

Programmable Thermostat

Let's develop a simple application, a programmable thermostat. The front panel will be used as a user interface. Let's define some design rules.

1. Four modes of operation for the front panel
 - MODE 1: Display current temperature.
 - MODE 2: Display current time.
 - MODE 3: Set temperature range of operation (low, high).
 - MODE 4: Set time range of operation (start, end).
2. Heat pump will turn ON, if the current time is within the range of operation and if the current temperature is within the temperature range.
3. MODE button will scroll through the different modes.
4. ENTER button will select between high and low temperatures and start and end times.
5. PB4, PB3, PB2 and PB1 will change the values of time and temperature only when in set mode.

Figure 1 shows the hardware schematic for the programmable thermostat. A 10K Ω thermistor will be used to measure temperature. The external heat pump requires a 24 VDC input, which will be triggered by output Y8 of the IP3416.

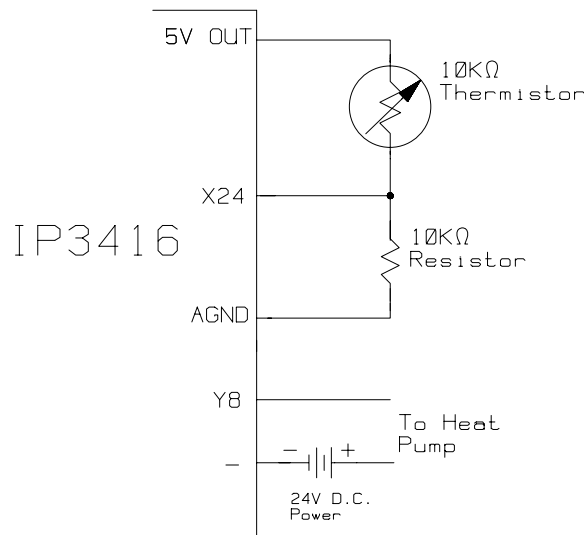


Figure 1. Hardware design of programmable thermostat

The first thing to develop will be the operator interface. The front panel of the IP3416 will be used for this purpose. The MODE button will scroll through the four modes of operation. Let's define these and the corresponding internal relays,

MODE	DESCRIPTION	RELAY
1	Display current temperature.	R0
2	Set low & high temperature.	R1
3	Display current time	R2
4	Set start & stop time	R3

On system power-up, the system will default to Mode 1. The following RLL will accomplish this,

```

1 |         |                                     | On system power-up initialize
  | R248   |         R0          | mode 1
3 |---| |-----| (SS)-----| R0 ON, R1...R7 OFF

```

R248 is only active during the first scan of the RLL. This only occurs during initial system power-up or reset, thereby setting system to mode 1 by turning ON relay R0 and turning off relay R1 through R7. When the user presses the MODE button, we want the mode to cycle from 1 through to 4 and back to one. One of the simplest ways of doing this is shown below,

```

2 | X39   R99                | MODE SELECTION
  |   ^| |-----| (SU1)-----| X39 Scrolls through modes
3 | R1    R2    R3          |
  |---|/|---|/|---|/|-----| ( )-----| Make sure one of the 4 modes
                               | is active.

```

Rung 2 does the actual cycling. The pulse contact makes sure the mode only cycles once per push of the mode switch. The Shift-up coil shifts R0 through R7 up by one. So if R0 is ON then R1 will become ON and R0 will become OFF and so on. Rung 3 cycles from mode 4 to 1.

The next step will be to translate the other push buttons to internal relays. We need to do this because, the status of the push buttons will be used throughout the RLL and they could go off anytime, therefore we have to trap them for the whole scan. This is accomplished by the following logic,

```

4 |-----| |-----| |-----| |-----| |-----| |-----|
  |                                               | LOGIC TO SET TEMPERATURE
5 | X34   R94                | R34   | Enter pressed so debounce by
  |   ^| |-----| ( )-----| using R34
6 | X35   R95                | R35   | PB4 pressed so debounce by
  |   ^| |-----| ( )-----| using R35
7 | X36   R96                | R36   | PB3 pressed so debounce by
  |   ^| |-----| ( )-----| using R36
8 | X37   R97                | R37   | PB2 pressed so debounce by
  |   ^| |-----| ( )-----| using R37
9 | X38   R98                | R38   | PB1 pressed so debounce by
  |   ^| |-----| ( )-----| using R38

```

We need a method for toggling between set low and set high temperature. The following rungs will perform this,

```

10 | R1    R34      R8      R10
   |---| |---| |---|+-----|/|-----| ( )-----| We are in MODE 2, SET TEMP.
   |                               | and ENTER was pressed, so toggle
11 |                               | R10   R8
   |                               |-----| |-----| (S)-----| between set high and low
   |                               |                               | temperature.
   |                               | R10   R8
12 |                               |-----|/|-----| (R)-----|

```

R8 will be used to indicate if the temperature that is being set is high or low. We have finished the necessary mapping of the external push buttons into internal relays. Now let's put them to use. The following rungs handle the setting for a new high or low temperature.

```

13 | R1   R35       R8       C5       |Change low temp LS digit
   | -| |---| |---+-----|/|-----[C5+1]----|
   |         |         |         |         |
14 |         |         | R8       C6       |Change high temp LS digit
   |         |         |-----+-----|/|-----[C6+1]----|
   |         |         |         |         |
15 | R1   R36       R8       C5       |Change low temp tenths digit
   | -| |---| |---+-----|/|-----[C5+10]---|
   |         |         |         |         |
16 |         |         | R8       C6       |Change high temp tenths digit
   |         |         |-----+-----|/|-----[C6+10]---|
   |         |         |         |         |
17 | R1   R37       R8       C5       |Change low temp MS digit
   | -| |---| |---+-----|/|-----[C5+100]--|
   |         |         |         |         |
18 |         |         | R8       C6       |Change high temp MS digit
   |         |         |-----+-----|/|-----[C6+100]--|

```

In the above code if the temperature is set to 950 degrees and the user presses PB2, then the temperature will increase to 1050 degrees. We do not want temperatures greater than 999 to be entered. Therefore, we must mask this. The best way of doing this is by the following code,

```

19 |-----+-----|/|-----[C5/1000]-|Clear any thousands digit
   |         |         |         |         |of low temperature.
20 |-----+-----|/|-----[D1]-----|
   |         |         |         |         |
21 |-----+-----|/|-----[C6/1000]-|Clear any thousands digit
   |         |         |         |         |of high temperature.
22 |-----+-----|/|-----[D1]-----|

```

When C5 is divided by 1000, the remainder is placed in D1, this is the result we want. I.e. if C5 is 1050, divided it by 1000, you get 1 with a remainder of 50. The remainder is what we want, so place that back into C5.

The same functionality could be done to change the time and the following code demonstrates this,

```

23 | R3   R34       R9       R10      |LOGIC TO SET TIME
   | -| |---| |---+-----|/|----- ( )-----|
   |         |         |         |         |
24 |         |         | R10      R9       |END time set mode
   |         |         |-----+-----|/|----- (S)-----|
   |         |         |         |         |
25 |         |         | R10      R9       |START time set mode
   |         |         |-----+-----|/|----- (R)-----|
   |         |         |         |         |
26 | R3   R35       R9       C13      |Change start time minutes
   | -| |---| |---+-----|/|-----[C13+1]----|
   |         |         |         |         |
27 |         |         | R9       C14      |Change end time minutes
   |         |         |-----+-----|/|-----[C14+1]----|
   |         |         |         |         |
28 | R3   R36       R9       C13      |Change start time minutes
   | -| |---| |---+-----|/|-----[C13+10]--|
   |         |         |         |         |
29 |         |         | R9       C14      |Change end time minutes
   |         |         |-----+-----|/|-----[C14+10]--|
   |         |         |         |         |
   |         |         |         | C0       |Make sure minutes is not

```

```

30|-----[C13/60]---|greater than or equal to 60
|
|          C13
31|-----[D1]-----|
|          C0
32|-----[C14/60]---|
|          C14
33|-----[D1]-----|
|
|  R3      R37      R9      C23      Change start time hours
34|-| |-----| |-----|/|-----[C23+1]---|
|
|          R9      C24      Change end time hours
35|          \-----| |-----[C24+1]---|
|
|  R3      R38      R9      C23      Change start time hours
36|-| |-----| |-----|/|-----[C23+10]--|
|
|          R9      C24      Change end time hours
37|          \-----| |-----[C24+10]--|
|
|          C0      Make sure not > 24
38|-----[C23/24]---|
|
|          C23
39|-----[D1]-----|
|
|          C0      Make sure not > 24
40|-----[C24/24]---|
|
|          C24
41|-----[D1]-----|

```

Now we need a way of informing the user what mode we are in. We will use the three mode LED's of the IP3416 (I/O, time and mode).

```

|          I/O L
|  R0          R206      TEMPERATURE indicator light
42|-| |-----|-----| ( )-----|
|
|  R1
43|-| |-----|
|
|  I/O L      TIME
|  R206      R204      TIME indicator light
44|-|/|-----|-----| ( )-----|
|
|  R1          PROG
|          R203      SET indicator light
45|-| |-----|-----| ( )-----|
|
|  R3
46|-| |-----|
|
|  TIME      COLON
|  R204      R205      TIME mode, so light-up colon
47|-| |-----|-----| ( )-----|

```

This completes the user interface. The next thing that needs to be done is to measure and calculate the current temperature.

```

|          CALCULATE CURRENT TEMP
|          C0
48|-----[A24]-----|calculate current temperature
|
|          C31      by using a spline approximation
|          of the curve generated by the

```

```

49|-----[1361-C0/7] thermistor
|
|          C30      | f(x)=(9939-(1361-x)/7*x/14)/61
50|-----[C0/14]----|          *x/10 - 38
|
|          C29      | where, x = A24
51|-----[C31*C30]-|
|
|          C28      |
52|-----[9939-C29/61]--|
|
|          C8        |
53|-----[C0/10*C28/73-38]----| C8 is in degrees Celsius
|Overf
| R31          C8    | If negative, light up upper
54|-| |-----[20000-C8]| green LED to indicate negative

```

We have the current temperature. Now the logic to control when the heat pump is turned on has to be generated. The heat pump should be ON when the current time is between start and end time and the current temperature is less then low temperature. The heat pump should turn off when we are not within the time range and when the current temperature is greater than the high temperature setting. The following RLL will do this,

```

|
|          C7        | Calculate the current time
55|-----[D35*100+D34]--| HHMM
|
|          C3        | Calculate start time
56|-----[C23*100+C13]--| HHMM
|
|          C4        | Calculate end time
57|-----[C24*100+C14]--| HHMM
|
|          C0        | Start time <= current time?
58|-----[C7-C3]----|
|Overf
| R31          C0    | Start time > end time?
59|-|/|+-----[C7-C4]----|
|
|          |Overf
|          | R31      | R17 means C3 <= C7 < C4
60|          `--| |----- ( )-----|
|
|          C0        |
61|-----[C8-C6]----| Turn on heater if current time
|          Overf    | is between start and end times
| R17   R31      R11  | and if current temperature is
62|-| |---| |----- (S)-----| between low and high temperature.
|
|          C31      |
63|-----[20000-C8]|
|Overf
| R31          C31  |
64|-| |-----[0]-----|
|
|
65|-----|
|
|          C0        | Stop the heater when temp. is
66|-----[C8-5-C6]-| 5 degrees more than the high
|          Overf    | temperature
| R17   R31      R11  |
67|-| |---|/|----- (R)-----|
|
| R11          Y8    |
68|-| |----- ( )-----|

```

The final step in our program is to display the time or temperature depending on the mode we are in. The following does this,

```

| R0
69|-|-----[C8]-----|Display current temperature
| R2
70|-|-----[C7]-----|Display the current time
| R3          R9
71|-|-----+|/|-----[C3+10000]|Display low time
|          | R9
72|          `|-----[C4+20000]|Display high time
| R1          R8
73|-|-----+|/|-----[C5+10000]|Display low cutoff
|          | R8
74|          `|-----[C6+20000]|Display high cutoff

```

The final step of this design is to design our own label for the programmable thermostat as shown in Figure 2. International Parallel will provide private labels at a nominal charge.

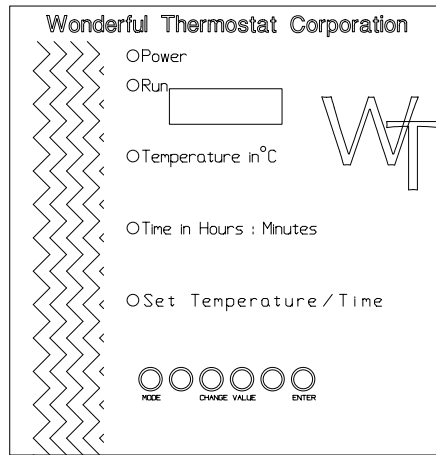


Figure 2. Private label for thermostat