Application Note - Team MD

Bill Dripps and James Merrill

Rowan University

December 2, 2018

1 Design Overview

This AN describes an MSP430-based fan cooling system designed to regulate the temperature of a 5-volt DC (direct current) voltage regulator. Temperature is sensed via a thermistor and onboard ADC (analog-to-digital converter) peripheral, and cooling is provided by a PID (proportional-integral-derivative) controller to determine fan speed. The desired temperature is input to the device via UART and a computer terminal. The PID calculates the correct duty cycle of the PWM (pulse-width modulation) signal that drives the fan, in order to maintain the desired temperature over time.

1.1 Design Features

The controller presents the following features:

- Thermistor-aided digital temperature reading
- 9600 baud UART interfacing
- PWM controlled fan speed

1.2 Featured Applications

- Power supply thermal management
- HVAC controller system

1.3 Design Resources

https://github.com/RU09342-F18/introtoembedded-f18-milestone2-md-team

1.4 Block Diagram

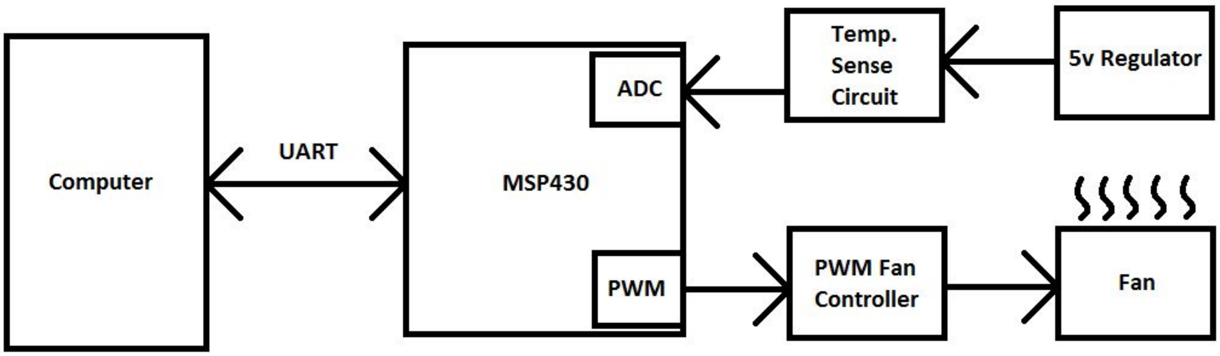

Figure 1.

1.5 Board Image

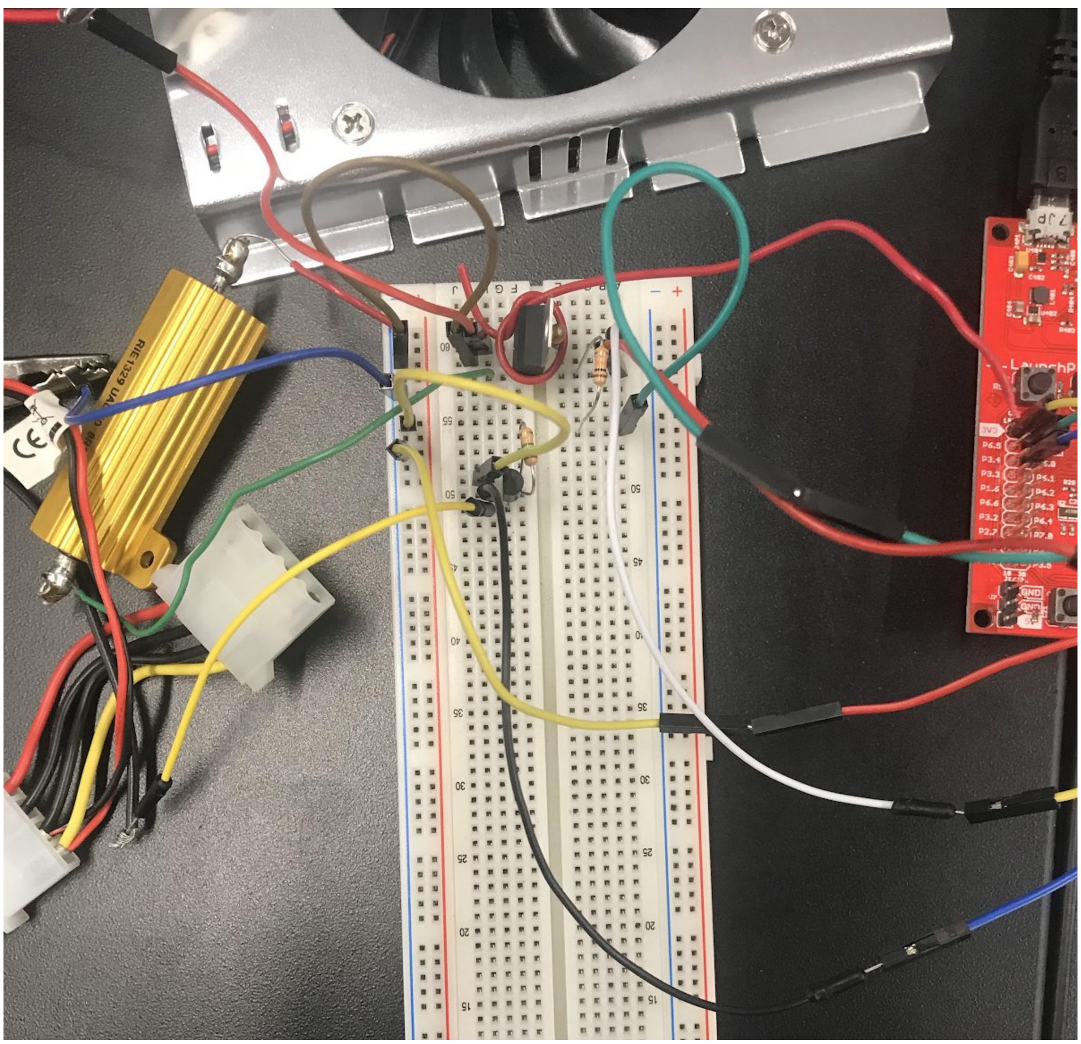

Figure 2.

2 Key System Specifications

Parameter	Specification	Details
Baud Rate	9600	*The rate at which each bit, or pulse, is passed through UART Communication*
Temperature range	35-60℃ (95-140℉)	*The temperatures the system is tested to regulate.*
Temperature Checking Frequency	3.815 Hz	*The rate at which the bits from the ADC are retrieved and converted to a temperature*

3 System Description

In its entirety, Milestone 2 is a closed-loop temperature control system for a 5v regulator. A thermistor and resistor are arranged into a simple voltage divider network, which the MSP430F5529 is then configured to interpret the output from the divider network into a corresponding temperature. A linearized approximation of the Steinhart-Hart characteristic of the thermistor is used to calculate the current temperature at the thermistor, in order to save the microprocessor from having to calculate "expensive" natural logarithmic functions. The value calculated is then compared to the user input desired temperature, which the microcontroller uses to determine the PWM signal that controls the fan. The duty cycle value is calculated by the PID calculation.

The PWM signal does not drive the fan directly, as the MSP430F5529 does not support 12-volt logic outputs. Instead the signal is used to drive the gate of an NMOS Low Side Switch. This switch closing and opening respectively turns on and turns off the 12 volt fan circuit. The 12 volt rail is also used to drive a 5 volt regulator, which is in turn driving an 8 ohm load. This results in a power dissipation of approximately 4.4 watts from the regulator. For electrical safety of surrounding components, a flyback diode is used across the fan, since it possesses inductive characteristics.

Simplified, the microcontroller tends to turn on the fan when it detects that the current temperature is higher than the target temperature, and tends to turn off the fan when the current temperature is lower than the target temperature. The microcontroller tries to run the fan at a fairly constant rpm to maintain the target temperature by determining to what degree the fan needs to be turned on or turned off.

3.1 Detailed Block Diagram

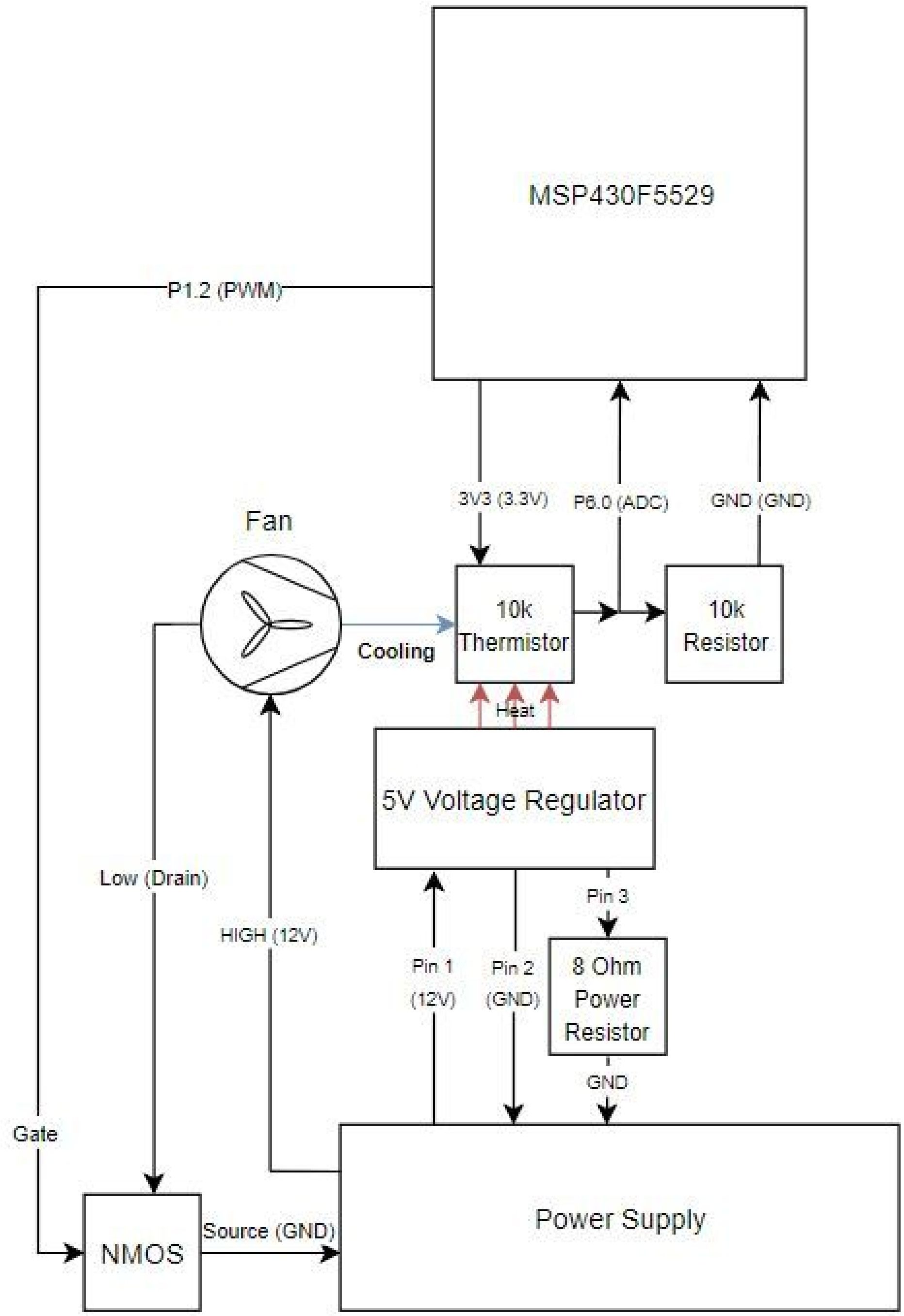

Figure 3.

3.2 Highlighted Devices

- MSP430F5529 - *The node of the system*
- NMOS Transistor - *Allows PWM to control the fan speed*
- Fan and Voltage Regulator - *Generates temperature of the system*

3.3 MSP430F5529

The MSP430F5529 is the centerpiece and brain of Milestone 2. This programmable processor is in control of all functionalities in the system. The MSP430F5529 receives all of the temperature readings via the on-board ADC (Analog-to-Digital Converter). The number of bits being inputted into the ADC corresponds to a specific temperature which is transmitted through the UCA1TXBUF (UART Transmit Buffer). The MSP430F5529 can also retrieve a user input (desired temperature through the UCA1RXBUF (UART Receive Buffer)) and change the characteristics of the system (control a fan's speed through PWM) based on the measured temperature and the desired temperature.

3.4 NMOS Transistor

The NMOS Transistor is the critical component to controlling the speed of the fan because it's the centerpiece of the PWM-controlled low-side switch. In the system, the low end of the fan is connected to the drain of the NMOS, the source is connected to ground and the gate is connected to the pin from the MSP430F5529 which outputs a PWM. When the gate has a high voltage (3.3V), current is allowed to pass through the NMOS to ground. If the gate has a low voltage (0V), current will to go to ground and the fan won't be able to run. This allows PWM to control when the fan is off and on, and therefore, the speed of the fan.

3.5 Fan and Voltage Regulator

The Fan and Voltage Regulator control the temperature of the thermistor. The voltage regulator is the heating element with the input voltage measuring 12V and the regulated voltage (5V) going through a high-wattage power resistor to ground. The heat sink on the voltage regulator was placed very close to the thermistor to heat it. The fan is cooling element of the system whose speed is controlled by a PWM from the MSP430F5529. This allows for multiple levels of fan speeds which corresponds to multiple cooling settings.

4 System Design Theory

The Milestone 2 system is centered around the concept of a closed-loop system. In a closed-loop, the system takes a measurement or series of measurements whose value/values are transmitted back to the controller of the system. The controller will induce a change in the system based on the measured values which will directly affect future measurements. For Milestone 2's case, the system continually measured the temperature of a thermistor and an input of the desired temperature was given by the user. Based on both values, the speed of the fan, which directly cooled down the thermistor, would changed.

4.1 Power Dissipation in a Voltage regulator

As stated previously, the 5v voltage regulator was subjected to an input voltage of 12 volts DC and drove an 8 ohm load. At the load side, since V = I * R and both V and R are known to be 5 volts and 8 ohms respectively, I can be calculated to be .625 amperes. The regulator experiences a voltage drop of 7 volts across its terminals and has the same current running through it as the load. Since P = I * V, the power can be calculated by simply multiplying these two values together, resulting in 4.375 watts dissipated through the regulator.

4.2 Thermistor Response and the ADC

The thermistor used is of an NTC (negative temperature coefficient) variety, meaning it would tend to decrease in resistance with an increase in temperature. The temperature of the thermistor as a function of time is determined by the following equation:

$$T_{(R)} = \left(A_1 + B_1 \ln\frac{R}{R_{ref}} + C_1 \ln^2\frac{R}{R_{ref}} + D_1 \ln^3\frac{R}{R_{ref}}\right)^{-1}$$

Figure 4.

The coefficients A_1, B_1, C_1, and D_1 are specific to the thermistor's construction, R_{ref} is the thermistor's nominal resistance at room temperature, and R is the measured resistance. This equation uses Kelvins as units of temperature measurement. In this milestone, the Kelvins were converted to degrees Fahrenheit for ease of use with American users.
The 12-bit ADC onboard the MSP430F5529 interprets analog voltages between its reference voltage and ground, which are then converted to an integer value between 0 and 4095, where an input of 0 volts would result in a value of 0 and an input of 3.3 volts would result in a value of 4095. Here, a reference of 3.3 volts was used, so each step is therefore equivalent to approximately 806 microvolts. Since the thermistor is part of a voltage divider network and its resistance varies with temperature, the input voltage also varies with temperature. Also, since the ADC's output varies with voltage input, the ADC output value can be mathematically related to the thermistor temperature. The conversion in a general sense looks like the following:

NTC Temperature -> NTC Resistance -> Divider Output Voltage -> ADC Output Value

Each of those segments is related to each other through a simple ratio multiplication.
Instead of using this equation exactly, since it would cause the microprocessor to calculate the natural log of the resistance ratio 6 times, 12 linear approximations of this function are instead used, each only called when in a range that would allow for accuracy to be within +-0.5 degrees Fahrenheit. This was determined to be sufficient to cover the expected range of measured temperatures with a high degree of accuracy.

4.3 Controlling the Thermistor's Temperature (PID Controller)

As previously described, the thermistor's temperature is dependant on two things, the heat coming from the voltage regulator and the cooling generated from the fan. The voltage regulator generates a consistent amount of heat but the fan's speed is controllable using a PWM generated by the MSP430F5529. Varying the fan speed in a specific manner will cause the thermistor to stay at a consistent temperature (the temperature received via UART). To control the temperature a PID (Proportional-Integral-Derivative) Controller for the PWM was implemented.

The PID Controller is a common closed-loop system controller that takes three different measurements (shown in the name) that change as the system runs. The first measurement (Proportional) is the current error (expected value minus the current value) times a constant. The second measurement (Integral) takes the error and continually adds it to itself each time a sample is taken. This value is multiplied by the time between samples and a constant. The last measurement (Derivative) which takes the current error and subtracts it from the samples last error. This value divided by the time between samples and again like the other portions, multiplied by a constant.

For Milestone 2, the values for the PID equation were estimated using knowledge of how the PWM was configured. Due to the hardware PWM using OUTMOD_2 (the signal is initially off and turns on with CCR1), the higher the output of the PID Controller, the lower the fan speed. With lots of experimentation and testing, the constants for each respective portion were 55 (Proportional), 5 (Integral), and 0.1 (Derivative).

To counter any issues involving CCR1 being out of range, limits were set. If the PID output was greater than 1000 or less than 0, the CCR1 value would correct itself to 1000 and 0 respectively. Another issue was an integral value that would get too large (both positively and negatively) causing the fan to stay at either 0 (fully on) or 1000 (fully off) for much longer than a few minutes. For this issue, 2 counters were implemented that would trigger if the PID output was out of range. These counters would add up for every time cycle. If either counter added up to 60, the integral and the counter would be reset. Lastly, this feature caused a few problems on the high end (>1000) with the fan not staying off. To counter this, a buffer of 500 was set. This means that the counter would only be activated when the PID output was greater than 1500.

5 Getting Started/How to use the device

To use the (insert name here), apply source voltage to the circuit and connect to the MSP430F5529 via RealTerm or other terminal software. The UART connection is configured to run at 9600 baud, with (name other settings). The desired temperature in degrees Fahrenheit should be entered into the terminal. The device will output the current temperature every time the temperature reading interrupt is called. Over time the target temperature should be reached, within a margin of error of 5 degrees Fahrenheit.

It should be noted that at higher temperatures, the fan will tend to stop running, and briefly blip on, when trying to maintain temperature. This is due to the fact that the fan has to meet a certain threshold duty cycle in order to run, and when it eventually turns back on, it will overcool the voltage regulator and shut off again. This process will repeat itself in a predictable manner.

6 Test Setup

In order to test the node, a terminal program capable of sending data into the node should be used. In this case, the program RealTerm was used, which can be found free of charge at:

- https://sourceforge.net/projects/realterm/

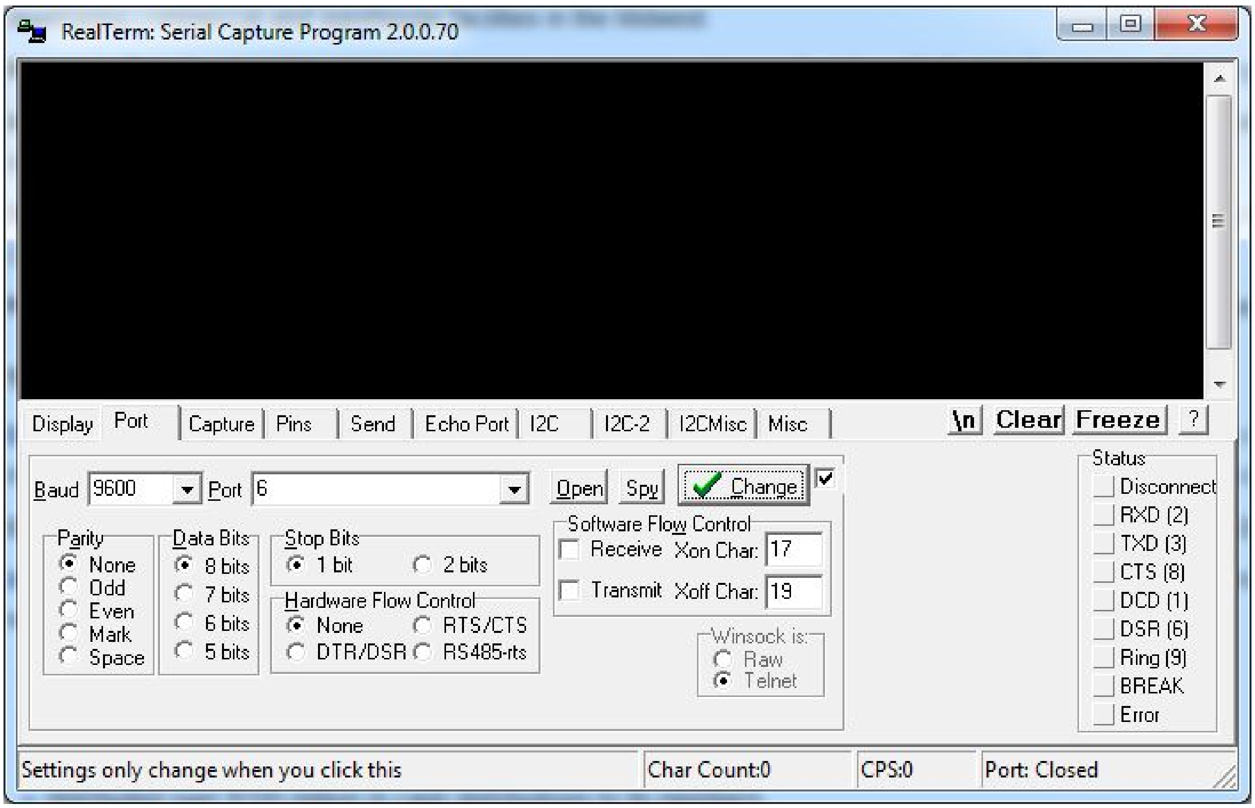

Figure 5.

6.1 Test Data

6.1.1 60℃

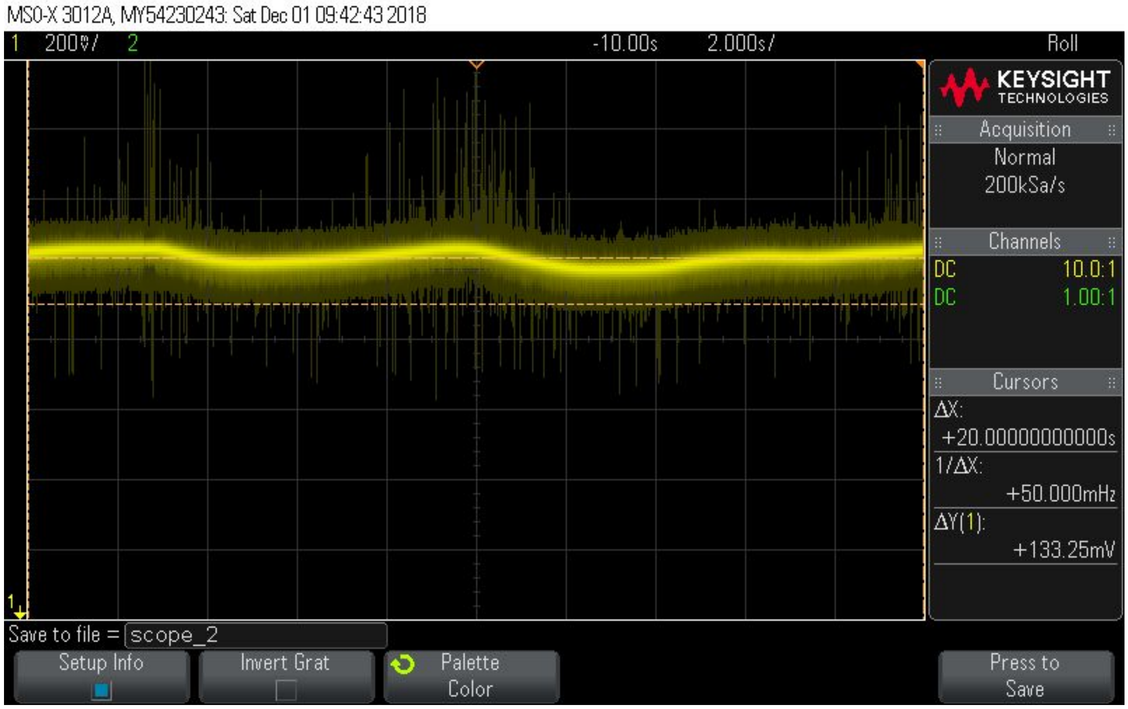

Figure 6.

6.1.2 48℃

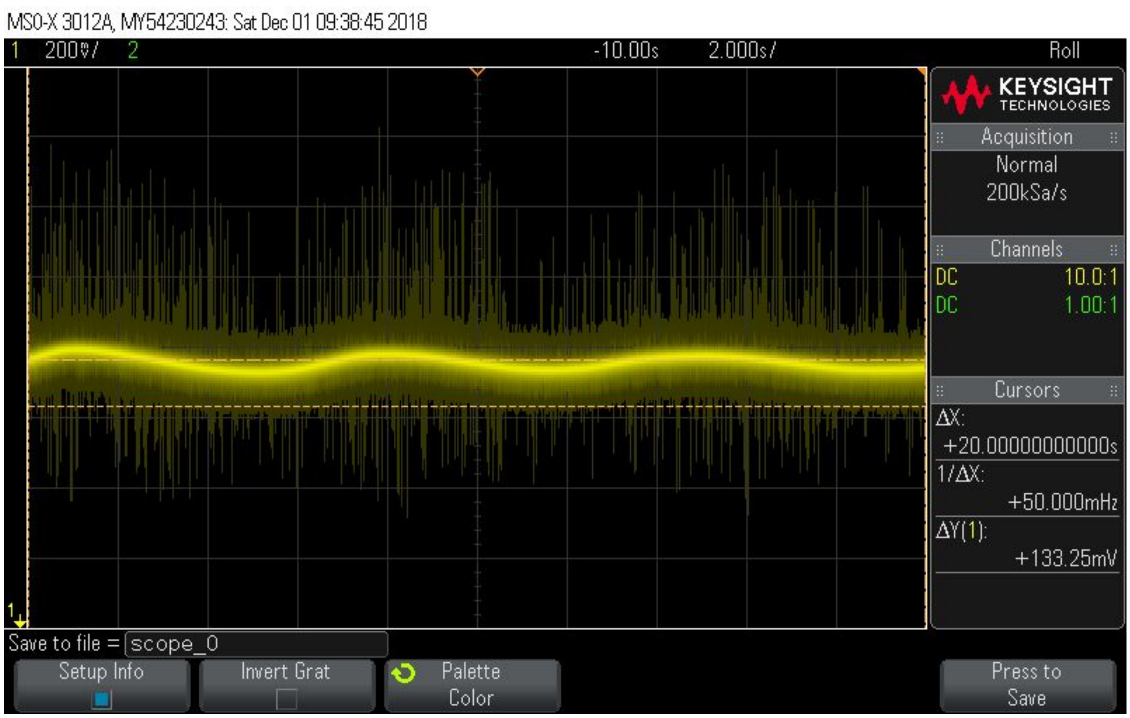

Figure 7.

6.1.1 35℃

MSO-X 3012A, MY54230243: Sat Dec 01 09:40:05 2018

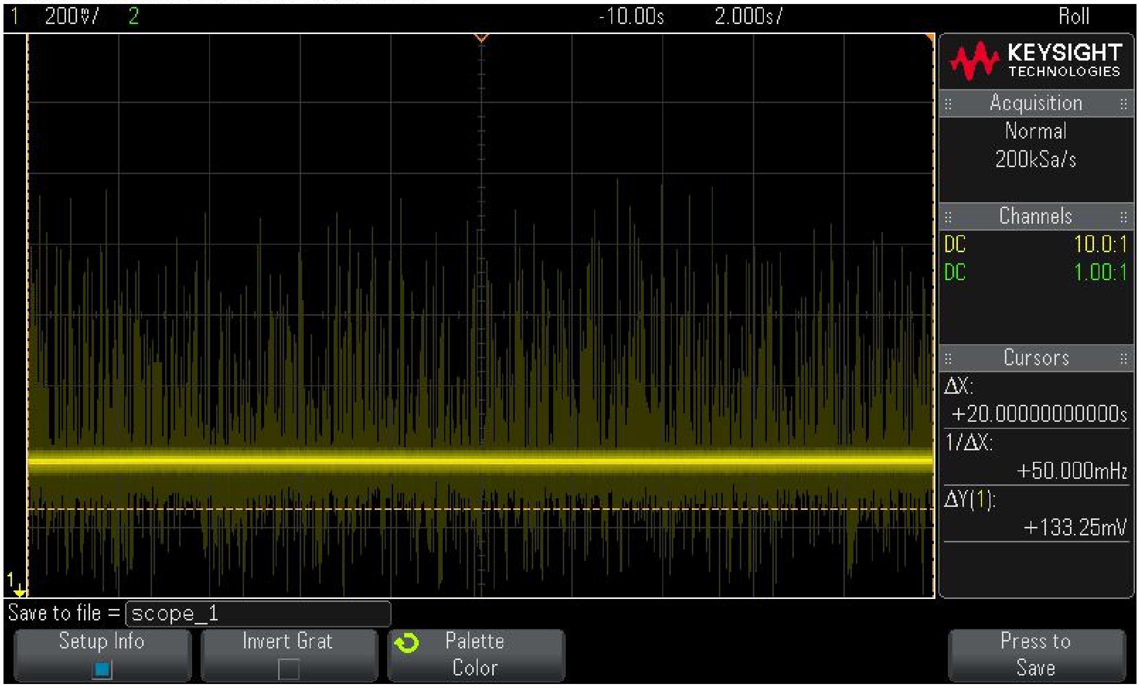

Figure 8.

7 Design Files

7.1 Schematics

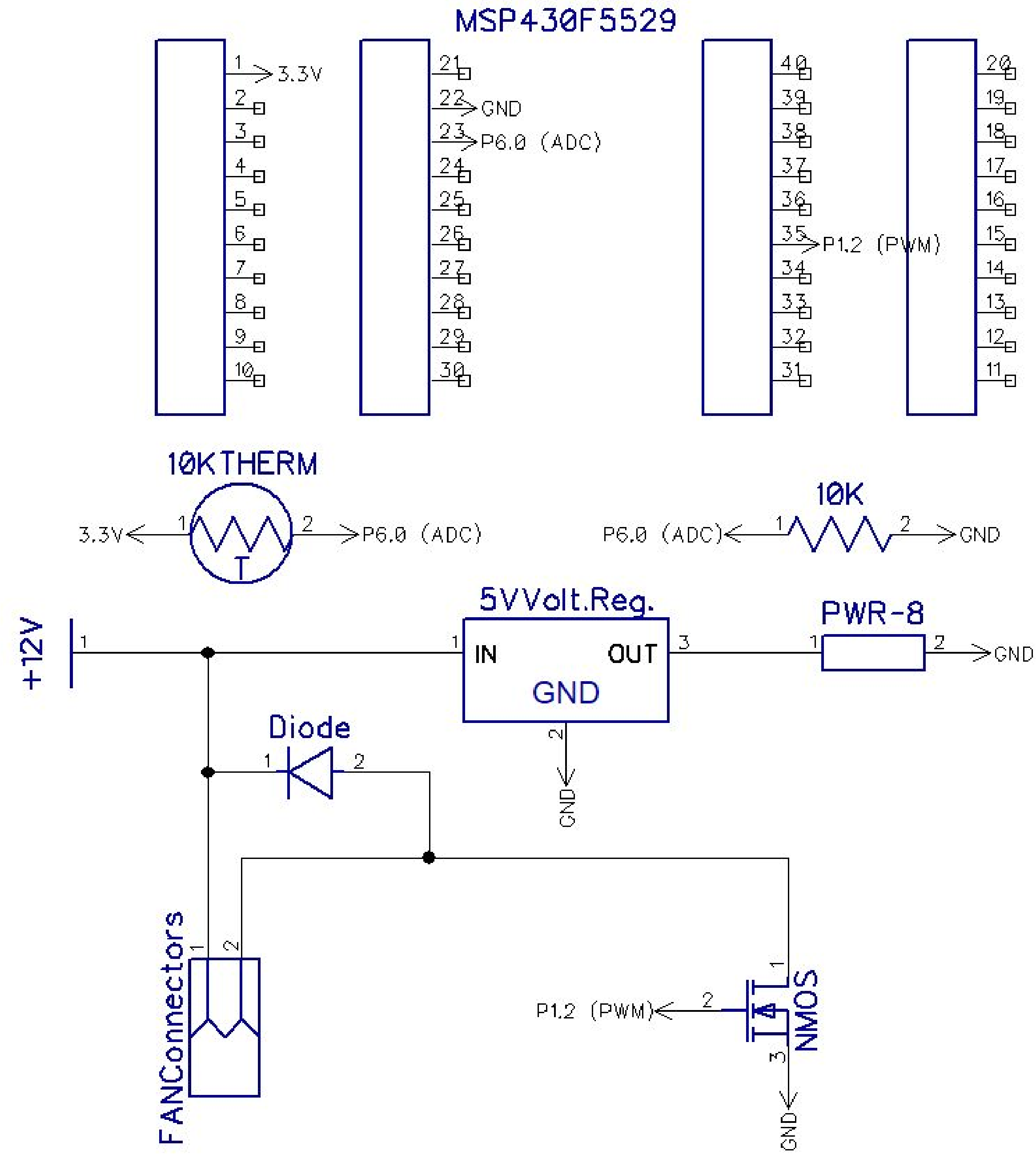

Figure 9.

7.2 Bill of Materials

Component	Description	Tolerance
MSP430F5529 Development Board	n/a	n/a
Voltage Regulator	LM7805	0.2v
Resistor	8Ω	5% 5-watt
Fan	12v DC Fan	n/a
NMOS transistor	2N7000	n/a
Thermistor	NTCLE100E3103JB0	5% ¼-watt
Resistor	10kΩ	5%
Diode	1N4004	n/a