as below.

**Calibration of ADC =**

These calibrations were defined in the software as follows:

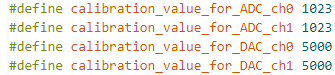


Figure 14. The calibrations for ADC and DAC for both channels.

In this program, every change of the torque and speed values by using the encoder are updated through the function of **void write\_dac(float torgue, int rpm)**. In other words, this function calculates new DAC values when there are any changes occuring, then updates the appropriate channels.



Figure 15. Updating the torque and speed changes through the function of void write\_dac(float torque, int rpm).

The encoder has two functions such as push-button and changing encoder states when controlling the desired input values. In this project, the **void read\_push\_button\_state(void)** function manages the state of a push-button from the encoder. It uses a static variable ‘timex’ for 200ms debouncing to avoid multiple triggers from one press. When the button is pressed (indicated by digitalRead(SW\_PIN) = = 0), and the debounce time has passed, it toggles the DAC\_select variable between true and false, switching between DAC1 and DAC2. The timex variable is then updated to the current time using millis() for accurate debouncing in subsequent presses.

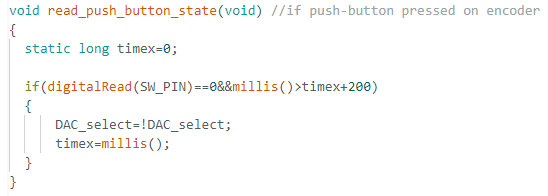


Figure 16. The state of a push-button.

An external interrup is the **void ISR\_encoderChange(void)** function, which is responsible for processing the changes of the encoder’s state to modify the torque and rpm values. Here is a part of this function shown in Figure 17. In this function, we used a lookup table of the encoder state to determine the rotation direction such as a clokcwise direction (CW) or a counterclockwise direction (CCW).

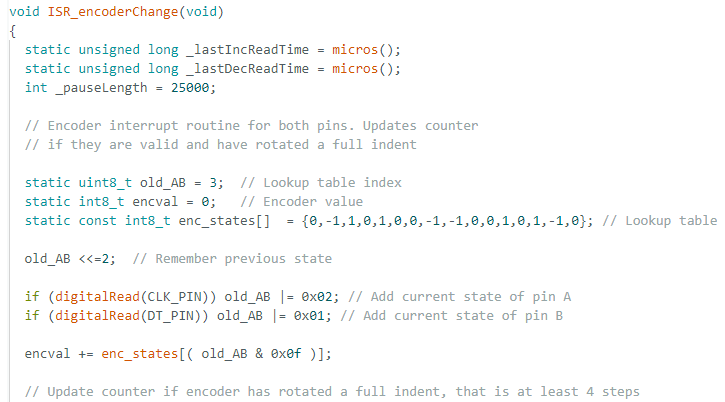


Figure 17. The state of the encoder changes as an external interrupt.

Due to the encoder has two outputs, generating four possible states: 00, 01, 10 and 11, the lookup table could be represented as below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Previous State** | **Current State** | **Index**  **(Binary)** | **Index**  **(Decimal)** | **Change** |
| 00 | 00 | 0000 | 0 | 0 |
| 00 | 01 | 0001 | 1 | -1 |
| 00 | 10 | 0010 | 2 | 1 |
| 00 | 11 | 0011 | 3 | 0 |
| 01 | 00 | 0100 | 4 | 1 |
| 01 | 01 | 0101 | 5 | 0 |
| 01 | 10 | 0110 | 6 | 0 |
| 01 | 11 | 0111 | 7 | -1 |
| 10 | 00 | 1000 | 8 | -1 |
| 10 | 01 | 1001 | 9 | 0 |
| 10 | 10 | 1010 | 10 | 0 |
| 10 | 11 | 1011 | 11 | 1 |
| 11 | 00 | 1100 | 12 | 0 |
| 11 | 01 | 1101 | 13 | 1 |
| 11 | 10 | 1110 | 14 | -1 |
| 11 | 11 | 1111 | 15 | 0 |

Table 2. Lookup table of the encoder changes.

Furthermore, this function implementes a feature of fast or slow increments or decrements based on the time elapsed since the last change, ensuring smooth and responsive adjustments. The Figure 18 illustrates an example of the feature of fast or slow increments for four step forward.



Figure 18. Fast or slow increments featuring for four steps forward.

Additionally, the software program is also included safety checks to keep the torque and rpm values within their defined limits.

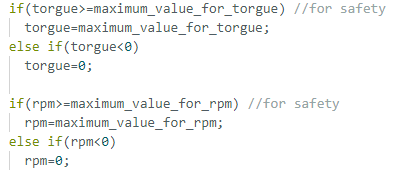


Figure 19. Safety checks within defined limits.

Besides, we also added one function for detecting possible errors in a control system by comparing the desired torque and rpm values with their corresponding ADC feedbacks, which was **int open\_loop\_error\_check(void)** as follows. For torque values, it flags an error if the desired value exceeds approximately 100 while the feedback is below 20. A similar check is performed for rpm. The function returns a combined error code: 1 for a torque channel error, 2 for an rpm channel error, or 3 if both channels have errors. This error checking mechanism helps identify potential issues in the open loop control system, such as disconnections.



Figure 20. Open loop error check featuring.

## I2C Communication Protocol

For data communication and transfer in this project, we leveraged the I2C (Inter-Integrated Circuit) protocol. This choice was particularly apt as key components of our system - namely the DAC DFR0552, I2C LCD, and Arduino Nano - all support I2C addressing, allowing for integration.

The I2C protocol is a serial communication protocol that uses two wires: SDA (Serial Data) and SCL (Serial Clock), simplifying the wiring and reducing the number of pins required. This protocol is used for communication between a master or controller (or multiple masters) and a single or multiple slave devices. The SDA line carries the actual data being transmitted, while the SCL line provides the clock signal that synchronizes the data transfer. Here is an example of I2C network in Figure 21.[[12]](#footnote-12)

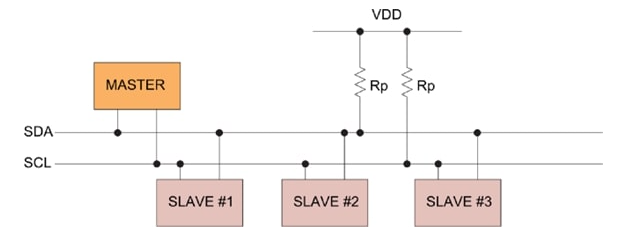


Figure 21. I2C network.

# Testing and Troubleshooting The Prototype on Breadboards

As our project evolved, we continually refined and enhanced our software to address increasingly sophisticated requirements dictated by the evolving prototype design. In the initial stages of development, our primary focus was on creating software capable of generating a precise output current of 20mA and a voltage of 5V. These parameters formed the foundation of our early prototyping efforts and served as the baseline for our initial testing phase. Consequently, when we conducted our preliminary evaluations using breadboards, we utilized this initial version of the software to power and control the prototype.

To ensure the reliability and accuracy of our design, we embarked on a thorough testing phase. This involved constructing and evaluating the critical components of both the main board and the encoderLCD circuits. We utilized breadboards for this purpose, as they provided an ideal platform for prototype testing. Our main goal was to verify if the actual output results matched our expected outcomes and simulation results in LTspice. We also aimed to ensure these results were accurately displayed on the LCD interface, which was crucial for user interaction and system monitoring.

Here are the main components of the prototype tested on breadboards.

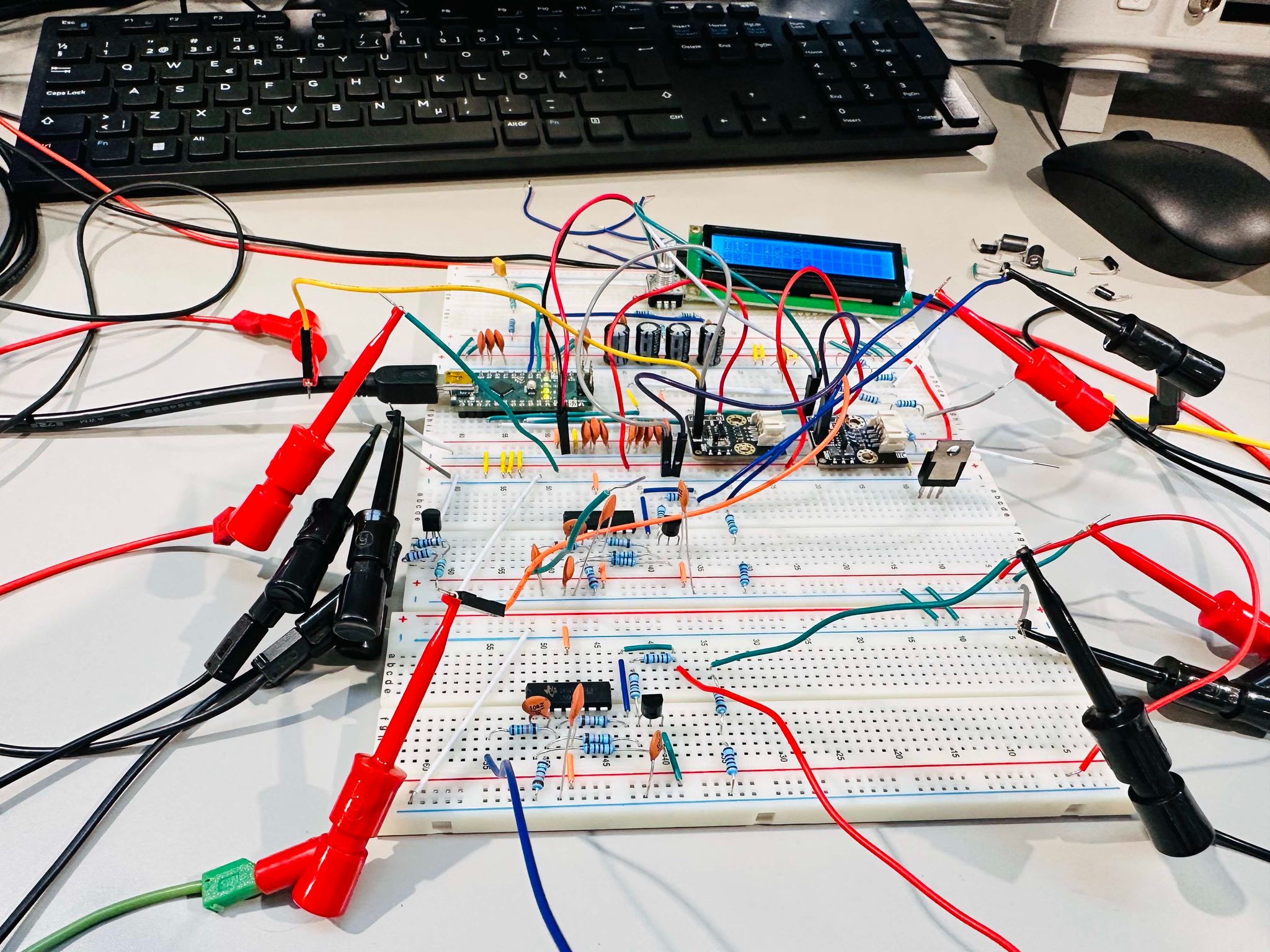


Figure 21. The main components of the prototype testing on breadboards.

During the testing phase, we encountered several significant issues that required careful attention and resolution. One of the primary challenges we faced was related to ground connections. We discovered it was crucial to connect all ground points together to effectively eliminate ground noise and achieve more accurate outputs. This realization came after a costly learning experience in our initial testing rounds. After three DACs were broken or burned, we found that interference from ground points made the input voltages of the DACs abnormal. This occurred because the output grounds of the DACs were not connected to other ground points, and the grounds of each breadboard were not linked together. Consequently, the DACs' output voltages were out of the operational range required in the datasheet. As a result of this incident, we properly connected all ground points in the prototype to create a stable reference point for our circuits.

After addressing this issue, the final output values were still higher than our calculated values, and another DAC was broken. We found that noise generated within the breadboards was the main cause of this problem. When we tested the prototype with the reference voltage created by the TL431 chip, the output values were much higher than expected, contradicting the simulation results. However, by using a 5V external power supply as the reference voltage, the output results were closer to the expected values.

Consequently, after conducting thorough testing and troubleshooting procedures as above, we recognized the need for a more stable and reliable platform for our prototype. This prompted us to initiate the process of designing PCB layouts.

# Design and Manufacture of Printed Circuit Boards

Our transition to PCBs was driven by the need for a more stable, noise-resistant platform than the breadboards we had been using. This crucial step in our development process ensured the reliability of the entire system design.

In this prototype, we designed and manufactured three distinct printed circuit boards (PCBs) based on the three comprehensive schematics mentioned earlier. This approach optimized the layout and functionality of each circuit, maximizing efficiency and performance in a finished product.

## Design PCB layouts

### The Filter’s PCB

The filter’s PCB was designed as below:

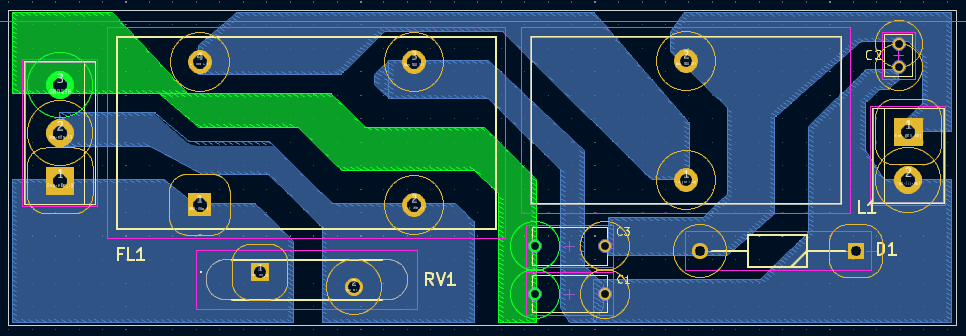


Figure 23. The filter’s PCB.

### The Main Board’s PCB

We designed a PCB for the main board part in two layers by using both SMD and through hold components.

A critical aspect of our design process involved thorough thermal analysis. To mitigate the risk of component overheating, we conducted detailed power consumption calculations for resistors. Our benchmark for thermal safety was set at a conservative one-fifth of the maximum rated power consumption for some resistors. In this device, through-hole resistors have 600mW and SMD resistors have 250mW according to the power consumption indicated by the manufacturers. Hence, we could calculate the maximum rated power consumption for through-hole and SMD resistors:

The maximum rated power consumption for through-hole resistor:

The maximum rated power consumption for SMD resistor:

If the real power consumption is smaller or equal to and bigger than , the resistor should be through-hole.

Here is the PCB layout of the main board.

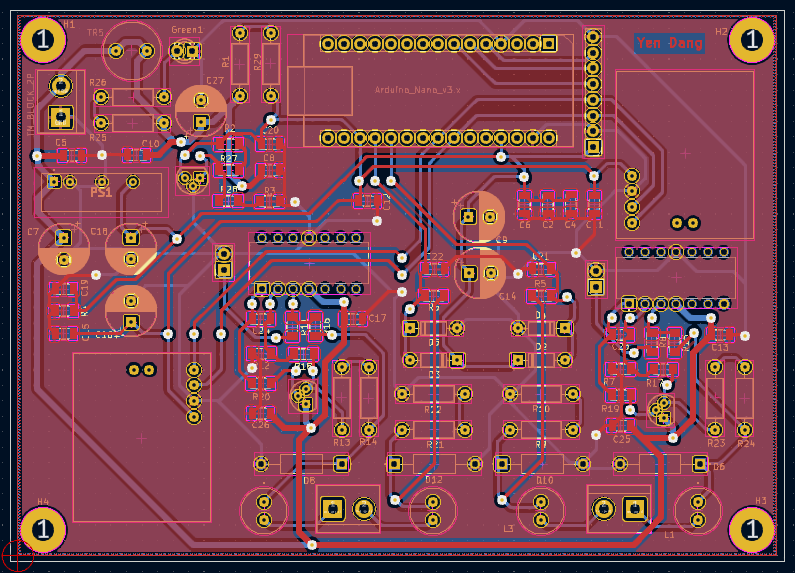


Figure 24. The mainboard’s PCB.

### The EncoderLCD’s PCB

The PCB of the encoderLCD circuit was designed in one layer.

Here is the PCB layout of this part.

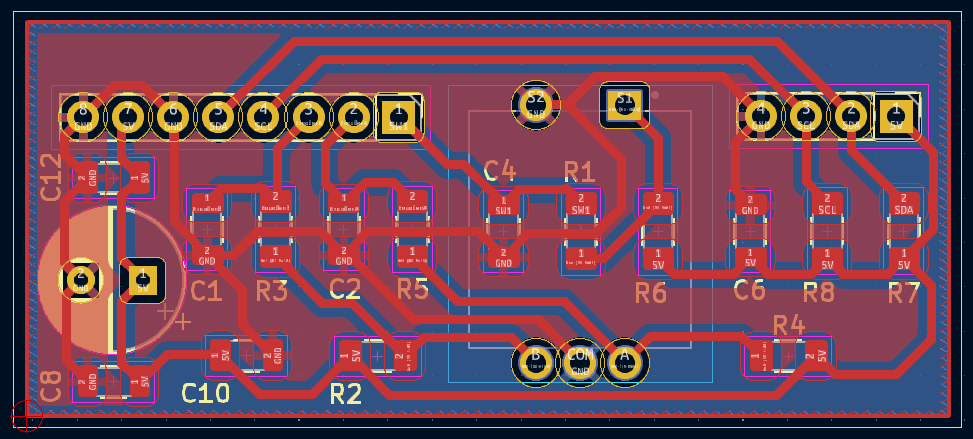


Figure 25. The encoderLCD’s PCB.

## PCBs Fabrication and Assembly

Our next step involved the manual fabrication of PCBs for the main board, encoderLCD and filter circuits. Following this, we assembled and soldered the electronic components onto these PCBs. The resulting physical PCBs are illustrated in Figure 26.

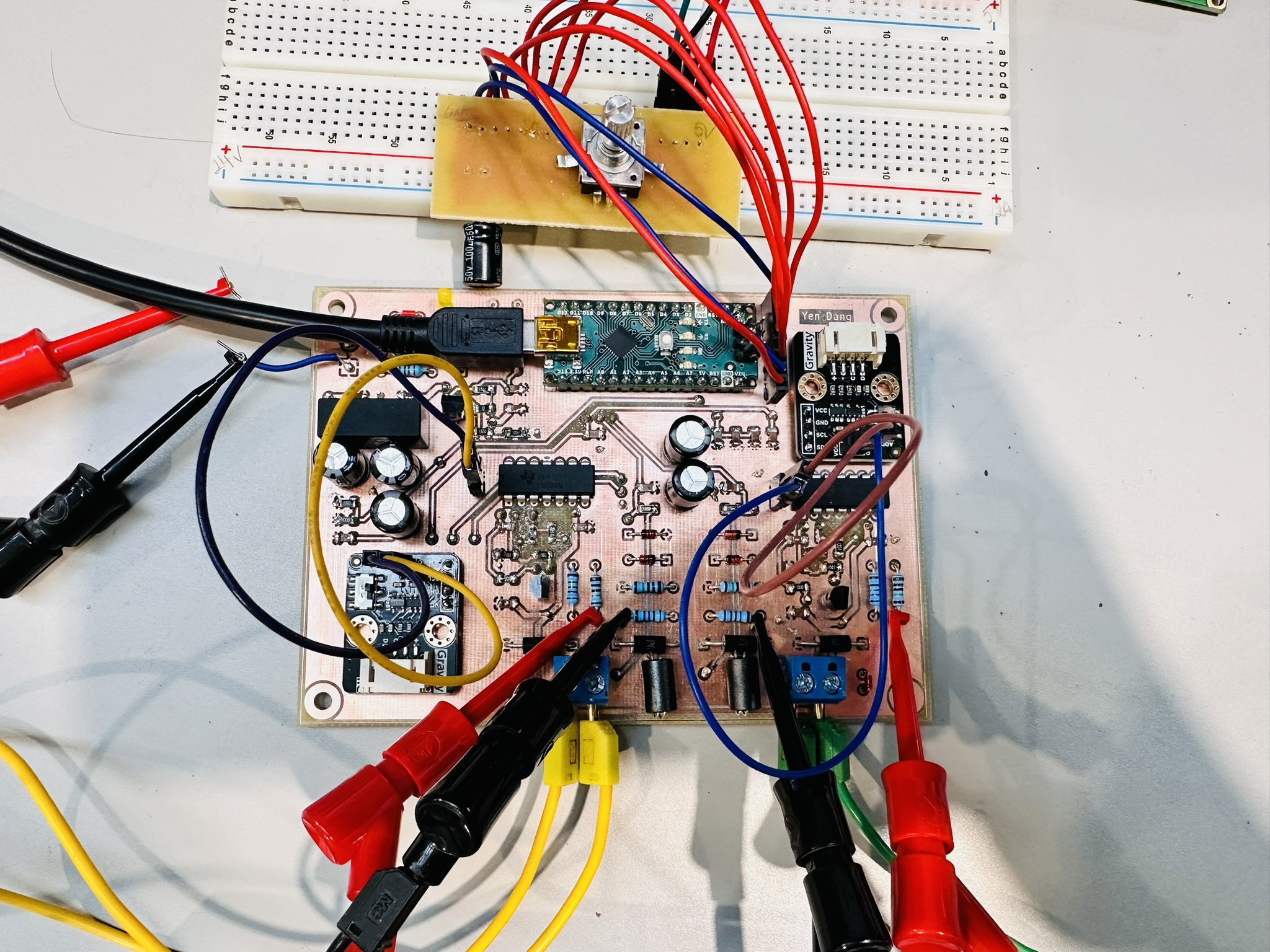


Figure 26. The assembled physical PCBs.

Upon testing these PCBs, we observed a significant improvement in the performance of our prototype. The output results demonstrated a marked increase in precision and accuracy, a clear indication of the benefits of moving from breadboards to PCBs. Especially, the values we obtained aligned much more closely with our calculated theoretical values, providing a strong validation of our design approach. This confirmed the validity of our previous testing and troubleshooting stages.

Furthermore, the wanted input values displayed on the LCD interface showed a high degree of consistency with the actual output results. This correlation between the displayed and measured values was particularly encouraging, as it confirmed the effectiveness of our user interface and the overall system integration. The improved accuracy and reliability achieved through the use of PCBs represented a significant milestone in our project's development, bringing us closer to a finalized, production-ready design. In fact, we placed an order of physical PCBs for ten devices from a professional manufacturer. This step aimed to validate our design's reproducibility and enhance the overall reliability and robustness of our project.

Then, we assembled these PCBs into two metal boxes to make a finished device as shown below:

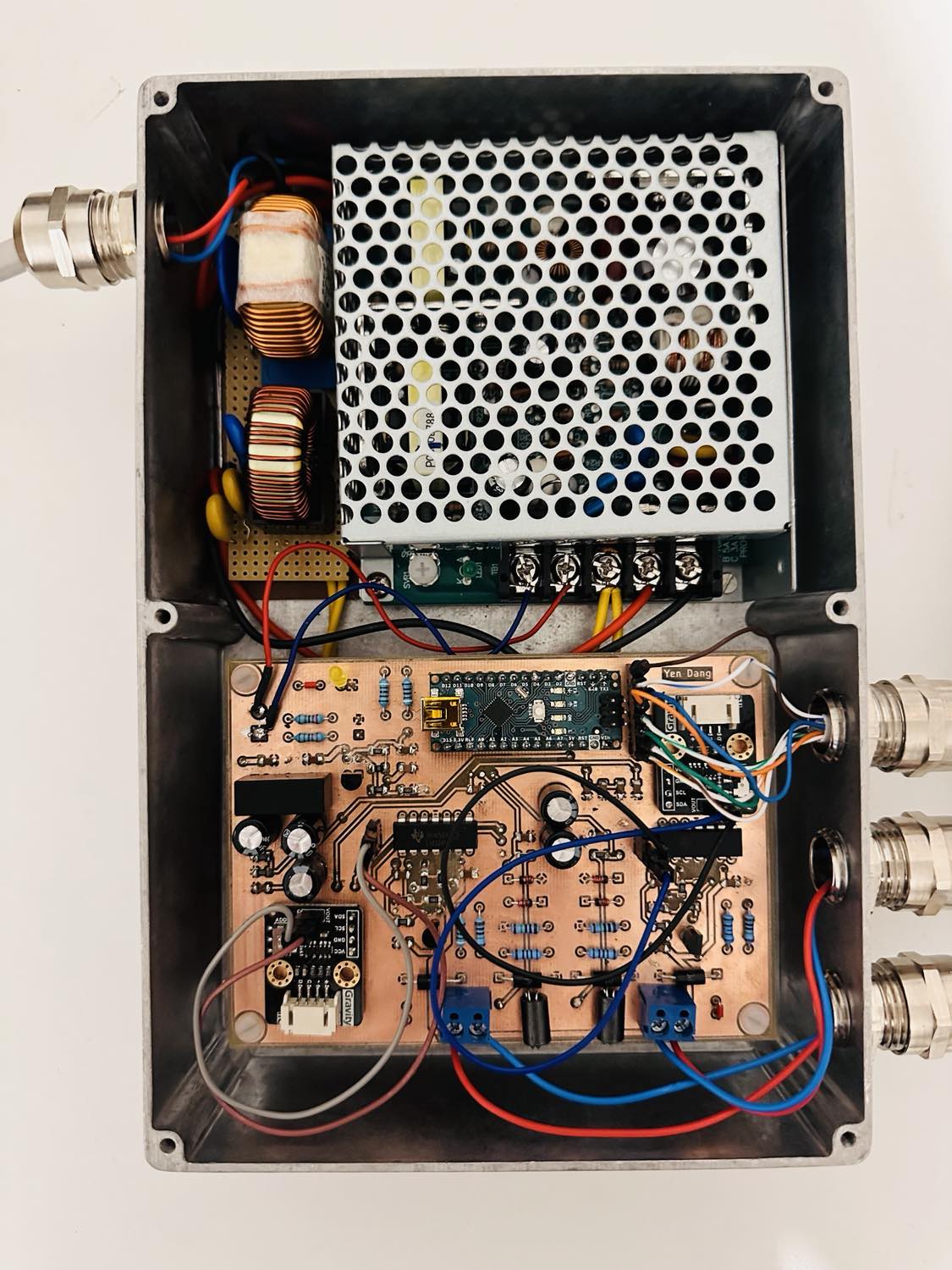
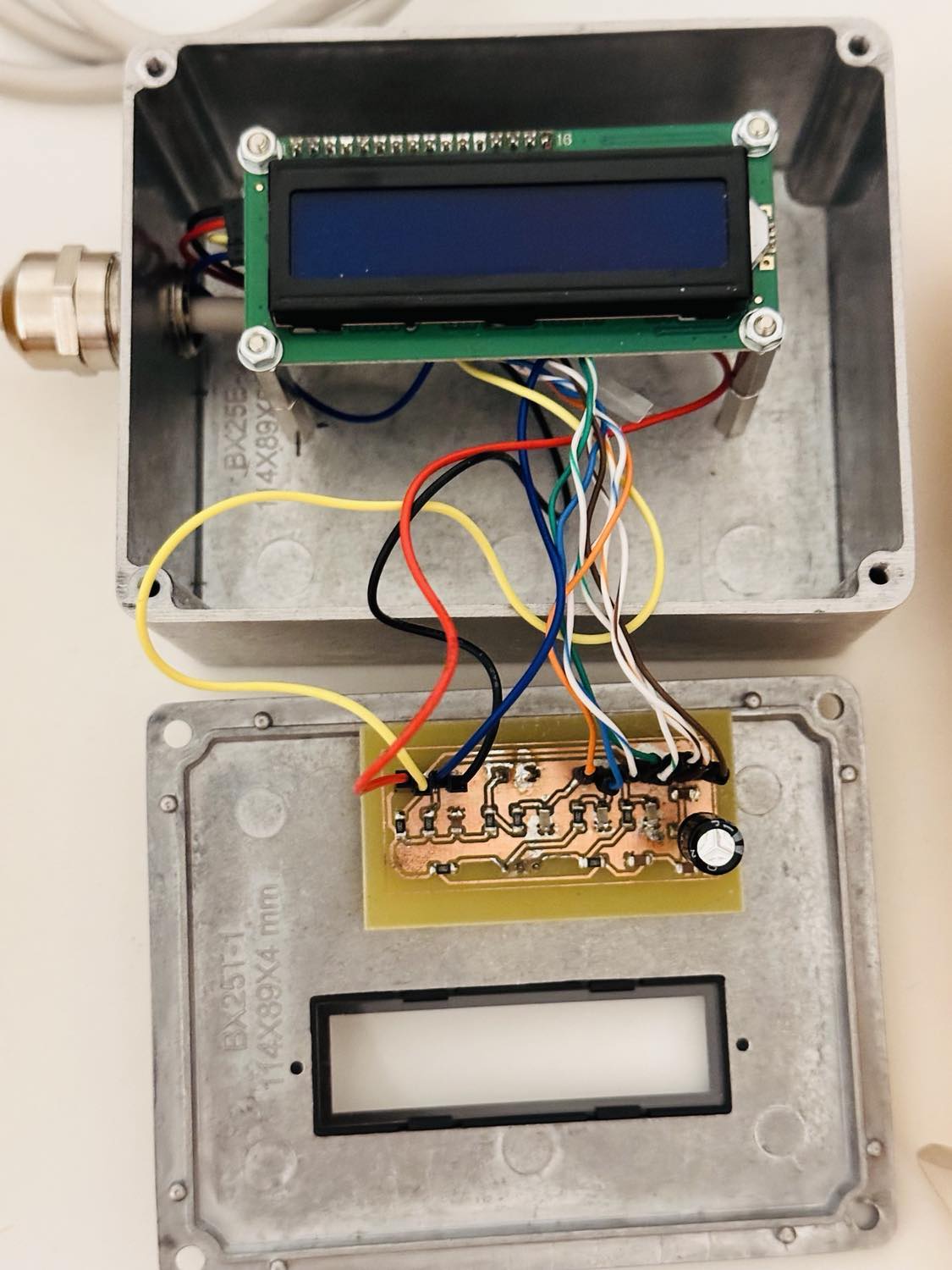


Figure 27. The finished current loop device.

However, during the development of our finished device, we faced a significant challenge related to signal integrity, specifically a crosstalk issue arising from the cable connection between two metal boxes. We used a CAT6 cable to connect the main board and encoder-LCD circuits, which included four pairs of wires. Initially, we randomly connected wires to signal pins. Consequently, without adjusting the encoder, some values on the LCD fluctuated randomly, indicating interference and noise. To solve this issue, we carefully separated different signals across the available wire pairs, ensuring that related signals were grouped together within the twisted pairs. In this device, we assigned dedicated pairs for critical signal combinations such as SDA and SCL for I2C communication, two ground (GND) connections for proper referencing, encoderA and encoderB for rotary encoder functionality, and a pair for the switch signal and 5V power supply. This approach helped to minimize the impact of capacitive coupling between different signal types, thereby preserving the integrity of our data transmission. As a result, the finished device worked properly.

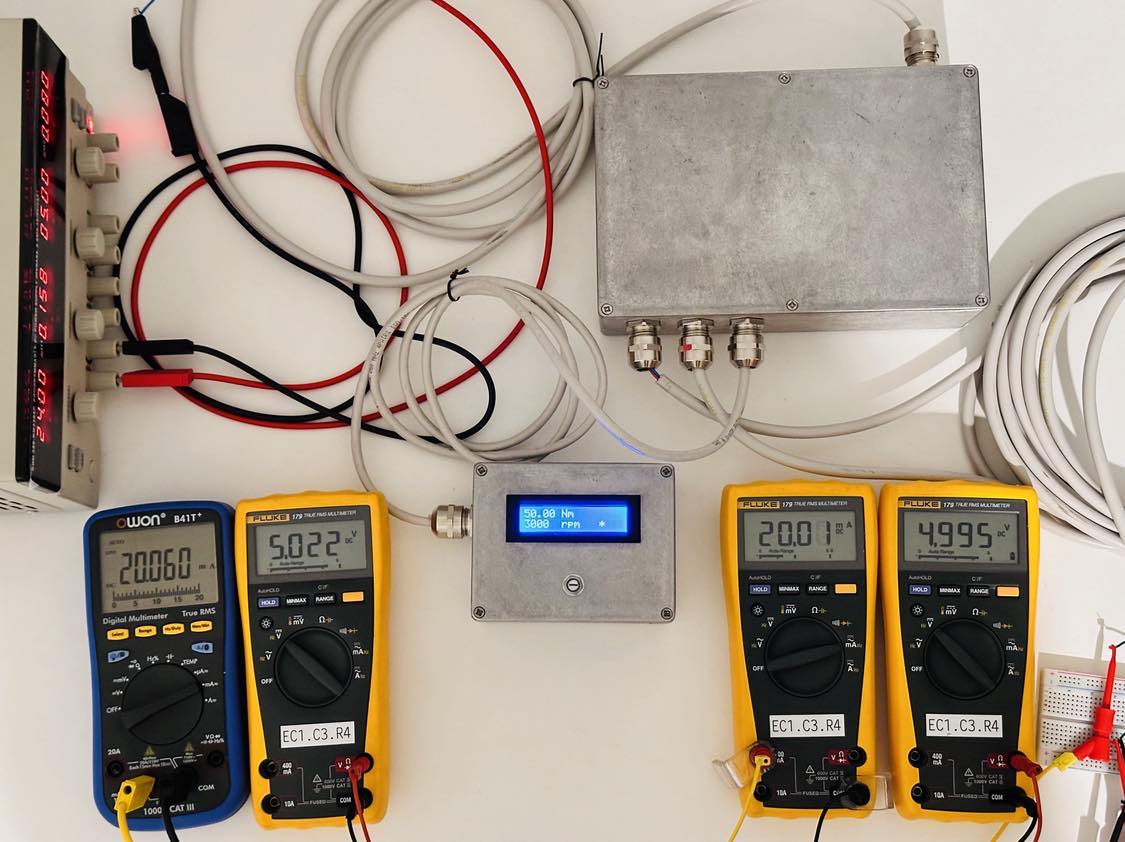


Figure 28. Testing results of the finished current loop device.

We observed that when the desired torque and speed values were set to 50Nm and 3000rpm respectively, the corresponding current and voltage values measured approximately 20mA and 5V for both channels. These measurements aligned with the simulation results of the receiver circuit.

# Testing and Comprative Analysis

This chapter described the output results of ten devices featuring professionally manufactured PCBs, and compared them to our main prototype.

## Testing Finished Current Loop Device with A Real Motor

After the finished current loop device generated stable and accuracy output values, we started testing it with a real motor by connecting it via a frequency converter. The setup for testing the finished device with the frequency converter is shown in Figure 29.



Figure 29. Setup for testing the finished device with the frequency converter.

The real-time outputs were also visible in a NCDrive software, as shown in Figure 30 below.



Figure 30. Real-time outputs displayed in the NCDrive software.

Despite our initial success, we faced challenges with the prototype's output stability due to frequency interference. This led to errors or abnormal characters on the LCD display. Upon investigation, we identified the root causes: unshielded CAT6 cable and EMC conductors lacking proper grounding. These issues allowed electromagnetic interference to affect the signal integrity, compromising the reliability of our output results. To eliminate the frequency interference, we implemented EMI shielding tapes around the CAT6 cable and connected all EMC conductors of both metal boxes to the ground. As a result, the device operated with significantly improved stability.

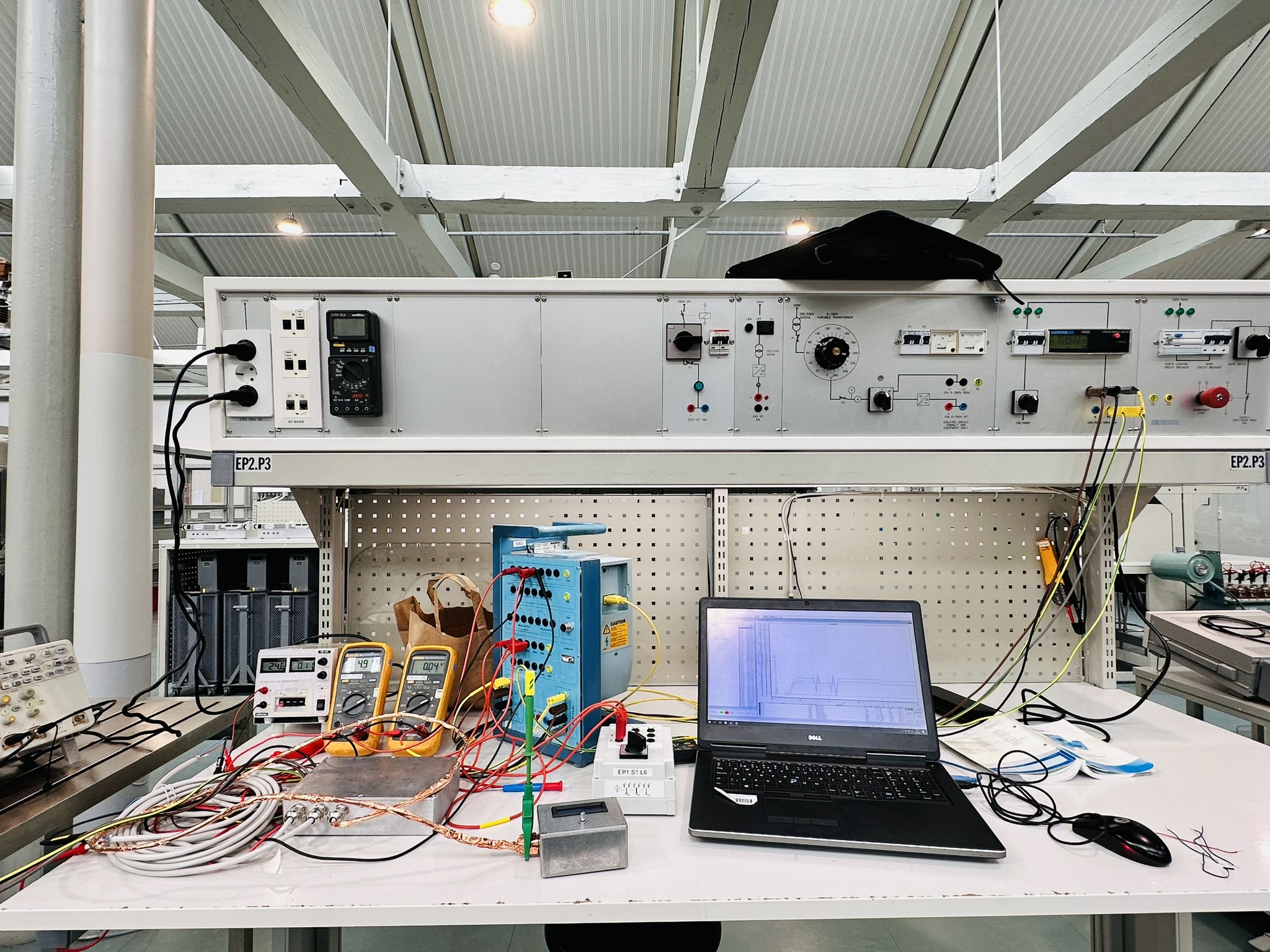


Figure 31. EMI shielding tapes applied to the CAT6 cable and EMC conductors grounded.

Moreover, the current and voltage diagrams were measured in an Oscilloscope as below:

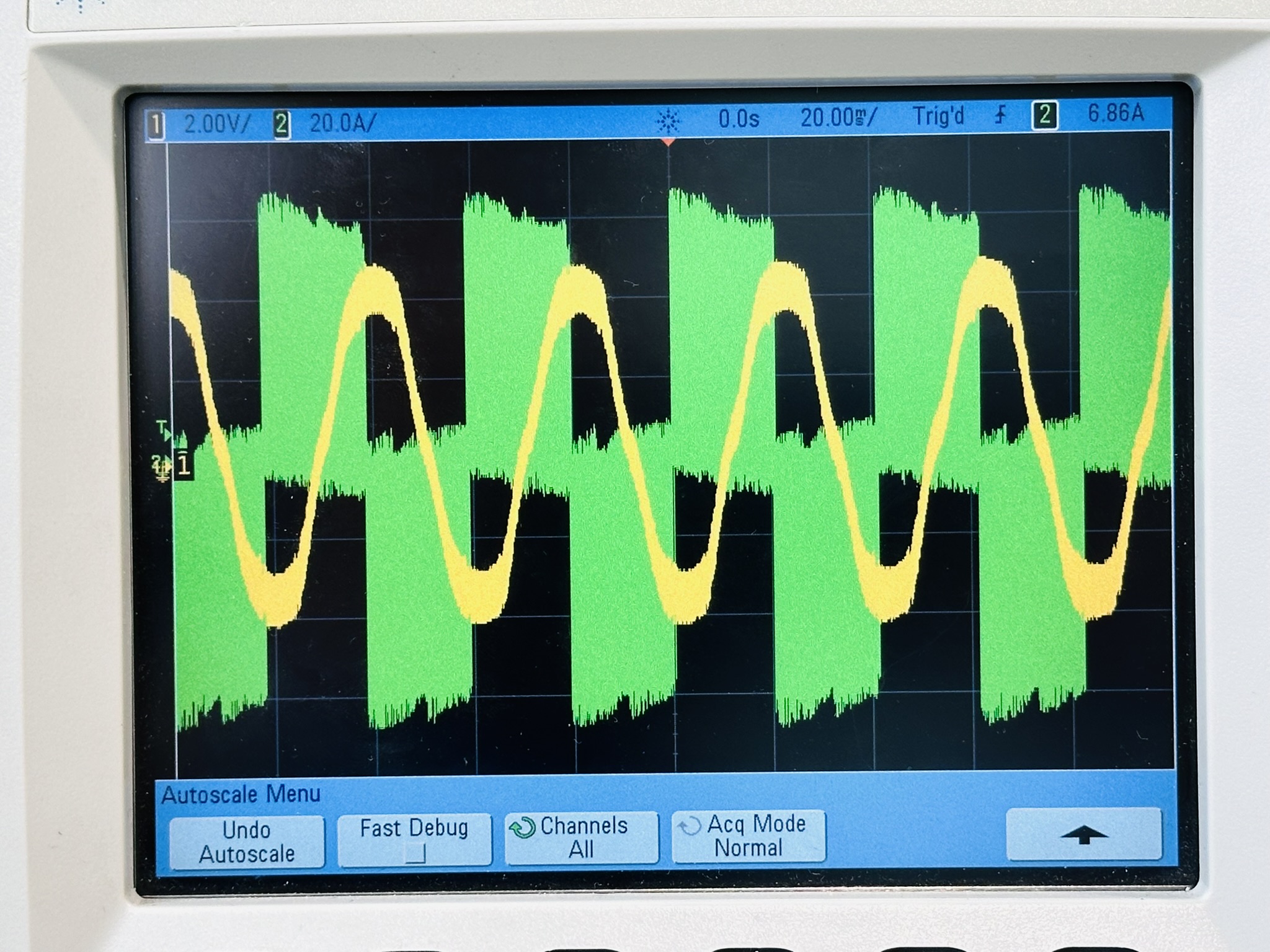


Figure 32. Current and voltage diagram in the Oscilloscope.

## Comparative Analysis of Output Results of Ten Devices and Main Prototype

In this project, we used Fluke 179 multimeters to measure output values such as current. According to the Fluke 179 datasheet (page 11), the accuracy was calculated using the following formula:

**Accuracy = ± ([% of Reading] + [Counts]) = ± (1.0% + 3)**

Using this formula, we calculated the total possible measurement error of 0.26mA and then converted to an actual current values range of 19.74mA to 20.26mA as shown in Table 3. These calculations provided a framework for interpreting our results, accounting for equipment limitations when assessing our digital current loop device's accuracy and reliability.

|  |  |  |
| --- | --- | --- |
| **Description** | **Current value** | **Unit** |
| Range of currents | 20.00 | mA |
| Percentage error | 0.20 | mA |
| Count error | 0.06 | mA |
| **Total possible error** | **0.26** | **mA** |
| **Actual current values** | **19.74** | **mA** |
| **20.26** | **mA** |

Table 3. Error measurement range for actual current values.

We documented torque, speed, and current values for the main prototype across 200 steps, with the full dataset in Appendix 1. Two graphs illustrated our findings: Figure 33 showed the relationship between desired torque and actual current output, while Figure 34 depicted the desired speed versus real current output.

Figure 33. Desired torque and real current diagram.

Figure 34. Desired speed and real current diagram.

Additionally, we recorded torque, speed, and current values for ten devices, comparing them to the results of the main prototype at 200 steps. The full dataset was presented in Appendix 2. To illustrate our findings, we created two graphs. Figure 35 showed the relationship between desired torque and actual current output for all devices, while Figure 36 depicted desired speed versus real current output. These diagrams were useful to identify performance patterns and deviations among the manufactured devices.

Figure 35. Desired torque and actual current output across all devices.

Figure 36. Desired speeds and actual current output across all devices.

According to these graphs above, the output torque and speed values of the ten professionally manufactured devices closely aligned with those of the main prototype. Minor variations were observed, potentially due to manufacturing tolerances, component differences, or assembly variations. This close alignment between the prototype and manufactured units demonstrated a high level of the robustness and reproducibility of our design.

# Conclusion

The purpose of this project was to design and implement a digital 0-20mA current loop device for frequency converters in motor technology applications. The development process involved three main stages: circuit design, software programming, and PCB design and manufacturing. The finished current loop device could read the desired values input by controlling the rotary encoder and convert them to analog signal values, creating the 0–20mA current loop transmitting to Arduino Nano. When connecting this device to both the actual frequency converter and simulated environments using a 250 Ω resistor for testing purposes, we observed positive relationships between the desired torque (Nm) and speed (rpm) values and the current values measured by Fluke multimeters. Furthermore, we noted the same positive correlations with current values for the additional ten devices featuring professionally manufactured PCBs. The following data illustrated these correlations in detail:

* The desired torque values range from 0-50Nm, with corresponding current values measured from 0-20mA.
* The desired speed values range from 0-3000rpm, with corresponding current values measured from 0-20mA.

The project faced several challenges, including ground noise, poor quality of breadboards, capacitive cross-talk issue and frequency interference. These were addressed through the manufacture of integrated PCBs and the implementation of proper shielding and grounding techniques. The use of CAT6 cables with twisted pair wiring helped minimize capacitive cross-talk and external interferences. After eliminating these problems, the finished current loop device worked properly.

In summary, the project successfully achieved its goal of generating a reliable digital current loop device. However, for further enhancement on the device’s performance, we would strongly recommend that this device could be implemented EMC testing. This crucial phase would serve to verify the device's long-term operational stability and control potential issues that may arise during extended periods of use. In addition, since the DC power supply converter might generate noise, we also recommend EMC testing for the filter circuit in future developments. This focused approach would effectively eliminate interference and noise, further enhancing the device's accuracy and precision in real-world applications.

# List of References

1. Back to Basics: The Fundamentals of 4-20mA Current Loops: <https://www.predig.com/indicatorpage/back-basics-fundamentals-4-20-ma-current-loops>
2. Why a 0 mA Signal is Not Practical: <https://www.predig.com/whitepaper/why-0-ma-signal-not-practical>
3. Ground loops & Non-Isolated Commons: [https://www.predig.com/indicatorpage/ground-loops-non-isolated-commons](https://www.predig.com/indicatorpage/ground-loops-non-isolated-comm)
4. Full control and optimised operation with frequency converters: <https://hoyermotors.com/about-hoyer/information/product-and-market-news/full-control-and-optimised-operation-with-frequency-converters/>
5. What is an incremental encoder: <https://www.dynapar.com/technology/encoder_basics/incremental_encoder/#:~:text=What%20is%20an%20Incremental%20Encoder,most%20commonly%20used%20rotary%20encoders.>
6. Datasheet of PEC11R-4220F-S0024 Incremental Encoder.
7. Datasheet and Manual of I2C LCD1602.
8. About Digital to Analog Converter (DAC) and Its Applications: <https://www.elprocus.com/digital-to-analog-converter-dac-applications/>
9. DFRobot Gravity 12-Bit I2C DAC Datasheet.
10. LM324 Datasheet.
11. What is BC547 Transistor Working and Its Applications: <https://www.elprocus.com/bc547-transistor-working-and-its-applications/>
12. 1N4148 Datasheet.
13. TR5-K Datasheet.
14. COIL3-100uH Datasheet.
15. TL431 Datasheet.
16. Arduino Nano Datasheet.
17. TVS Diodes Datasheet (P6KE Series)
18. AM2D-2405SZ Datasheet.
19. Specification for Approval of Coil3.
20. TVS Varistor Datasheet.
21. Filter B82725S2103N003 Datasheet.
22. Toroidal Inductor (Vertical, 100uH) Datasheet.

# List of Appendices

APPENDIX 1:

APPENDIX 2:

1. Back to Basics: The Fundamentals of 4-20mA Current Loops: [https://www.predig.com/indicatorpage/back-basics-fundamentals-4-20-ma-current-loops](https://www.predig.com/indicatorpage/back-basics-fundamentals-4-20-ma-current-loops" \o "https://www.predig.com/indicatorpage/back-basics-fundamentals-4-20-ma-current-loops) [↑](#footnote-ref-1)
2. Why a 0 mA Signal is Not Practical: <https://www.predig.com/whitepaper/why-0-ma-signal-not-practical> [↑](#footnote-ref-2)
3. Ground loops & Non-Isolated Commons: [https://www.predig.com/indicatorpage/ground-loops-non-isolated-commons](https://www.predig.com/indicatorpage/ground-loops-non-isolated-comm) [↑](#footnote-ref-3)
4. Full control and optimised operation with frequency converters: <https://hoyermotors.com/about-hoyer/information/product-and-market-news/full-control-and-optimised-operation-with-frequency-converters/> [↑](#footnote-ref-4)
5. What is an incremental encoder: <https://www.dynapar.com/technology/encoder_basics/incremental_encoder/#:~:text=What%20is%20an%20Incremental%20Encoder,most%20commonly%20used%20rotary%20encoders.> [↑](#footnote-ref-5)
6. Datasheet of PEC11R-4220F-S0024 Incremental Encoder. [↑](#footnote-ref-6)
7. Datasheet and Manual of I2C LCD1602. [↑](#footnote-ref-7)
8. About Digital to Analog Converter (DAC) and Its Applications: <https://www.elprocus.com/digital-to-analog-converter-dac-applications/> [↑](#footnote-ref-8)
9. DFRobot Gravity 12-Bit I2C DAC Datasheet. [↑](#footnote-ref-9)
10. LM324 Datasheet. [↑](#footnote-ref-10)
11. What is BC547 Transistor Working and Its Applications: <https://www.elprocus.com/bc547-transistor-working-and-its-applications/> [↑](#footnote-ref-11)
12. I2C Protocol: <https://www.prodigytechno.com/i2c-protocol> [↑](#footnote-ref-12)