

Assignment #2

Heat/Cool Temperature Control

Proportional Control- Spring 2024

Dr. Raed Alqadi

In this assignment, you will design and Control a Cooler in order to Control the temperature of a house by using Pulse Width Modulation and Proportional Control as Follows. We also Use a heater but we will NOT control the Temperature in this case. We will use Compare mode to generate PWM for the Heater. We will Implement 4 modes: *OFF, Cool, Heat* and *Auto-Cool*

Analog Input 0 (AI0): It is the (Pot. P1), It will be used to set the **Set Point**. It is a Potentiometer of 10K Ohm. Scale the value read to 0 – 100C. This will be used in Auto-Cool Mode. It should be displayed on Line 2(see Pages, 4,5,6) We will Refer to it as **SP (Set Point Temperature)**.

Analog Input 1 (AI1): It is the (Pot. P2), It will be used to set the **Heat Percentage in Heat Mode and Cool Percentage in Cool mode**. It is a Potentiometer of 10K Ohm. We will use it as Raw value (0 – 1023) to set the Cooling in Cool mode and will use it to set the Compare Value in Heating Mode

Display it as a percentage on Line 3 to reflect the amount of Cooling in Cool mode or Heat Mode. Read it always and display it on Line 3 regardless of the Mode.

Scale it as **HC = [(AI1 Voltage Value)/5.0]*100.0%** (see Pages, 4,5,6)

Analog Input 2 (AI2): This is already connected to a simulator that reads a voltage already connected to a sensor. Read the Voltage (0 –5) then scale it to temperature by just multiplying it by 100 since an increase of 10ms represent an increase by 1C. The minimum Value that it reads is around 27 degrees without doing extra changes. We will leave it as is to simplify things and we will control Temperatures above the minimum Value. The Value of this sensor is the actual temperature of the room.

We shall Refer to it as **T (Temperature) or RT (Room Temperature)**.

Heater: This is already connected to **RC5**. We will Control the RC5(Heater) by using the Compare Mode to generate a Pulse width Modulated Signal by using Time3 Interrupt and the CCP2IE interrupt.

Cooler (Fan): This is already connected to **RC2** which is the CCP1. We will use actual Pulse width Modulation to control the amount of Cooling.

Hysteresis (HS): This is a Value that we should set by INT2 at RB2 The HS should between **0 – 4** (use Enumeration). INT2 increments HS until it reaches 4, after that rollover to 0 if RB2 is pressed again. Hysteresis has meaning only when we are Controlling the temperature to reach the Set-point which we will o in the Auto Cool Mode.

Operation Modes: This is a Value that we should set Digitally through Interrupt INT0 at RB0. There shall be **4 Modes: OFF mode, Cool mode, Heat mode, Auto Cool mode**. We will use **INT0 (RB0)** will circulate through the Modes (You can use 4 values e.g. (0 – 3) (use Enumeration) and circulate through them. Rollover when the value exceeds 3.

The details of the modes are explained below:

1. **Mode 0:** OFF mode. The system shall start in this mode on Power up. The System Shall be turned OFF Heater OFF and Cooler OFF(0%) if the user Presses RB3 which shall be read in the main Loop. This is Mode 0 below. Turn the CCPIE Interrupt off in this mode
2. **Mode 1 (Cool):** In this mode, use PWM2 module(CCP1) to set the Cooling amount.
PWM raw-value = **AI1-raw_Value**. Use **Prescaler of 16 for Timer 2**
3. **Mode 2 (Heat):** In this mode, set the Heating amount by using the Compare mode and its Interrupt and Timer 3 Interrupt to generate PWM . We are not using actual PWM because the Simulator has the heater on RC5. Also we cannot use directly compare because the simulator does not implement the hardware Compare.

In the Initialization code do:

Initialize the CCP2 mode in Compare mode 9 (CCP2CON = 9).Use Timer 3 for CCP2 only, Prescaler of 4. Enable Timer 3 Interrupt.

In the main code do:

In this mode Turn the Cooler Off and Enable the CCPIE2 Interrupt.

Set the **Compare-Value = AI1_RAW_Value * 64**. We multiplied the value by 64 (just shift left by 6 bits) to make the compare value represent the percentage, since the period will correspond to $2^{16} = 65536$ when Timer 3 overflow, so we scale the **AI1_raw_Value** to 16 bit and hence the multiplication by 64

The work will actually be in the Interrupt as explained in the following Figure:

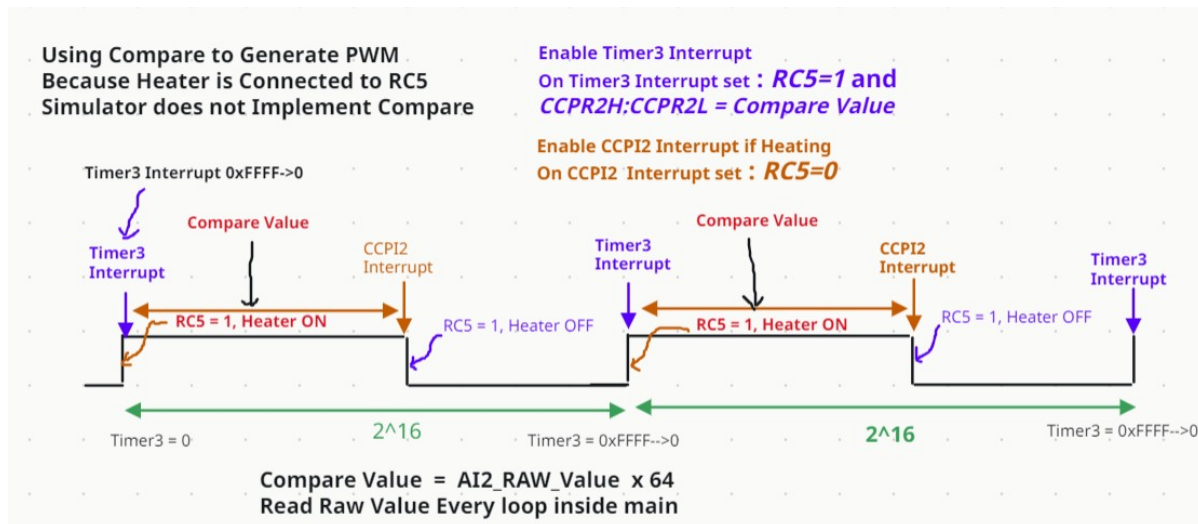


Figure 1

4. **Mode 3 (Auto Cool):**

In this mode, the amount of Heating or Cooling will be proportional to the difference between the Actual Temperature and the Set Point as explained below.

In this Mode, we will control cooler such that the Temperature will be automatically Controlled to become Almost Equal to the Set Point. In other words, we bring it to the Set Point to within a Hysteresis(H) Value (A small value, we will restrict to a Max of 4 as explained in Page 1 it can be from 0 --4). This Means that the Difference between the actual Temperature and the Setpoint should be very small (No more than the H Value). We should Not Heat and Cool at the Same Time.

Auto Cool: The Temperature(T) should be controlled such that the Actual Temperature should be around the Set Point referred to as **SP (SetPoint)**. The **Cooler** should be controlled by the Pulse Width Modulation signal CCP1 at RC2. The Value of the PWM percentage should be proportional to the **CoolError = T – SP** if T > SP and should be turned Off if T < (SP-HS). The percentage of Cooling (PWM percentage value) should be set as follows.

```
CoolError = T – SP
If( CoolError > 0){
    PWM percentage value = CError*100/10;
    Set the CCP1 PWM to PWM percentage value
    If this value is less than 25% set it to 25%
    heater should be off here
}
If( T < (SP -HS) )
    PWM percentage value =0 ;// Cooler OFF (turn heater here, see below)
```

In real life, the **Heater** should be **always off** but **here** we are **forced** to make it **(50% see note below)** when T fall below (SP – HS), that is (if T < (SP -HS)) **because we cannot increase the temperatureh. If we do not do that. in real life, in summer, the temperature rises when we turn the cooler off, we cannot do that here.**

Note: So we will need to turn the heater ON as we did on mode 2 (heat) by the same Compare method explained above. It is up to you, to force the heater to 50% or simply turn AI2 to almost 50% . Turn the heater on by the Code by enabling the Compare interrupt but only when t

Notes:

1. The temperature should be automatically updated as well as HS and the Mode if changed.
2. See the figures on this page and following pages to see what and where to display.
3. The main program should have an infinite loop and there should be noticeable flicker in the screen. It should be continuously reading the Analog Inputs and displaying them. You can put a small delay if needed.
4. The Interrupt code should be minimal, just change the Variables) and do actual displaying in the Main Program. Do not read any analog values or display inside the Interrupts.
5. In the auto mode use set point between 35 and 46 for best performance because of the special cases. So not worry about special cases of the heater.

There are special cases for the Heater, since this is a simulator. You cannot make the Temperature more than 76 and cannot be less than 27.

6. Make sure when you enable your Interrupts to Enable the Most Two significant bits bit 7 and 6 in INTCON. (GIEH, and GIEL) Otherwise your Timer 3 interrupt will not work correctly. Disable the Priority.
7. Work in Groups of 2 and start as soon as possible.

