This Fan Blows

Kyle McKeown and Daniel Morano
Rowan University

December 3, 2018

1 Design Overview

The overall goal featured in this project was to design circuitry from specific components given to us in the laboratory ,that is capable of detecting the temperature of a heat source and responding to that temperature accordingly. In the most basic description of the overall system, A computer fan is used to control the temperature of a voltage regulator based on the reading that is passed to the micro-controller from an NTC Thermistor. If the temperature exceeds the desired value that is received as an input to the software, the fan will speed up as a response, in order to lower the temperature. At this point the fan will try to maintain at or below that temperature by increasing/decreasing the fan speed. This project incorporates PWM (Pulse Width Modulation) and ADC (Analog to Digital Converter) methodology to control the speed of the fan and to receive and convert temperature readings into a format that is use able for the UART interface to interact with.

1.1 Design Features

These are the design features:

- Proportional Fan Speed Control
- Quick Thermistor Cooling and Heating

1.2 Featured Applications

These are the featured Applications:

- Introduction to Thermistors
- Introduction to Analog to Digital Converter
- Base for Large Scale Thermostat Project

1.3 Design Resources

Raw System Code

1.4 Block Diagram

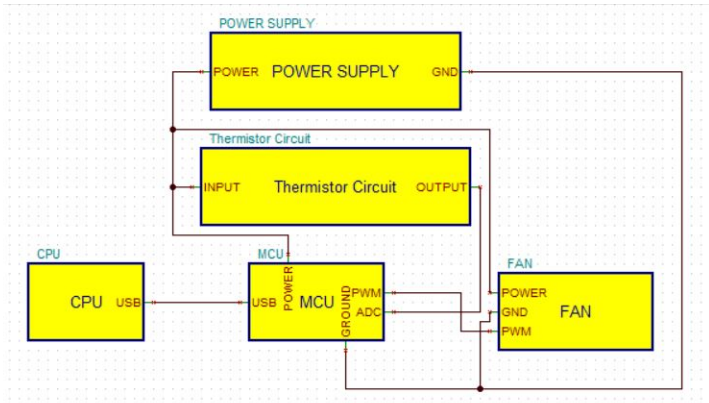

Figure 1: Simple System Block Diagram

1.5 Board Image

Figure: 2 shows the circuit as it is connected to the MSP430 and the Fan. As shown the flow of air is concentrated directly onto the voltage regulator and the thermistor so that the system is more capable of quickly regulating the temperature of the connected device.

Let it also be noted that one of the challenges that was faced in designing circuitry to be able to handle over an Ampere of current without overheating the circuit was to include a power resistor capable of handling that type of load. To overcome this, A 6 Ohm, 2 Watt power resistor was used in order to have as little resistance as possible which in turn creates a larger current draw. This device got slightly warm during testing, however previous attempts with lower wattage resistors caused slight melting of the breadboard before this solution was found.

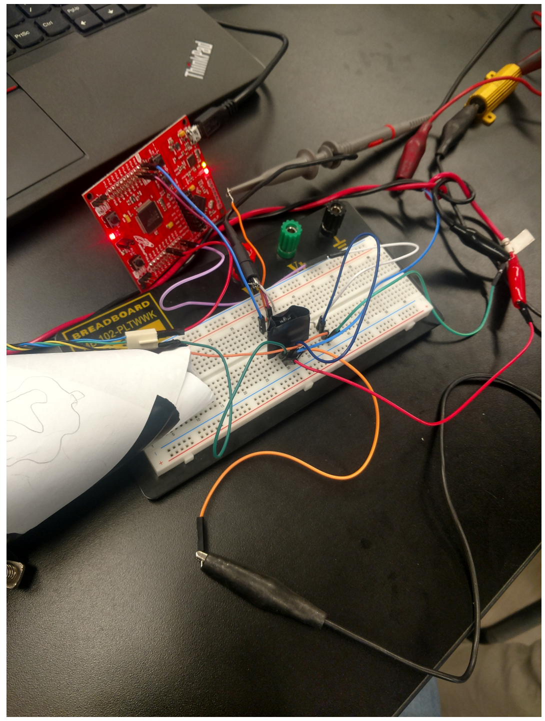

Figure 2: The Final Prototype on a Breadboard

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Temperature	Room Temp - 70℃	The device should be able to heat and cool to these temperatures.
Time	5 Minutes	The device should be able to heat and cool to specified temperatures in no more than 5 minutes from.

Organized table of system capabilities.

3 System Description

The developed system is a variable temperature controller, similar to a thermostat. This system was designed as a learning tool however it has real life applications.The main use for the system is to control the temperature of a thermistor. While this doesn't fix any problems the system could be expanded into a smart thermostat, which solves a myriad of problems.

3.1 Highlighted Devices

- Texas Instrument MSP430F5529LP
- (NTC)Termistor
- ADC

3.2 ADC - Analog to Digital Converter

The basic principle for an analog to digital converter is to turn an analog value into something that can be read and processed by a micro-controller.
The 12 bit ADC converter used in the MSP430FR5529LP contains Internal reference, sample and hold, and an auto scan feature. It has multiple interrupt channels that trigger according to switch statement logic.
The requirements of the project were to read a value from an analog device such as a PTAT or a thermistor and convert that analog value into usable data for the Micro-controller to make decisions based off of. Had this not been the requirement we could have easily simplified the design with a digital device such as a digital temperature sensor, however part of the challenge was to incorporate the ADC to work in our favor with components that we had on hand in the lab. Ultimately the thermistor was chosen based on the limited supply of the PTAT devices and based on the fact that at that time our group had already calculated the different regions of the curve to linear-lize the slope of the thermistor for temperature conversions

3.3 Thermistor

Thermistors, also known as resistance thermometers, are devices that have resistance which is dependent on the temperature of the device. There are 2 types of thermistors, NTC (Negative Temperature Coefficient) and PTC (Positive Temperature Coefficient). The difference between these to types of thermistors is the way temperature affects the resistance. A PTC works when temperature increases, resistance increases and an NTC works when temperature increases, resistance decreases. In this lab an NTC thermistor was used, because it is easier to read an increase of temperature resulting in a decrease in resistance. Since, resistance decreases as temperature increases, using Ohms Law, it can be deduced that voltage increases with temperature for an NTC thermistor. This made it much easier to read on an oscilloscope for testing.

4 SYSTEM DESIGN THEORY

The system is capable of heating up the thermistor to approximately 70℃ and cooling it to just above room temperature. The heating system is completely separate from the cooling system. For heating a linear voltage regulator IC was used. When dumping voltage and current through the IC it will heat up quite drastically. The thermistor was then taped to the voltage regulator to heat up the thermistor through conduction. The thermistor was then used to read the temperature as a voltage. This voltage was then sent to the built-in ADC on the MSP430F5529.

The on board ADC then transformed the signal into a steady voltage that the micro controller could read and handle. Since the incoming voltage is transformed considerably before the micro controller reads it an equation must be used to turn the received voltage back into a temperature reading. In fact, 5 equations were used to transform the ADC-Voltage into a human readable temperature. These equations were as follows:

- 15℃-30℃
  $Temp = \frac{-(165000-(171225*ADCVoltage))}{2729*ADCVoltage}$
- 30℃-45℃
  $Temp = \frac{-(330000-(262460*ADCVoltage))}{2729*ADCVoltage}$
- 45℃-60℃
  $Temp = \frac{-(330000-(169120*ADCVoltage))}{1383*ADCVoltage}$
- 60℃-75℃
  $Temp = \frac{-(330000-(169120*ADCVoltage))}{737*ADCVoltage}$
- 75℃-90℃
  $Temp = \frac{-(330000-(145640*ADCVoltage))}{411*ADCVoltage}$

The equations are different depending on the temperature because the thermistor used does not correspond linearly between output voltage and temperature. Below is a graph of the internal resistance of the thermistor vs. the temperature of the thermistor. The resistance of the thermistor is linear to the voltage so the graph would look the same if it was voltage vs. temperature. So, this graph shows why multiple equations were needed to determine the temperature. If this graph was linear than there would only have been one equation for all temperatures.

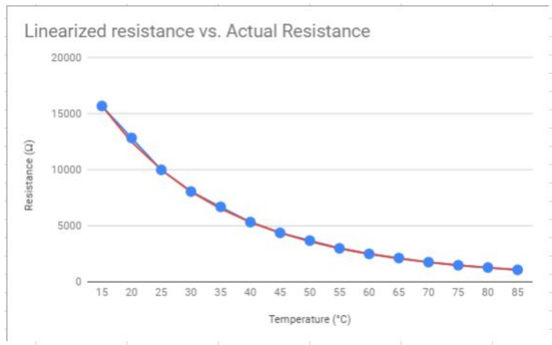

Figure 3: Thermistor Resistance vs. Thermistor Temperature

4.1 Thermistor Heating and Cooling Requirement

The first design requirement was to make the device capable to heat and cool to specific temperatures. The cooling aspect was done by controlling the speed of a computer fan. The fan was controlled using the PWM of a micro controller. Essentially, the fan was turned on and off extremely fast based on the duty cycle of the micro controller. To the human eye the fan being is just changing speed but in reality the fan is being turned off for a longer or shorter period of time which gives the illusion of the fan slowing down or speeding up. The duty cycle is determined by two internal timers on board the micro controller.

The heating element was very simple as well. It was simply a voltage regulator that was given excess voltage and current that was dissipated through the IC as heat. The thermistor was then taped onto the back of the regulator and the heat from the regulator was transferred to the thermistor through conduction.

4.2 Fan Control Requirement

As far as implementing the design is concerned, our group was fortunate to work with a 4-pin fan that had PWM capabilities built into it, while many other groups needed to incorporate a MOSFET switch in order to accomplish the various duty cycles required to consistently regulate the temperature of the system. Typically a fan of this type can be driven between about 30 percent and 100 percent of the rated fan speed, using a signal with up to 100 percent duty cycle. when full duty cycle is reached, it is to quickly cool down the fan from a high temperature to a low temperature as specifies through the user in UART. Otherwise the system that has been built is very efficient and operates at low duty cycle levels based on the P controller to maintain the temperature at the desired value without much oscillation in voltage at all.

5 Getting Started/How to use the device

The device is fairly easy to utilize. The first step is to connect the fan to power, ground and the PWM of the MSP430F5529. The fan used should have 4 wires of different colors. These colors should be black, yellow, green, and blue. The PWM for the micro controller should be connected to the blue wire, while a 12V power supply should be connected to the yellow wire. The black wire should be tied to the common ground of the system and the green wire is left open. Below is an image of the wiring system of the fan.

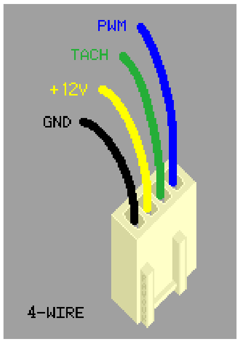

Figure 4: Fan Connection Wires

The next step is to connect the micro controller to the rest of the system, as well as a power supply. Connecting the micro controller to a computer via a development board is the easiest way to power it.

6 Getting Started Software/Firmware

If the micro controller hasn't been programmed yet the next step would be to use CCS and the correct script to program it. Connecting it to a computer also allows for quick debugging and re-programming.

Once everything is connected correctly open up RealTerm and set the baud rate to 9600, the port to whatever port the micro controller is connected to, and the output as an Int. Once the setup is complete press change then open the port. Once the port is open RealTerm should start showing the temperature read by the micro controller. To send a target temperature simply go to the send tab in RealTerm and input the number you wish the system to cool or heat the thermistor to. The inputted number must be an integer.

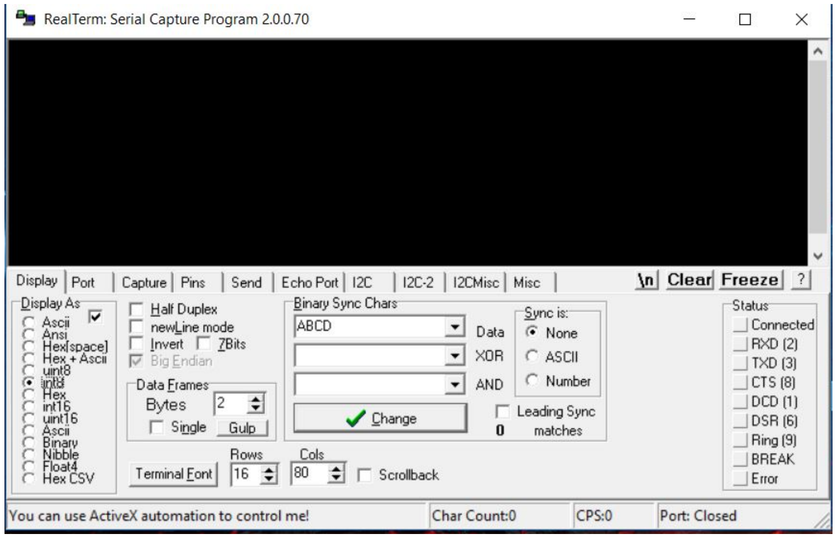

Figure 5: RealTerm

6.1 Communicating with the Device

Communication with the device comes from several different environments. The device communicates directly with the software, between the environment and the software, and between the user and the software through the integration of UART connectivity. The way the system operates is based on several of different stages that update at different times. Some of which are based on polling and others that are based on user input that change as the user changes the desired temperature of the system. On the most basic level, communication exists directly with the software whereas the program is instructed to check the value of it's registers at a specified interval. The complexity of the program comes into play when it is comparing values stored into memory and comparing them with an analog value that was converted by the ADC to be read by the micro-controller and adjust the PWM to regulate the temperature to a value that is specified by a user through UART communication. This is what is meant essentially by ”turning useless into useful” Alone; Pulse Width Modulation, a thermistor, Analog to Digital Conversion, a computer fan, and UART communication protocol are all useless. But when combined in a way that they are able to preform such a specific task as regulating temperatures efficiently, they become very useful.

7 Test Setup

The set-up for testing is the same as the set-up to use the device the only difference would be when utilizing an oscilloscope to observe exact values. The only other set-up needed to use the oscilloscope was to connect the BNC O-Scope probes to ground and the point of testing. For example, when testing the actual voltage coming off the thermistor the probe can be hooked directly to it or to a jumper connected to the thermistor. The PWM signal coming off the micro controller can also be observed using the oscilloscope. To do this the O-Scope probes must simply be connected to ground and the PWM out pin on the micro controller. In this case it was pin 1.3.

7.1 Test Data

During testing it was found that the system could heat the thermistor to approximately 70℃ and it could cool the thermistor to just above room temp. It took about 3 minutes to heat the device from room temperature to 65℃. It took less than a minute to cool it from 65℃ to 30℃, and again less than a minute to go heat from 30℃ to 45℃ The temperature response graphs observed on the oscilloscope were fairly noisy, however the overall curve was steady and smooth. This told us that the system was working as expected and turned the fan to a higher speed the further from the target temperature the thermistor was. The noise was unavoidable and caused mostly by the fan.

8 Design Files

8.1 Schematics

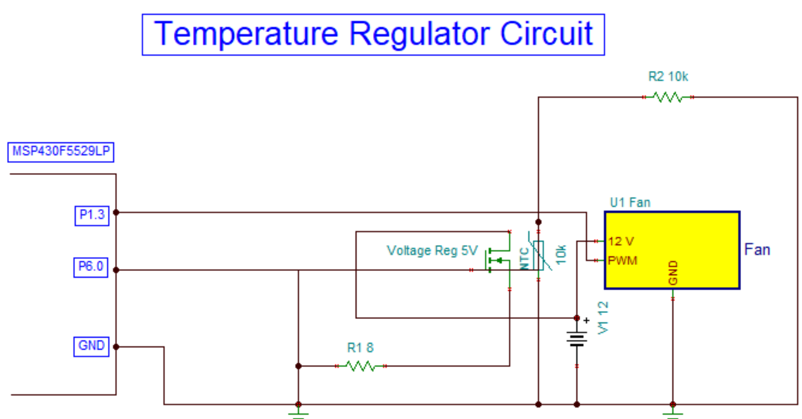

Figure 6: Schematic of The Fan Circuit

8.2 Bill of Materials

HARDWARE

- Texas Instruments - MSP430F5529LP
- Breadboard
- 1 Watt Resistor (10k ohms)
- 2 Watt Power Resistor (8 ohms)
- 5V Voltage Regulator (Used as the Heat Source)
- Heat Sink for the Regulator
- 4-Pin Computer Heat Sink Fan
- A Bad Ass Funnel to Concentrate Air-Flow
- Miscellaneous Jumper Wires

SOFTWARE

- CCS (Code Composer Studio)
- RealTerm for UART communication between hardware and software