Milestone 2: Closed Loop Temperature Controlling System

Shane Price and Kieran O'Connor
Rowan University

December 2, 2018

1 Design Overview

This project involved controlling the temperature at a steady state for a small space. The task was to utilize a thermistor to read a temperature to then set a desired temperature and hold steady at that temperature through the use of an external CPU fan. The communication of sending and receiving temperatures was done through UART, and for increasing and decreasing the temperature with the fan was done by using a fan set to a gradual PWM. The end goal was to increase temperature up to 60 degrees celsius from room temperature and hold that temperature steady, then to drop temperature to 30 degrees celsius and hold, and increase and hold one more time; the complicated part came with keeping the fan from audibly oscillating the fan speed.

1.1 Design Features

- Receive and Transmit temperatures over UART
- Change the temperature the thermistor is reading based on PWM of a CPU fan
- Set the PWM of a fan to hold a steady temperature without hearing the fan change speeds
- Use a 5V regulator to heat a thermistor

1.2 Featured Applications

- HVAC
- Cooling Systems
- Cooking Ovens

1.3 Design Resources

In this project the team used two resources to help build our code, one was a the TI Resource Explorer for echo information, configuring ADC, and our previous code to configure hardware PWM.

http://dev.ti.com/tirex/#/Device/MSP430F5529/?link=Software%2FMSP430Ware%2FDevices%2FMSP430F5529%2FPeripheral%20Examples%2FRegister%20Level%2FMSP430F55xx_uscia0_uart_03.c http://dev.ti.com/tirex/#/Device/MSP430F5529/?link=Software%2FMSP430Ware%2FDevices%2FMSP430F5529%2FPeripheral%20Examples%2FRegister%20Level%2FMSP430F55xx_adc_05.c

https://www.embeddedrelated.com/showarticle/420.php

The code for the finished product of the project can be found below.
https://github.com/RU09342-F18/introtoembedded-f18-milestone2-russells-muscles/blob/master/Milestone2

1.4 Block Diagram

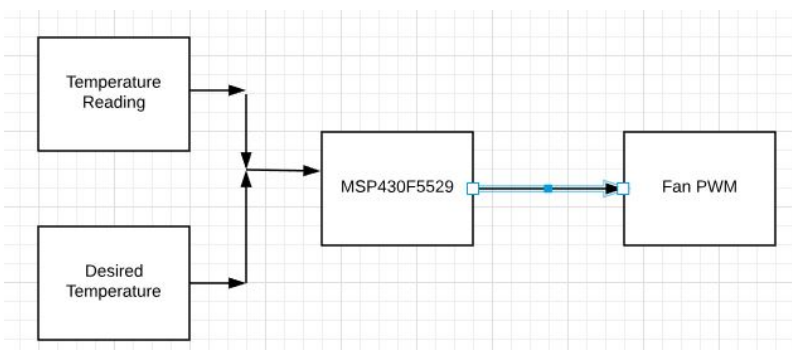

Figure 1: System Block Diagram

1.5 Board Image

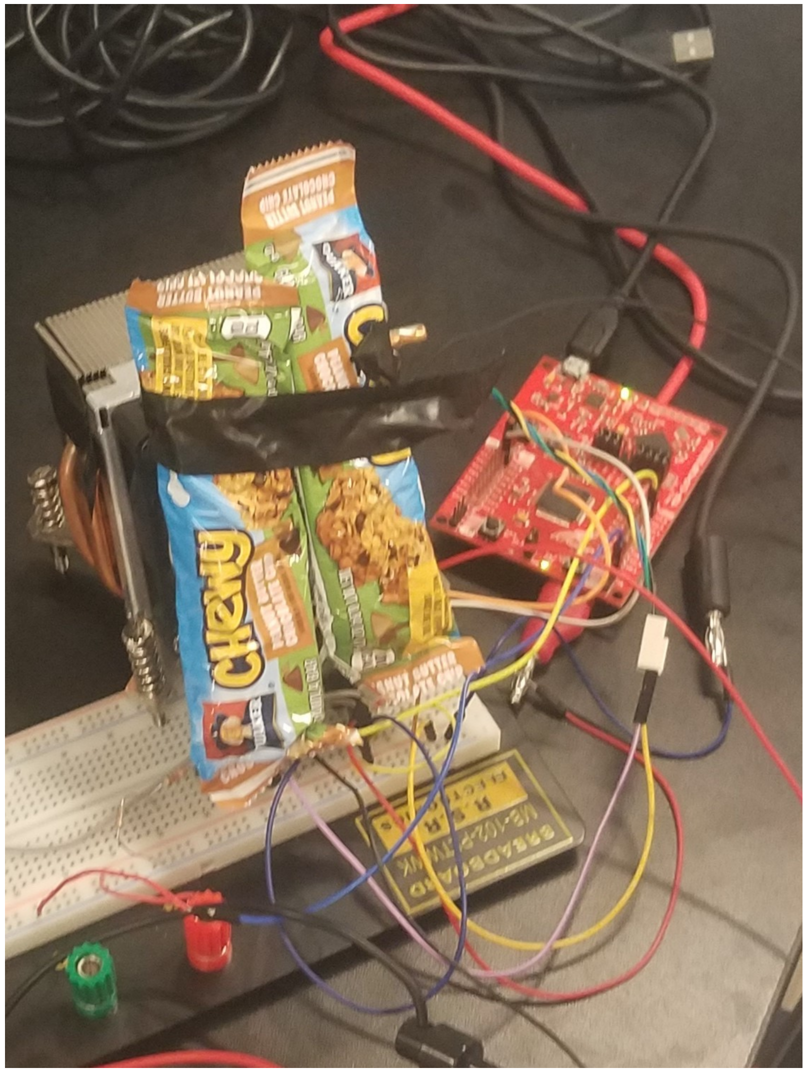

Figure 2: MSP430F5529 Launchpad CPU Fan with High Tech Air Deflectors

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Dissipation of 5V Regulator	2W	Dissipation used to heat thermistor
Input Voltage of Voltage Divider	3.3V	Voltage used for the voltage divider supplied by the MSP430F5529
Input Voltage/Current of Low side Switch	12V, 0.045A	Input voltage and current of the low side switch determined by the 12V fan used.
Input Voltage of 5V Regulator	12V	Input voltage of the 5V Regulator same source as low side switch

3 System Description

This project is trying to control to the temperature in a small area. The project was able to cool down to room temperature and heat up to roughly 70 degrees celsius at an absolute max.

3.1 Detailed Block Diagram

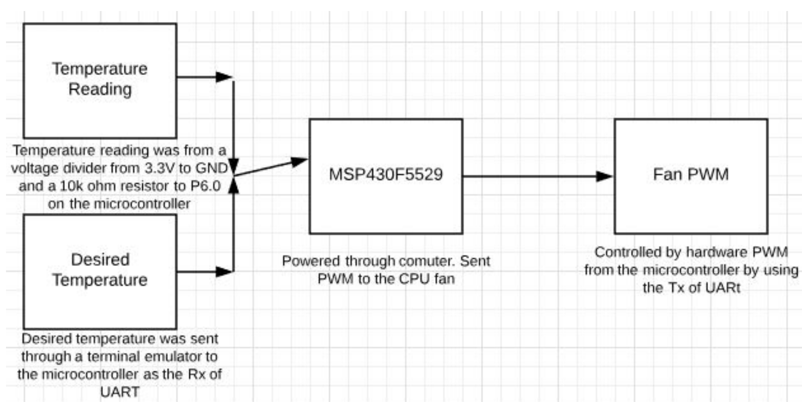

Figure 3: Detailed System Block Diagram

3.2 Highlighted Devices

- MSP430F5529

  - Processes the desired temperature and the current temeperature, then sends a PWM to the CPU fan to reach the desired temperature.

- Temperature Reading
  - The temperature was read by using a thermistor and a 10k ohm resistor to make a voltage divider. The voltage was read, converted to a resistance and using a linear approximation of the Stein Hart-Hart equation a temperature was calculated.

- Heating Element
  - In order to heat the thermistor, a 5V regulator was used with 12V on the input and 10 12.5 ohm resistors in parallel, dissipating approximately 2W of power. With the thermistor touching the conducting pad of the 5V regulator.

- Low Side Switch
  - the low side switch was used to control the voltage of the CPU fan.

3.3 MSP430F5529

This TI micro-controller used in the system processes the code and the input temperature from a terminal emulator. Using this component's timer and two capture compare registers the timer so the desired PWM was possible. The clock used for each of the two timers was the Sub Main clock which is a 1 MHz clock, in up mode.

4 SYSTEM DESIGN THEORY

This project has four main components. The microcontroller, voltage divider, 5V regulator, and the low side switch to the CPU fan. Each are described below on their functionality and how they each interact in the system.

4.1 MSP430F5529

The main driver of this project is the MSP430F5529 micro-controller. The micro-controller receives a desired temperature from a terminal emulator then processes that information compared to the temperature the thermistor is reading to determine the PWM of the CPU fan. If the desired temperature was lower than the temperature that was being read, the fan would gradually turn on until it comes in range of the desired temperature and gradually slows the fan down to hold that temperature. If the desired temperature was greater than the current temperature the PWM would gradually turn off the CPU fan and slowly turn it on once in range of the desired temperature to prevent it from going beyond the temperature.

4.2 Voltage Divider

The Voltage divider was used to get a voltage reading between the 10k ohm resistor and the thermistor to then calculate the temperature at the thermistor. The temperature was calculated by determing the resistance of the thermistor as determined by the input voltage, then using the Stein Hart Hart equation to determine the temperature at that point in time.

4.3 5V Regulator

The 5V regulator was used to generate heat to heat up the area around the thermistor. This was done by putting a load from ground to the output of the 5V regulator while applying 12V to the input. This was able to heat the thermistor up to about 70 degrees celsius when the thermistor was secured to the regulator.

4.4 Low Side Switch

The low side switch was just as crucial as the other components as it controlled the PWM of the CPU fan. The low side switch controlled the voltage going to the CPU fan, which was dictated by the MSP430F5529. The fan would turn on and cool the thermistor, high tech air deflectors, Chewy Granola Bars, were necessary to direct the onto a singular point of the thermistor to ensure proper cooling.

5 Getting Started

Once the device is programmed and connected to the voltage divider, the 5V regulator is connected, and drain of the low side switch, the next step is to send a desired temperature to the microcontroller and have that information processed. This is done by following the information below.

5.1 Terminal Program

In order to send information to the micro-controller you will need a terminal program on your computer. The terminal program used by the team was Real Term, once on the program specify the port the controller is connected to. The controller should be connected using a UART communication cable to each of the required pins; 3V3, GND, P3.3 (RXD), P3.4 (TXD). Using real term send the desired temperature to the system, an example would be 60. This example temperature would be processed in the microcontroller and turn off the CPU fan to allow for the thermistor to heat up and then gradually turn the fan back on to hold that temperature. The final test to ensure it is completely working, the current temperature read by the thermistor will be transmitted back to Real Term. This is the value the microcontroller will use to determine the PWM.

6 Getting Started Software

As stated before the main way to communicate with the system is by using a terminal program using UART. UART was used due to its simplicity to implement with the MSP430 boards as well as it being a common form of data communication.

6.1 Communicating with the Device

The devices communicate using UART, UART is a standard simple form of communication for data. UART consists of four wires, two for power and ground and two for transmitting and receiving. These are unidirectional data lines, meaning that they will not function properly if reversely connected.

The UART transmitted over USB telling the computer what the temperature read was and received the desired temperature sent using Realterm.

6.2 PWM on Device

A necessary function on the device was using the hardware PWM. This was used to set the timers to a specified frequency (BAUD rate of 9600) to allow the LED to be toggled at certain intervals. The BAUD rate was specified to be 9600, from the requirements of the project. The interval for the LED to be toggled was the frequency of the clock used (SMCLK)/9600; 1MHz/9600.

6.3 Device Specific Information

The micro-controller used was the MSP430F5529 used for its capable analog to digital converter and Timer. Time 1 was used with the SMCLK in Up Mode with no divider with a frequency of 1 MHz. The pins used for the ADC was P6.0 and the timer pin used for the PWM for the fan was pin P2.0 for the PWM output. The green LED used for testing was P4.7 and the UART pins used included P4.4 for output and P4.5 for input.

7 Test Setup

Testing was done by using the terminal program, Real Term. Necessary steps to take before testing was to specify the ports of the device, using device manager and input that to Real Term. Then sending a desired temperature.

7.1 Test Data

The data collected for this project was to see the output of the code which was displayed by the oscilloscope and the terminal program. Once a desired temperature was set and the range, +- 3 degrees celsius, was determined in voltages and set on the oscilloscope the voltage should remain within the range to show proper functionality, as seen in Figure 4. When examining the terminal program, the temperature displayed should remain at the desired temperature with minimal fluctuation. The final test was to tune down the voltage supplied to the fan and 5V regulator and send a new desired temperature and observe it once again hold a steady voltage.

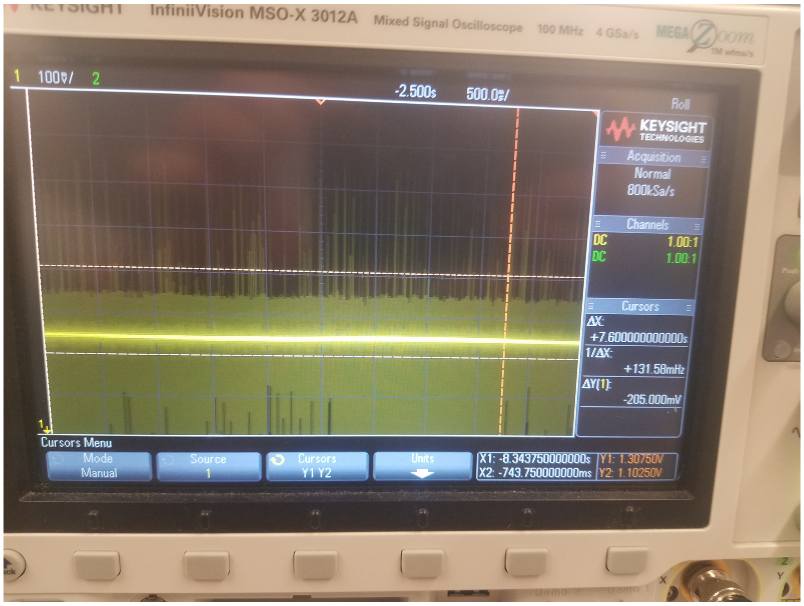

Figure 4: Ideal Voltage Reading within the Bounds

7.2 Schematics

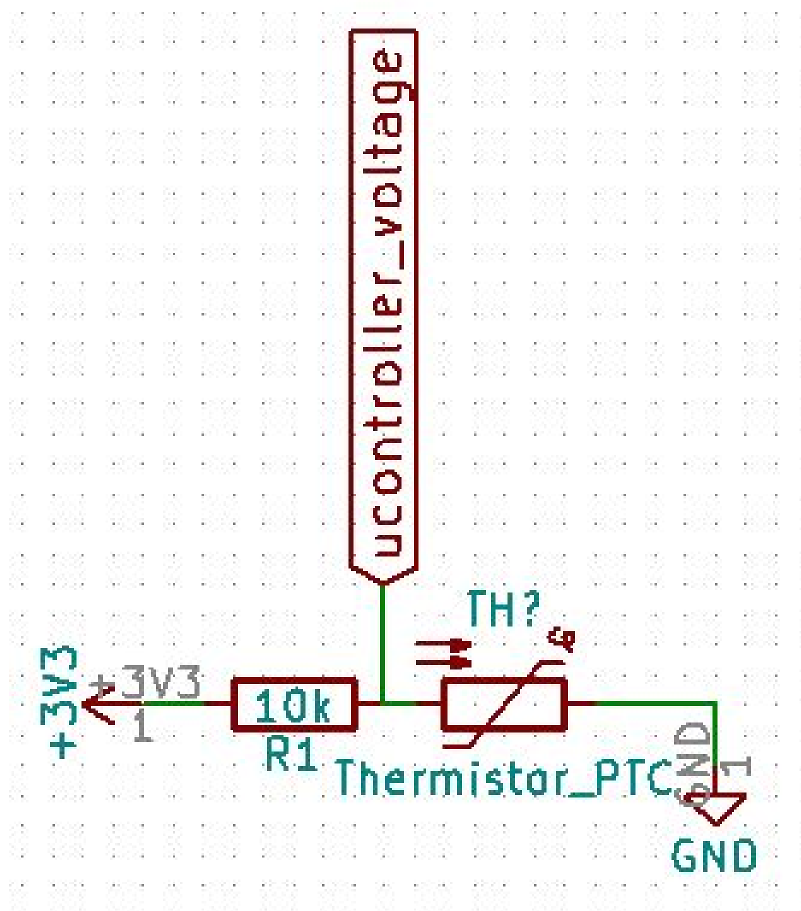

Figure 5: Voltage Divider for Current Temperature Reading

This figure shows how the microcontroller would get an input voltage to determine the temperature at a given time.

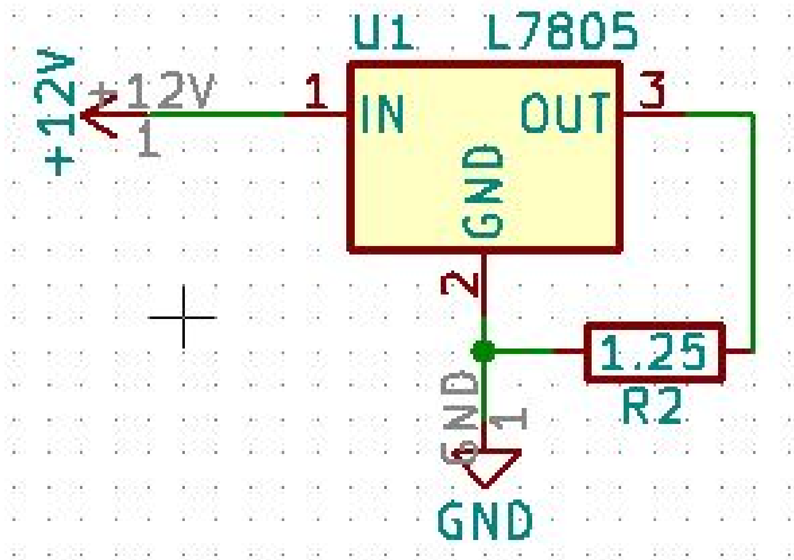

Figure 6: 5V Regulator as the Heating Element

This figure dictates how the thermistor would be heated up, with a load equivalent of 1.25 ohms, 10 12.5 ohm resistors in parallel.

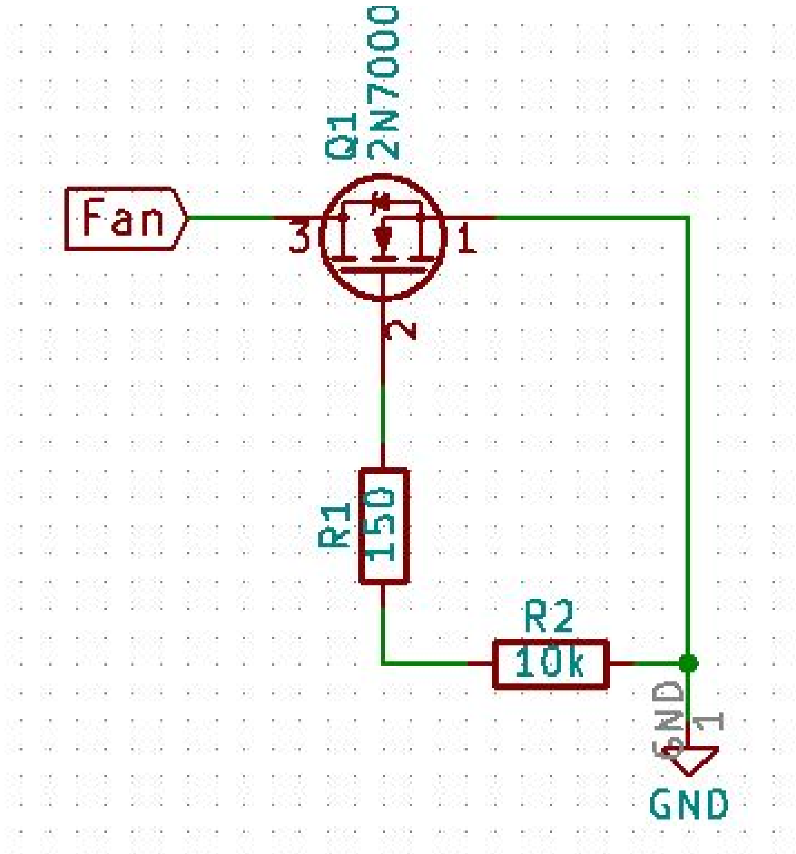

Figure 7: Controlling the PWM of the CPU Fan

This figure dictates how the low side switch controlled the voltage going to the fan from the PWM sent from the microcontroller.

7.3 Bill of Materials

- 2 10k ohm resistor
- 10 12.5 ohm resistors
- 1 MSP430F5529
- 2N7000 mosfet
- Real Term Terminal Program
- 5V Regulator, L7805
- 1 Breadboard

- 1 Thermistor
- 1 150 ohm resistor
- 1 12V CPU Fan
- 2 Peanut Butter Chocolate Chip Chewy Granola Bars

8 References

- Family User Guide
  http://www.ti.com/lit/ug/slau208q/slau208q.pdf
- MSP430F5529 Datasheet
  http://www.ti.com/lit/ug/slau533d/slau533d.pdf
- 2N7000 Mosfet Datasheet
  https://www.onsemi.com/pub/Collateral/2N7000-D.PDF
- TI Resource Explorer UART 9600 Echo Example Code
  http://dev.ti.com/tirex/#/Device/MSP430G2553/?link=Software%2FMSP430Ware%2FDevices%2FMSP430G2553%2FPeripheral%20Examples%2FRegister%20Level%2Fmsp430g2xx3_uscia0_uart_01_9600.c
- L7805 5v Regulator Datasheet
  https://datasheet.octopart.com/L7805CV-STMicroelectronics-datasheet-7264666.pdf
- NTCLE100E3 Thermistor
  http://www.vishay.com/docs/29049/ntcle100.pdf