Milestone 2: Closed Loop Systems

Scott Wood and David Sheppard
Rowan University

December 3, 2018

1 Design Overview

Milestone 2 involved the design of a user controlled fan with utilization of UART communication. This design required that the MSP430F5529LP be programmed with software PWM to control the speed of a fan according to a temperature that is read by the inbuilt 12 bit ADC on the MSP430F5529LP or that is sent by a user over UART. This design required that the circuit be designed with a PTAT (Proportional To Absolute Temperature) sensor and that a voltage regulator be used to heat the PTAT such that the temperature data read by the 12 bit ADC is first read by the PTAT and fan speed is ultimately controlled with software PWM such that a stable temperature is maintained. As mentioned, a user can control the speed of the fan by sending a desired temperature in Celsius over RealTerm (UART), and stability is achieved with utilization of a closed loop control system described as a modified Bang-Bang Controller.

1.1 Design Features

These are the design features:

- Fan Speed Control with Software PWM
- 9600 Baud Rate
- LMP0 (Low Power Mode 0)
- 12 bit ADC
- Flyback Diode
- Bulk Capacitor for noise reduction

1.2 Featured Applications

Possible applications:

- Protection against overheating of Electronic Devices
- Stable Fan Source for Industries and Businesses
- General Comfort

1.3 Design Resources

The code and README for the project can be found at the following link: `https://github.com/RU09342-F18/introtoembedded-f18-milestone2-how-s-it-going`

1.4 Block Diagram

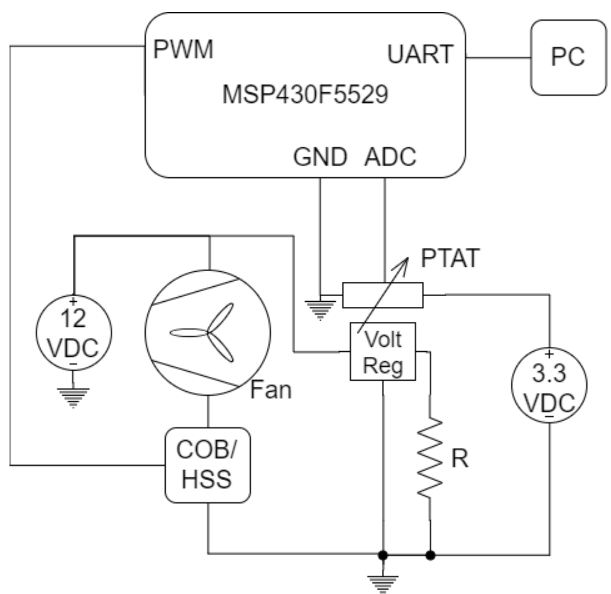

Figure 1: Block Diagram of Closed Loop System

1.5 Board Image

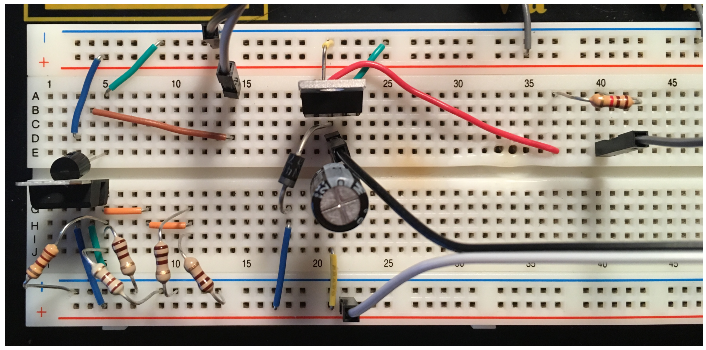

Figure 2: Image of the breadboard during usage

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
ADC	12 BIT	Reference voltage = 3.3 V Conversion time: 2.3-3.1 μs
Low Power Mode	0	ACLK and SMCLK High
Convert-o-Box	LSS	Allows for MCU to control 12V DC Fan
Allowed Temperature Offset	$\pm 1°C$	PWM only changes when temperature is at least 1° off

3 System Description

With many electronics, heat can often cause operational problems to arise if not dealt with properly. If an electronic device surpasses its heat rating, it could malfunction or be permanently damaged. Many devices such as laptops and desktop computers rely on fans to keep their components from overheating. The system presented here is able to control heat levels in that way. Using a 12 V fan and a microcontroller, the device is able to maintain a desired temperature by adjusting the fan speed using pulse width modulation (PWM). The device relies on temperature data from a PTAT sensor to determine the temperature of the desired element. After calculating the difference between the desired temperature and the real temperature, the program then increases the fan in a manner proportional to the temperature difference. This results in allows for the fan to respond quickly to large temperature changes without overreacting to small interferences.

3.1 Detailed Block Diagram

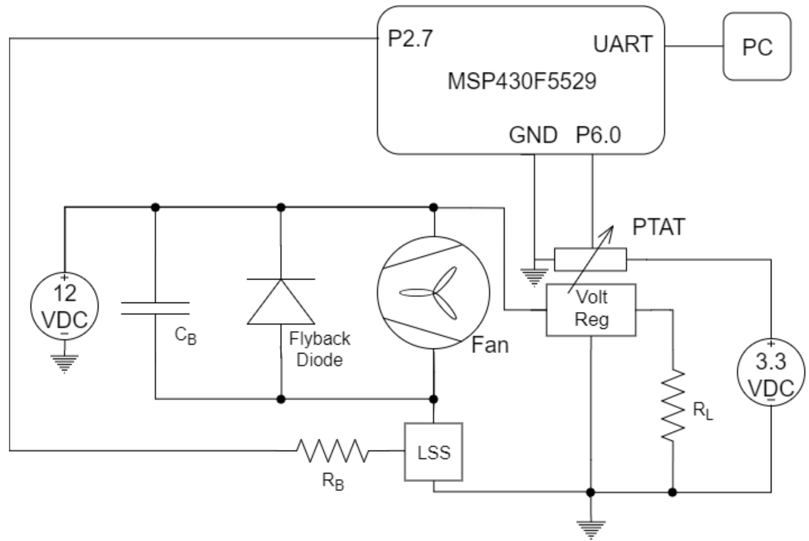

Figure 3: Detailed Block Diagram of Closed Loop System

3.2 Highlighted Devices

- MSP430F5529 - for controlling fan speed
- Fan - for cooling the heating element
- 200 μF capacitor - to reduce noise from the fan
- IN4007 diode (flyback diode) - to prevent wear on the fan
- L7805CV voltage regulator - to act as a heat source
- 110 Ω resistors (5) - to draw current from the 5 V regulator and produce heat
- LM60CIZ PTAT sensor - to measure the temperature
- TIP122G BJT - to allow the microcontroller to drive the fan
- 1.1 kΩ resistor - the base resistor for the BJT
- 3.3 VDC power supply - to provide a reference for the PTAT
- 12 V power supply - to power the fan and 5 V regulator

3.3 MSP430F5529LP

The selection of the MSP4305529LP to act as the microcontroller to read the temperature and control the fan speed was determined with consideration of the inbuilt 12 bit ADC within the MSP430F5529LP and the vast number of ADC source code available for the MSP430F5529LP within the TI Resource Explorer. For example, unlike the MSP4305529LP, the MSP430G2553, a microcontroller initially considered for the role only has a 10 bit ADC available for use. In fact, none of the other microcontrollers available to us within the Launchpad family have a 12 bit ADC.

3.4 Fan Setup

The fan that was used for testing was rated for up to 600 mA and was supplied with a 12 V source. While this fan had a built-in PWM function, the testing setup utilized only the power and ground wires on the fan because PWM had been built into the code. Since the fan was controlled by a low-side switch, the 12 V power supply was connected directly to the fan's positive terminal and the fan's negative terminal ended at the collector of the low side switch. The low-side switch consisted of an npn BJT and a 1.1 kΩ resistor leading to the base of the BJT. This allowed the microprocessor to control the fan without drawing more than the maximum 6 mA of current from the microprocessor. The general setup of the system can be seen in Fig. 3.

4 SYSTEM DESIGN THEORY

The overall closed loop system required the utilization of several important components such as: PTAT Temperature Sensor, Voltage Regulator, Low Side Switch, Modified Bang Bang Controller, and the inbuilt 12 ADC within the MSP430F5529LP. In general this system was implemented to achieve a stable temperature with the control of a fan according to a temperature read by a PTAT or by a user over RealTerm. In general, the voltage regulator supplies heat to the PTAT such that a temperature is read by the PTAT. The fan is then set to a specific speed such that a stable temperature is maintained.

The utilization of components such as the PTAT allowed for the ability to measure the temperature, which is converted to a voltage within software and within the MSP430F5529LP's inbuilt ADC. Stability of this temperature is maintained with the utilization of the modified Bang-Bang controller. Without the implementation of the Low Side Switch, control of the 12V DC fan would not be possible with such a low powered microcontroller.

The utilization of the PTAT sensor to measure and read the temperature came with the decision to not use a NTC (Negative Temperature Coefficient) thermistor or a PTC (Positive Temperature Coefficient) Thermistor to act as the temperature sensor. In general, NTC or PTC thermistors behave such that their resistance changes with temperature. For example, with the PTC thermistor, the resistance increases as the temperature increases, and the opposite is true for the NTC thermistor, the resistance decreases as the temperature increases. The use of the thermistor required use of the Steinhart-Hart equation for accurate temperature control, and calculation of the thermistor's Steinhart-Hart coefficients for a specific temperature range and overall stability and control of the fan speed. The use of the PTAT avoided the use of such unnecessary calculations of such coefficients and allowed for a corresponding temperature to be converted to analog to digital voltage directly in software with an equation specific to the PTAT.

4.1 Design Requirement 1: Modified Bang-Bang Control

The method chosen to control the fan speed was a proportional type of PWM control (modified Bang-Bang). After reading the temperature from the ADC, the difference between the real and desired temperatures was calculated and the result was used to set the new PWM duty cycle. Running in Up Mode, the timer controlled the PWM module such that the output was set low when CCR1 overflowed and set high wan CCR0 overflowed. For each degree of temperature difference, the timer's CCR1 value was increased or decreased by 1. Since the CCR0 value was set to 262, an increase in the CCR1 value translated to an increase in duty cycle of about 0.4% for every degree Celsius that the real temperature was off by. Although that is a relatively small increase, the ADC is able to perform its conversions in about 3 ms. As a result, the fan speed was found to respond quickly enough to temperature changes without oscillating too quickly. Since the requested accuracy was $\pm 3^\circ$, it was decided that the PWM would not be changed if the temperature reading was within 1° of the desired temperature. This helped to keep the fan stable while the temperature was reasonably close to the desired one.

One of the advantages to using the proportional modified bang-bang control is simplicity. Another method that could have been used to set the PWM is the proportional integral derivative (PID) control. This method prevents the PWM values from oscillating too quickly by tracking the rate at which the duty cycle is changing. This results in a very steady fan speed, but at the cost of time and efficiency. During testing, the proportional modified Bang-Bang control proved to be very steady and effectively maintained the desired temperature without major oscillations in the fan speed. As a result, this method was deemed sufficient. The simplicity of this method allows for faster response times and uses less of the power-consuming calculations that are necessary for PID control.

Listed below is an example of the Modified Bang-Bang Control implemented in software, which is utilized in the operation of the stability and overall control of the fan. For example, one can observe that the basic functionality of the code is to increase or decrease the respective ccr value and in doing so increasing or decreasing the speed of the fan depending on the difference of the real temperature and the desired temperature. The implementation of this algorithm required only two simple conditional statements since the goal is to either increase or decrease the fan speed depending on the temperature difference value.

```c

void setPWM()
{
  //proportional control: change PWM based on difference in temperature ************************************
    if(difference > 1)                                    //if temp is too high
        ccr += (difference);                              //increase ccr value to speed up fan
    else if(difference < -1)                              //if temp is too low
        ccr += (difference);                              //decrease ccr value to slow down fan (difference is already negative if realTemp is too low)
    //else if within 1 degree, don't bother to change PWM

    if(ccr < 0)                    //ensure ccr never gets negative
    {
        ccr = 0;
    }
    else if(TA1CCR1 > 250){        //ensure CCR1 never gets too close to CCR0 (CCR0 is 262)
                                   //if CCR1 is too close to CCR0, they won't both fire
        ccr = 250;
    }

    TA1CCR1 = ccr;                 //finally set the CCR1 value
}
```

4.2 Design Requirement 2: Low Side Switch

In order to drive the fan with the microcontroller, a low side switch needed to be implemented. This is due to the fact that the MSP4305529LP, the micronctroller selected to drive the fan was not designed to output more than 6 mA while the fan was designed for a much greater current - up to 600 mA. The low side switch was constructed with an npn-type TIP122G bipolar junction transistor (BJT) and a 1.1 kΩ resistor. The resistor helped to achieve an optimal β level to drive enough current through the BJT to power the fan without surpassing the 6 mA current limit of the microcontroller.

5 Getting Started/How to use the device

The device and the overall system's functionality can be described in such a that anyone can use it and appreciate a user controlled stable fan source within a few steps. For example, as long as one possesses the required devices listed in the Highlighted Devices section, the source code listed in the Design Resources section, and the correct circuit configuration listed as Fig. 5 in the Design resources; the device should function with no issue upon compilation of the code. More detail for information regarding the specific steps for functionality of the device can be found within the subsections below.

5.1 Connecting the Device

Prior to connecting the device so that the device can be used properly. The ports on the MSP430F5529LP must have the following configuration: 6.0: ADC input, 2.7: PWM output, 4.4: TX, 4.5: RX. The device can then be connected with a micro USB cable to a PC to power the MSP4305529LP. If the PC contains the proper software required to run the code for the device and a proper UART terminal such as RealTerm, or the potential to run such software, the device can be run with success as long as the proper code is compiled and the circuit described in Figure 3 is built correctly with the proper port connections mentioned above.

5.2 Circuit Setup

The circuit must be powered such that 12 VDC is supplied to the fan and voltage regulator and 3.3V DC is supplied to the PTAT. To ensure that the PTAT reads a temperature properly from the voltage regulator the two components were wrapped in thermal conductive material to ensure close proximity between the two for proper heating conditions.

6 Getting Started Software/Firmware

The program itself, which allows for the user control of a fan with the MSP430F5529 is written in C within an Integrated Development Environment (IDE) called Code Composer Studio (CCS). One would have to have CCS installed and the correct MSP430F5529LP Family Device library selected to run and build the program on the MSP430F5529.

In order to successfully transmit data with the use of RealTerm, the proper UART settings had to be calibrated within RealTerm, which required a 9600 baud rate to be selected and set from the drop down menu labeled as Baud. This Baud Rate defined the rate at which the binary packets were transferred in the UART communication channel. Furthermore, the user has to ensure that the correct COM port is selected for proper communication to be achieved from their device to another or to their own device, in order to do this device manager must be pulled up and the USB PORT number, which is connected to the UART cable and visible in device manager must be entered and selected from the drop down menu labeled as Port.

6.1 Hierarchy Chart

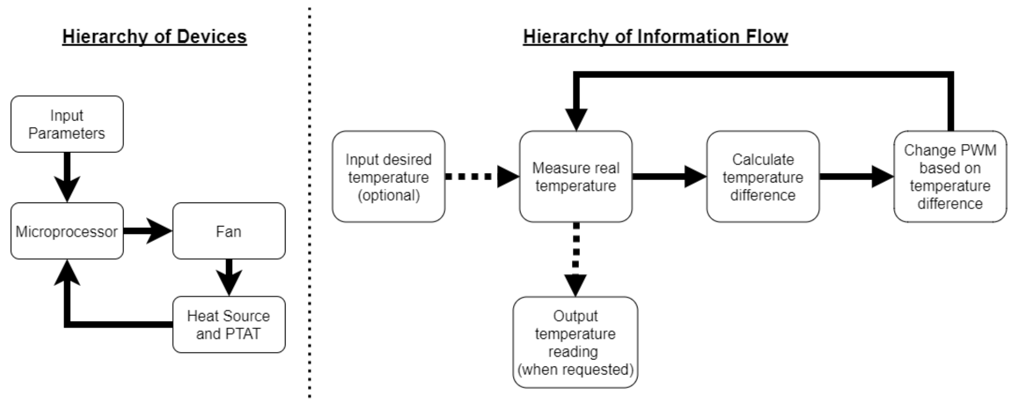

Figure 4: Hierarchy Chart of System

6.2 Communicating with the Device

Information can be sent to and received from the MSP430F5529LP using a micro USB cable and a terminal such as RealTerm. Prior to successfully sending any temperature values to the device and after connecting the MSP430FF29LP with a micro USB cable to a PC, the user must open the PC's device manager and find the COM Port number of the connected board. This is found under *Ports (COM & LPT)* in the device manager. The desired COM Port number is that which follows the term *UART* in the list.

Once the COM Port number is determined, RealTerm can be opened up and the BAUD rate (9600) and COM Port number specified under the *Port* tab.
The user is then able to successfully control the speed of the fan by sending specific temperature values in Celsius over RealTerm. The user is also allowed to verify that the device is functioning correctly with ease by sending over a value of 0 over RealTerm so that the current temperature is displayed.

6.3 Device Specific Information

The MSP430F5529 contains a 12-bit analog to digital converter. It can be powered by 2.4-3.6 V and should not exceed an input or output current of more than 6 mA per pin or 48 mA overall. It can be operated from -40°C to 85°C. It utilizes the SMCLK which has a frequency of 1 MHz which can be divided or multiplied as desired.

The program was written with a 3.3 V reference voltage in mind. As such, the PTAT must be connected to a 3.3 V power supply to obtain proper readings. While the fan that was used for testing was supplied by a 12 VDC source, a different voltage could be used with an appropriate fan (provided that an appropriate low-side witch is used), as the microprocessor simply drives the switch that controls the fan, not the fan itself.

7 Test Setup

To ensure that the device and the overall closed loop system was achieved such that a stable temperature was maintained, several tests were performed with the device connected to a PC for UART communication. In general, the tests involved sending a desired temperature in °C value over RealTerm such that the fan speed was set for the following three cases: Room Temperature to Medium/Hot , Medium/Hot to Slightly Warm, and Slightly Warm to Medium/Hot Cold.

With a default desired temperature of 20°C, success of the device was verified by sending in over RealTerm a new desired temperature in °C, which would accomplish the three cases mentioned above: Room Temperature to Medium/Hot, Medium/Hot to Slightly Warm, and Slightly Warm to Medium/Hot Cold. For each case, the desired temperature appeared on the terminal of RealTerm, which verified further success of the device's software functionality because it required sending a value of 0 over RealTerm so that the current temperature was displayed on the RealTerm Terminal.

For the initial tests of the device, the temperature value recorded by the PTAT would increase at a low rate. To resolve this, five resistors of value of 110 Ω were placed in parallel and connected to the voltage regulator. This is shown in Fig. 5. After this connection was completed, the rate of temperature would increase or decrease at a substantially increased rate, which improved the accuracy and efficiency of the tests and the overall test setup.

7.1 Test Data

The fan control and overall stability of the fan was tested by sending the following temperature values in Celsius over UART to satisfy the following three cases: Room Temperature to Medium/Hot , Medium/Hot to Slightly Warm, and Slightly Warm to Medium/Hot Cold.

- Room Temperature to Medium/Hot: 65°C
- Medium/Hot to Slightly Warm: 55 °C
- Slightly Warm to Medium/Hot: 65°C
- Display Current Temperature: 0°C

8 Design Files

8.1 Schematics

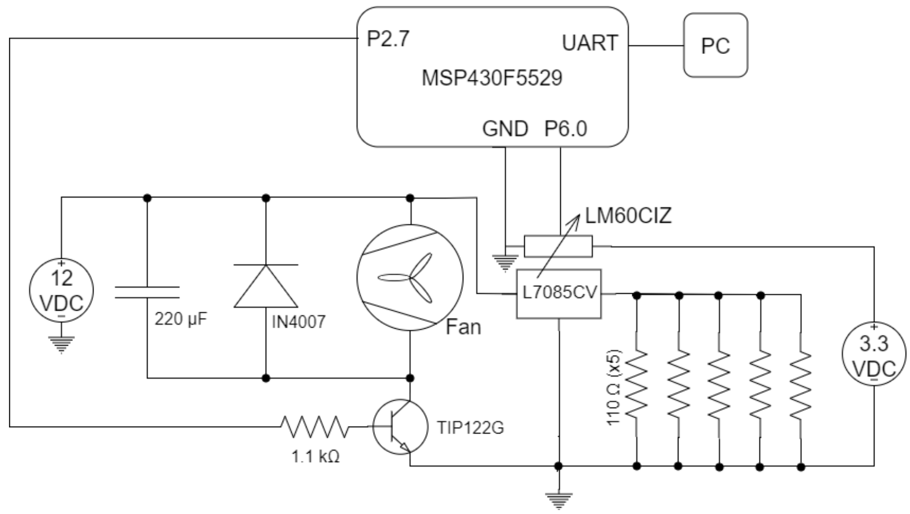

Figure 5: Schematic of Closed Loop System

8.2 Bill of Materials

Device	Cost	Quantity
MSP-EXP430F5529LP	$12.99	1
Intel 12V DC Fan	$16.90	1
Breadboard Kit	$3.99	1
110 Ω Resistors	$0.04	5
1.1 kΩ resistor	$0.112	1
IN4007 diode	$ 0.18	1
TIP122G BJT	$0.70	1
LM60CIZ PTAT sensor	$1.02	1
L7085CV voltage regulator	$1.70	1
200 μF capacitor	$0.82	1