Milestone 2: Closed Loop Systems

Nick Klein, Chris Amling, Scott Gordon
Rowan University

December 3, 2018

1 Design Overview

This project creates a closed system with the ability to regulate temperature by reading the input from a temperature sensor and control the speed of a cooling fan based on that input. This is accomplished by heating the temperature sensor by touching it to a 5V regulator running on 12v, the temperature of the sensor is read by measuring the voltage output by the sensor into the analog to digital converter (ADC) pin on an MSP-EXP430F5529 microprocessor. The current temperature is compared to a desired temperature given to the system via a UART serial connection to a computer using a serial terminal program. A fan is sped up or down using pulse width modulation (PWM) based on whether the sensor is too hot or cold so that the fan may cool the sensor or allow it to heat up, based on the desired temperature.

1.1 Design Features

- Dictate desired temperature using Serial Terminal Program.
- Displays current temperature in Serial Terminal Program.

1.2 Featured Applications

- Regulate temperature automatically based on system feedback.

1.3 Design Resources

All code, resources, and instructions are hosted here:
`https://github.com/RU09342-F18/introtoembedded-f18-milestone2-two-milestoned`

1.4 Block Diagram

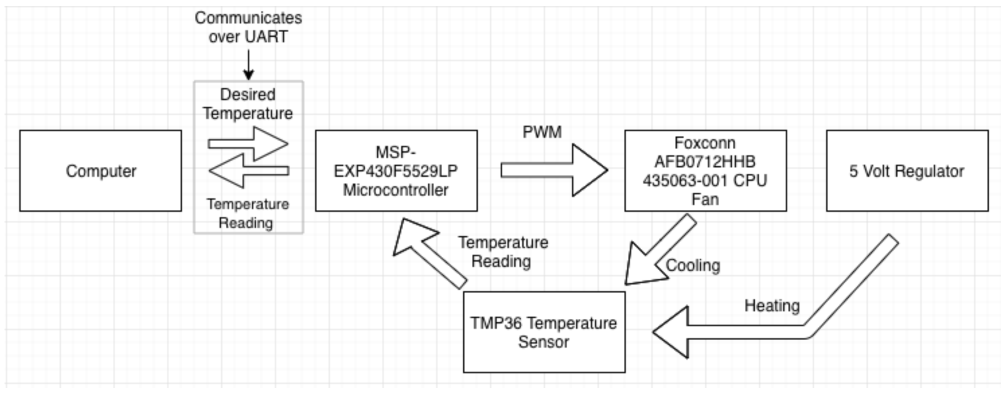

Figure 1: Block Diagram

Figure 1 shows a simple view of the block diagram. The voltage sources and board pins are not shown but the main connections between devices are. The components included in the basic block digram are; the computer, the micro-controller, the CPU Fan, the 5 Volt regulator, and the temperature sensor.

1.5 Board Image

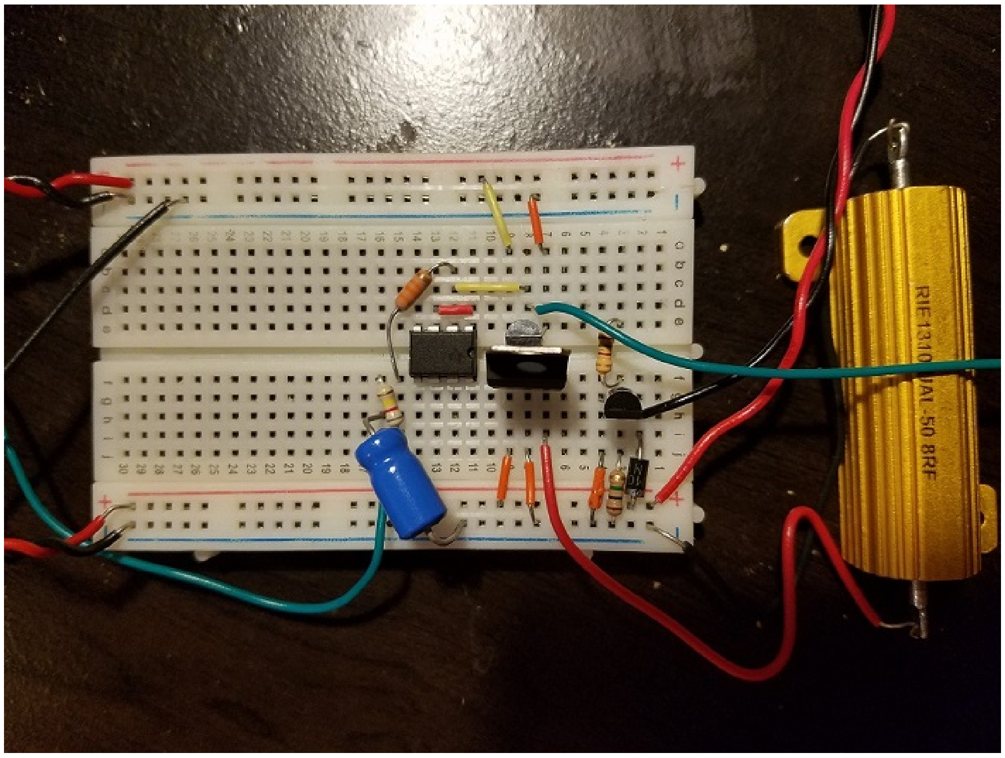

Figure 2: Circuit Board Image

An overview of the circuit board is shown in figure 2. Seen here are the components including the 5 Volt regulator, the TMP 36 temperature sensor, the NE5532P Op-Amp, RIE1310 8W power resistor, the 2N7000G MOSFET, a 15uF Capacitor, and several resistors.

2 Key System Specifications

Parameter	Value	Details
Max Temperature	60C	Max Temp of sensor reachable in current configuration.
Min Temperature	28C	Min Temp of sensor reachable in current configuration.
C Error	5C	Possible mismeasurement of temp in C.
Sensor Output	10mV/C + 0.5V	Voltage output from temperature sensor read by ADC.
Minimum Duty Cycle	2%	Minimum PWM so that the fan is never off.

3 System Description

The system is comprised of four subsystems, the microprocessor, the fan, and 5V regulator, and the temperature sensor. The fan is connected to 12V and is used to cool the temperature sensor. The 5V regulator is also connected to 12V so it will disperse the remaining power as thermal energy. It is positioned on the board directly across from the TMP36 temperature sensor, close enough that they are touching so the regulator will heat the temperature sensor. The MSP-EXP430F5529 microprocessor is used to send a PWM signal to the fan, to translate the output of the temperature sensor using an Analog-to-Digital Converter, and to process the information to regulate the temperature of the sensor. Realterm is a serial terminal program which is used to send the board a desired temperature for the temperature sensor to reach as well as to display the current temperature that is currently being read. If the temperature is too high, the board will increase the duty cycle of the PWM signal to increase the fan speed, which cools the temperature sensor. Once the sensor is too cool, the fan will slow down again to allow the regulator to heat the sensor. If the temperature stays in an acceptable range, the PWM signal stays the same.

3.1 Detailed Block Diagram

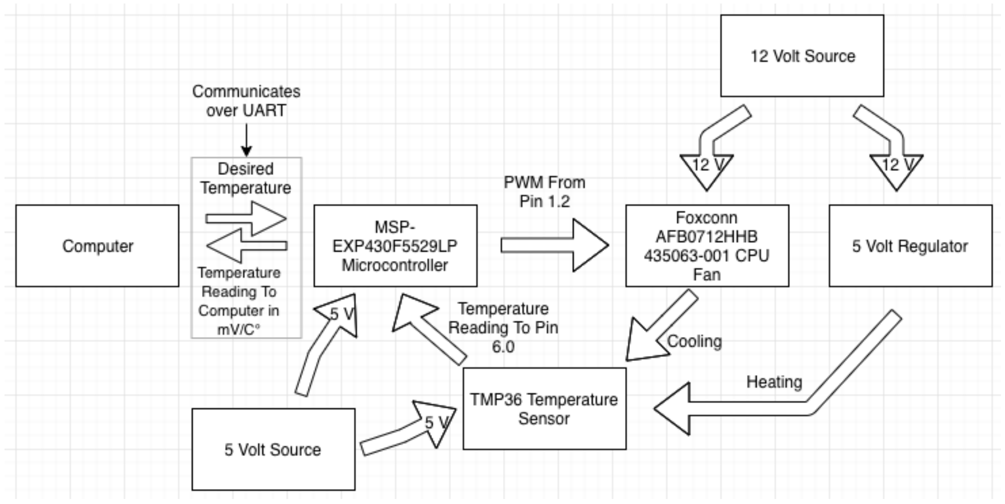

Figure 3: Detailed Block Diagram

Figure 3 shows the detailed block diagram. This figure creates an easy-to-read diagram of all the components being put together. Components included in the diagram are; the two voltage sources, 12 Volt and 5 Volt, the computer, Micro-controller, CPU Fan, 5-Volt Regulator, and the temperature sensor. Left out of the diagram are minuscule circuit components such as resistors and capacitors.

3.2 Highlighted Devices

- MSP-EXP430F5529: The processing unit for our closed system. The 5529 controls the fan, takes in the temperature reading of the sensor, and regulates the temperature.
- TMP36 Temperature Sensor: The sensor outputs a particular voltage per degree Celsius of the sensor, this is interpreted by the 5529's ADC.
- LM7809AC 5V Regulator: The regulator's function is to be overpowered and to heat up. Its close proximity to the sensor allows the sensor to constantly have a heat source influencing the temperature.
- Foxconn AFB0712HHB 435063-001 CPU Fan: The fan is controlled by a PWM signal from the 5529 and cools the sensor until it is in balance with the heating coming from the regulator.

3.3 MSP-EXP430F5529

The microprocessor serves to control the PWM signal controlling the fan speed, interpret the voltage being sent by the temperature sensor as a temperature in degrees Celsius, to communicate with a computer over UART to receive the desired temperature as well as to send the current temperature. The PWM signal comes from Pin 1.2 and the ADC is connected to Pin 6.0.

3.4 Fan Circuit

A CPU cooling fan is connected to a 12V VCC. 12V is chosen as the fan needs 12V to run. A flyback zener diode in parallel controls power spikes. They are connected to an NMOS drain. This NMOS serves as a low-side switch which controls the PWM of the fan. The source of the NMOS is connected to ground and the gate connects to two resistsors. The first goes to Pin1.2 of the microcontroller where the signal for PWM origionates. The second is a very large pull-down resistor that goes to ground so the gate is off by default. A paper cone is taped to the fan to better focus the air movement which is pointed directly onto the temperature sensor.

3.5 Heating Circuit

A 5V regulator is connected to the 12V VCC on the Vin pin. 12V are used since the power supply already has that supplied and the voltage needs to be more than the 5V that will be regulated to create heat. The Vout Pin is connected to an 8W power resistor which is then connected to ground. This pulls more current so the regulator can heat up. The ground pin of the regulator connects directly to ground. The regulator must be placed so it can touch the temperature sensor.

3.6 Sensor Circuit

The Temperature sensor's Vin pin is connected to a 5V VCC as the TMP36 sensor used only operates from 2.7V to 5.5V. The ground pin is connected directly to ground, and the Vout Pin is connected to the negative terminal on an op-amp which is used as a buffer. The buffer opamp serves to reduce noise by stopping extra voltage from being pulled. The output of the Op-Amp is connected back to the positive terminal and to a 850 resistor. That resistor is connected to a 15uF capacitor and Pin6.0 which is the input of the ADC. The resistor capacitor combo is a low-pass filter to reduce noise. The Sensor has to be touching the 5V regulator to heat up.

4 SYSTEM DESIGN THEORY

4.1 Heating

The heating system's purpose is to create ambient heat that raises the temperature of the temperature sensor. This heat is mostly constant unless the fan begins to blow directly on it. A small air gap is left between the 5V regulator which serves as the heating element and the temperature sensor, this is so the heating element doesn't overheat the sensor so much that the fan cannot cool it sufficiently. The heat is created by using a 12V power source on the VCC pin and putting a power resistor between the 5V regulator's Vout pin and ground. The extra voltage is turned into heat since the goal of the regulator is to only output 5V. The power resistor creates a much larger current flow because it's able to handle more power without burning out, represented by the power equations below.

$$P = IV, P = \frac{V^2}{R}, P = I^2 R \tag{1}$$

4.2 Cooling

The cooling system uses a fan to create airflow around the temperature sensor, removing ambient heat from the 5V regulator and allowing the sensor to cool. This is accomplished by varying the duty cycle of the pulse width modulation signal controlling the fan speed. This signal is sent from the microprocessor to the fan based on the difference between the current temperature of the sensor and the desired temperature sent to the microprocessor over the UART connection. The registers which control the duty cycle are TA0CCR0, which controls the number of clock cycles in a given period for the PWM signal based of TIMER_A, and TA0CCR1 which dictates how many cycles the power is high. The remainder of the period is in a low power mode. This particular configuration of PWM is called reset/set and is controlled by the TA0CCTL1 register, which is set to OUTMOD_7. Reset/set implies that the PWM cycle begins high and goes low as TA0CCR1 is reached.

4.3 Temperature Regulation

Temperature regulation is accomplished by comparing the current temperature of the sensor to the desired temperature sent over UART. The current temperature is obtained by the voltage coming off of the temperature sensor going into the ADC pin on the microprocessor, pin 6.0. That input is stored in the resister ADC12MEM0. The voltage is then interpreted by the microprocessor as a temperature in degrees Celsius using the equation...

$$((\frac{Nadc}{4095} * 2.75) - 0.5) * 100) + 10 \tag{2}$$

where Nadc is the raw voltage reading the ADC interprets given by ADC12MEM0. This equation was originally...

$$((\frac{Nadc}{4095} * 1.5) - 0.5) * 100) \tag{3}$$

which was derived using the information given in the parts' datasheets however the readings were incorrect and so the equation was modified using experimental data to give more accurate readings.

4.4 Input and Output (Display)

Input and output to the system is accomplished over a Universal Asynchronous Receiver Transmitter (UART) connection from a computer using a serial terminal program, in this case RealTerm, and the microprocessor via a USB connection. The serial terminal must be configured to match the Baud rate of the microprocessor, set here to 9600, and to communicate over the correct com port (which is individual to the computer). The serial terminal can send an integer to the microprocessor which it will interpret as a new desired temperature. By default this value is set to 25 degrees Celsius.

At each reading of the ADC which occurs every 1ms, the current temperature interpreted by the microprocessor is output to the serial terminal and displayed so the user can see the change in real time.

5 Getting Started/How to use the device

To be able to use the system, only a few things have to be done. First the circuit board needs to be connected to a power supply, with one rail powering 12V, the other powering 5V, and a common ground between the two. Next, the microprocessor needs to be hooked up to the device that is going to be running Code Composer Studio and Realterm. The board needs to be flashed with the code and it needs to be running. Realterm is used to send and receive the temperatures, but this is further detailed in the next section. With the microprocessor connected, the code running, and the circuit being powered, the system will be fully functional. As the 5V regulator heats up the temperature sensor, the system will evaluate whether or not it needs to increase or decrease the fan speed to adjust the temperature.

6 Getting Started Software/Firmware

To be able to send and receive temperature values to and from the system, a program called Realterm is used. The user simply needs to connect the program to the port that the microprocessor is connected to on the laptop and then set the baud rate to 9600. All readings are displayed on the main part of the screen.

In addition to using Realterm, Code Composer Studio is used to code all of the setup and operations carried out by the microprocessor. This includes the setup of the ADC, the setup of UART, the calculations of temperature, the sending and receiving of those values, and the incrementing/decrementing of the fan speed. For all of this to work, the board needs to be flashed.

7 Test Setup

This section will reference Figure 2.

Connect VCC and GND wires to the 12V and 5V power supplies. Be sure a common ground is created by connecting the two grounds with a banana connector wire.

With the circuit assembled on the breadboard, attach the wire coming out of column d that is attached to the Vout pin of the temperature sensor to the ADC pin of the microcontroller, Pin 6.0. attach the wire connected to the capacitor to the PWM pin, Pin 1.2. The bottom rails, on the side with the capacitor need to be connected to 12V and common ground. The top rails need to be connected to 5V and common ground. The twisted wires coming off of the NMOS and the 12V rail on the right side of the board connect to the Vin and GND pins of the Fan. Often Vin is shown with a yellow wire and the GND is shown with black but consult your individual datasheet to be sure. The black wire connected to the ground rail goes to the ground pin on the MSP-EXP430F5529 so that everything has common ground. Place short wires in the same row as the output of the temperature sensor and one in the common ground. Connect an oscilloscope probe with the positive end on the output pin and the negative end on the ground pin. This will allow a comparison between the number being read by UART and the actual output of the sensor.

Connect the Microprocessor via USB to a computer with a serial terminal program. Set the program to 9600 Baud and to the com port that the computer has connected to the usb connection. Ensure the port is open in the terminal program. Place the tip of the paper cone attached to the cooling fan as close as possible to the sensor and at such an angle that the air will blow between the sensor and regulator as well as around the sensor.

Flash the board with the program and information should start appearing in the serial terminal program.

7.1 Test Data

During testing the desired temperature was set to 60 degrees to observe how the system heats up followed by a desired temp of 30 degrees to see it cool then a desired temp of 45 degrees to see how the system deals with regulating one temperature. Figure 4 shows the rolling output at 60 degrees Celsius. The result is very stable and is observed to be about as hot as the sensor is going to get while the fan is still present.

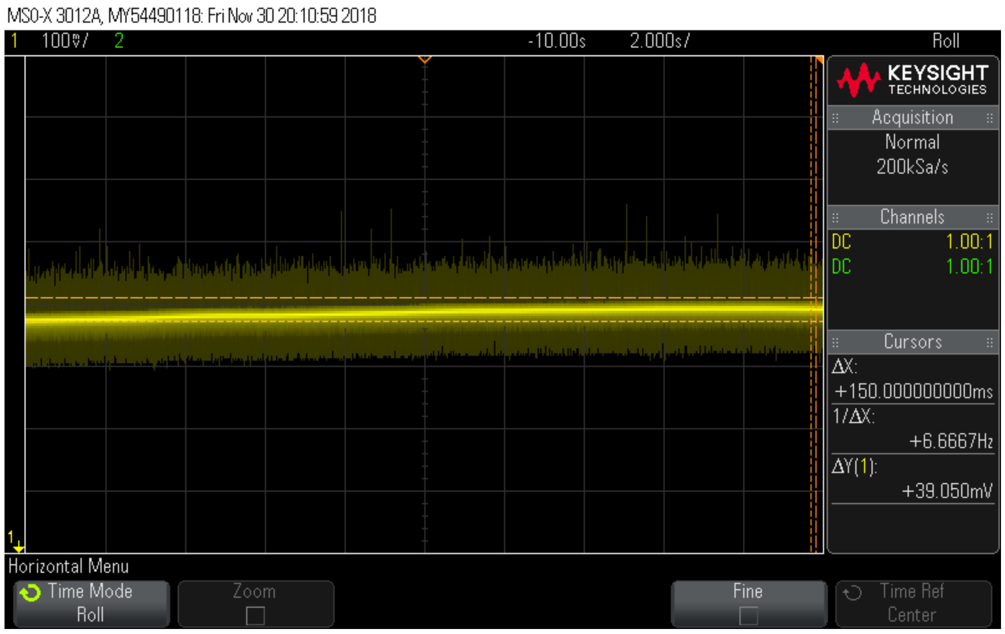

Figure 4: Room temp to 60 degrees

Figure 5 shows the output as the sensor temp reaches 30 degrees. Like the last figure this reading is very stable since it is as cool as the sensor will get in the presence of the heating element.

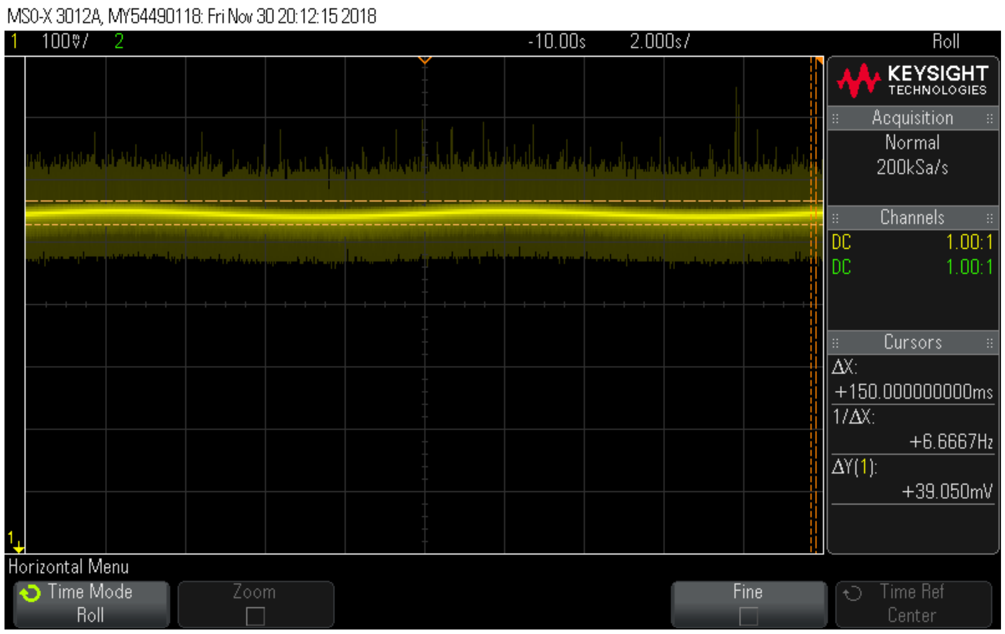

Figure 5: 60 degrees to 30 degrees

Figure 6 shows the oscillations as the system attempts to keep the sensor at 45 degrees Celsius. There is cyclical variations as the system allows the sensor to heat and cool, slightly overshooting the desired temperature each time within a degree or two.

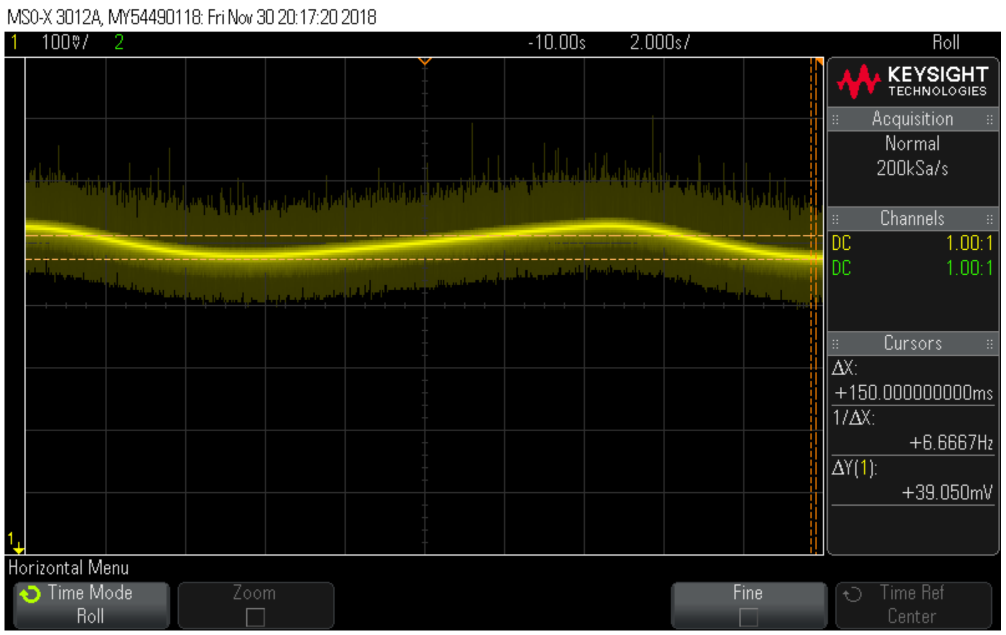

Figure 6: 30 degrees to 45 degrees

8 Design Files

8.1 Schematics

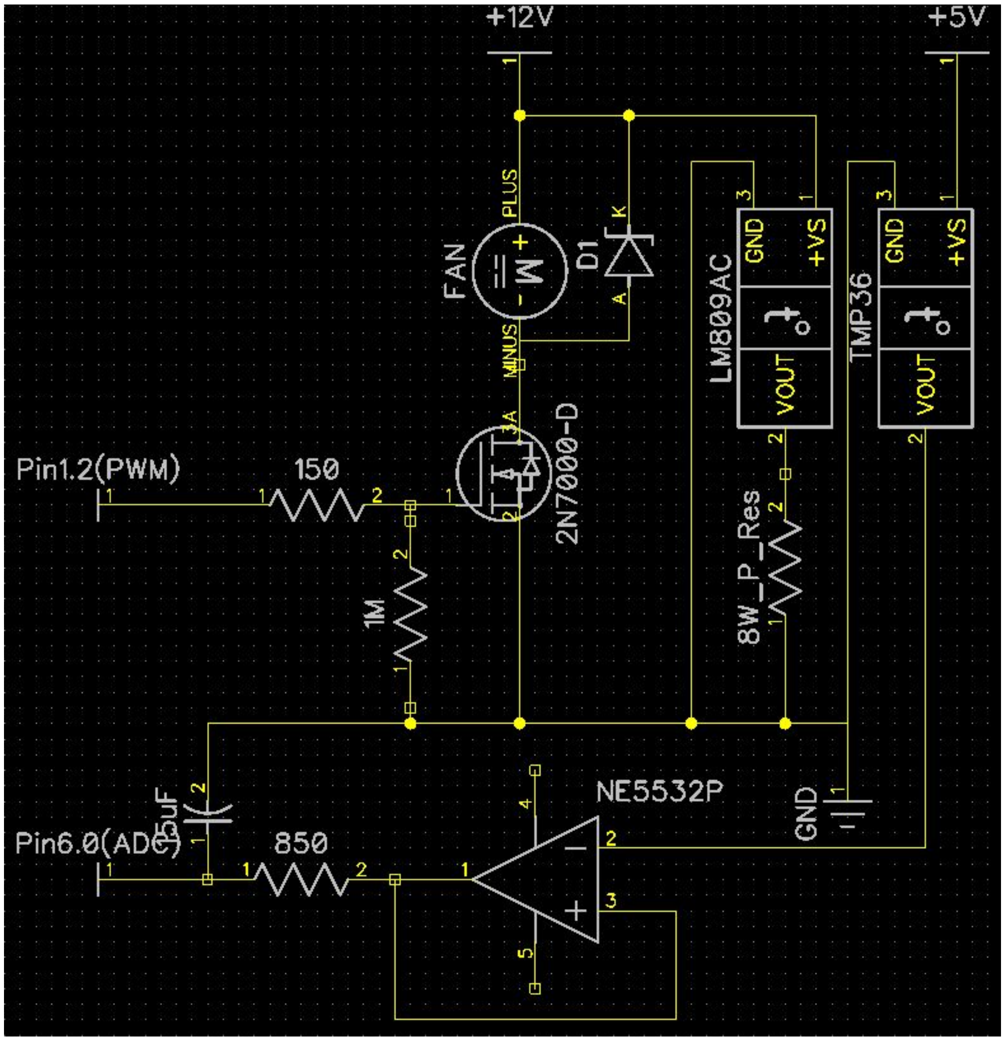

Figure 7: Circuit Schematic

8.2 Bill of Materials

Following is a bill of materials including the part name and model number:

- MSP-EXP430F5529LP Microcontroller
- Solderless Breadboard

- TMP36 Temperature Sensor
- LM7809AC 5V Regulator
- Foxconn AFB0712HHB 435063-001 CPU Fan
- 2N7000G MOSFET
- NE5532P Op-Amp
- RIE 1310 8W power resistor
- 150 Resistor
- 1M Resistor
- 850 Resistor
- 15uF Capacitor
- Zener Diode

8.3 References

- MSP-EXP430F5529 Family User Guide
  http://www.ti.com/lit/ug/slau208q/slau208q.pdf
- MSP-EXP430F5529 Datasheet
  http://www.ti.com/lit/ds/symlink/msp430f5514.pdf
- LM7809AC Datasheet
  https://tronixstuff.com/wp-content/uploads/2010/06/fairchild-lm7805c.pdf
- TMP36 Datasheet
  http://ctms.engin.umich.edu/CTMS/Content/Activities/TMP35_36_37.pdf
- 2n7000 Datasheet
  https://www.onsemi.com/pub/Collateral/2N7000-D.PDF
- NE5532P Datasheet
  http://www.ti.com/lit/ds/symlink/ne5532.pdf
- TI Resource Explorer ADC12 Example Code
  http://dev.ti.com/tirex/#/Device/MSP430F5529/?link=Software%2FMSP430Ware
  %2FDevices%2FMSP430F5529%2FPeripheral%20Examples%2FRegister%20Level
  %2FMSP430F55xx_adc_01.c