Milestone 2

Eric Schroeder and Ian Nielsen
Rowan University

December 2, 2018

1 Design Overview

The goal of this milestone is to create temperature controlling system. This is done by controlling the PWM of a fan to maintain a constant temperature on a voltage regulator. A thermistor is used to read the temperature of the voltage regulator to ensure the fan is keeping it at a constant temperature. Along with this the user can enter a temperature via UART that the system will then keep the voltage regulator at.

1.1 Design Features

These are the design features:

- Temperature (C) of a voltage regulator is displayed through UART
- A fan will keep a voltage regulator at 30C when both are powered
- The user can enter a desired temperature through UART that the fan will maintain on the voltage regulator

1.2 Featured Applications

While this design was created with specific specifications in mind it can be used in a wide range of applications.

Featured Applications Include:

- Computer cooling system to keep electronics at certain temperatures
- The fan control can be used on a larger scale for room temperature control
- Safety measures to initiate when electronics are nearly over heated

1.3 Design Resources

Our code can be found on GitHub at the following link: GitHub Code

1.4 Block Diagram

A block diagram showing the connections of the system can be seen in Figure 1.

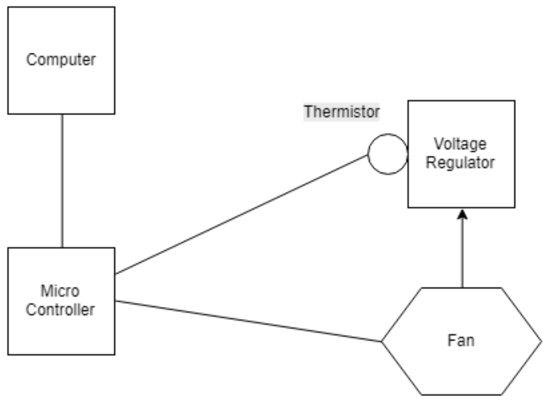

Figure 1: System Block Diagram

1.5 Board Image

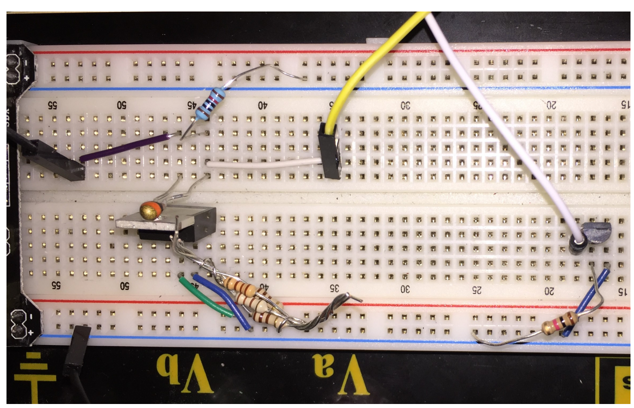

Figure 2: Picture of the circuit on the breadboard

2 Key System Specifications

This project has two main parameters which are used. These two parameters encompass the functionality required to complete this lab. This first parameter is a temperature reader which uses a thermistor in a voltage divider circuit. The second parameter is a temperature setter which allows for the user to choose a temperature to be maintained by the fan PWM.

PARAMETER	SPECIFICATIONS	DETAILS
Temperature Reading	15°C to 70°C	Takes temperature reading of the voltage regulator based off of the voltage reading from the ADC12 port
Temperature Setting	20°C to 70°C	Takes desired temperature of the voltage regulator through UART and controls the PWM of the fan based off of the temperature reading parameter.

3 System Description

The goal of this system is to create a temperature regulating system with a fan and a temperature sensor. The system as a whole uses a voltage divider with a thermistor in order to calculate temperature at the voltage regulator. The system also uses a low-side switch in order to create a PWM driver and control the fan speed. The final part of this system is UART, which sends and accepts temperature information to the computer's UART port.

3.1 Detailed Block Diagram

The voltage divider uses a 10kΩ resistor in series with a thermistor connected from the 3.3V on the MSP430 to the common ground. The MSP430 Pin 6.0 is connected in between the resistor and the thermistor which is the ADC12 port.

The next device in the system is a low-side switch which uses a 2N7000 N-channel MOSFET. The 12V power supply is connected to the positive lead of the fan. The negative lead of the fan is the connected to the drain of the nMOS. The gate is connected to Pin 4.0 of the MSP430 which is the PWM driver. The source is connected to the common ground.

The next device is a UART I/O connection which connects to the UART COM9 or COM8 on the laptop. The UART has a baud rate of 9600 and sends out temperature readings through the TX port of the MSP430 and also accepts desired temperature through RX port on the MSP430.

The final device in this system is the circuitry for the voltage regulator. The voltage regulator is connected to the 12V power supply into the IN port and the OUT is connected to a low resistance resistor which is connected to ground. This is done to draw the most current. The COM is then connected to ground. The effect of this is that the voltage regulator heats up drastically in order to maintain the voltage.

3.2 Highlighted Devices

In order to complete this lab the MSP430 is used as well as the 2N7000 nMOS and a voltage regulator. These components each are used in different parts of the lab explained above. Their detailed use in this circuit is explained in the next subsections.

- MSP430F5529
- 2N7000

3.3 Device MSP430F5529

This system uses the MSP430 processor in order to process, send and receive data relating to the analyzing and cooling the circuit. The processor in this project is used as a node for which the UART code is passed through and is the vehicle for the message to be processed and implemented. This device was chosen as it contains more capture compare registers for a single timer than the other MSP boards available.

3.4 Device 2N7000

The 2N7000 is a nMOS which is used in the low-side switch in order to control PWM. This nMOS is closed until the gate is on. The gate is controlled by P4.0 on the MSP430 and the duty cycle for this gate is decided by the calculations done by the processor. This results in the fan being faster or slower depending on the input from the processor. The drain to source current only flows when the gate is open, allowing for a higher voltage switch to be controlled with a lower voltage input.

4 SYSTEM DESIGN THEORY

This project uses several different theories in order to get the final output which is within 1°C of the desired temperature. The first requirement and calculation that needed to be done with this lab is to calculate the temperature of the thermistor. This used an equation which straightened the curve and corrected for the y-intercept. The second requirement is to calculate a PWM for the fan based off of the temperature information found in the first requirement and PID calculations which were made in the processor.

4.1 Design Requirement 1

The first design requirement was to make a circuit and to create a program which would calculate the temperature of the thermistor. In order to do this we needed to correct for the slope. Since the voltage readings based on the thermistor temperature are exponential the slope needed to be corrected for at different rates at different points in the curve. In order to do this we split the curve into five parts and corrected for each of the slopes at each segment using if statements in order to check which part of the curve is currently at.

4.2 Design Requirement 2

The second design requirement was to implement a PWM driver which would keep the thermistor reading within 3°C of the desired temperature. We ended up getting within around 1°C of the desired temperature. In order to do this we used the principles of PID (Proportional Integral Derivative) control. Although we did not end up using the integral or derivative part of the control we still calculated them and tested partially with them. The integral and derivative control did not prove necessary ;however, as the fan is not precise enough at maintaining temperature they were not required. The integral and derivative control introduced error into the code that was not necessary. However the proportional control worked by multiplying the difference between the desired temp and the current temp, then multiplied that number by a constant. The resulting value is the PWM, which is sent to the P4.0 port. The integral control is calculated using Riemann sums of the proportional control. The derivative control was calculated by calculating the difference between the previous proportional control calculated and the current one divided by the amount of time between the two proportional controls. Again, the integral and derivative control are calculated in the code, but they are not added to the final output.

5 Getting Started/How to use the device

In order to interface with this device you must power the board and the circuit. The fan and the voltage regulator must be hooked up to a 12V power supply and the MSP430 must be powered using the micro USB port on the board. From there, the rest of the interfacing with the board becomes software based, and that is discussed in the next section.

6 Getting Started Software/Firmware

In order to interface with this project you must use Realterm in order to send/receive information from the MSP430. In order to do this you must find what port is used for UART on your computer. To do this you must open device manager and find the COM port which says UART. In our case this is COM8. This must then be put into the port section in the Port tab in Realterm. Next the baud rate must be set to 9600 and the port can then be open. The display must also be changed to uint8 in the Display tab. In order to send a desired temp go to the Send tab and type in the desired temp between 20°C and 70°C and hit send numbers. Only the number needs to be typed (i.e. 43°C is 43).

6.1 Hierarchy Chart

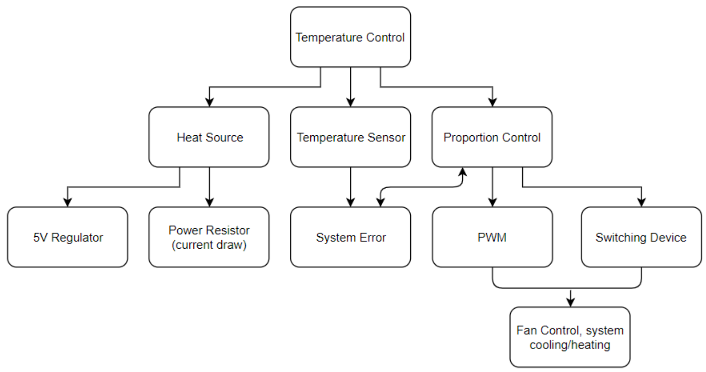

Figure 3: Hierarchy Chart of Project

7 Test Setup

In order to test this program the schematics shown in Section 8 Design Files must be created. Once they are created the MSP430F5529 must have pin 4.0 connected to the gate of the MOSFET. Then pin 6.0 must be connected between the thermistor and the 10k Ohm resistor. Once this is done the program is run onto the micro controller. The default temperature the program will keep the regulator at is 30C. Although if the user desires they can send a value in UART and the regulator will be kept at the desired value.

7.1 Test Data

The testing data below shows the UART display of the regulator at room temperature. The temperature would be similar for a user entered number however the temperature would be constant at the selected value.

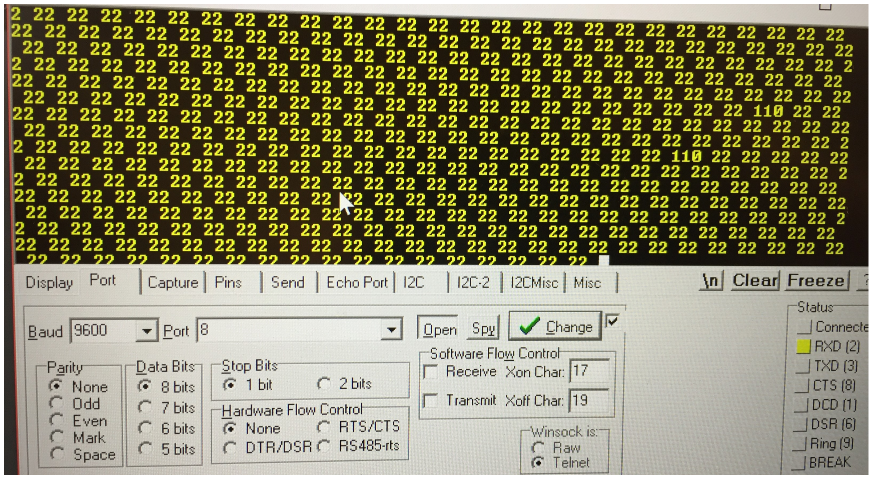

Figure 4: Test data of regulator at room temperature

8 Design Files

8.1 Schematics

The Schematics for the circuits built to test this program can be seen in the following three figures. The first of these shows the circuit created to control the PWM for the fan. This was done by creating a low side switch. This switch allows the fan to be left floating (off) when the regulator is at the desired temperature.

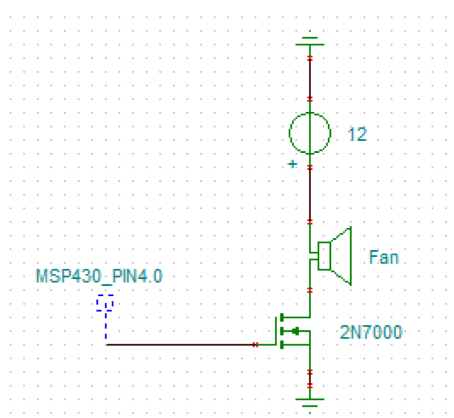

Figure 5: Fan Schematic

The input to the lowside switch gate for the fan control is pin 4.0 on the MSP430 micro controller. The next schematic shows the circuit used to read the voltage from the thermistor. This can be seen in the figure below.

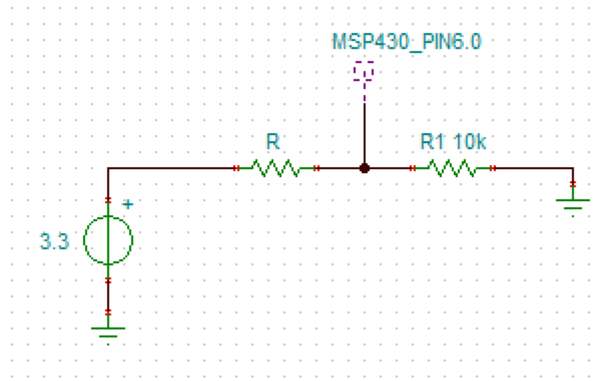

Figure 6: Thermistor Schematic

The thermistor is placed in series with a 10k ohm resistor to create a voltage divider. Then the MSP430 pin 6.0 is connected in between the thermistor and the resistor to read the voltage on the thermistor. This value is then put into the ADC to determine the temperature. The finally circuit created is the circuit to heat the voltage regulator. The circuit for this can be seen in the figure below.

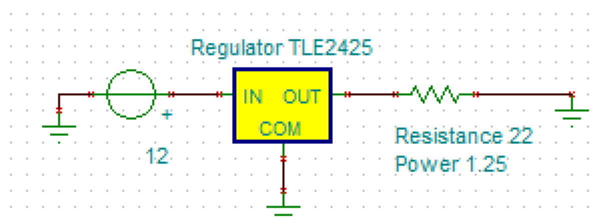

Figure 7: Regulator Schematic

The voltage regulator circuit is the simplest of all the circuits for this milestone. This simply connects the 5V regulator to a 12 volt power supply and the common pin to ground. Finally, there is a power resistor connected to the output of the regulator to ground. The power resistor for this was created by wrapping 5 110 Ohm 0.25 watt resistors together which puts them in parallel.

8.2 Bill of Materials

- 1 MSP430F5529 Micro Controller
- 1 10k Ohm resistor
- 5 110 Ohm resistors

- 5v Regulator
- 1 2N7000 MOSFET
- 1 12V Fan
- Power supply capable of 12V output