Closed Loop Systems - Milestone 2

Tiernan Cuesta and Kevin Purcell
Rowan University

December 2, 2018

1 Design Overview

This closed loop system's task is to maintain target temperature of a physical object, in this case a 5 volt regulator, using a feedback loop design with a microcontroller to perform control algorithms. The purpose of this is to determine if a control algorithm can be implemented using the MSP430F5529 to keep any sort of object at a certain temperature using a cooling system, in this project a PWM controlled fan was chosen as the cooling system. The PWM signal sent to the fan is calculated by the difference in system temperature and the set point temperature using proportional control. The temperature of the system is determined by a thermistor in a voltage divider topology to utilize the on-board ADC, analog to digital converter, for calculations and control. An image of the assembled project can be seen in Fig.1.

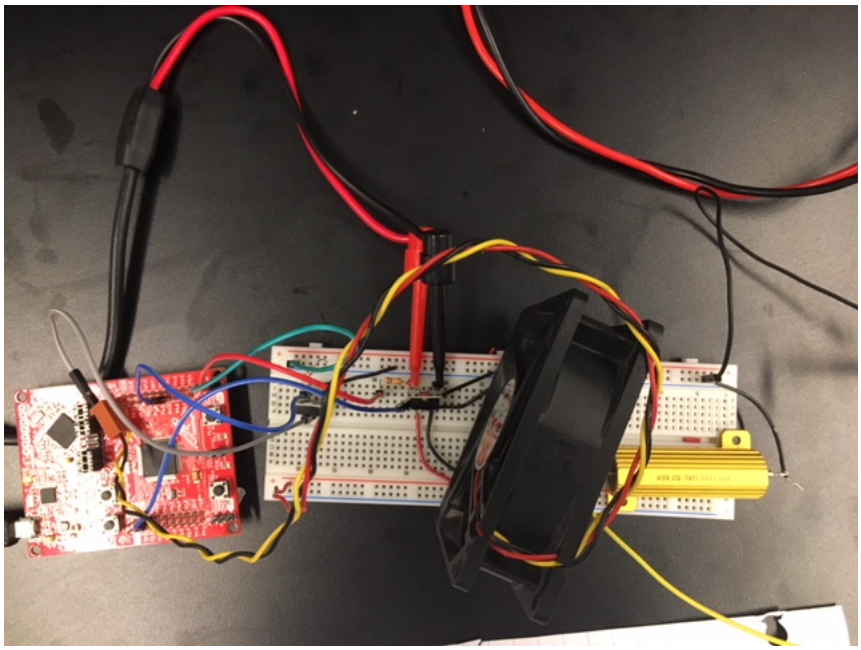

Figure 1: Fully-assembled system under test.

1.1 Design Features

These are the design features of the closed loop system:

- Serial Communication
- Single Channel, single port analog to digital conversion sampling
- Single port PWM
- Proportional Control

1.2 Featured Applications

These are some featured applications that this project could support given possible slight modifications per application:

- HVAC systems
- Motor/Generator temperature control
- Computer cooling systems
- Safety shutdown systems

1.3 Design Resources

This is a quicklink to the program files for the project:

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

1.4 Block Diagram

This very high level diagram explicitly shows that the system requires two converter subsystems where one converter is an output and another is an input to the microcontroller. This feedback loop is important for the overall design of the system since the temperature regulation of the system depends on it's current state.

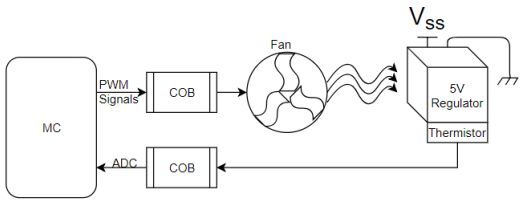

Figure 2: High-level block diagram of the closed loop system.

1.5 Board Image

The blue wires going out of the picture are data lines that are used for the PWM control and analong to digital coversion sampling.

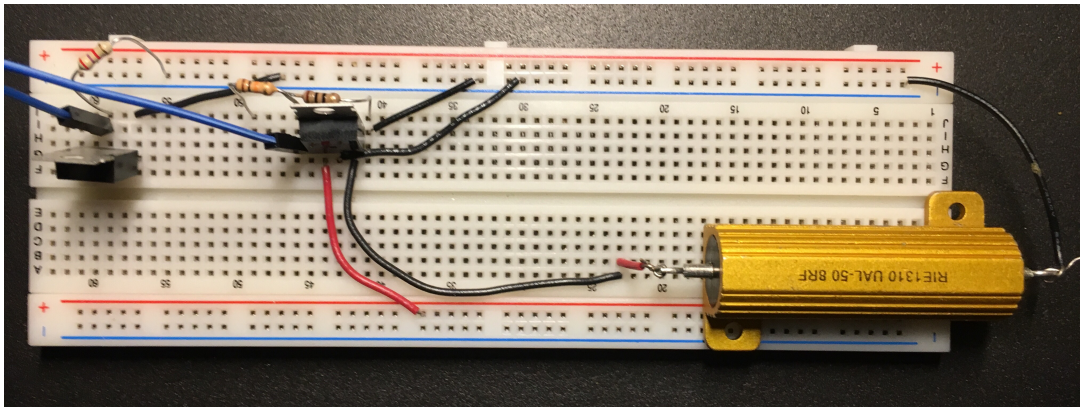

Figure 3: Photo of the protoboard setup excluding the fan and MSP430F5529

The device on the left most side of the board is the switching transistor that is

responsible for PWM transmission to the fan via a low side switch topology. The device on the right is the 5 volt regulator that is producing the heat for the system. The 12 volt supply is the red rail on the bottom of the board and common ground is the blue rail on the top side.

2 Key System Specifications

PARAMETERS	SPECIFICATIONS	DETAILS
Set Point	35-70°C	Desired temperature of system
True Point	25-80°C	Recorded temperature of system
Fan Speed	0-100 duty cycle	PWM control of the fan, cooling system
Voltage Max	12V	Maximum voltage for the regulator and LSS
Current Max	1.5A	Maximum current draw from the regulator and LSS

Table 1: Various parameter types required by the system

3 System Description

The main goal of this project is to emulate a system in which it needs to maintain a specified temperature. An example of a system that needs to maintain a specified temperature is an internal combustion engine found in most cars. Similar to this project a fan is turned on and off to cool down the radiator that pulls heat away from the engine block. The engine block in this project is a 5 volt regulator that is being powered by 12 volts, same as the fan, and an 8Ω 50W power resistor to draw current and power from the regulator, the heat source. The NTC thermistor is in contact with the voltage regulator's heat sink and is in series with a 10kΩ resistor for a voltage divider topology. The output voltage of the divider is read by the onboard Analog-to-digital converter of the MSP430. Temperature calculations based off of these voltages are made and sent to the TXBUF to be read and recorded with serial communication. Additionally, the RXBUF receives the target temperature of the system and the program takes the error of actual temperature and the desired temperature of the operator. This error is extrapolated and implemented in a proportion control algorithm to control a fan with hardware pulse width modulation. In Fig.4 there is a snippet of this algorithm.

```
volatile float P = diff * 70;                    // Proportion control constant
// Proportion control functions
      if (P == 0)
      {
          TA0CCR1 = 0;
      }

      else if (P >= 255)
      {
          TA0CCR1 = 255;
      }

      else if (P < 0)
      {
          TA0CCR1 = 0;
      }
      else
      {
          TA0CCR1 = P;
      }
        break;
```

Figure 4: Proportional control algorithm for fan speed control.

3.1 Detailed Block Diagram

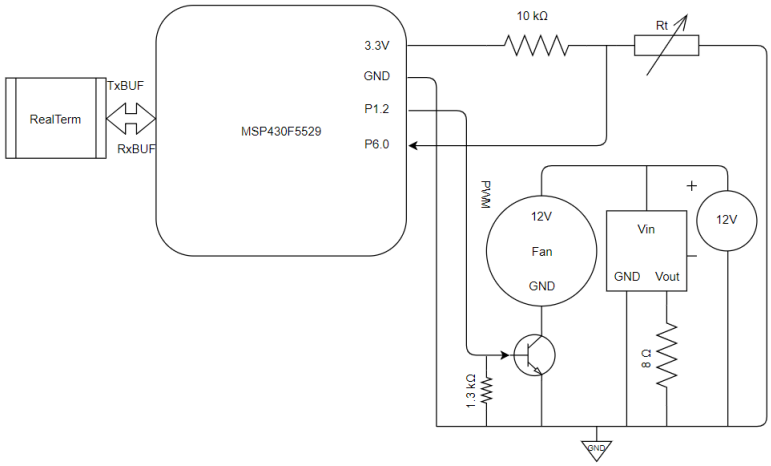

Figure 5: Detailed block diagram of the closed loop system.

3.2 Highlighted Devices

- MSP430F5529: The microprocessor contains the protocols for the ADC, PWM, and UART for the fan speed and temperature conversions for the thermistor.
- 5 Volt Regulator: A voltage regulator is a system designed to automatically maintain a constant voltage level. The voltage regulator was used as the heating mechanism.
- 50W 8Ω Power Resistor: The power resistor was used to draw current from the voltage regulator to dissipate power in the form of heat.

- TIP31C Transistor: The TIP31C is an NPN power transistor typically used for switching. It was used to control the PWM for the fan speed.
- NTCLE100E3 Thermistor: This device changes resistance due to temperature changes, and this resistance can be measured by applying a known voltage across it in a voltage divider topology. The NTCLE100E3 was used to read the temperature of the voltage regulator.

3.3 Device/IC 1

MSP430F5529 Microprocessor:

The MSP430F5529 was used as the brain of the circuit. It's role is to send PWM signals to control the fan's speed, send and receive UART messages, and read and convert the thermistor's nomimal voltage drop to a readable value.
Pulse width modulation is used to control the fan's speed to keep the voltage regulator at a constant temperature. If the difference between the actual temperature of the voltage regulator and the desired temperature is zero, the fan speed will remain constant. If the difference between the actual temperature of the voltage regulator and the desired temperature is less than zero, the fan speed will the fan speed will slow down and eventually stop, and if the difference is greater than zero the fan speed will increase until the difference again reaches zero.

```
void configurePWM() {
    // Sets P1.2 as the output pin
    P1DIR |= BIT2;                          // Sets P1.2 as output driver for pwm for fan speed control
    P1SEL |= BIT2;                          // Selects the port 1.2 as the timer A output
    TA0CTL = TASSEL_2 | MC_1 | TACLR;       // Sets timerA_0 to SMCLK, up-mode, clears the register
    TA0CCR0 = 255;                          // Sets CCR0 max pwm
    TA0CCR1 = 30;                           // Sets CCR1 to initial value;
    TA0CCTL1 = OUTMOD_7;                    // Output mode 7 for set/reset
}
```

Figure 6: PWM Setup.

Universal Asynchronous Receiver/Transmitter is used to send and receive signals from the MSP430F5529 to RealTerm or any serial communication firmware. The baud rate was adjusted to match the speed at which data is transferred in order to observe a readable value on the firmware. In this project a baud rate of 9600 was utilized, a Baud rate of 9600 means to have a transfer speed of 9600 bits per second achieved by clock division and modulation.

```
void configureUART() {
   P4SEL |= BIT5;                                  // Enables RX and TX buffer
   P4SEL |= BIT4;
   UCA1CTL1 |= UCSWRST;                            // Software reset enable
   UCA1CTL1 |= UCSSEL_1;                           // USCI clock source select - ACLK
   UCA1BR0 = 3;                                    // Baud rate clock divider1 for 9600 BR
   UCA1BR1 = 0;                                    // Baud rate clock divider2 for 9600 BR
   UCA1MCTL |= UCBRS_3 | UCBRF_0;                  // First and second stage modulation for higher accuracy baud rate
   UCA1CTL1 &= ~UCSWRST;
   UCA1IE |= UCTXIE;                               // Enables Transfer buffer interrupt
   UCA1IE |= UCRXIE;                               // Enables Receiver buffer interrupt
}
```

Figure 7: UART Setup.

The Analog to Digital Converter was needed to properly analyze the input from the thermistor to a readable value. However, in order to properly read the temperature value from the thermistor, the Stienhart and Hart equations were used to convert the resistance value to temperature in Celsius. In Fig.8 the various registers and control registers were initialized.

```
void configureADC() {
   P6DIR &= ~BIT0;                                 // Sets P6.0 to input direction for ADC_12A input from voltage divider
   P6SEL |= BIT0;                                  // Sets P6.0 as the input for ADC12_A sample and conversion
   ADC12CTL2 = ADC12RES_2;                         // AD12_A resolution set to 12-bit
   ADC12CTL1 = ADC12SHP;                           // ADC12_A sample-and-hold pulse-mode select - SAMPCON signal is sourced from the
   // ADC12_A Control Register 0 - 1024 cycles in a sampling period - Auto Trigger - Ref Volt off - Conversion overflow enable - Conversio
   ADC12CTL0 = ADC12SHT1_15 | ADC12SHT0_15 | ADC12MSC | ADC12ON | ADC12TOVIE | ADC12ENC | ADC12SC;
   ADC12IE = ADC12IE0;                             // Enables ADC12 interrupt
   ADC12IFG &= ~ADC12IFG0;                         // Clears ADC12 interrupt flag
}
```

Figure 8: ADC Setup.

3.4 Device/IC 2

TIP31C Transistor:

The TIP31C Transistor was used in the circuit as a low-side switch to control the PWM for the fan since the fan did not have a PWM control pin built in. Acting as a low-side switch the base of the transistor was connected to pin 1.2 on the MSP430F5529, the collector pin to the fan, and the emitter grounded. Depending on the average voltage supplied to the base pin by the MSP430F5529, the fan would speed at a speed proportional to the duty cycle. If the voltage was greater then the cut-off, the transistor will be in the active and saturation region, the fan will turn on, if the voltage supplied is less than the cut-off, the transistor will be in the cut-off region and the fan will turn off.

4 SYSTEM DESIGN THEORY

The system in this project is responsible for maintaining a subsystem at a specified temperature. A 5 volt regulator that regulates 12 volts and consumes power with a 50W power resistor to emulate the heat of a system such as an ICE. The temperature

of the regulator is monitored using an NTC thermistor in a voltage divider topology with a 10kΩ. The voltage output of the divider is converted to a digital value using the onboard ADC. This voltage is a resultant of the changing resistance of the thermistor and can be calculated using equations.

$$ThermistorResistance : R = \frac{ADC * R_s}{Bits - ADC} \tag{1}$$

$$R_t = log(\frac{R}{R_s}) \tag{2}$$

$$SteinhartandHart : T_k = (A_1 + B_1 * R_t + C_1 * R_t^2 + D_1 * R_t^3)^{-1} \tag{3}$$

Where A_1, B_1, C_1, and D_1 are constant values depending on material, see Appendix A for thermistor datasheet that contains these constants. ADC is the value read from the analog to digital converter from the voltage divider, Bits is the resolution of the ADC12 which is 2^{12}, and R_s is 10,000kΩ. After converting T_k from kelvin to celcius the error between the measured temperature and target desired temperature is calculated to implement the proportion control algorithm in Fig.4.

4.1 Design Requirement 1

Keeping a constant temperature: One of the main purposes of the milestone was to be able to keep the voltage regulator at a constant temperature. In order to do so, we needed to be able to successfully read the temperature on realterm from the thermistor. The UART was properly configured with a baud rate of 9600, pins 4.4 and 4.5 enabling the RX and TX buffer, and pin 6.0 being the input direction for ADC12A input from voltage divider. Equations 1, 2, and 3 were then used, by the ADC, to convert the resistance value of the thermistor to a temperature reading, in Celsius.
Once the temperature is able to be properly read, the proper PWM for the fan must be configured. Several If statements were used to implement the proportional control algorithm. If the difference between the actual temperature of the voltage regulator and the desired temperature is zero, the fan speed will remain constant. If the difference between the actual temperature of the voltage regulator and the desired temperature is less than zero, the fan speed will the fan speed will slow down and eventually stop, and if the difference is greater than zero the fan speed will increase until the difference again reaches zero.

4.2 Design Requirement 2

Power Resistor: The purpose of the power resistor is to withstand and dissipate large amounts of power. In the case of the 50W 8Ω used in the circuit, it is connected to the output of the voltage regulator and grounded as seen in Fig.5. By having the resistance of the power resistor small, 8Ω, a large amount of current is drawn from the regulator, up to 1 amp. This large amount of current is what produces and emulates the heat of an applicable system.

4.3 Design Requirement 3

Proportional Control System:

The heart of the system is the proportion control subsystem. This subsystem is responsible for delivering the proper PWM signal to the low side switch and subsequently the cooling system. In other words, the P-control's task is to keep the error of the system, difference in target temperature and true temperature, by correcting unplanned disturbances. This concept is generally referred to as negative feedack. The P-control loop calculates this error every sample period and is multiplied by a controller gain, or proportion constant. This constant can be fine tuned during testing depending on the type of fan being used as some are more powerful than others. Typically, the product of the error and constant are biased by some null value but in this project we didn't utilize the controller bias null value. However, since there was a significant deadzone for the fan used it would be a next step in the optimization process of this project if it were to be further developed.

5 Getting Started/How to use the device

To properly use this device, the circuit in Fig.5 must be constructed in a manner in which the regulator and the 8Ω are somewhat isolated because they will both emit heat. Also, the thermistor in the diagram is shown at a distance away from the regulator for diagram simplicity, however, it should be placed close enough to contact the heat sink of the regulator without shorting the the circuit. The transmission lines with an arrow head are data transmission lines that the microcrontroller is using for the control system. To read the temperature values and send a target temperature to RealTerm or some other serial communication firmware, it must be properly set up. Further details of the RealTerm setup can be read in section 6. Furthermore, all devices used including the 12V supply require a common ground reference in order for all subsystems to have a similar reference voltage.

6 Getting Started Software/Firmware

To set up Realterm properly so that the design in section 1.3 is compatible a couple of steps need to be taken. First the baud rate needs to be set to 9600. The communication port under device manager indicates the port that the MSP430 is instantiated to and this needs to be set to the corresponding port on RealTerm. Also, the display must be set to unit8, or unsigned integers of 8 bits, and half duplex checked off to see the sent target temperature by the user. After the program is flashed to the board, the circuit is fully and properly built, and the program is running RealTerm should automatically be receiving the temperature values of the system. Furthermore, under the send tab the user can enter a temperature value in the set point range specified in section 2 and the system will automatically adjust to the new desired temperature. In

Fig.9 below is an example of how a serial communication software setup might look like.

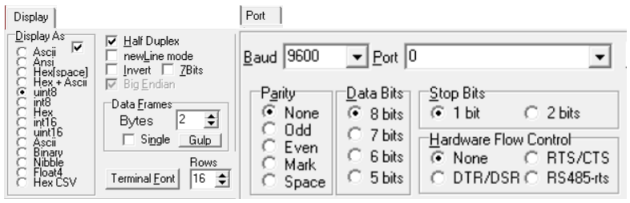

Figure 9: RealTerm setup of display and port menus where Port 0 is the port that the board is connected to in the user's computer.

6.1 Hierarchy Chart

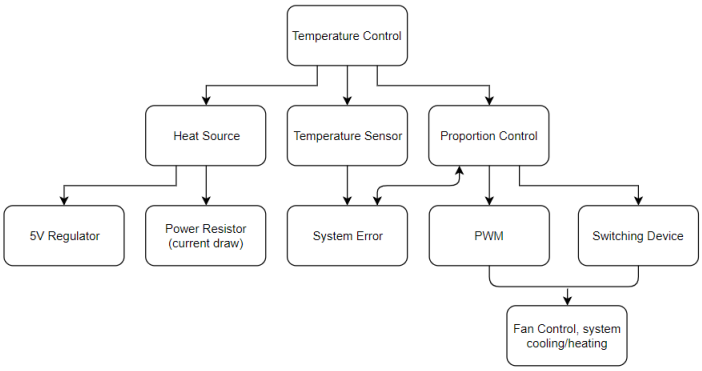

Figure 10: Hierarchy chart of the closed loop system.

6.2 Test Data

Below in Fig.11 is a sample of the test data obtained from RealTerm.

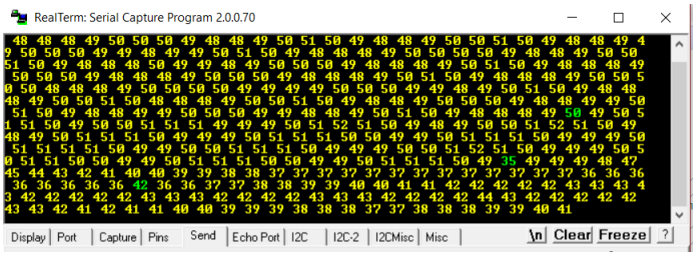

Figure 11: RealTerm test data

The yellow integer values are the temperature values of the system at the time of the sampling. The green integers are the new target temperature values sent from RealTerm to the device. It can be observed in the figure how the values recorded after a new temperature is sent that the system begins to move towards the set point and maintains this temperature within 3°C. In Fig.12 is the oscillating voltage output of the voltage divider circuit that contained the thermistor.

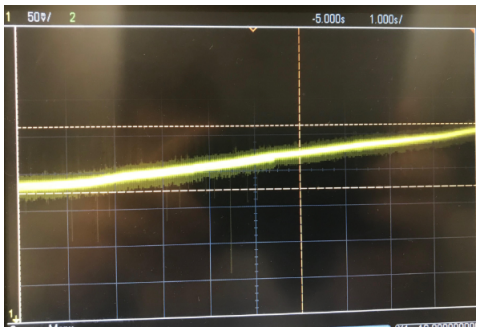

Figure 12: Rolling waveform of voltage divider output.

The wave is oscillating because the resistance of the thermistor is changing due to the imperfect cooling control system. Since the control algorithm is only Proportional, does not include integral or derivative control, it is not as accurate as it could be causing some oscillation about the set point. Also, due to the dead zone of the fan the control loses even more precision. When the system should be cooling down, hence the fan should be spinning slowly, it actually isn't performing correctly since the fan will only begin to spin at a minimum pulse width. Although the circuitry and control systems aren't perfectly accurate and precise they are with a the qualified range of 3°C, or 215mV range. The scope reads an oscillation of 90mV which is well within the equivalent voltage range and determines that a successful control system was implemented.

6.3 Bill of Materials

- 1 - 10kΩ 1/4W resistor
- 1 - 8Ω 50W resistor
- 1 - TIP31C power transistor
- 1 - NTCLE100E3 Thermistor
- 1 - MSP430F5529 Launchpad kit
- 1 - 1.3kΩ 1/4W resistor
- 1 - 12V DC fan
- 1 - 5V Voltage Regulator

7 Appendix

7.1 Appendices A

Below is a quick link to the datasheet of the thermistor used in this project.
http://www.vishay.com/docs/29049/ntcle100.pdf

7.2 Appendices B

Below is a quick link to the datasheet of the switching transistor used in this project.
https://www.st.com/resource/en/datasheet/tip31c.pdf

7.3 Appendices C

Below is a quick link to the users guide for the MSP430F5529 launchpad.
http://dev.ti.com/tirex/#/DevTool/MSP-EXP430F5529/?link=Device20Documentation2FMSP430
6XX2FMSP430F55292FUsers20Guide