Closed Loop Systems

Jake Fraser and Tomas Uribe
Rowan University

December 1, 2017

1 Overview

The goal of any electronic control system is to measure, monitor, and control a process. A closed loop system is one way to accurately monitor the output of a system, and use it as a "feedback" to control the system in real time, in a self sustained, closed system. In this situation, the temperature of a five volt regulator is measured and is fed into the system that controls the speed of a fan. Using the temperature measurement as an input, the fan is pointed at the five volt regulator to control the speed of the fan and ultimately, the temperature of the regulator. Through UART communication ports, the user is able to enter a desired temperature and the closed loop system adjusts the temperature and is able to sustain it.

2 Introduction

2.1 Pulse Width Modulation

Pulse width modulation is a term used to describe a certain type of digital signal. In most cases, a PWM is a square wave that varies frequency in order to change the power being used on a load. Some common applications for a PWM include dimming control on RGB LEDs, varying the speed of a servo motor, as well as varying the speed of a fan. In order to allow for more functionality, the PWM allows the user to vary how much time the input signal is high for. While the signal can only be high or low (because its a square wave) at any time, the proportion of time the signal is high versus low can be changed. Below, various example signals for a PWM can be seen, each one with a different duty cycle. Duty cycle is the actual ratio of time that relates the length of analog high and low in the signal. A duty cycle of 50 percent means that the signal is high for the same amount of time it is low in one period.

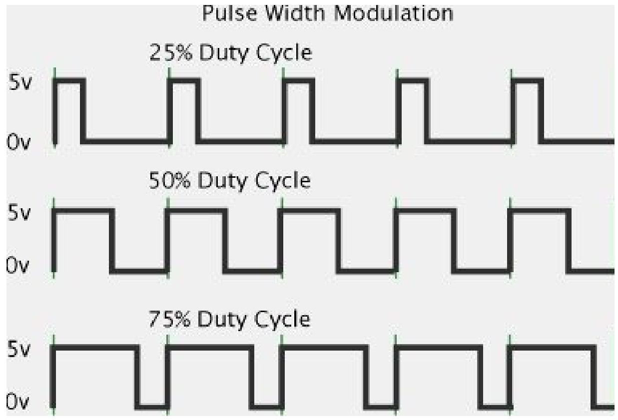


Figure 1: A PWM signal with varying duty cycles.

From this, it can be seen that in order to power a load at different speeds, the PWM will actually turn the device on and off at different speeds, constantly changing the state of the load. In order to do this in code, a timer is most often used. In this case, the capture compare register (CCR) of a timer is given a value to assign a duty cycle to the PWM. When the CCR value is reached, the PWM will change states, effectively changing the state of the load. In the case of this project, a fan is controlled using a PWM. The duty cycle of the PWM is adjusted according to the input of the closed loop system.

2.2 Analog to Digital Converter

An analog to digital converter is a system that converts an analog signal, such as a sound picked from a microphone or the output of a sensor measuring real world data, into a digital signal. In most cases, this digital signal is then processed by a micro controller and the data is used to control another peripheral. An ADC may also provide an isolated measurement such as an electronic device that converts an input analog voltage or current to a digital number representing the magnitude of the voltage or current.

2.3 Closed Loop Systems

A system in which the output will effect the input to control the process is called an closed loop control system. The goal of any electronic control system is to measure, monitor, and control a process. One way to monitor the process is to have a feedback loop in the system. This allows the input of the system to know what is happening at the output and adjust the process in order to get a more desirable output. Oppositely, an open loop system is one in which the input does not know what the output is producing, and will only perform the task it is being asked to do based on the input. The difference between an open loop system and closed loop system can be seen in the general system flowcharts below.

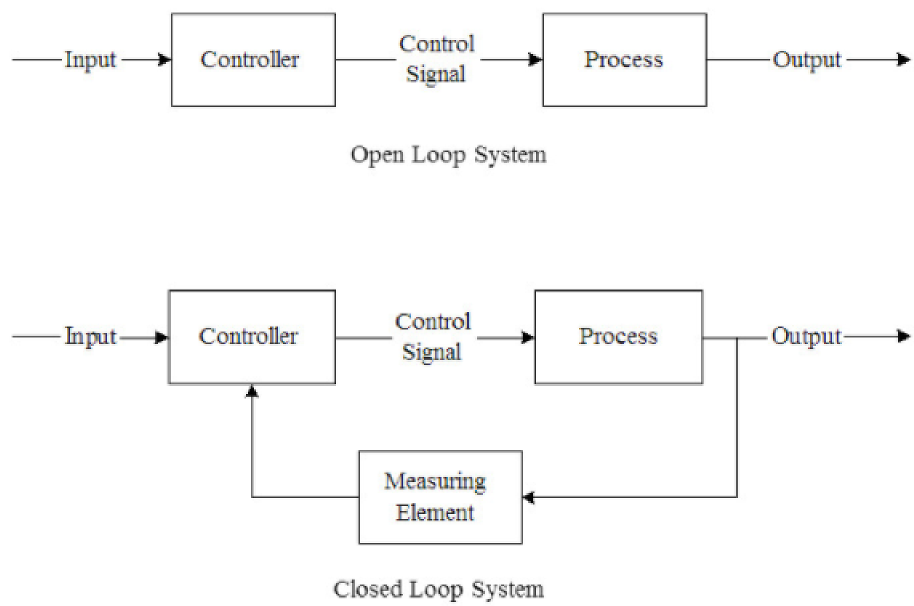


Figure 2: An example system flowchart for an open loop and closed loop system.

Closed loop systems are designed to automatically achieve and maintain the desired output condition by comparing it with the actual condition. To do this, it generates an ”error” signal, which is the difference between the reference and the output signal. The use of this feedback error signal allows the system to function completely isolated from any other user input. The accuracy of the output highly depends on the path of the feedback, and in cases where a short route is taken, the output can be fairly accurate. Because of this, closed loop systems and feedback control are more commonly used than open loop systems.

Closed loop systems have many advantages over open loop systems. One of the primary advantages of closed loop feedback is its ability to reduce error and sensitivity from external disturbances. Closed loop systems will generally improve the stability of a system, automatically correct errors through the feedback signal, and always produce a reliable and repeatable performance.

3 Application

The closed loop system that was designed and built in this experiment uses a temperature sensor to measure the temperature of a 5 V regulator. This measurement is then sent to the MSP430G2553 micro controller which controls the fan to adjust the temperature of the 5 V regulator to the desired temperature. The system diagram for the closed loop fan system can be seen below in Figure 3.

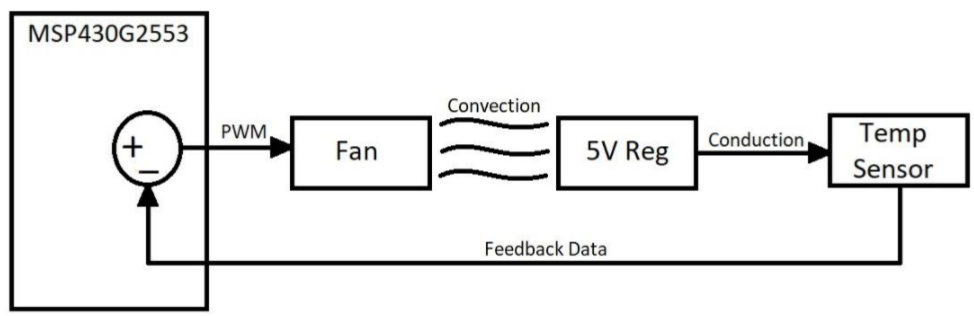


Figure 3: The system flowchart for the closed loop fan system.

As shown in the flowchart above, the fan is indirectly controlled by the temperature of the 5 V regulator. The system begins when the user inputs a desired temperature through the UART serial communication. When this happens, the system automatically begins to measure the temperature of the 5 V regulator. It does this by taking in the output voltage of the temperature sensor through the analog to digital converter on the micro processor. This signal is considered to be feedback in this specific system. The temperature sensor is placed adjacent to the 5 V regulator in order to get an accurate reading.

When this is completed, the MSP430G2553 will use the value stored in the ADC to produce an accurate PWM duty cycle for the fan. It does this by using linearized data from an open loop experiment. In this experiment, temperatures were measured at different PWM duty cycles. From here, linearization equations were derived from the data. These equations are used to derive an appropriate PWM duty cycle for each desired temperature.

Once a proper duty cycle is produced for the PWM, the fan is pointed at the 5 V regulator to cool it down to the appropriate temperature. The temperature is measured and sent back to the processor to further adjust the duty cycle of the PWM. This is constantly repeated until the current temperature matches the desired temperature entered in by the user. Once this condition is met, the system will maintain the current temperature. This method reduces the oscillation of temperature when a steady state is met, and is able to produce a reliable and accurate system. The system is able to react to outside disturbances because the speed of the fan is dependent on the temperature of the regulator. Therefore, if there is a disturbance on the fan, the temperature will increase and the fan will adjust to compensate.

The schematic from of the closed loop system relies on multiple power op amps to apply the appropriate voltages to each component. This is to reduce the amount of power supplies needed to run this system as well as to appease our lord and savior, Phil Mease. The schematic diagram for this system can be seen below in Figure 4.

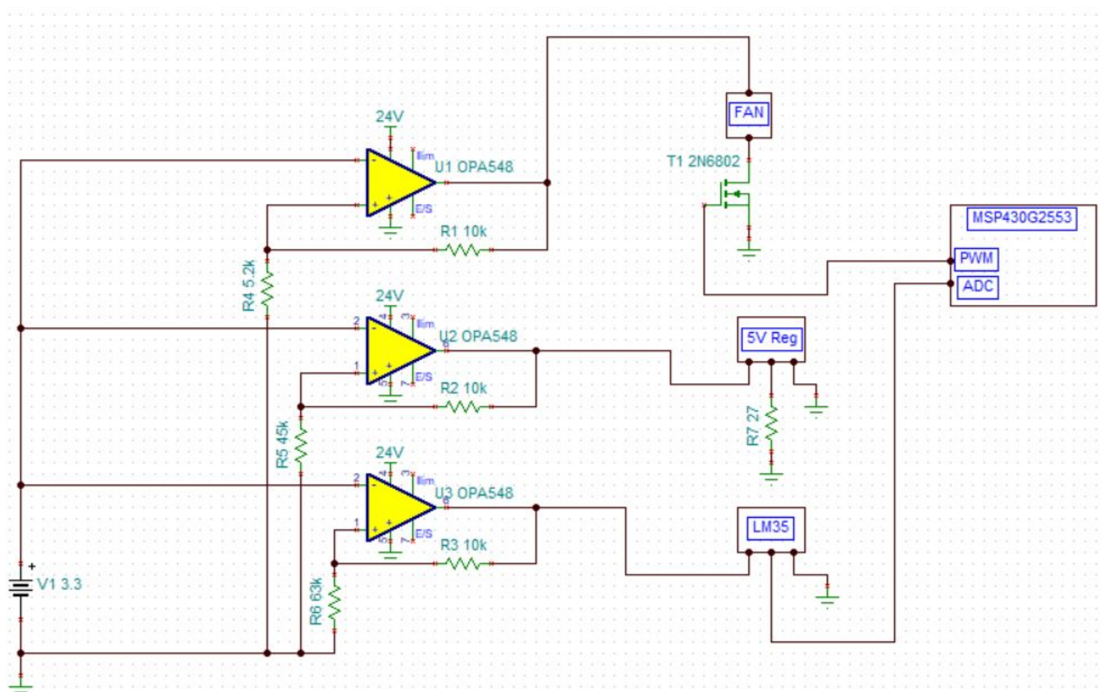


Figure 4: The schematic diagram for the closed loop fan system.

Each op amp in this system will amplify an input of 3.3 V to achieve an appropriate output voltage for the device it is powering. Therefore, each op amp is configured in a non inverting amplifier setup, each with a different gain. The devices powered from the op amps include the temperature sensor, the 5 V regulator, and the fan. Since the fan requires the most power, the rails of each op amp were given 24 V, the appropriate voltage for the fan.

The positive terminal of the fan is connected to an NMOS which acts as a switch. This NMOS takes in the PWM signal from the MSP430 as an input in order to control when the fan is on, and how long it is on for. The 5 V regulator has an input of 18 V and automatically outputs 5 V. Because the regulator has to dissipate so much power, this device will heat up fairly quickly. Therefore, the temperature sensor is placed on it to measure its temperature. The temperature sensor outputs a voltage which if multiplied by ten, is converted to degrees Celsius. Therefore, the output of the temperature sensor is connected to the ADC pin of the MSP430. From there, the MSP430 will convert the hex value taken in, back to a decimal value that the user can read. The current value that the sensor outputs is displayed to the user in decimal through UART, and is sent back to the processor in hex as a feedback signal.

4 Functionality

4.1 Hardware

4.1.1 Closed Loop System Circuit

The circuit that was designed for this portion of the lab was built using three OPA548 power op amps to amplify a 3.3V input. Since the voltage regulator, temperature sensor, and fan all had different operational voltages the 3 OpAmps were used to provide each component with the voltage necessary to function. In order to properly operate the fan, a MosFet switch was implemented using a 2N7000 NMOS. By using this method the fan was able to simulate the duty cycle of the PWM signal that was applied to the gate of the NMOS. This signal continuously turned the NMOS on and off allowing the fan to function at a speed that correlated to the duty cycle of the PWM. As for the voltage regulator, a 27 ohm power resistor was attached to its output as a load in order to allow it to heat up faster. The temperature sensor was then used to read the temperature around the voltage regulator in order to allow the user to read the temperature of the regulator through UART. The circuit used for the closed loop system can be seen below in Figure 5

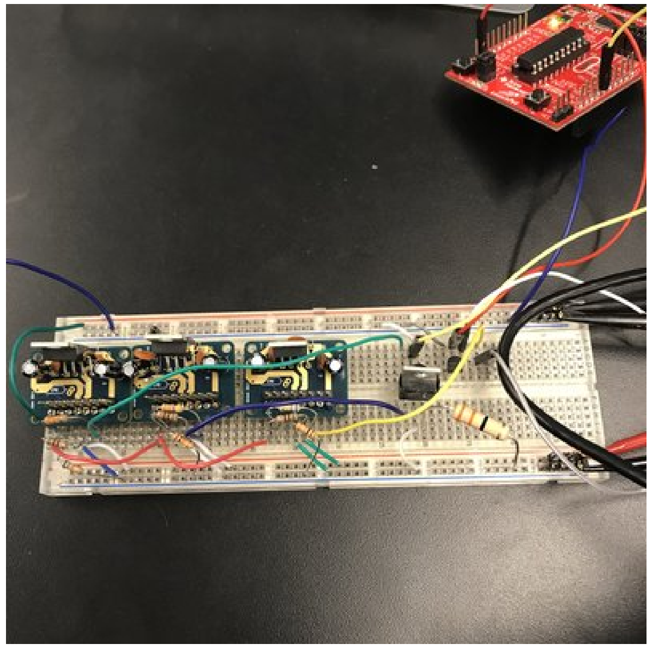


Figure 5: Closed Loop Control System

4.1.2 Devices/Sensors

- **MSP430G2553 Microcontroller**
  The MSP430G2553 was chosen for its versatility and furthermore, its replaceability. Since this processor can be removed from the dev. board and is replaceable, it was seen as favorable because the processor was likely to get damaged using such high voltages.

- **OPA548 Power OpAmp**
  This device was chosen due to its ability to supply up to 60V. Additionally, this device is unipolar, meaning if it had rails of +25 and ground, it would still allow for an input voltage of 3.3V, unlike the TL072.

- **2N7000 NMOS**
  The decision to go with this specific MosFET was made simply because this FET is able to withstand a maximum gate-to-source and drain-to-source voltage of $\pm 20V$ and $\pm 60V$ respectively. Since the circuit designed for the closed loop system only supplied the NMOS with a V_{GS} of 3.3V-0V and a V_{DS} of 24V, the specifications for the 2N7000 NMOS were deemed acceptable.

- **LM35 Temperature Sensor**
  The LM35 was chosen because of its ability to give an accuracy of $\pm 1\,°C$ within a range of $-55\,°C - 150\,°C$.

- **LM7805 Voltage Regulator**
  The LM7805 was arbitrarily chosen since its output was not needed. The main purpose of this device was simply to radiate heat which could then be read by the temperature sensor.

4.2 Software

4.2.1 Pulse Width Modulator

The main function of the software for the closed loop control system was to produce a PWM signal with a duty cycle depending on the desired temperature set by the user. Through UART, the user was able to send a desired temperature to the MSP430G2553. Within the code, the received temperature was then passed to a function called setDutyCycle which correlated the given temperature to an appropriate duty cycle. This duty cycle was then used in a function called setPWM which was used to set the capture compare register that was output to the NMOS switch, controlling the fan. In order to correctly model the data, an Excel spreadsheet was made comparing the duty cycle of the PWM and the temperature read by the temperature sensor in an open loop setup. The data plotted in Excel can be seen below in Figure 6

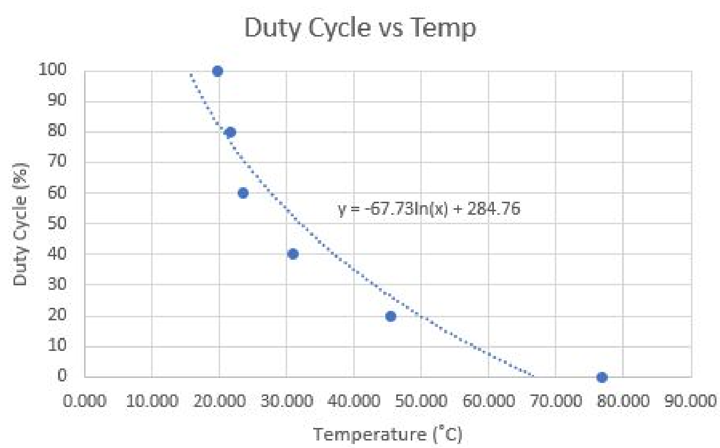
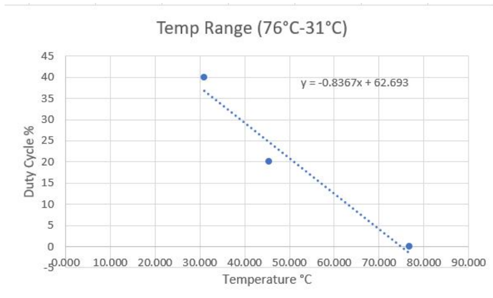
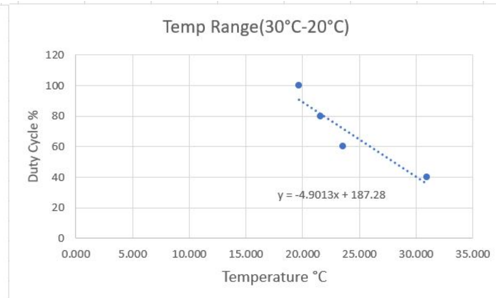


Figure 6: Temperature Vs. Duty Cycle

This data was then split up into 2 different data sets and then independently linearized. The temperature regions that were determined by the graph above were 76 °C- 31 °C and 30 °C-20 °C. Each produced a line of best fit of y = -0.8367x + 62.693 and y = -4.9013x + 187.28 respectively. This process allowed the processor to determine at what duty cycle the PWM had to be set, depending on the temperature that the analog to digital converter read. Ultimately, the linearization of data was what caused the fan to change speeds in order to let the voltage regulator reach the temperature assigned by the user. Figure 7 below depicts the lines of best fit of the 76 °C-31 °C and 30 °C-20 °C temperature regions.



(a) Temperature dataset between 76 °C and 31 °C



(b) Temperature dataset between 30 °C and 20 °C

Figure 7: Linear relationship of temperature and duty cycle

Within the setDutyCycle function, the linear relationship between duty cycle and temperature was used in order to predict the speed at which the fan should spin over a wide range of temperatures. For example, when a temperature value was passed to the setDutyCycle function, conditional statements were used to determine what equation to use depending on what range the given temperature fell within. Figure 8 depicts the setDutyCycle function.

```
void setDutyCycle(int temp){
    if(temp > 76){
        setPWM(10);
    }
    else if(temp > 31){
        long pwm = ((temp * -84) / 100) + 63;
        setPWM(pwm);
    }
    else if(temp > 20){
        long pwm = ((temp * -49) / 10) + 187;
        setPWM(pwm);
    }
    else{
        setPWM(100);
    }
}
```

Figure 8: SetDutyCycle Method

4.2.2 Analog to Digital Converter

Previously, for the MSP430 to process the temperature readings from the LM35, they had to be passed through the ADC on the processor. This converted the analog voltages read by LM35 into digital signals that the MSP430G2553 could use. Since the values converted by the ADC directly correlated to the temperature of the voltage regulator, it was essential to retrieve the values that were stored in the ADCs memory and transmit them to the user through UART. For this reason, a function called ADC10Interrupt was written. Within this function an array called tempBuf added the value that was stored by ADC10MEM every time the ADC interrupt triggered. Once the interrupt fired a variable called tempBuf_index would increment until its value reached 10 which would cause the values in tempBuf to be added together and assigned to a variable called average. This variable was then divided by 10 in order to find the average of the last 10 temperature readings. However, the value stored in average only represented a fraction of the reference voltage of the ADC (3.3V), and furthermore had to be converted in order for the average to represent the actual temperature reading. Since the ADC has a resolution of $\frac{V_{ref}}{1024}$ and the LM35 has a conversion rate of $\frac{10mV}{°C}$ the average was first multiplied by 330 and then shifted to the right 10 times to simulate a division of 1024. This section of code is depicted in Figure 9

```
if(tempBuf_index < 10){
    tempBuf[tempBuf_index] = ADC10MEM;
    tempBuf_index++;
}
else{
    long average = 0;
    int i = 0;
    for(i = 0; i < 10 ; i ++){
        average += tempBuf[i];
    }
    average /= 10;
    long temperature = (average * 330) >> 10;
```

Figure 9: ADC Portion of Code used to average sets of 10 temperature readings

The most important function of the ADC10Interrupt method was to account for any type of disturbance. This means that if any external factor effected the temperature of the voltage regulator the code in this function had to be able to counteract these effects by accounting for the error. The code written to account for error can be seen below in Figure 10

```
int newtemp = 0;
long error = 0;
error = (goaltemp - temperature);
newtemp = (k*error) + goaltemp;
setDutyCycle(newtemp);
UCA0TXBUF = temperature;
tempBuf_index = 0;
```

Figure 10: Error Correcting Portion of Code

The error was calculated by subtracting the temperature assigned by the user from the average temperature read by the ADC. Once the error was established it was multiplied by a constant k, in order to reach the desired temperature sooner. This product was then added to the desired temperature and output to the setDutyCycle function to adjust the speed of the fan. This functionality allowed the closed loop system to maintain the temperature set by the user and furthermore removed the chance of external factors affecting the system.

5 Testing

The code written for the closed loop system was tested at four different temperatures. The trials were deemed successful if the system could fall within ±3 °C and did not oscillate significantly within the accepted range of temperatures. The first and largest temperature that was tested was 62 °C. The voltage read by the LM35 was measured on the oscilloscope and the results can be seen in Figure 11

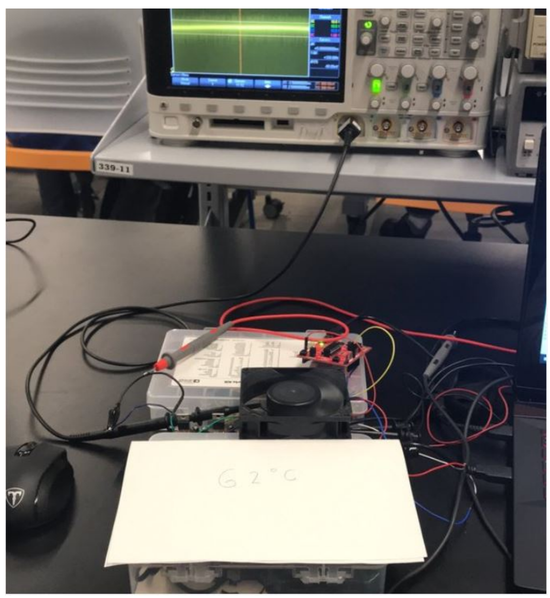
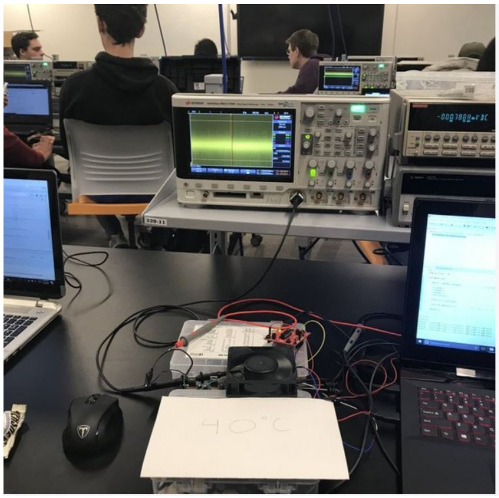


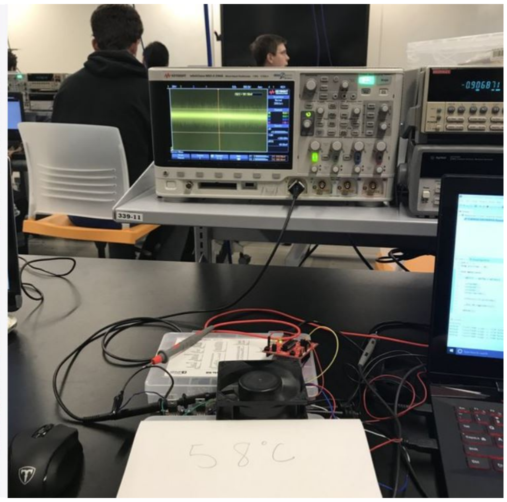
Figure 11: Temperature Readings When Code was Set 62 °C

On the oscilloscope the cursors were use to set the ±3 °C threshold in order to identify if the closed loop system functioned properly. As Figure 11 demonstrated, at 62 °C the system was successful.

The next two tests consisted of setting the temperature of the system to 40 °C and 58 °C. This was done to examine how well the system was able to increase the speed of the fan in order to drop in temperature and then lower the speed to allow the regulator to heat up. The results of both the 40 °C and 58 °C test can be seen in Figure 12



(a) Temperature Readings When Code was Set 40 °C



(b) Temperature Readings When Code was Set 58 °C

Figure 12: Test 2 and 3 of Closed Loop System

In both tests the temperature read by the oscilloscope fell within the acceptable range. Some oscillations were observed but the temperature did not fluctuate significantly, therefore the tests were deemed successful.

The last test that was performed called for the temperature to be dropped to 50 °C and for the fan to be set on a 45° incline. This was done to see if the system could correct itself without the fan being directly over it. The results are shown in Figure 13

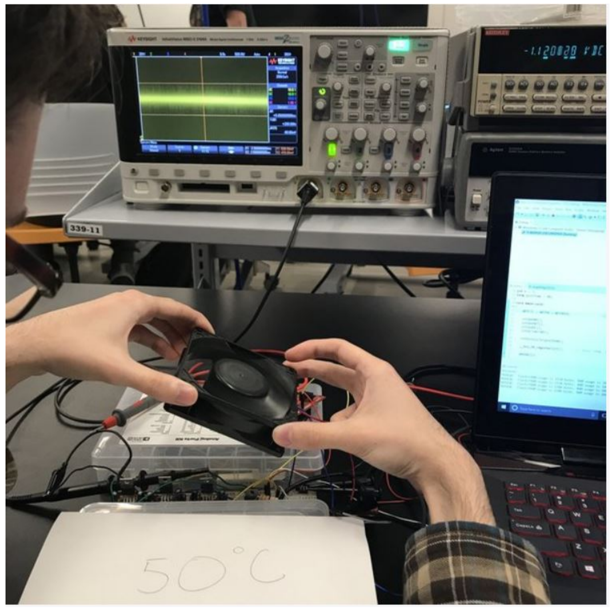


Figure 13: Temperature Readings When Code was Set 50 °C

This test revealed that the closed loop system was able to correct itself regardless of the position of the fan. However, the temperature oscillated significantly within the accepted temperature threshold because of the algorithm that was written for the setDutyCycle function. Since the relationship between the duty cycle and the temperature was linearized with the fan close to the regulator, the equations used in the setDutyCycle function did not exactly correlate as the temperature approached the desired value. This resulted in the processor trying to quickly speed and slow down the fan in order to reach the desired temperature, causing oscillations.

The most important observation that was made for every single test was that the processor was able to get the voltage regulator to a steady state temperature within a minute. In this system it was very favorable to produce a desired steady state temperature fairly quickly as it meant that the system could reach homeostasis faster and avoid any malfunctions. The reason that the system was able to change temperatures so rapidly was because of the constant, k, which was multiplied by the error. This constant tricked the system into thinking that there was more error than was actual produced which caused the duty cycle of the fan to dramatically change to accommodate for the large error. Ultimately this caused the regulator to reach its intended temperature at an accelerated rate.

6 Conclusion

Through the completion of this project, a closed loop fan system that measures, monitors, and controls the temperature of a 5 V regulator was designed and built. This system uses the output of a temperature sensor that is very close in proximity to the regulator as a feedback signal to control the duty cycle of a PWM. The PWM signal is then fed into a fan that blows air onto the 5 V regulator. This adjusts the temperature and the process is repeated until the temperature of the regulator matches the desired temperature input by the user. This feedback signal characterizes the closed loop system methodology, and is crucial to producing an accurate, reliable, and reproducible system. A closed loop system such as this is ideal in a situation where system isolation and the reduction of external disturbance effect is desired.