Milestone 2 - Closed Loop Systems

Will Cronin and Matt Whitley
Rowan University

December 18, 2018

1 Design Overview

A closed loop system is the idea of using system feedback to control that same system. Examples of this type of system can be found everywhere, and is a vital part of embedded system design. One such application of a closed loop system is the temperature sensing assembly that will be covered in this app note. In this application, an MSP430F5529 microprocessor is used in conjunction with a thermistor, LM7809AC regulator, YwRobot board, fan, and other components. The MSP430 sets a desired temperature to keep the thermistor at. The regulator, with current flowing through it, heats up the thermistor indefinitely, as the thermistor sends voltage back to the MSP430 to be converted to a readable temperature. When the desired temperature is exceeded, the MSP430 tells the fan to turn on, cooling the thermistor. When the temperature cools to the desired value, the fan turns off, and this cycle repeats.

1.1 Design Features

These are the design features:

- Set a temperature for the system to maintain
- Transmits the current temperature of the thermistor over RealTerm
- Once the temperature begins to go above the set temperature, the fan speed increases
- If the temperature is lower than the set temperature, the fan speed decreases.

1.2 Featured Applications

- Can be used to regulate temperature in multiple applications such as computer hardware, refrigerator and something similar to a home thermostat.

1.3 Design Resources

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

1.4 Schematic

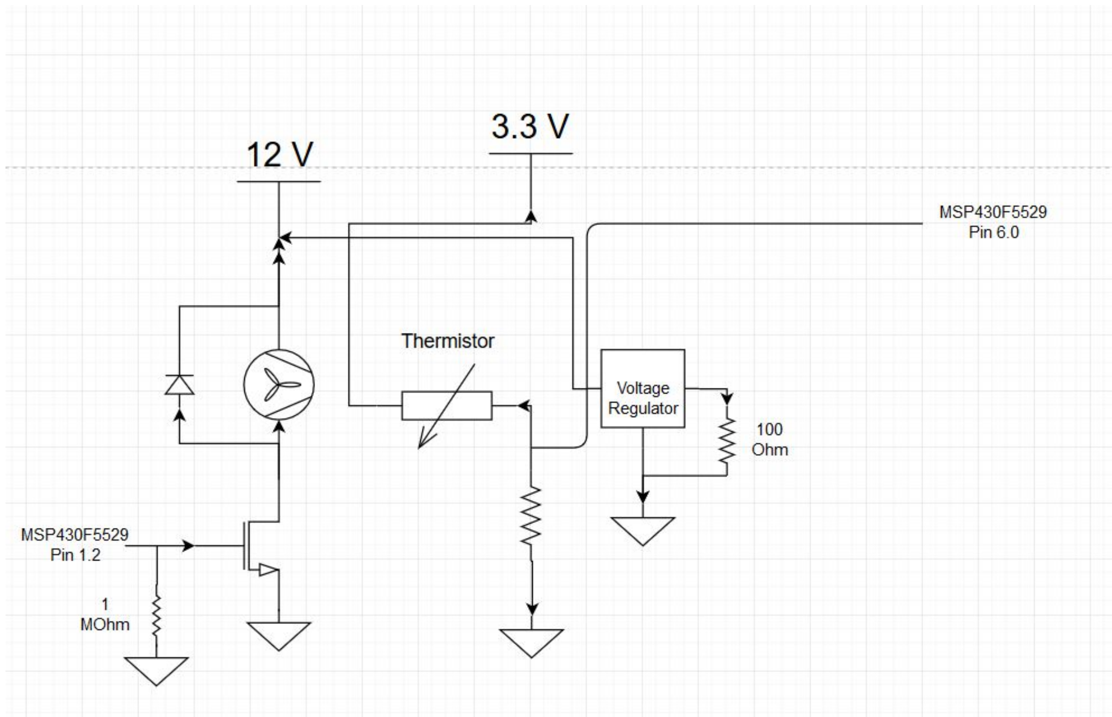

Figure 1: Schematic representation of the MSP430 interfacing assembly

1.5 Board Image

See Figure 2

2 Key System Specifications

3 System Description

The problem address by this lab is to create a program that would control the temperature of a thermistor with a fan. The temperature would be determined by the user.

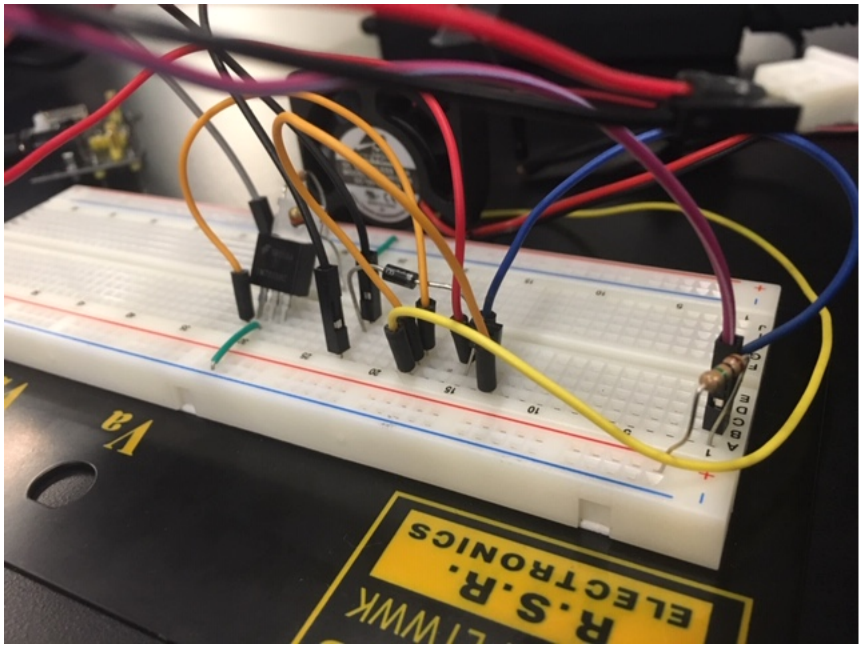

Figure 2: Board Image

PARAMETER	SPECIFICATIONS	DETAILS
SMCLK Frequency	1 MHz	PWM timer speed
TA0CCR0	0xFF	Period of PWM
TA0CCR1	Pin 1.2	Red LED time
TA0CCR2	Pin 1.3	Green LED time
TA0CCR3	Pin 1.4	Blue LED time

The MSP430 processor would read the temperature of the thermistor, determine if the temperature is higher or lower than the desired temperature, and set the fan speed accordingly.

3.1 Detailed Block Diagram

3.2 Highlighted Devices

- MSP430F5529 - This microprocessor is used to transmit and receive signals and use a portion of the signal to power the fan on or off.
- Fan and driving circuitry - A fan was used and connected to the timer output of the board. The fan's driving circuitry includes a fly-back diode and a low-side switch.

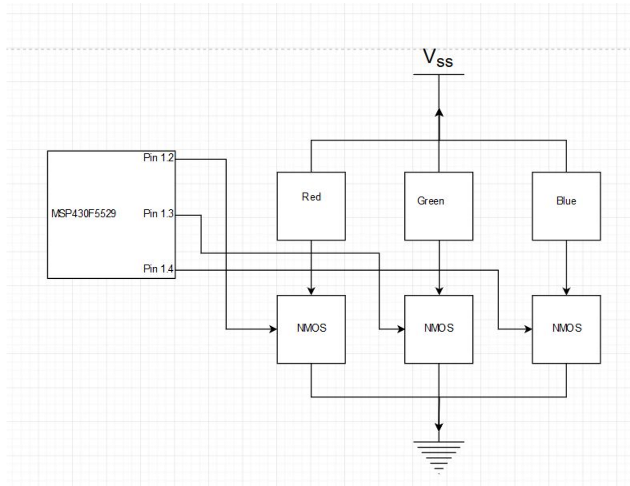

Figure 3: Block Diagram

- Thermistor and Voltage regulator - A voltage regulator was used for generating heat while the thermistor was connected to 3.3 Volts and the output voltage was fed back into Pin 6.0 of the board.

3.3 MSP430F5529

The processor used for this device is the MSP430F5529. The 5529 is better for this application due to its four timer registers. Other MSP430 boards only have three. The 5529 is configured with UART TX and RX (transmitting and receiving) nodes, which is how the strings of hexadecimal values are sent to the fan to cool the thermistor or let it heat up.

LED nodes. As seen in Figure 1, pin 1.2 sends its value to the fan. The initial RX node receives its value from the thermistor. This value is then transmitted to the next node via the TX path.

3.4 Device/IC 2

The fan was connected to a 12 volt power supply with the ground of the fan connected to an NMOS. The NMOS's gate was connected to PIN 1.2 of the 5529. This design allows the fan to power on depending on the voltage supplied by the pin as well as control how fast the fan spins with the amount of voltage supplied. The diode connected to the fan's connections insures that the 12 volt power supply only goes through the fan and does not go directly into the microprocessor.

3.5 Device/IC 3

The thermistor used is an NTC100E3 model, and changes its resistance based on its surrounding temperature. This thermistor is used in series with a resistor to create a voltage divider. This allows an output to be received that is dependent on temperature. Using the datasheet's tables, and the Steinhart Hart equation, a reference voltage is found to turn the voltage supplied by the thermistor into a readable Fahrenheit value.

The voltage regulator is an LM7809ac. It is fed a voltage, and its output is fed to ground through a resistor. Though there are several applications to a voltage regulator, its self-limiting characteristics make it a great heat source for use in this circuit. It supplies a steady source of heat, and when bound to the thermistor with thermal tape, heats the thermistor at a quick rate.

4 SYSTEM DESIGN THEORY

The main design challenges with this assembly include interfacing the MSP430 with the fan and thermistor. With the use of a low-side switch and a fly-back diode, a useful voltage was achieved and the fan was able to successfully integrate with the MSP430. The thermistor was able to be integrated simply by connecting it's output to Pin 6.0 on the board. The voltage sent to pin 6.0 was used in a calculation to determine the resistance of the thermistor, and therefore calculate the temperature.

4.1 Design Requirement 1

The hardware requirement for this assembly consists of a fan and thermistor to maintain a desired temperature, along with supplementary components such as a schotsky diode, NMOS transistor, and a voltage regulator. The voltage regulator is powered by a 12V supply, and a 100 ohm resistor to ground. This allows the voltage regulator to supply enough heat to raise the temperature of the thermistor. The thermistor is bound to the regulator by thermal tape, to conduct more heat and raise the temperature more quickly.

The fan is powered by a 12V supply, and has a schotsky diode in parallel to act as a flyback diode. The fan is controlled by the MSP430, via an NMOS. The MSP430 sends power to the fan when the desired temperature is exceeded by the thermistor. When the fan is turned on, it cools the thermistor indefinitely, until the desired temperature is reached.

The thermistor is supplied 3.3V by the MSP430, and is connected to a receiving pin on the board. This pin continuously reads the voltages given by the thermistor.

4.2 Design Requirement 2

In order to send and receive the data from the MSP430, a UART lines need to be utilized to send the string of hex values. The UART RX line receives the string of bytes, while the UART TX line transmits between each node. The UART configuration allows the MSP430 to read the voltages being supplied by the thermistor.

5 Getting Started/How to use the device

5.1 Materials Needed

- MSP430F5529 and attached design assembly
- Micro-USB to USB connector
- Computer or laptop with Code Composer Studio open and running the supplied code
- RealTerm Serial/TCP Terminal software running on the same computer

5.2 Connecting the Assembly

The MSP430 and breadboard assembly will be given. To begin experimentation, connect the assembly to the computer or laptop using the Micro-USB to USB connector. A small green LED should flash on the MSP430 indicating that it is being supplied power.

6 Getting Started Software/Firmware

6.1 Software Needed

- RealTerm Serial/TCP Terminal
- Code Composer Studio, with given code files

6.2 Using Code Composer Studio

The code needed to run this assembly will be given, and should be opened in Code Composer Studio (CCS). On the main page of CCS, just click the ”Build” icon in the toolbar at the top of the page. This compiles the code. Next, ensure the assembly is plugged into a USB port and run the code. The LEDs on the MSP430 should flash momentarily, indicating that the code is now running on the board.

To set the desired temperature, simply enter the temperature value in Fahrenheit on the line that defines ”settemp.”

6.3 Monitoring Temperature in RealTerm

In RealTerm, click on the tab labeled ”Port” and click the drop-down menu marked by the same name. Click the port from the drop-down menu in which the MSP430 assembly is connected to the computer. You can find which port this is in the Device Manager program in your computer’s settings.

With RealTerm open, the temperatures being sent by the thermistor will be shown in the display window in real time. This is how the temperature is monitored and observed as the program runs.

7 Test Setup

The device is set up for testing in 5 steps

- Connect 3.3V and ground from the microprocessor board or an external source to the breadboard
- Connect Pin 1.2 to the gate of the NMOS
- Connect Pin 6.0 end of the thermistor
- Connect the oscilloscope probes to the end of the thermistor and ground
- Plug the MSP430F5529 into a computer and flash the code
- Open RealTerm and open the MSP430's UART port to view the temperature of the thermistor or use the oscilloscope to view the change in voltage

8 Design Files

8.1 Bill of Materials

- MSP430F5529 x1
- NTC Thermistor(Brown, Black, Orange) x1
- Breadboard x1
- 12 Volt, 0.15 Amp Fan x1
- LM7809AC Voltage Regulator x1
- NMOS x1
- 100 Ohm resistor x2
- 1 MegaOhm resistor x1