# Assignment #2

**Heat/Cool Temperature Control**

**Summer 2023**

**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 as described below. We also Use a heater but we will NOT control the Temperature in this case

**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.

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 PWM value ( 0 – 100) to set the cool amount to be proportional to the Duty cycle ( exactly as we did in the PWM example). This will only be used in Mode 1 ( cool) described below.

**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 turn Control the RC5 by generating a Pulse width Modulated Signal. We will not use an actual PWM signal but we by use Time3 Interrupt to generate a Pulse Width Modulation

**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 Digitally through Interrupt INT1 at RB1 . The HS should value in the range 0 – 3. INT1 will increment HS until it reaches 3 and will rollover back to zero . In other words, INT1 circulates the value of HS in the range 0 – 3.

**Cooling Amount:** The amount of cooling depends on the mode as described in detail below. Here is the main points:

- In the Simple Cool Mode (Mode 1): PWM is used such that the amount of cooling is proportional to the raw value of Analog Input 1

- In the Auto Cool mode (Mode 3): PWM is used such that the amount of Cooling will be proportional to the difference between the Actual Temperature and the Set Point as explained in the Modes below.

**Operation Modes:**

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

Mode 0: *OFF*, Mode 1: *Cool*, Mode 2: *Heat*, Mode 3: *Auto Cool*

The details of the modes are explained below.

1. Mode 0: OFF. The system shall start in this mode on Power up. The System Shall be turned OFF Heater OFF and Cooler OF(0%)
2. Mode 1: Cool. The cooler will be on and the amount of cooling should be proportional to the Analog Value of AI1 ( 0 – 5 volt will correspond t0 0–100%). *Exactly as we did* in the PWM example. The Heater must be turned off in this mode
3. Mode 2:(Heat): In this mode, simply turn ON the heater. The Cooler should be turned Off (PWM % = 0) .
4. Mode 3: Auto Cool:

   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 – 3). This Means that the Difference between the Actual Temperature and the Setpoint should be very small (No more than the H Value)..
   Therefore, the Temperature should be controlled such that the Actual Temperature should be around the Set Point referred to ad ***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.

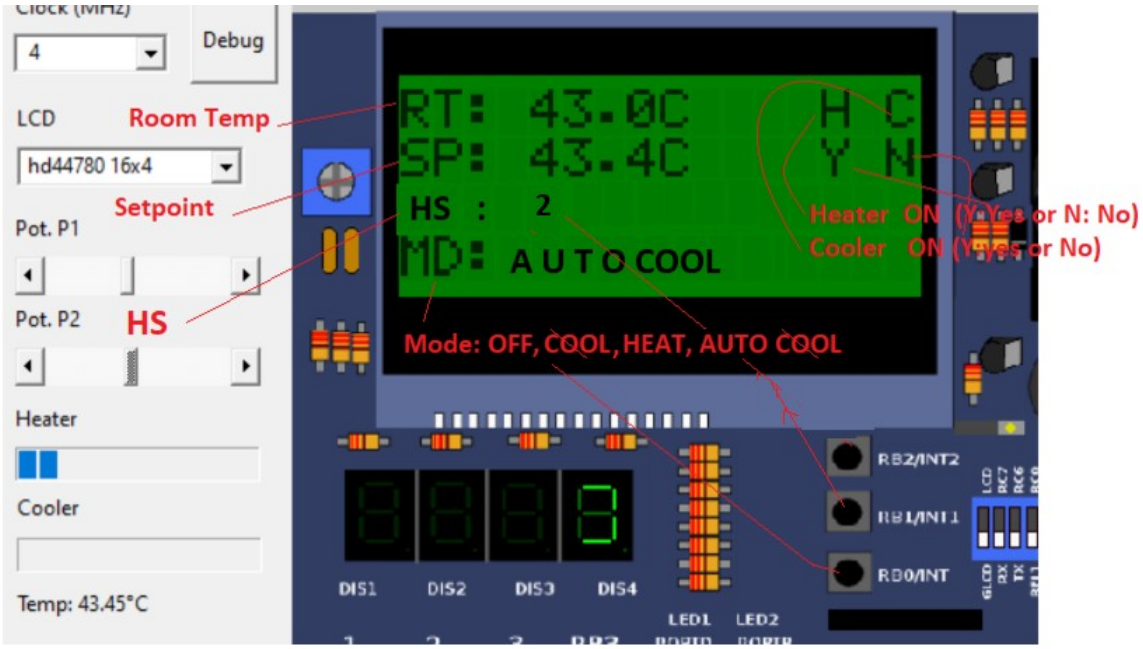
```
CoolError = T – SP
 If( CoolError > 0)
      PWM percentage value = CError*100/10;
      Set the CCP1 PWM to PWM percentage value
      If this value is less than 25% set it to 25%
  If( T < (SP -HS)  ){
      PWM percentage value =0 ;// Cooler OFF
      // Turn the heater here ON with 50% as described below
  }
```

**Why Do we turn the heater ON during the Auto Mode:**

In real life, the ***Heater*** should be always off but ***here*** we are ***forced*** to make it ***ON half the time in Mode 3( the Auto Cool mode).***

***Toggle the heater ON and Off*** **when T fall below (SP – HS), that is ( T < (SP -HS) )** ***because we cannot increase the temperature 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.***

***To set the Heater to 50% ON (half the time) No pulse modulation here. You can use Timer 3 to create an Interrupt say every 200 ms. Use the Interrupt toggle the Heater between ON and OFF on. Make sure that you do that Only if this condition explained in the previous paragraph occurs. So you can set a Variable if we are cooling and ( T < (SP -HS) ) and make the Interrupt toggle in this case.***




Notes:

1. The temperature should be automatically updated as well as HS and the Mode if changed.
2. The main program should 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.

3. The Interrupt code should be minimal, just change the Variables) and do actual displaying in the Main Program.
4. Keep your set points less than 50 in the Auto Cool for best performance. 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.
5. 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.
6. Work in Groups of 2 and start as soon as possible.