AN-Milestone 2: Closed Loop System

Nick Scamardi and Nick Setaro
Rowan University

December 3, 2018

1 Design Overview

For Milestone 2, the ultimate objective was to create a closed loop system that could detect the temperature and maintain a user-defined target temperature. The three main components of this milestone were ADC, open loop system characterization, and closed loop system characterization. The ADC portion involved detecting temperature using a PTAT by converting voltage to a temperature. This temperature was displayed over UART. Once a temperature could be detected and displayed, a 5V regulator was used to produce heat. Based on the difference between the desired temperature and the detected temperature, the PWM of a fan was controlled using proportional control to keep the temperature at the desired value. If the temperature was above the set temperature, the PWM would increase in order to cool the system and vice versa. The microcontroller used for this project was the MSP430F5229.

1.1 Design Features

A few of the key desgin features implemented by our system include:

- Maintain constant temperature on PTAT
- Set and Display Temperature using UART
- Proportional Control of Fan PWM
- Analog to Digital Conversion Sampling

1.2 Featured Applications

The basic concepts behind this system can be used in various real-world applications. A few of these applications include:

- HVAC Systems in Buildings
- Electronic Device Cooling (Computer, Game Systems, etc.)
- Constant Temperature Control
- Protection from Overheating

1.3 Design Resources

The design components (code) can be accessed using the following link. The Milestone2.c contains the C code for the milestone.

`https://github.com/RU09342-F18/introtoembedded-f18-milestone2-phat-stax`

1.4 Block Diagram

The block diagram for this system is displayed in Figure 1. The four main blocks contain the microcontroller, PTAT circuitry, 5 Volt Regulator, and the Fan Circuitry. The arrows coming from the fan represent it blowing on the PTAT.

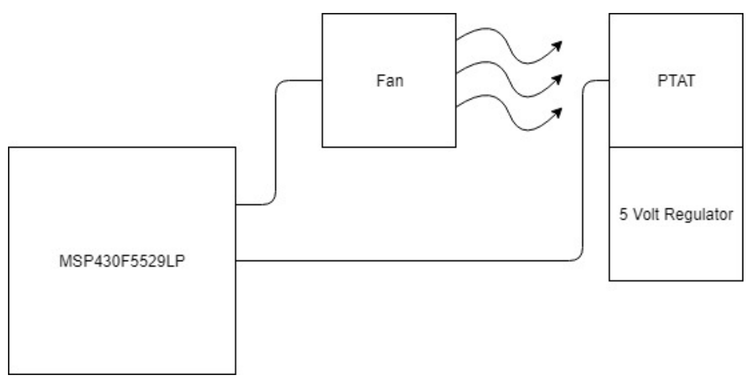

Figure 1: System Block Diagram

1.5 Board Image

The board image of our system is displayed in Figure 2 below. The system was constructed on a breadboard using the various components shown. The main devices featured are the MSP430F5229 microcontroller (Left), the PWM-enabled fan (Right), 50W power resistor, 5V regulator, power transistor, and the PTAT.

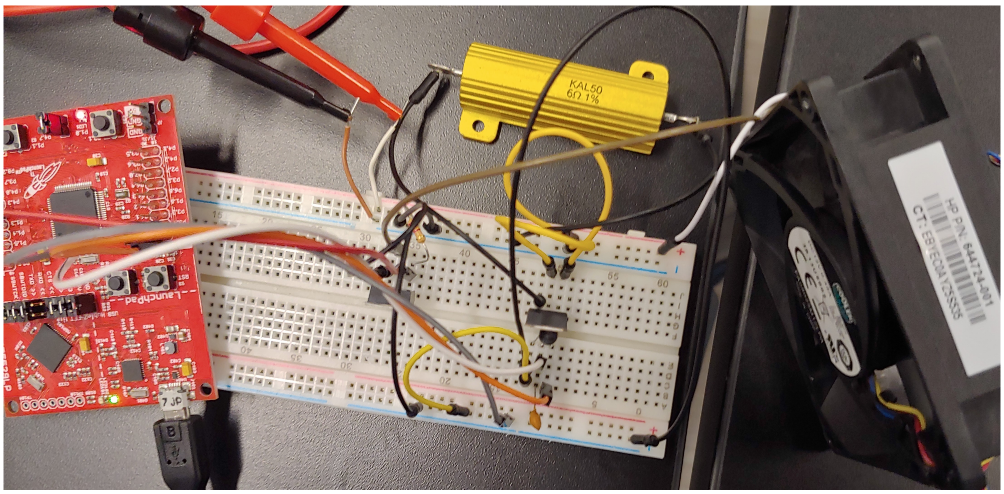

Figure 2: Board Image: Closed Loop System

2 Key System Specifications

The key specifications at which the system can perform are displayed below:

PARAMETER	SPECIFICATION	DETAILS
Set Temp. Range	20°C to 80°C	Range of temperatures that the system is able to operate. Ideally, the system should be able to perform within this range.
Oscillation Range (Steady State)	3°C	When setting a target temperature, this is the maximum error allowed in the temperature oscillations.
Voltage and Current Limits	12V and 1.5A	Maximum operating voltage of the fan and the maximum current drawn from the regulator.

3 System Description

The purpose of designing this system was to regulate the temperature of a device. This is the broad-scale purpose of the system that was created for milestone 2. In the case of our system in particular, the PTAT represented the device being cooled. 12V was used to power the fan, as well as the voltage regulator. The voltage regulator and PTAT were placed in close proximity so that the PTAT could be heated up by the regulator. The user is able to input a target temperature over UART, which is then stored in the receive buffer register and replaces the previous set temperature. Once this occurs, the fan PWM will adjust in order to allow the PTAT to heat up or cool down. Once the PTAT is at the desired temperature, the fan will adjust in order to maintain that temperature with minimal oscillations. The system was tested to operate within approximately 3 degrees of the set temperature.

3.1 Detailed Block Diagram

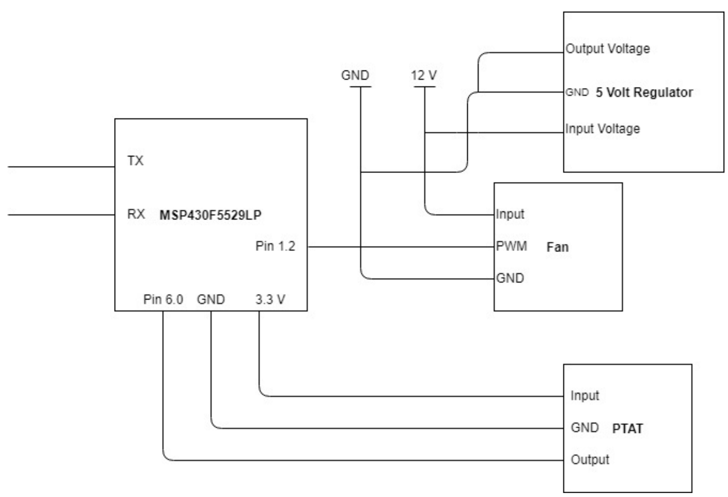

Figure 3: Detailed Block Diagram

3.2 Highlighted Devices

Below is a list of the parts used in this system.

- **MSP430F5529LP**: The MSP430F5529LP is a mixed-signal microcontroller with analog to digital and digital to analog capabilities. Used for UART communication and analog to digital conversion.

- **TMP36GZ**: The TMP36GZ is a PTAT temperature sensor that transmits a voltage proportional to the temperature and input voltage. The operating range of this device is -40°C to 125°C. The device outputs 750 mV at 25°C and 10 mV per degree Celsius. This device was used to detect the temperature by providing a voltage that was converted.
- **L7805CV3**: The L7805CV3 is a voltage regulator that limits the output voltage to 5 Volts by dissipating the rest of the energy as heat. The purpose of this device in the system was to heat up the PTAT in order to get higher temperature readings.
- **TIP31C**: The TIP31C is a power transistor. This device was used for switching the PWM of the fan in our system.

3.3 Device/IC 1: MSP430F5529LP

The processor used in this experiment was the MSP430F5529LP. This device is responsible for controlling a pulse width modulation output that drives the fan. The processor takes a voltage from the PTAT through its analog to digital converter and converts the voltage to a temperature. It then adjusts its pulse width modulation output which determines the speed of the fan. The MSP430F5529 is also responsible for UART communication using its RX and TX buffers. This allows for data to be received and transmitted, which is ultimately what controls the system.

3.4 Device/IC 2: TMP36GZ (PTAT)

The TMP36GZ is a temperature sensor with an operating range from -40°C to 125°C. This device operates in a linear fashion, which made the characterization fairly easy. The output voltage of the TMP36GZ is 750 mV at 25°C. The scale factor for this device is 10 mV per degree Celsius, which means that the voltage increases or decreases by 10 mV every time the temperature rises or falls by one degree. With this information, the voltage output by the PTAT was used in converting to a temperature reading. The output voltage was returned to the processor so that the ADC conversion could take place. The PTAT allowed for very accurate temperature detection due to its linear behavior.

3.5 Device/IC 3: Cooling Fan

The fan used for this system was a Cool Master 12V and 0.45A fan with part number FA08025M12LPA. The power and ground ports were utilized on the fans wire harness with 12V going into the fan. The PWM was controlled using a power transistor configuration and the MSP430F5529. The purpose of the fan in the system is to cool the PTAT as it heats up in order to maintain constant temperature. With the 5V regulator always dissipating heat, the fan was required to keep temperature constant, as well as to get down to lower temperatures. The PWM adjusts based on the desired behavior. Maintaining a current temperature requires a much lower PWM (fan speed) than decreasing the temperature.

3.6 Device/IC 4: L7805CV3 (5 Volt Regulator)

The L7805CV3 is a 5 Volt regulator used to heat the PTAT when voltage is applied. The regulator takes an input voltage up to 35 Volts and limits its output voltage to 5 Volts. The leftover electrical energy is then converted to thermal energy which is used to heat the PTAT. The metal side of the L7805CV3 is placed in direct contact with the PTAT to insure that there is maximum heat transfer.

4 SYSTEM DESIGN THEORY

Given the linear behavior of the PTAT, the system was fairly easy to characterize in terms of displaying the correct temperature. The voltage being read from the PTAT is converted into a temperature using the characteristics provided on the data sheet. After being able to read the temperature, the PWM was set to adjust based on the temperature difference. The target temperature and the current temperature are taken into account and the fan adjusts accordingly so that the temperature reaches the target. The following sections provide more in depth information on the design requirements as well as how the system works.

4.1 Design Requirement 1: Hold the PTAT to a target temperature

The temperature of the PTAT must stay at one value which is the target value. To achieve this the processor reads the output voltage of the PTAT through its analog to digital converter. This value is then converted to get the temperature of the PTAT. Using the current temperature, the processor calculates its error from the target temperature. The error is then used to set the fan speed with a pulse width modulation output. To ensure that the analog voltage input was a clean signal a low pass filter was placed on the output of the PTAT. This ensured that the fan was set to the correct speed.

4.2 Design Requirement 2: Reduce oscillations in temperature

When the PTAT reaches its target temperature the desired result is the PTAT remaining steadily at that temperature. Proportional control was used to achieve this design requirement. Proportional control sets the temperature by multiplying the error in the temperature by some proportional constant. The pulse width modulation is then set to this new value. This prevents oscillations in temperature by settling on one fan speed that holds that temperature.

5 Getting Started/How to use the device

This section covers the initialization and setup of the system from a mostly hardware standpoint. This section, along with its subsections, will provide instruction on how to configure and optimize the circuit, as well as how to use the overall system. The software aspect of this system is covered in Section 6.

5.1 Configuring the Circuit

To configure the circuit, refer to the block diagrams and schematics provided in this report. The block diagram is located in Figure 3 and the schematic is displayed in Section 8.1. When configuring the circuit, ensure that the 5V regulator and PTAT are very close together, if not touching. This is required in order to heat the PTAT to the desired temperatures. Also, the fan should be oriented in a way that will cool the device sufficiently. Both the fan and voltage regulator must be connected to a 12V input and there must be a common ground established. The microcontroller should be powered through the computer via USB with the correct code running. The next section discusses the setup from a software perspective.

6 Getting Started Software/Firmware

Once the circuit is configured, the rest of the setup is done through software. The following sections will cover executing the code and setting the target temperature. A hierarchy chart of the system is also provided. On the software end, two programs are needed to interface with the system.

- Code Composer Studio 8.1.0
- RealTerm or PuTTY

Code Composer is used to run the code and send it to the microcontroller. Once the code is flashed to the controller, code composer is no longer needed as the program will be stored on the microcontroller until removed or overwritten. As for RealTerm, this is used to communicate with the system and provide target temperatures as well as display the current temperature.

6.1 Executing the Code

The required code for this system is provided in the team repository located on Github. The link for the repository is located in Section 1.3. The required file is the Milestone2.c file, which is the complete code for the system. Once the code is inside Code Composer, execute the Debug function. After debugging successfully without errors, press the play/resume button to flash it to the board. This completes the code execution, as it is now running on the microcontroller.

6.2 Communicating with the Device (Target Temperature)

RealTerm or PuTTY is used to change the target temperature of the system over UART. After connecting RealTerm or PuTTY to the correct port the temperature can be set by sending it from the send tab. The system will then adjust the temperature of the PTAT to reach the new target temperature.

6.3 Hierarchy Chart

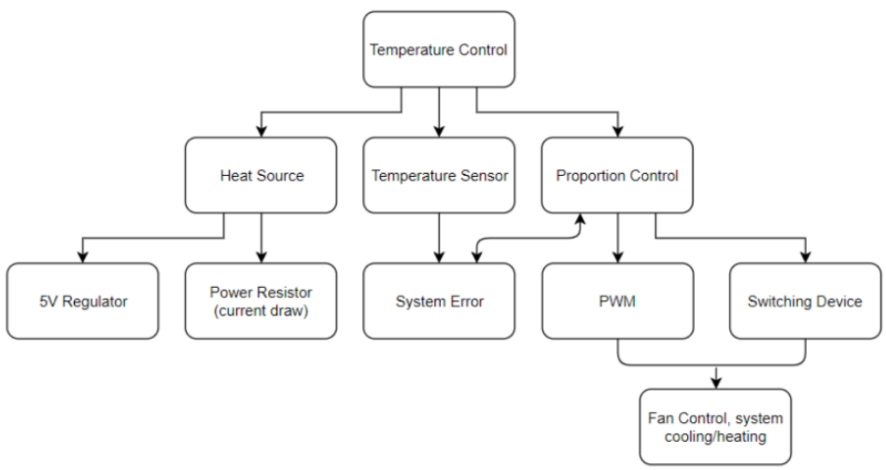

Figure 4: Hierarchy Chart of Closed Loop System

7 Test Setup

Test Setup includes flashing the code to the microprocessor and using RealTerm to send and receive values. Once the code is on the device, open RealTerm and select the correct port and open it. Next, set the BAUD to 9600 and the display type as uint8. Half duplex mode can also be enabled in order to see what is being sent and received. Assuming everything is connected properly, there should be a feed that is displaying the current temperature reading. Make sure the 12V power supply is connected and on as well so that the regulator will heat up. To set a target temperature through UART, go to the send tab, type in the desired value, and press send. The value entered should appear in green inside of the RealTerm window to signify it is being sent to the device and not output from it.

7.1 Test Data

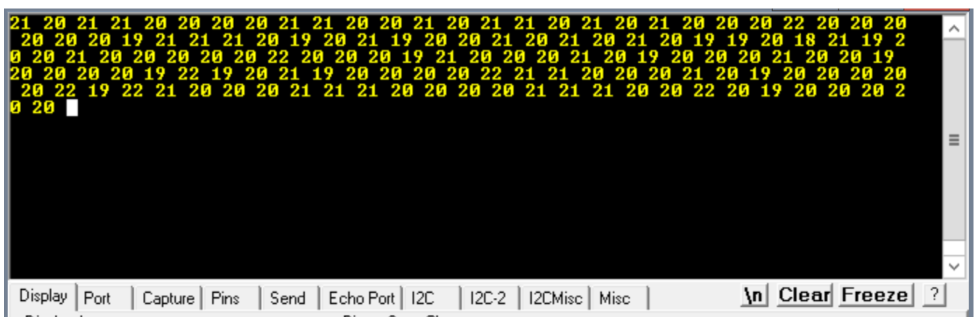

Figure 5: Room Temperature detection test data

Figure 5 shows the feedback from the microcontroller at room temperature. The reading hovers around 20°C, which is approximately room temperature. The yellow indicates that the values are being output from the system. If a target temperature is entered, the temperature would change until it reached the target and would then display the same behavior.

8 Design Files

8.1 Schematics

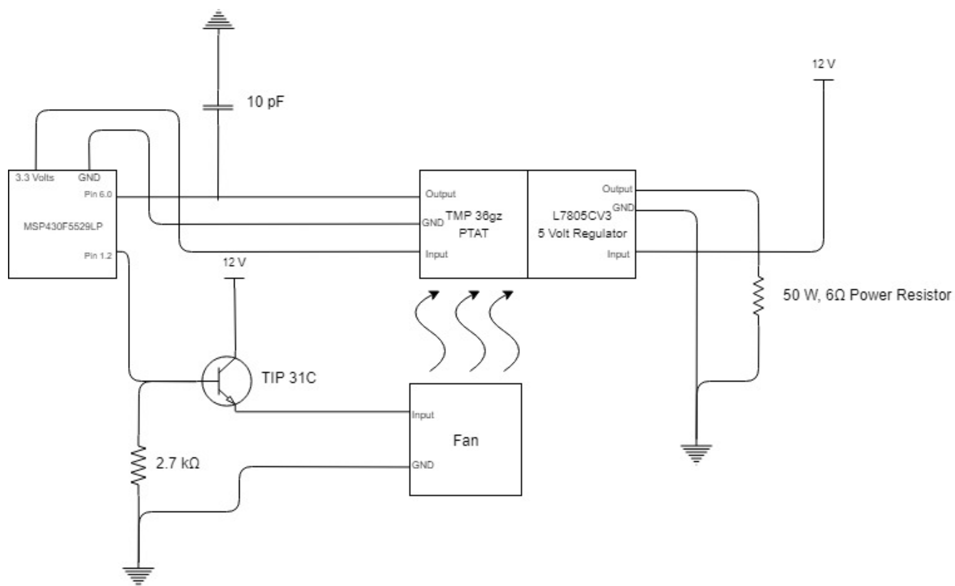

Figure 6: Schematic of System

8.2 Bill of Materials

- MSP430F5529LP Microcontroller
- 12V Cool Master FA08025M12LPA Fan
- TMP36GZ PTAT
- L7805CV 5 Volt Regulator
- TIP31C Power Transistor
- 6Ω 50W Power Resistor
- 2.7kΩ Resistor
- 10 pF Capacitor