– ECE 09.321: Milestone 2 –

MSP430 Thermostat

J.Matteo, R. Stelts, A. Hollabaugh
Rowan University

November 27, 2019

Jacob Matteo, Richard Stelts, Andrew Hollabaugh

1 Design Overview

This project is a device that controls a fan's speed to regulate the temperature of a heat source. This device is controlled by an MSP430FR6989 processor. The heat source used is an L7805 voltage regulator. A thermistor is physically attached to heat source and connected to a voltage divider circuit, which is then connected to one of the MSP430FR6989's pins. The MSP430FR6989 takes the analog voltage reading of this pin and this value is used to calculate the temperature of the thermistor. This is then used to regulate a fan's power via a PI control loop and PWM. The end goal is to have this fan regulate the temperature of the thermistor. The MSP430FR6989 was chosen because it has enough ROM to store the algorithms required for floating point arithmetic.

1.1 Design Features

- UART communication through MSP430FR6989 board to allow for temperature and fan speed setting, as well as temperature data reporting
- Temperature sensing via a thermistor in a voltage-divider circuit, allowing for accurate temperature data to be reported in Celsius
- Pulse Width Modulation (PWM) system to control the speed of the fan, allowing for a variation of different fan speeds to be used
- Closed loop control via a PI (Proportional Integral) loop to automatically adjust the fan speed to efficiently cool the heat source

1.2 Featured Applications

- The ability to cool a heat source using a fan
- Active regulation of the temperature of a heat source
- Real-time reporting of temperature data

1.3 Design Resources

Shared Github Repository:
https://github.com/Intro-To-Embedded-Systems-RU09342/milestone-2-front-row-squad

1.4 Block Diagram

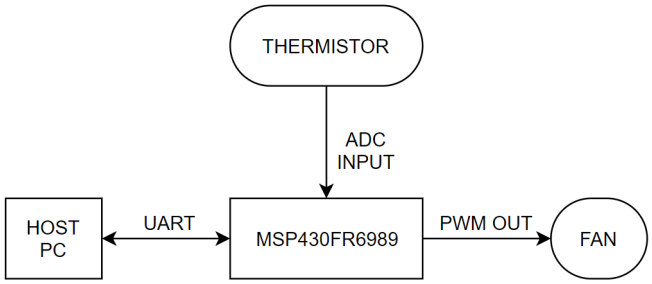

Figure 1: Simple Block Diagram

1.5 Board Image

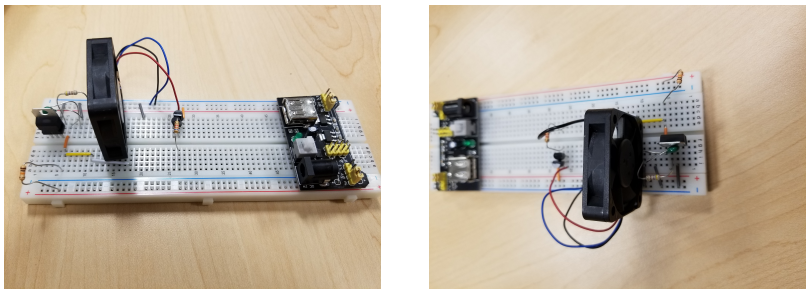

Figure 2: Lone Breadboard Circuit

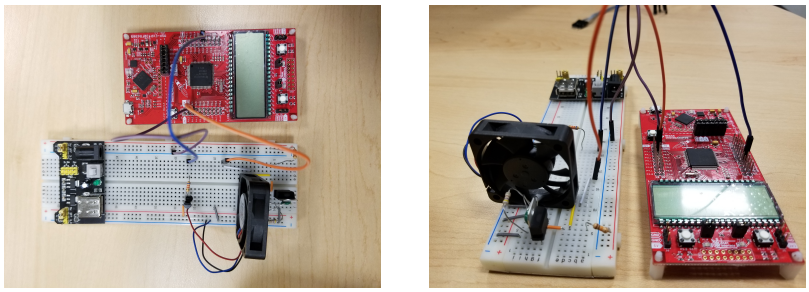

Figure 3: Connected to Micro-controller

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Output Pin	P1.6	PWM for fan
Tempature circut Input	P8.4	Input voltage

3 System Description

The problem trying to be solved is to create a cooling system based on a thermistor and a fan through the use of a microcontroller. The thermistor data is converted from a resistance to a voltage using a voltage divider and this is read by the microcontroller through pin P8.4. The microcontroller then calculates the temperature in Celsius based on the thermistor's resistance vs. temperature response. The microcontroller runs a PI control loop to calculate what speed to set the fan to based on the temperature. It outputs to the fan via the PWM output on pin P1.6. The UART communication protocol is used to send the desired temperature, manually set the fan speed, and report temperature values.

3.1 Detailed Block Diagram

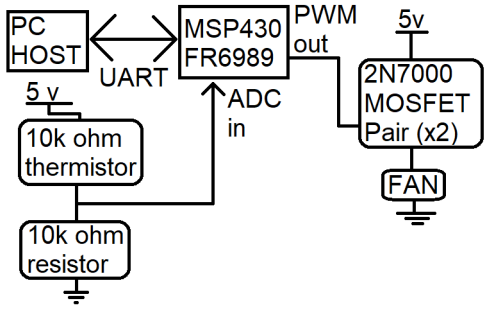

Figure 4: Detailed Block Diagram

3.2 Highlighted Devices

- MSP430FR6989: The microprocessor used for this project. The code is loaded onto this device via a UART serial connection, reads the voltage divider's voltage, and outputs PWM for the fan.
- Thermistor: Senses temperature by changing resistance based on its temperature
- Fan: Blows cool air onto the voltage regulator, cooling it down; runs on 5V DC
- Breadboard: The thermistor was placed on the breadboard along with the resistors, transistors, voltage regulator, and fan. The microprocessor was then routed here.
- Voltage Regulator: Produces heat, changing the thermistors's temperature.
- Transistors: Pair of 2N7000 transistors amplify the PWM signal from 3.3V from the MSP430FR6989 to 5V for the fan

3.3 Device- MSP430FR6989

The MSP430 family of microprocessors includes many different boards with hundreds of features. The MSP430FR6989 family is a part of the MSP430 series that includes multiple 16-bit timers, 40 accessible pins, ultra-low power, and UART communication capability. This specific board includes a 12 bit analog to digital converter. The reason this was the board chosen for this project is because the timers and clocks are easily programmable. The other option for a board was the MSP430G2553, which did not have enough ROM to store the underlying arithmetic for floating-point numbers.

3.4 Device- FAN

The fan is the main output for the MSP430FR6989. The MSP430 outputs a PWM signal to the two 2N7000 transistors. These transistors up the voltage from 3.3 to 5 volts for the fan. The fan speed is determined by how high the voltage. As the PWM varies the width of its pulse, the average voltage goes higher or lower from 5v, changing the fan speed.

3.5 Device- THERMISTOR

The thermistor is the main source of data for the MSP430FR6989 to read from. The thermistor used is rated at 10k ohm and this value changes based on temperature; when the thermistor gets hotter, its resistance goes down. It is connected from ground to pin P8.4 of the MSP430FR6989. Also connected to this pin is a 10k ohm resistor pulling it to ground, creating a voltage divider for the MSP430FR6989's ADC to read.

3.6 Device- TRANSISTORS

In order to convert the PWM output from the MSP430FR6989's PWM pin to the 5 volts required by the fan, two 2N7000 MOSFET transistors were used. The 2N7000 was used due to availability and two were required because the fan's maximum current draw exceeds the maximum current for a single 2N7000.

4 SYSTEM DESIGN THEORY

The system design for this project contains two major components, the MSP430FR6989 code, and the breadboard circuit. The breadboard supporting circuit is there in order to take the inputs from the thermistor to the microcontroller, as well as run the fan at an appropriate speed.

4.1 Pin Setup

Pin P8.4 is used to input the analog signal from the thermistor. This pin corresponds to A7, pin 7 of the analog-to-digital converter. Pin P1.6 is used to output the PWM signal to the fan. This pin corresponds to TA0.1, capture/compare pin 1 of Timer A0. The UART pins, TXD and RXD, are used for transmitting data over the UART communication protocol to a host computer. Pin P2.0 is used for RX and P2.1 is used for TX.

4.2 UART Communication

The MSP430FR6989 processor communicates with a host computer over universal asynchronous receiver-transmitter, or UART, communication. UART's transmission speed to a baudrate of 9600, 8 data bits and 1 stop bit.

Two commands can be sent to the MSP430 over UART, T and F. Each command is followed by an argument, which is a number in decimal. For example, T25 is a valid command. T sets the controller to automatically control the fan to cool the heat source to a specific temperature. The temperature is specified in the command's argument, in Celsius. F sets the fan speed, a decimal number between 0 and 10,000.

The processor uses an interrupt to receive incoming bytes, then pushes them to a queue. The main loop of the program runs a function that checks the queue, executes commands that are sent. This includes converting the command argument from a string of bytes to a number.

The current temperature of the thermistor is output constantly. The floating point value representing the temperature is multiplied by 1000 and rounded to the nearest whole number. It is then sent into an algorithm which converts the number to a string of bytes to be sent to the UART's transmission buffer. Since the value was multiplied by 1000, the temperature output is in thousandths of a degree Celsius.

4.3 Pulse Width Modulation

The first requirement is the output of the PWM. PWM stands for Pulse Width Modulation, and allows for control of the fans speed. The PWM works by modifying the duty cycle of the fan to a certain percentage. The lower the percentage, the slower the fan will be. With the

In the code, this is implemented using the timers capture and compare values. The fan pin is connected to Timer A in the FR6989, and are controlled via a PID control loop. This repeatedly makes an error calculation between the current temperature and the desired temperature and will adjust the PWM output appropriately.

4.4 PI Loop

A PI loop, also known as a proportional integral loop, is function fundamentally finds the difference between the current temperature value and the desired temperature. It uses this as a way to exponentially control the speed of the fan, the higher the error the faster the fan speed, and this will in return slow down at a non linear rate as well. This code also includes a ”kick” function to get the fan to start spinning originally as it takes a higher PWM to turn it on then it does to keep it turning.

5 Getting Started/How to use the device

To interface this device with a computer, the TXD line of the first or only board in the chain must be connected from a computer or other controller via two wires, the computer's RX connected to pin P2.1 of the MSP-EXP430FR6989 and the computer's TX

connected to pin P2.0, or the integrated USB to serial UART controller built into the MSP-EXP430FR6989 board from Texas Instruments.

NOTE: If using the integrated USB to serial UART controller on the MSP-EXP430FR6989 board, a USB type A to USB type micro B must be connected between the host computer and the MSP-EXP430FR6989 board.

6 Getting Started Software/Firmware

In order to interface with this board, a serial terminal is needed on the host computer. This terminal will be used to send and receive data from user to the MSP430 board. RealTerm was used for testing.

6.1 Hierarchy Chart

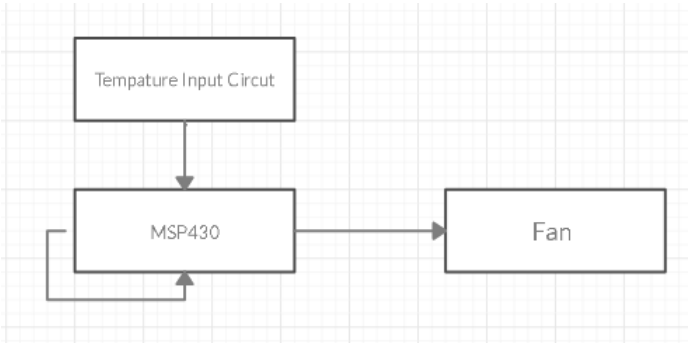

Figure 5: Heirarchy Chart

The MSP430 takes in the temperature from the thermistor circuit and either waits since the temperature is correct or turn on the fan to an appropriate speed until the device is cooled.

6.2 Communicating with the Device

The boards can be connected up to a user terminal through the USB port on the test board, allowing it to link up to a user's PC. From the terminal a temperature or a fan speed can be selected.For more details see UART communication.

7 Test Setup

The setup of the device starts with plugging the board into a computer in order to load the code. Once the code is debugged, it can be flashed onto the board so it will run when connected to other boards.

Using RealTerm, bytes can then be sent to the board and the Fan. In RealTerm, the baudrate should be set to 9600, the display set to HEX[space], and the Half Duplex box checked. In the laptop's Device Manager, the COM port associated with UART can be viewed. This is the port that it should be set to in RealTerm. Pushing the "Change" button in the Display and Port tabs saves these settings. The figures below show what RealTerm should look loke before bytes are sent.

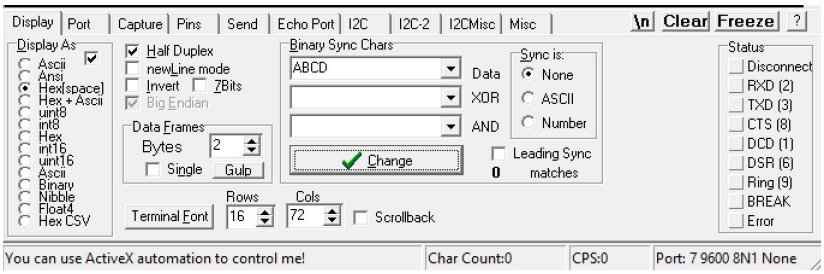

Figure 6: RealTerm Display Tab Setup

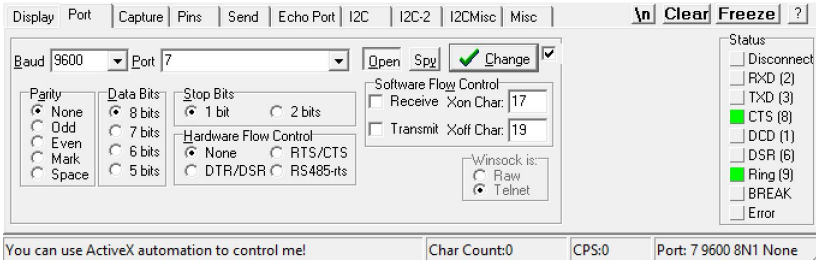

Figure 7: RealTerm Port Tab Setup

7.1 Test Data

Using Realterm to send data, the string "T0" which will turn on the fan and it will run at varying speeds unless it manages to get that low. for a more accurate test use T"current temperature in Celsius" in the environment you are testing such that it is possible for the fan to cool the thermistor to the temperature. Upon reaching this temperature the fan should stop until the temperature rises back above the set value.

8 Design Files

8.1 Schematics

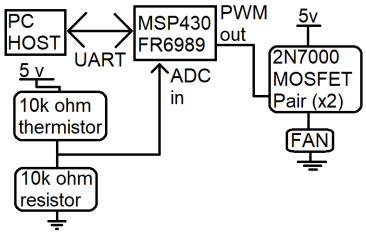

Figure 8: Schematic for the MSP-connected circuit