Daniel Diamont

TA: Saadallah Kassir

EE 445L

**Lab 9 Report**

1. **OBJECTIVES**

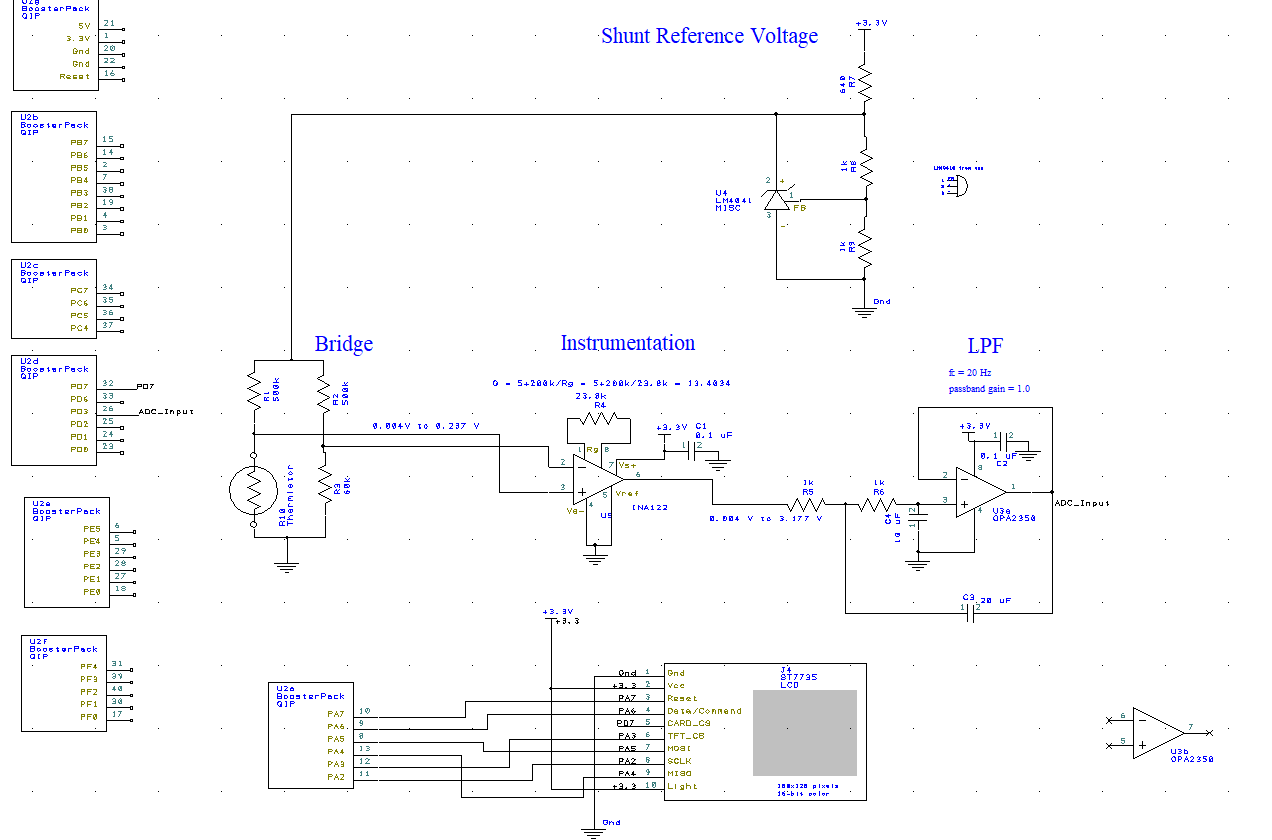
**Goals**

The goals of this lab are to study ADC conversion, the Nyquist Theorem, and to develop a data acquisition system involving transducers, instrumentation amplifiers, and filters. Specifically, the lab involves the design and implementation of a digital thermometer with a temperature range from 20 to 40 degrees Celsius with an accuracy of 1 degree Celsius and a resolution of 0.1 degrees Celsius or better. The knowledge needed to do this job include – but is not limited to – the following:

* Understanding bridge circuits for precise instrumentation
* Understanding the non-linear temperature-resistance curve of thermistors
* Ability to construct a calibration procedure for a transducer
* Signal conditioning with instrumentation amplifiers and filters
* Sampling analog signals and understanding the conditions for full reconstruction of a signal
* Time-domain and frequency domain analysis of signals, noise, and the effect of filtering while sampling
* 2-pole Butterworth LPF design for signal conditioning
* Using shunt reference voltages for instrumentation
* Performing software conversion of transducer data to a meaningful digital format using equations and/or look-up tables
* Real time sampling using a hardware timer to trigger ADC conversion
* Graphic display of temperature vs. time on LCD screen

1. **HARDWARE DESIGN**

**Circuit Schematic *(check Lab09\_Artist.sch in .zip file for better resolution)***



1. **SOFTWARE DESIGN**

**Calibration Calculations**

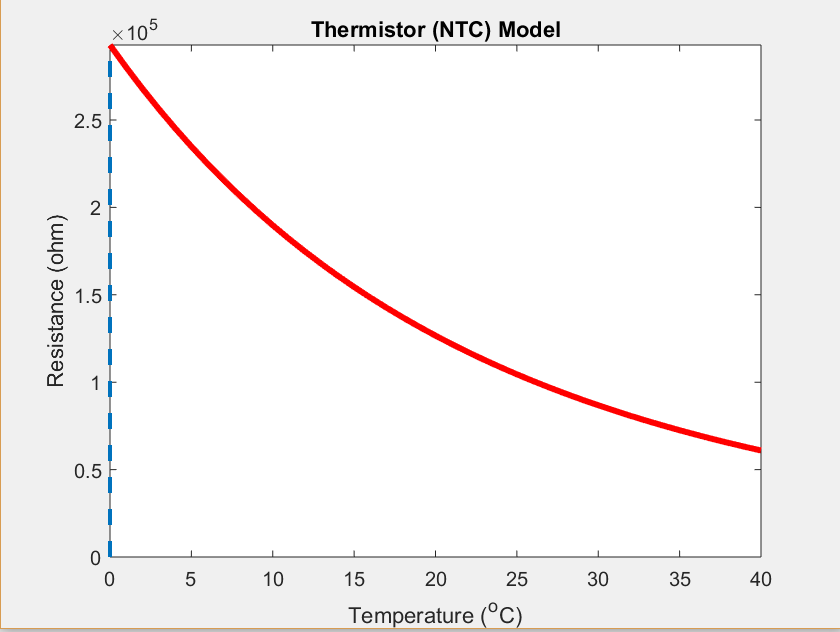
The NTC thermistor that we used for this lab (Part #: BC2432-ND) follows a resistance-temperature relationship that can be modeled by the following equation:

|  |  |
| --- | --- |
| **Resistance** | **Temperature** |
| 104.5 kΩ |  |
| 70 kΩ |  |

To obtain the β-term of the NTC Thermistor equation, I measured the resistance of my thermistor at two temperatures as follows:

Turning the temperature into Kelvin, the β-term (in Kelvin) can be calculated to be:

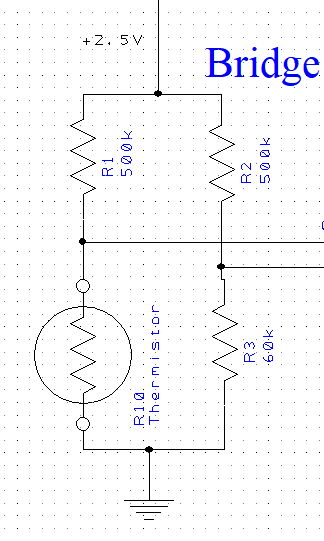
This produces our approximate thermistor curve as follows:



|  |  |
| --- | --- |
| **Resistance** | **Temperature** |
| 126.629 kΩ |  |
| 60.932 kΩ |  |

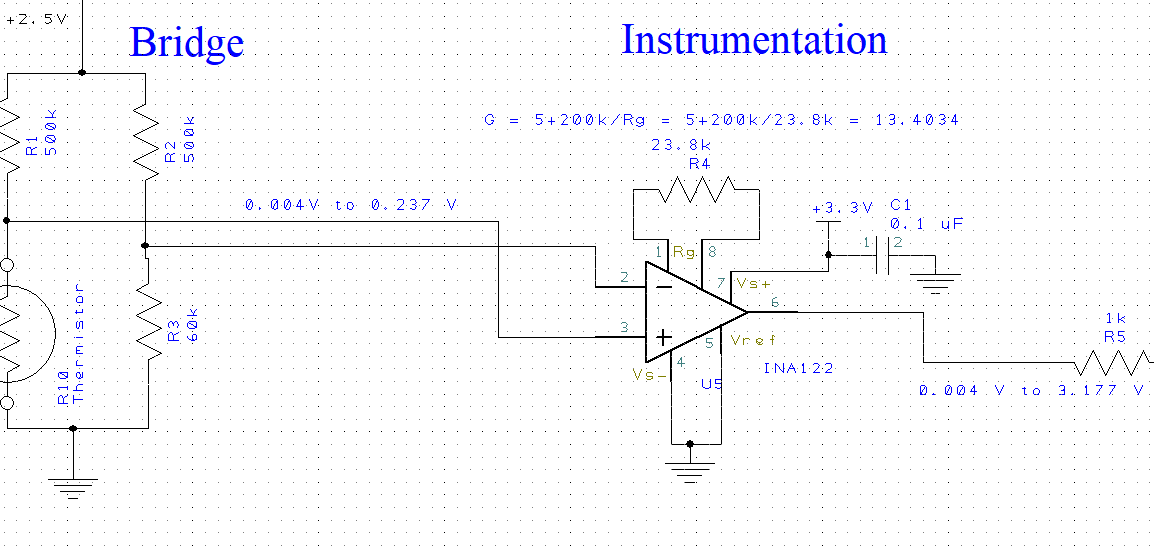
We can see that at the end of our temperature range of interest (, the resistance of our thermistor approaches ~ 60 kΩ, and at the beginning of our region of interest () it approaches ~ 125 kΩ. Indeed, when calculating the resistances with our acquired β-value, we arrive at:

Given these values, we can choose the resistor values of the Wheatstone bridge used to transform the variable resistance of the thermistor into a voltage that we can condition and process. The bridge circuit is as follows:

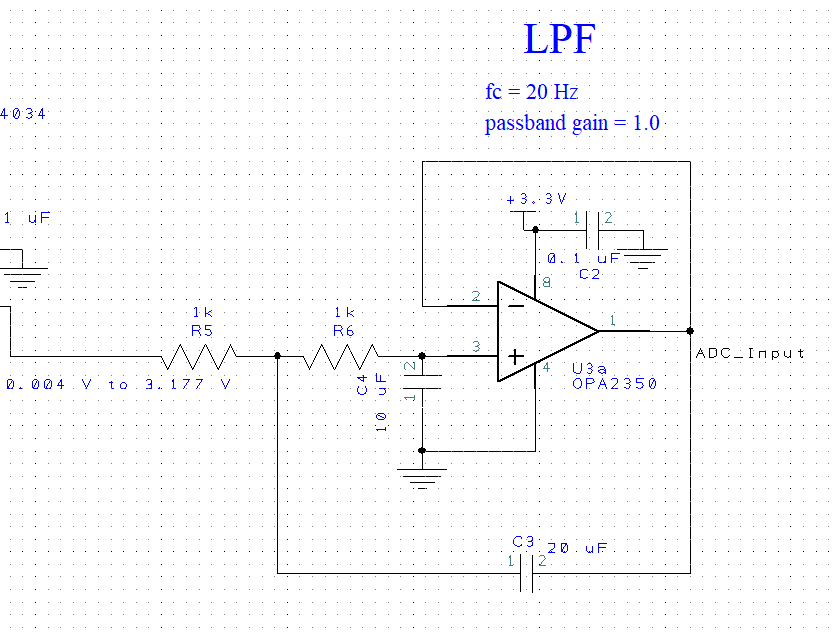


Using voltage division equations, the range of input resistances to the bridge can be represented as a voltage as follows:

Since our Analog-to-Digital converter module can only take voltages from 0.0V to 3.3V as inputs, we run the instrument signal from the resistor bridge through an instrumentation amplifier with a gain factor of 13.4034 to map the voltage range from 0.004 V to 3.177V. The gain of the instrumentation amplifier is calculated as shown in the diagram on the next page:



The instrumentation amplifier has the advantage of neglecting common-mode voltage between its inputs. This increases the signal-to-noise ratio by removing any noise shared by the two input nodes and only amplifying the voltage between them. Additionally, the signal conditioned by a 2-pole Butterworth Low-Pass-Filter chosen with a cut-off frequency of ~20 Hz to eliminate any unwanted high-frequency noise from our transducer signal. Note the frequency of our transducer signal is expected to be less than or equal to 10 Hz. The passband gain of the filter is 1.0. After this conditioning stage, the signal is ready to be sampled by our Analog-to-Digital converter.



**LUT Generation**

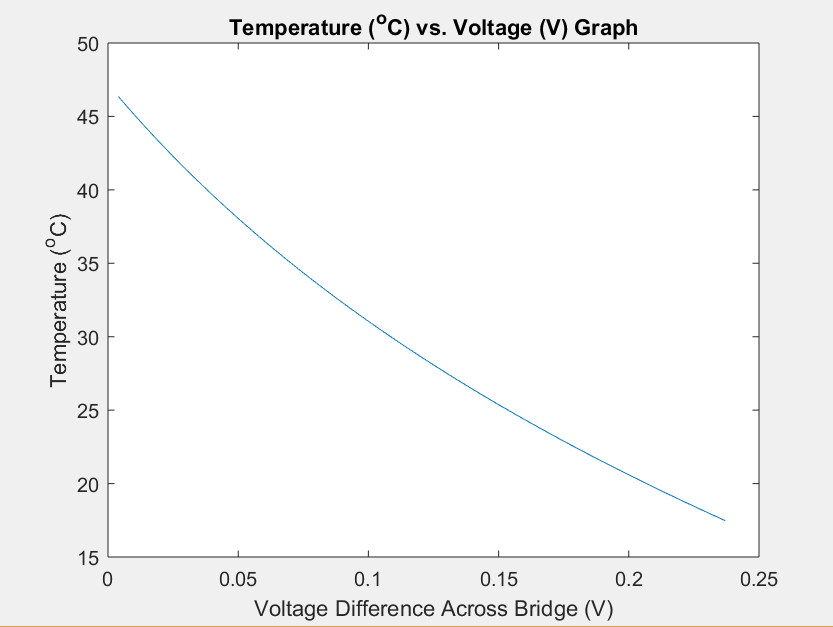
The ADC on the TM4C123GH6PM has input codes (4096 alternatives) available to convert an analog signal. Since the format of interest of the data is in units of temperature, we must find some function *f[n]* that maps the discrete input codes of the ADC to temperature.

We can do this by first finding the transfer function *T* which maps the continuous transducer resistance values to temperatures. This is the function obtained from running the signal through the bridge circuit.

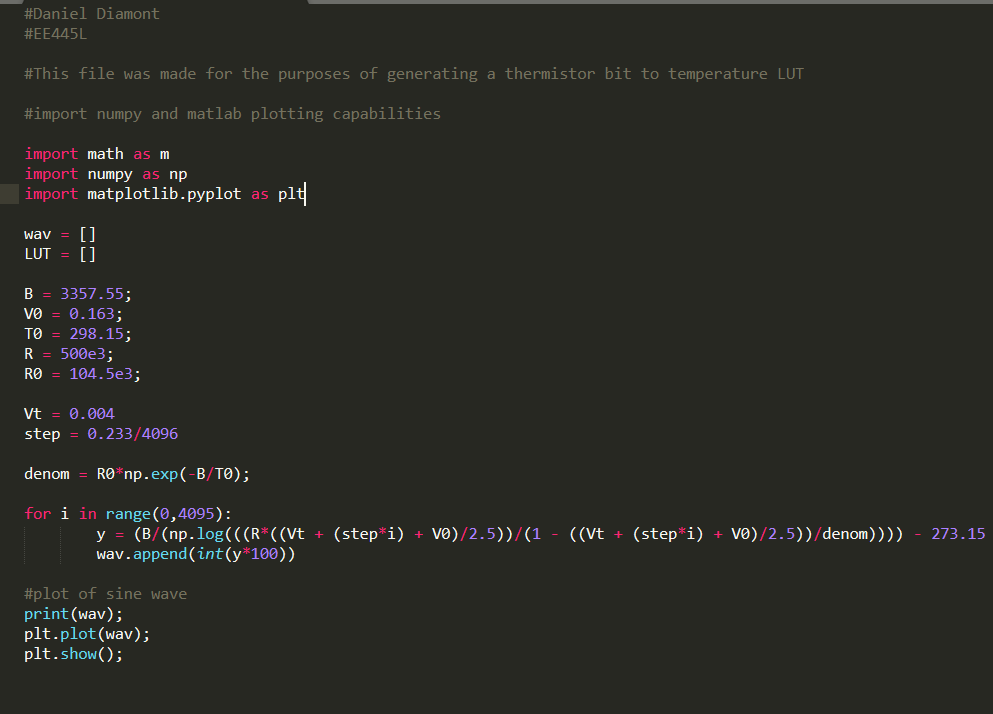
This requires rearranging the β-term equation to the form:

The calculation of is dependent on the thermistor resistor input as follows:

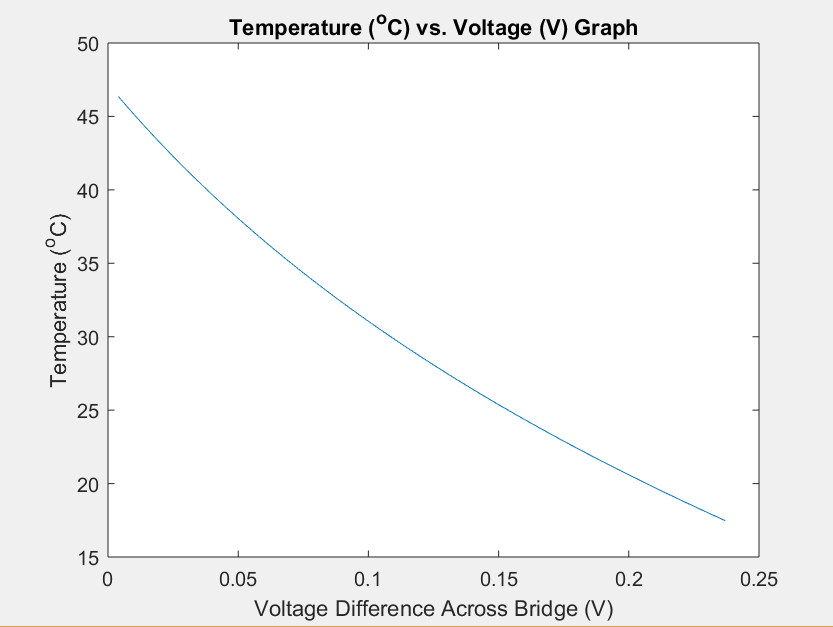
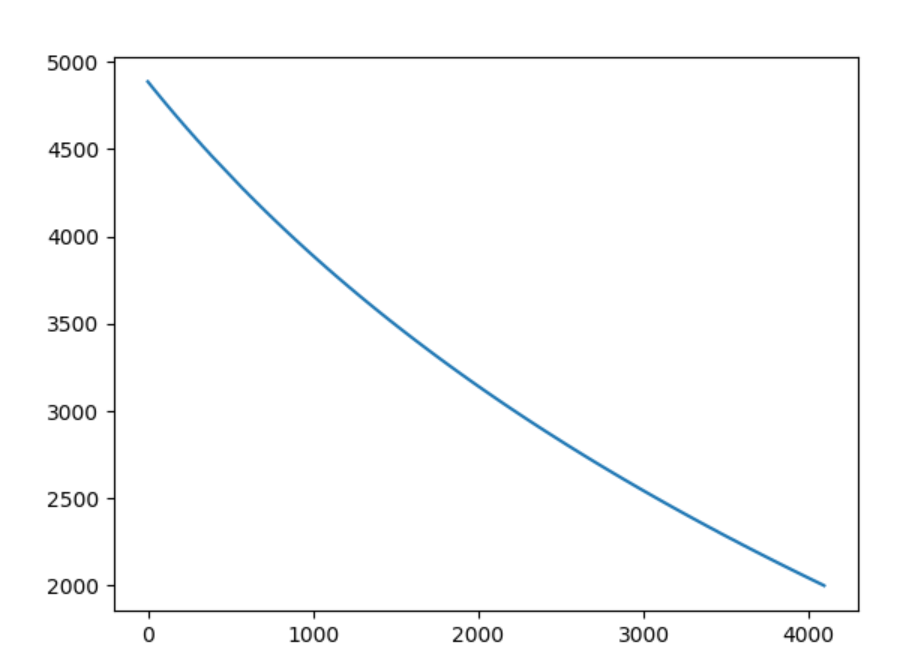
Now, we can substitute the resistance into the rearranged β-term equation and plot the temperature response for our full-range of transducer voltage signals as such:



This table can then be discretized to generate function *f* as follows to generate a 4096 element look-up table that will map the values of the ADC to the temperature measured by the thermistor.

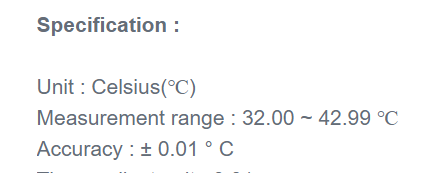


This code generates the temperature vs. voltage plot except it is discretized:



**Calib.h**

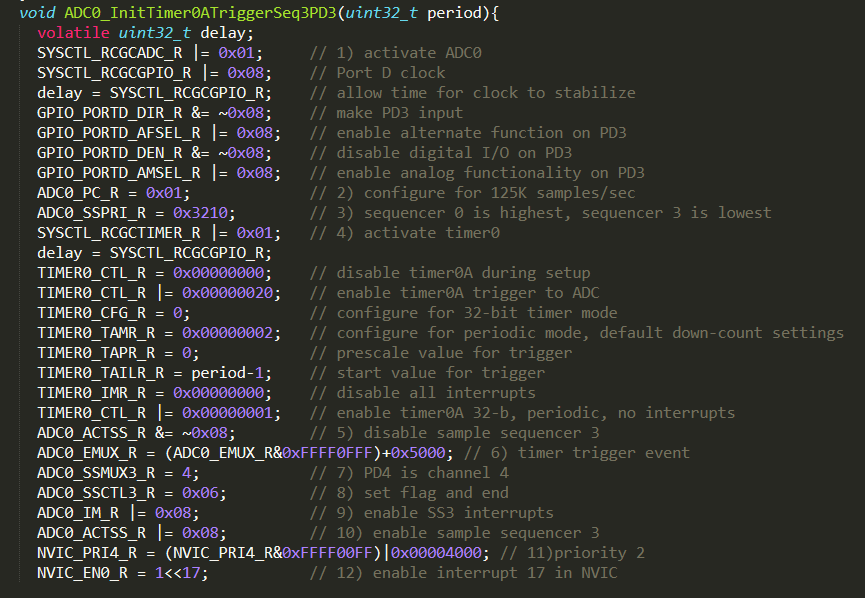
When implemented on the micro-controller, to achieve a similar curve as shown above by using the look-up table, the function had to be implemented with an offset of in order to achieve the same temperature as the following thermometer under the same conditions:

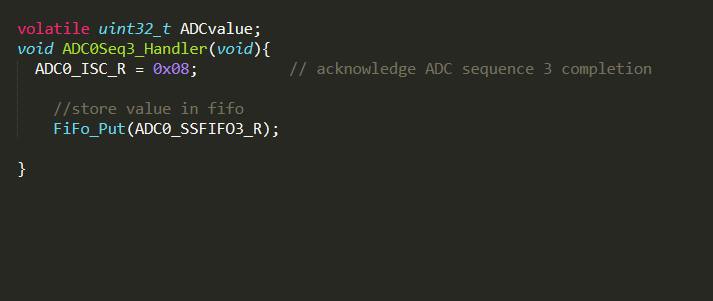


[thermometer link](https://www.banggood.com/Loskii-YI-202B-Digital-Women-Baby-Basal-Measuring-Ovulation-Thermometer-Silicone-Probe-Soft-Head-p-1258239.html?gmcCountry=US&currency=USD&createTmp=1&utm_source=googleshopping&utm_medium=cpc_elc&utm_content=frank&utm_campaign=pla-elc2-us-pc-0409&gclid=Cj0KCQjw5LbWBRDCARIsALAbcOcWSI3OClWQLlD1o_BVa0QJLKGi7xjpGHD5vWhSKhqbrSSOhD5MY74aAsSqEALw_wcB&cur_warehouse=CN)

This offset was stored as a #define in calib.h

**ADC.c and ADC.h (check attached files to .zip folder)**





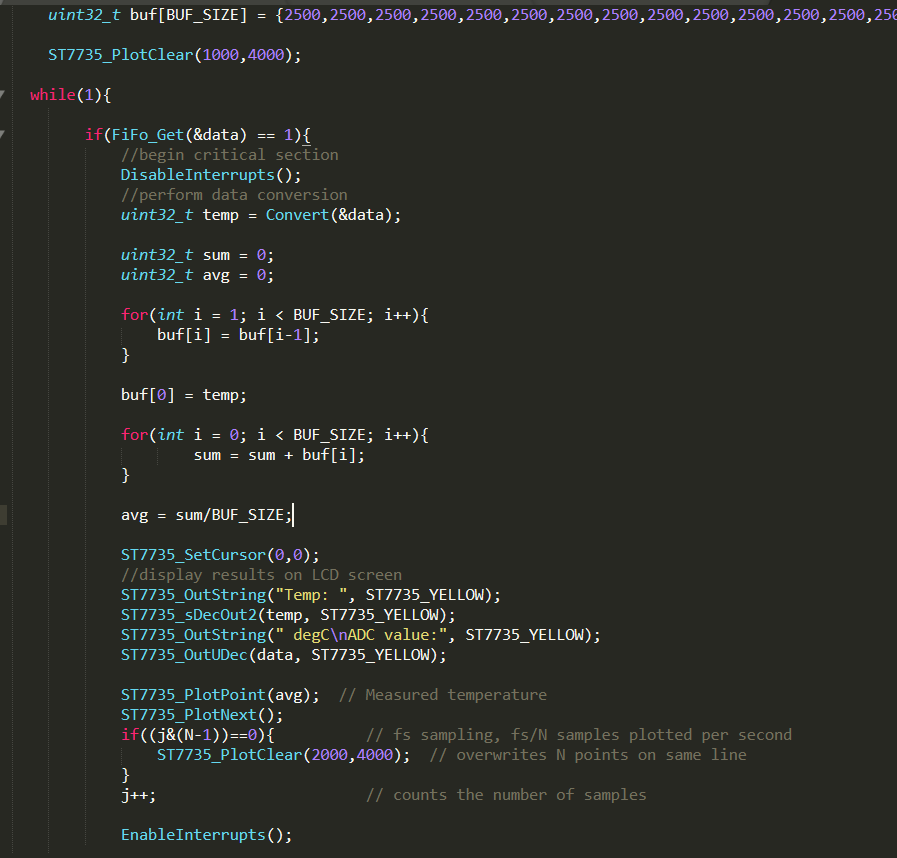
The key to the ADC conversion process lies in triggering the ADC with a general-purpose timer. When the conversion process is finished, the value of the ADC is put into a software FIFO for later access by main.c

**Main.c**

Main performs the initialization routine to begin collecting samples at 1kHz.



Before plotting the temperature, the values are fed through a digital low-pass filter implemented as the average of the most recent 50 values in order to cause gain drop-off after 10 Hz.



1. **MEASUREMENT DATA**

**Waveform Analysis**

**Static Circuit Performance**

**Dynamic Circuit Performance**

**Accuracy**

**Reproducibility**

1. **ANALYSIS AND DISCUSSION**