ENEL 351: Microcontroller System Design

Filament Monitoring System Report

Mubashir Hussain

200396797

April 11th 2023

Table of Contents

1.0 Introduction / Functional Specification review

I have developed a filament monitoring sensor project that incorporates an infrared sensor and a temperature sensor to monitor the status of the printer and alert the user when the filament is running low. In addition, this project includes a solenoid to automatically turn off the printer based on the infrared sensor reading, providing an extra layer of safety for the user. The project also includes an LCD display to showcase the state of the 3D printer and the temperature of the bed. A WiFi module was added later on, enabling remote monitoring of the printer's status, temperature data, and filament levels. This final report expands on the information provided in the functional specification, includes a full electrical schematic for the device and a detailed procedure of the testing carried out during the development of the project. Additionally, a photograph of the final device is included. Finally, the report discusses potential improvements and future developments that could be made to enhance the project's functionality and usefulness.

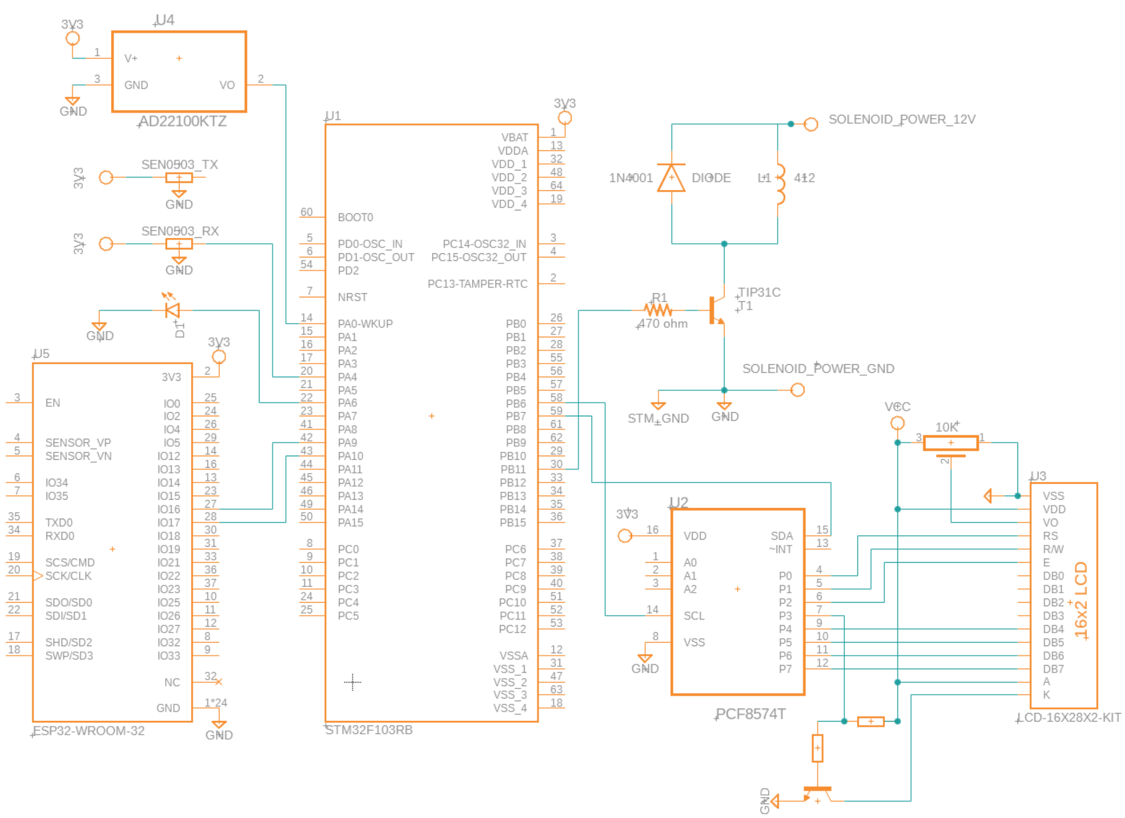


Figure 1 : Filament Sensor Project Schematic (Sketched on Fusion Electronics)

2.0 Changes and enhancements to the design

The functional specification of the project differed from the original proposal by using an AD22100 voltage-based temperature sensor instead of a thermistor due to two main factors: accuracy and ease of installation. Initially, the plan was to connect a thermistor to the hot end and display temperature readings on the LCD. However, adding an extra thermistor to the hot end proved to be difficult as the hot end itself needs to be removed and the thermistor is locked in with a screw (shown in Figure 2). This is problematic because I wanted to add in another thermistor on top of the existing one and it could cause firmware issues with the 3D printer I am trying to read from the STM32. The AD22100 temperature sensor avoided these challenges and provided better accuracy.

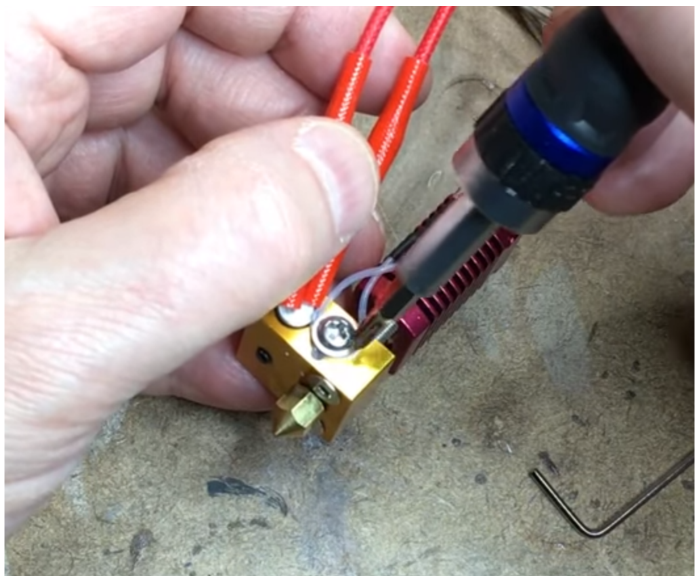


Figure 2: Hot end Thermistor replacement

Reasons to use AD22100 versus 10k NTC thermistor

1. Linear response: the AD22100 has a linear response to temperature changes, which can make it easier to obtain accurate temperature readings over a wider range of temperatures. In contrast, thermistors have a non-linear response, which can require more complicated calibration procedures to obtain accurate temperature measurements. (Comparison shown in Appendix A)
2. Stability: The AD22100 is a solid-state sensor and is less susceptible to drift and aging effects over time compared to thermistors, which can experience changes in resistance

due to long-term exposure to high temperatures. This can result in more stable temperature measurements which is ideal for a 3D printer.

3. Easy to interface: The AD22100 outputs a voltage signal that can be easily interfaced with an analog-to-digital converter (ADC) or microcontroller, without the need for additional signal conditioning circuits. In contrast, thermistors require a signal conditioning circuit to convert their non-linear resistance into a linear voltage signal, which can add complexity to the design and increase the potential for errors in the measurement system.

ESP32 Module with Wifi Connectivity

After completing the primary tasks outlined in the functional specification for the filament monitoring system, the next step involved adding WiFi connectivity. The main objective was to enable users to remotely turn off the printer and monitor the temperature of the bed. This addition was crucial as bed adhesion is a crucial element in achieving successful 3D prints, and this modification would significantly improve the system's practicality. To establish communication, the STM32 would take readings from the Infrared break beam sensor and AD22100 and send them to the ESP32 module, which would then be connected to the local WiFi router. By accessing the IP address of the ESP32 module through the router, users could turn off the printer and monitor the temperature sensor. The STM32 and ESP32 communicated using UART at a baud rate of 9600.

The ESP32 has two modes to connect to WiFi, Station mode and Access Point mode. In Station mode, it functions as a client and links to an existing access point, providing access to the internet and enabling communication with other devices on the same network. In Access Point mode, it serves as an access point and allows other devices to link to it, which is useful in creating a network in remote locations or temporary setups without an existing access point. To test the system, I configured the ESP in Station mode to link with my local wifi router. The Arduino sketches were obtained from the ESP32 library in the Arduino IDE and were combined to ensure that the UART communication between the STM32 and ESP was established. Although there was a half-second delay between WIFI to ESP to STM32 after the implementation, it was reasonable but not desirable.

3.0 Testing Procedure and Methodology

Version Control

Throughout my project, I used a modular methodology to tackle specific tasks by creating separate branches in GitHub (Appendix B). For instance, my first branch was called sensorMotorFanfare, which I used to get my IR sensor working and trigger my solenoid based on the sensor's readings. Once that was complete, I created the LCDFanfare branch to display the sensor's status on a parallel connection LCD. However, I later realized that there were too many wires needed for the parallel LCD, so I created a new branch called I2C_LCDfanfare to work on getting an I2C LCD to work instead. Once I had the LCD working, I merged the changes into the main branch. Finally, I created a UART_wifi branch to establish Wi-Fi connectivity on the ESP32 board and communicate with my STM32 board over UART. This modular approach allowed me to focus on one aspect of the project at a time, ensuring that each part worked correctly before integrating it into the final system.

By breaking the project down into separate branches, I was able to isolate and focus on specific areas of the project. When I encountered a problem with the I2C LCD in the I2C_Fanfare branch, I could easily switch over to another branch, like the SensorMotorFanfare branch, to test the functionality of the sensors and the solenoid. This helped me quickly determine that the issue was with the LCD, not the sensors or motor, saving me a lot of time in troubleshooting. Additionally, if I ever got stuck on a particular problem, I could easily switch to another branch and come back to the problem later with a fresh perspective. Overall, a modular design approach made the project more manageable, efficient, and easier to troubleshoot.’

Keil uVision Debugger

I used the Keil uVision debugger quite extensively. Whenever I encountered an error with my code or some logical mistake, I would use the debugger to step through my code and set watch variables, which helped me maintain an understanding of my code and control flow. This was especially useful when I was implementing a toggle function for the solenoid to prevent it from triggering constantly when there was no filament. I had a problem with my logic for that code, but by using the debugger, I was able to step through and fix the logical error that I made. Overall, the debugger was an invaluable tool in helping me identify and fix errors in my code, and I believe that it significantly contributed to the success of my project.

4.0 Conclusion and Future improvements

In summary, the filament monitoring sensor project was designed and implemented to address the challenge of monitoring filament levels during long 3D printing jobs. The project incorporates both digital and analog sensors, a solenoid, and an LCD to provide real-time data on the printer's status and temperature, as well as an alert when the filament is running low. The addition of a WiFi module enables remote monitoring of the printer's status and data. A future enhancement for the project would be to integrate the STM32 board with the Marlin firmware so that if the filament is not present, the bed and hot end can be moved to their home positions and the temperature can be decreased before turning the printer off. Another potential future enhancement could be to add a camera module to the project to allow for remote monitoring and visualization of the printer's progress. Overall, the project's modular design and testing methodology proved to be effective in addressing design challenges and ensuring a successful implementation.

Appendix A

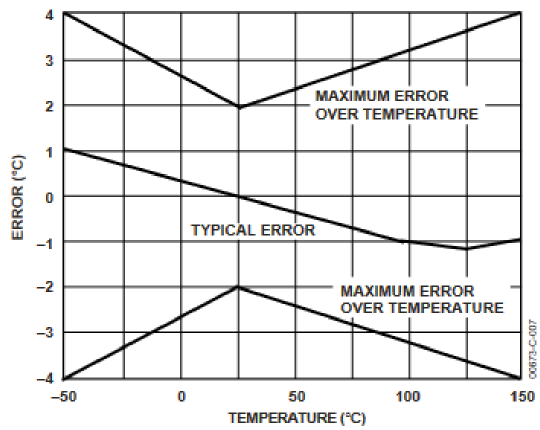
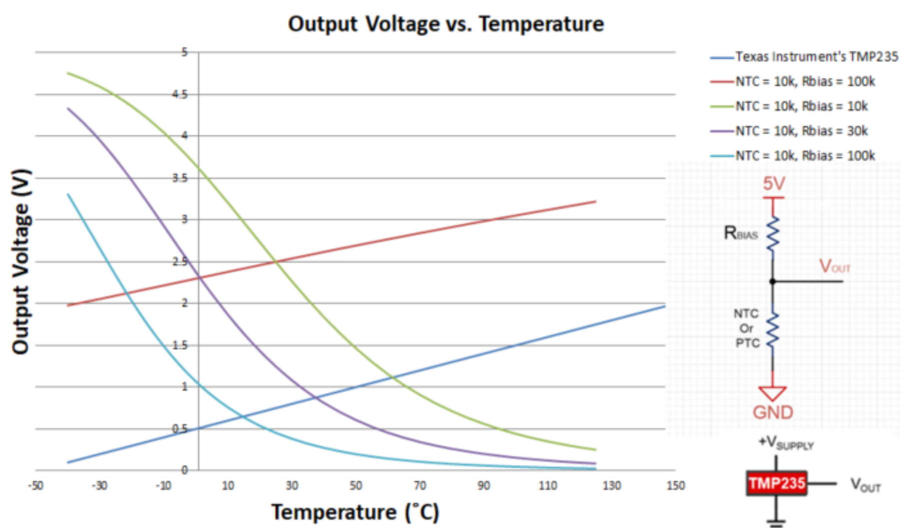


Figure 8. Typical AD22100 Performance

voltage output of AD22100

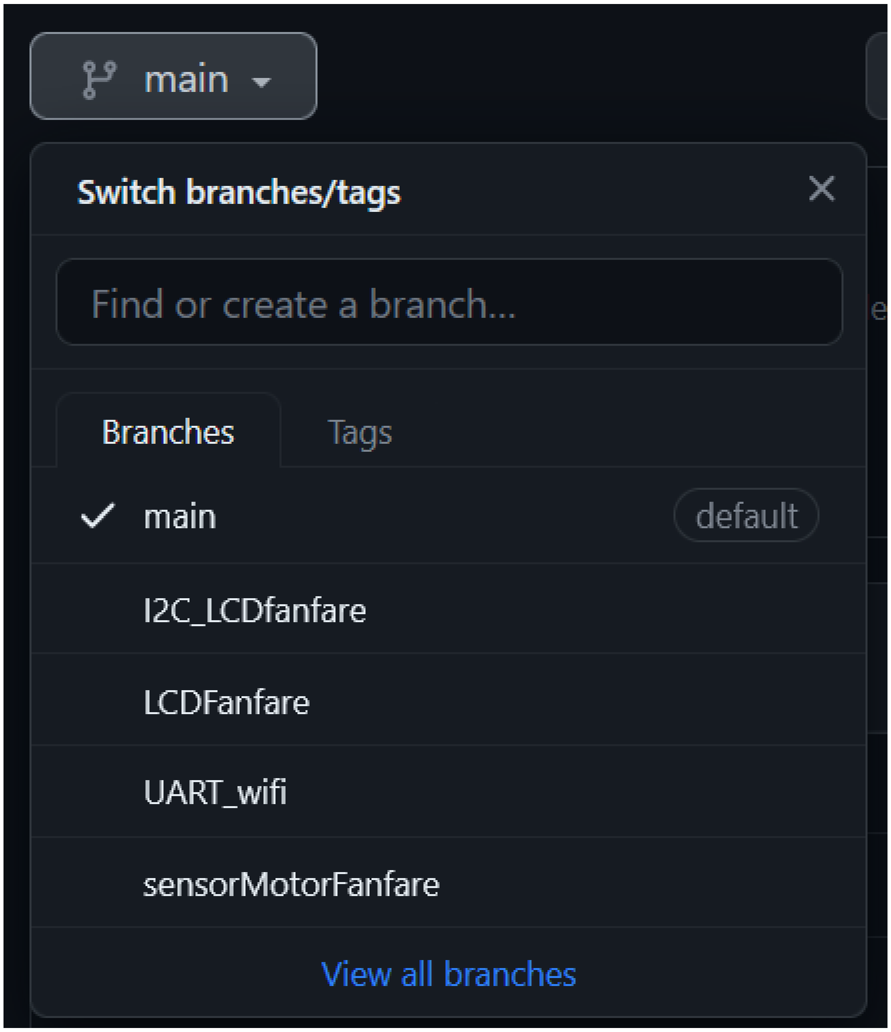
The primary sources of error for the sensor are offset, span error, and deviation from the theoretical output of 22.5 mV/°C. However, these errors can be calibrated to achieve improved performance. Figure 8 shows the guaranteed performance.



Voltage Output of thermistor

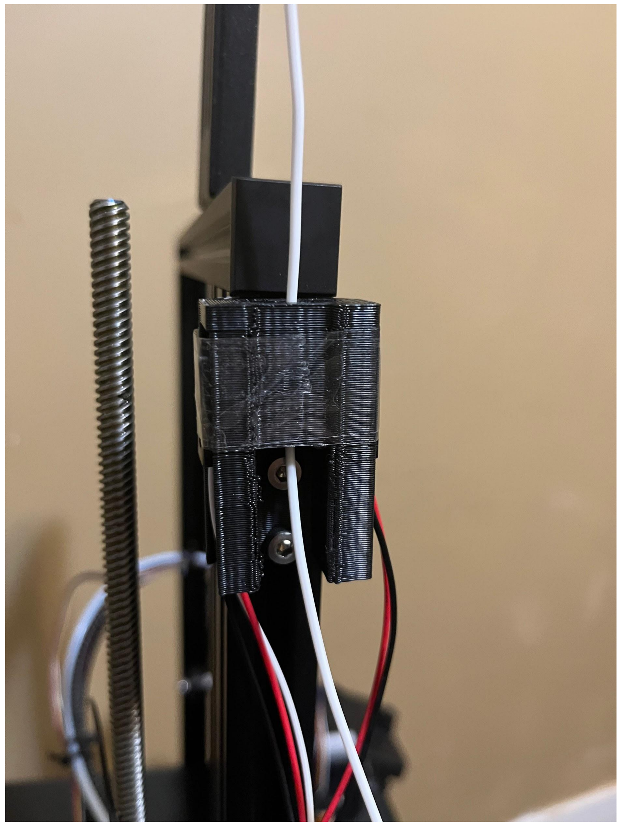
Appendix B

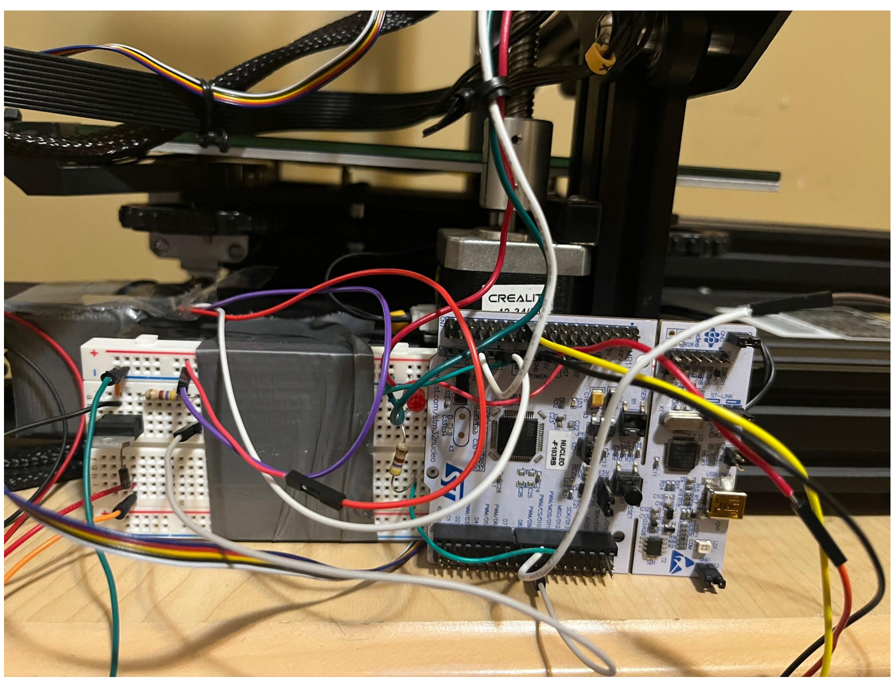
https://github.com/MubashirHu/FilamentSensorProject



Appendix C

Pictures of Project setup





Appendix C