This article offers an introduction to programming PIC chips using the free version of Flowcode (limited to 2k of compiled code), and shows you how to use it to create two simple programs for the PICkit 2 44-pin Demo Board. To start with we will examine a very simple LED flashing program so you can get to grips with the basic operation of Flowcode and your PICkit 2, after that we will look at a more complex program using a thermistor.

First of all – the flowchart

Program design usually starts with a flowchart, planning the sequence of steps which the microprocessor will follow. With Flowcode, it is a very simple matter to convert the flowchart into a Flowcode program.

This first example develops a system which outputs one of two lighting patterns, depending on whether or not a switch is pressed. You can see this in figure 1:

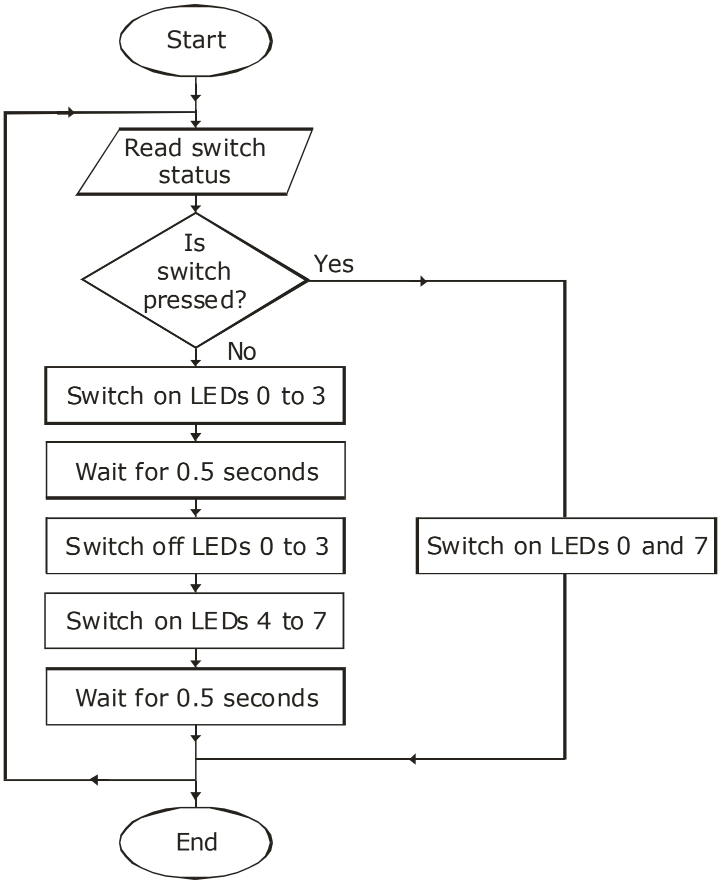


Figure 1

Flowcode 1 – step-by-step

Let’s now convert this into a Flowcode program.

A free demo version of Flowcode is available from:
http://www.matrixmultimedia.com/temp/FlowcodeV3.exe

When you install it, make sure that the “Microchip PICkit2” option is selected.

- Run the Flowcode application.
- Click on the ‘OK’ button when the Reminder screen opens.
- Next you have the option of creating a new Flowcode flowchart, or opening an existing one. Select the option to create a new one.
- You now have to choose a target PIC chip. The 44-pin Demo Board is fitted with a PIC 16F887, so choose that from the list.
- A new workspace called ‘Main’ opens.
- Click and drag icons from the Icons Toolbar running down the left-hand edge of the screen, to make the Flowcode flowchart shown below. For the moment, do not worry about configuring them – we will do that next.

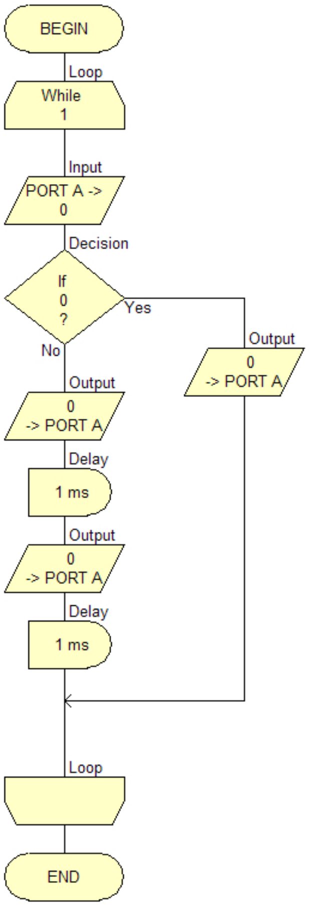
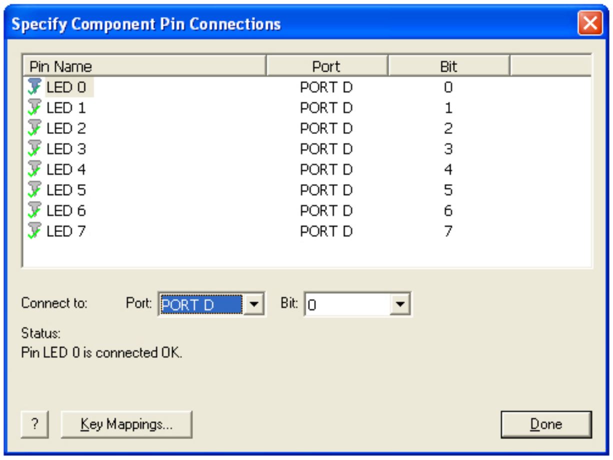


Figure 2

- Now add the hardware. Click on the LED icon in the Components Toolbar, running just inside the Icon Toolbar. Also click on the switch icon. You now have two more items on the workspace.

- Select the LED component in the workspace, c lick on the drop down menu button and select the 'Component Connections' option. In the 'Connect to Port' drop down menu, select Port D, (because the LEDs on the 44-pin Demo Board are connected to Port D.) The Connections dialogue box should now look like the following:



Specify Component Pin Connections

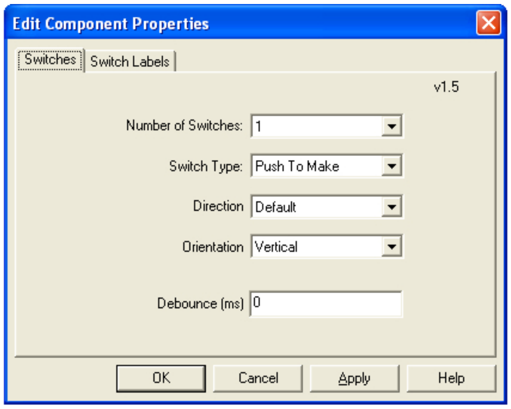
Pin Name	Port	Bit
LED 0	PORT D	0
LED 1	PORT D	1
LED 2	PORT D	2
LED 3	PORT D	3
LED 4	PORT D	4
LED 5	PORT D	5
LED 6	PORT D	6
LED 7	PORT D	7

Connect to: Port: PORT D Bit: 0

Status:
Pin LED 0 is connected OK.

Figure 3

- Next select the Switch component in the workspace. Click on the drop down menu button, select the 'Properties' option, and set the number of switches to one.



Edit Component Properties

Switches | Switch Labels

v1.5

Number of Switches: 1

Switch Type: Push To Make

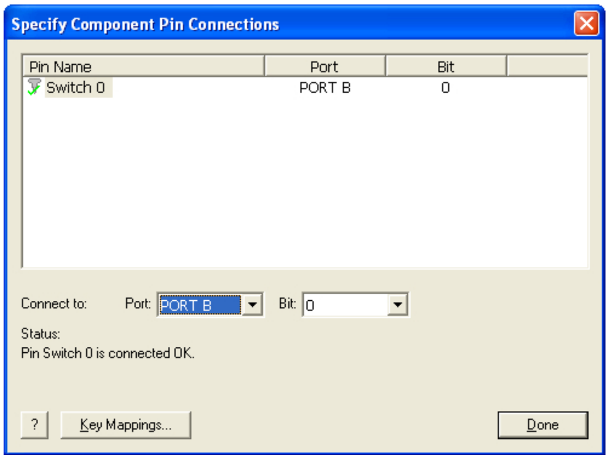
Direction: Default

Orientation: Vertical

Debounce (ms): 0

Figure 4

In the ‘Connect to Port’ drop down menu, select Port B, (because the push switch on the 44-pin Demo Board is wired to bit 0 of Port B.) The Connections dialogue box should now resemble the following:



Specify Component Pin Connections

Pin Name	Port	Bit
Switch 0	PORT B	0

Connect to: Port: PORT B Bit: 0

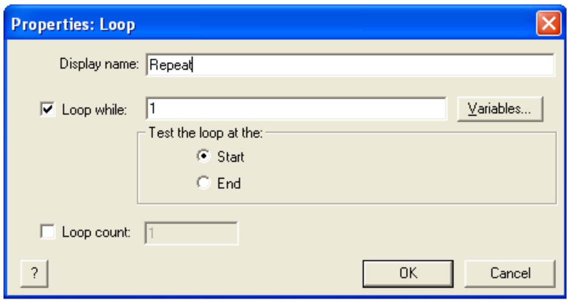
Status:
Pin Switch 0 is connected OK.

Figure 5

Configuring the icons

Now it is time to configure the icons that make up the Flowcode flowchart. To do this, double-click on each icon in turn and set up the configurations shown in the following diagrams. In the ‘Display name’ boxes, type the text from the corresponding element of the flowchart. This makes it easier to check your progress.

The Loop icon:



Properties: Loop

Display name: Repeat

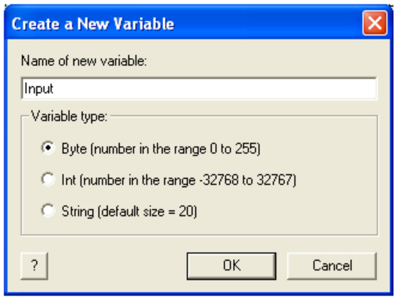
☑ Loop while: 1

Test the loop at the:
● Start
○ End

□ Loop count: 1

Figure 6

The Input icon:
First of all you need to create a variable to carry the switch information. Click on the ‘Variables…’ button to open the ‘Variable Manager’ window. Click on the ‘Add New Variable’ button, and type ‘Input’ as the name of the new variable, as shown in the next screenshot.



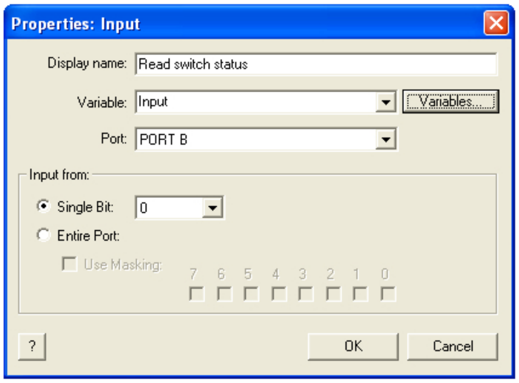
Create a New Variable

Name of new variable:
Input

Variable type:
● Byte (number in the range 0 to 255)
○ Int (number in the range -32768 to 32767)
○ String (default size = 20)

Figure 7

Then click 'OK'. Back in the Variable Manager window, click on 'Use Variable'. Back in the Input Properties window, select Port B, and 'Single Bit', so that the dialogue box now looks like:



Display name: Read switch status

Variable: Input

Port: PORT B

Input from:

● Single Bit: 0

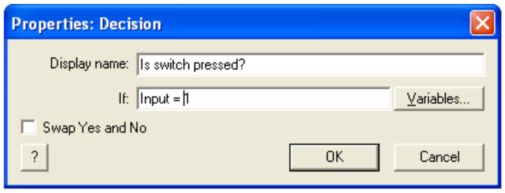
○ Entire Port:

□ Use Masking: 7 □ 6 □ 5 □ 4 □ 3 □ 2 □ 1 □ 0 □

Figure 8

The Decision icon:

In the Properties dialogue box, click on the 'Variables' button and select the 'Input' variable for use. Type 'Is switch pressed' in the 'Display name' box. You should now have:



Display name: Is switch pressed?

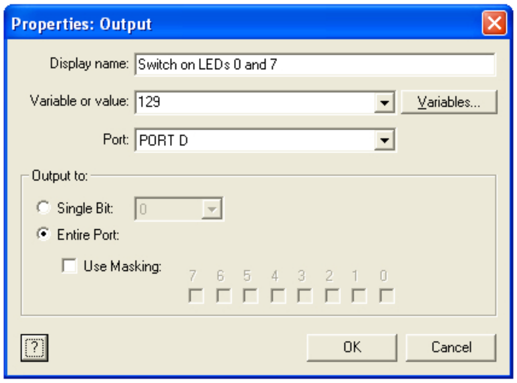
If: Input = 1

□ Swap Yes and No

'**Figure 9**

No' Loop- The Output icon:

LED 7 is the most significant bit of the display, and so has a place value of 128 (= 2^7).
LED 0 has a place value of 1 (= 2^0). To make both light, you output a value of 129 to Port D, where the LEDs are connected.



Display name: Switch on LEDs 0 and 7

Variable or value: 129

Port: PORT D

Output to:

○ Single Bit: 0

● Entire Port:

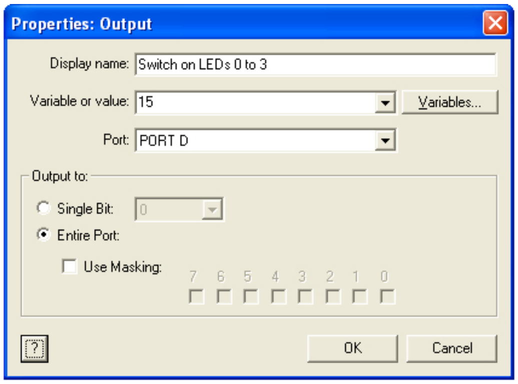
□ Use Masking: 7 □ 6 □ 5 □ 4 □ 3 □ 2 □ 1 □ 0 □

Figure 10

'Yes' Loop:

First Output icon:

The first four LEDs, 0 to 3, have place values of 1, 2, 4 and 8, (= 2^0, 2^1, 2^2 and 2^3 respectively). You can light all four by outputting a value of 15 (= 1 + 2 + 4 + 8) to Port D, as shown in the next screenshot.



Properties: Output

Display name: Switch on LEDs 0 to 3

Variable or value: 15

Port: PORT D

Output to:

○ Single Bit: 0

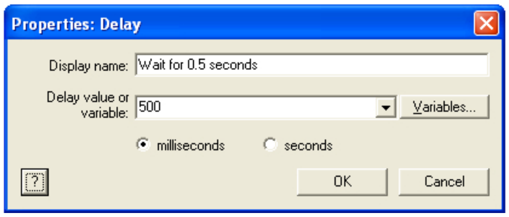
● Entire Port:

□ Use Masking: 7 6 5 4 3 2 1 0 □ □ □ □ □ □ □ □

Figure 11

Both Delay icons:

The duration of the delay can be specified either in milliseconds or seconds. To create a half second display, type 500 as the 'Delay value', with the 'milliseconds' button selected, since 500ms = 0.5s.



Properties: Delay

Display name: Wait for 0.5 seconds

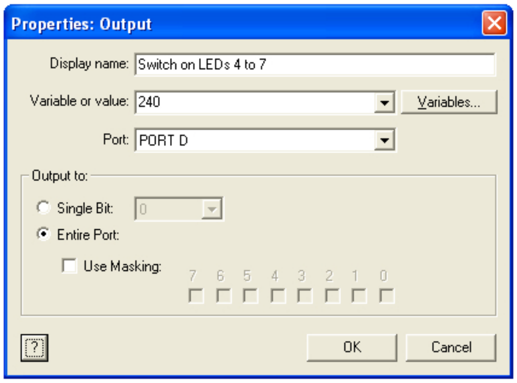
Delay value or variable: 500

● milliseconds ○ seconds

Figure 12

Second Output icon:

The LEDs 4 to 7, have place values of 16, 32, 64 and 128, (= 2^4, 2^5, 2^6 and 2^7 respectively). To light all four, output a value of 240 (= 16 + 32 + 64 + 128) to Port D, as shown in the next screenshot.



Properties: Output

Display name: Switch on LEDs 4 to 7

Variable or value: 240

Port: PORT D

Output to:

○ Single Bit: 0

● Entire Port:

□ Use Masking: 7 6 5 4 3 2 1 0 □ □ □ □ □ □ □ □

Figure 13

Now your Flowcode flowchart should look like:

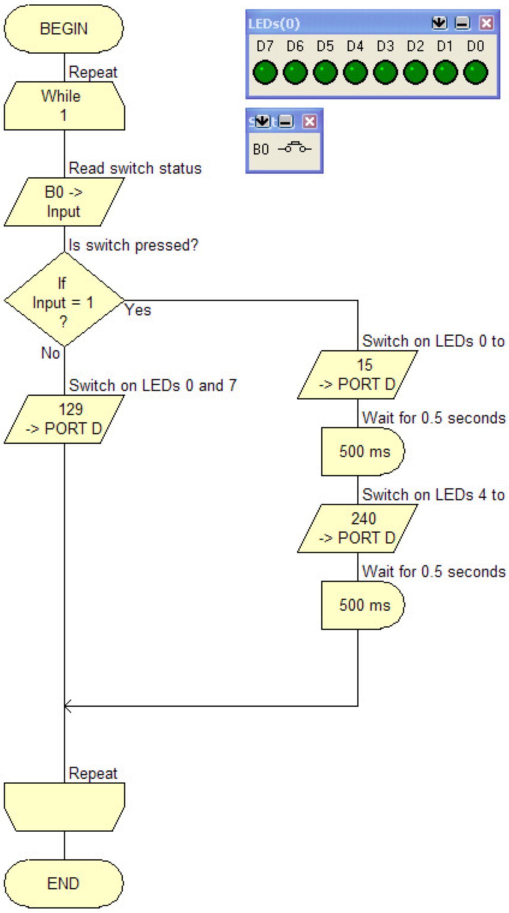


Figure 14

Simulate the program

Flowcode allows you to check whether you program works by running an on-screen simulation. This can be done in two ways, simulating the full program, or stepping through the simulation, icon by icon. In both cases, two windows, the ‘Variables’ window, and the ‘Call Stack’ window, appear while the simulation runs. When you run the program simulation at full speed, the values in these windows are not updated as the program runs. If you slow down the full program simulation, or if you step through the program, then you will see the effect of each stage on the variables you created, shown in the ‘Variables’ window.

For now, look at the full simulation, running at full speed. Click on the ‘Run’ command in the Menu toolbar. Choose the ‘Go/Continue’ option. The ‘Variables’ window, and the ‘Call Stack’ window appear, and the LEDs 0 an 7 light. Click on the push switch icon, and the LEDs should flash on and off, with LEDs 0 to 3 alternating with LEDs 4 to 7. The program works!

(If it does not, then go back and check the configuration details given above. You could simulate the program step-by-step, by clicking on the ‘Step Into’ option in the ‘Run’ menu, and then using the F8 key to move to the next step.)

Programming the PIC chip

The groundwork is done – you have a working Flowcode program. The rest is easy! Here are instructions for getting you program into the chip:

- Connect the PICkit 2 module to your computer using the USB cable.
- Plug the 44-pin Demo Board into PICkit 2.
- Click on the 'Chip' command in the Menu toolbar, and select the 'Compile to Chip...' option. (You may be asked to save the Flowcode program – click on the 'Yes' button.)
- A 'Complier Messages' window opens to show progress, as various elements are launched. The Target and Busy LEDs on PICkit 2 light eventually, when the program is transferred. Finally, the 'Compiler Messages' window shows the word 'FINISHED', at which point you can click on the 'Close' button.
- Your program should now be running on the 44-pin Demo Board. LEDs 0 to 3 and 4 to 7 are alternately on then off. Press the push switch, and the LED pattern should change, with only LEDs 0 and 7 lit. That's what your Flowcode program said!
- Unseen by the user, the Flowcode programmer first compiled the graphical flowchart to a C file, and then compiled that to assembler. These can be viewed by C and assembler buffs either by using the 'View C' and 'View ASM' commands under the 'Chip' command on the Menu toolbar, or by opening the files *prog1name*.c and *prog1name*.asm in the folder where you saved the Flowcode program (where *prog1name* is the name you gave to that program.)

Program 2

OK now you are up and running let's look at a more complex problem and show you how to solve it using Flowcode and PICkit 2. Our goal here is to build a digital temperature display for an incubator. It gives a visual warning if the temperature is too hot (LEDs 4and 5 flash alternately with 6 and 7,) or too cold (LEDs 0 and 1 flash alternately with 2 and 3.) In between these limits it lights either two LEDs (0 and 1) or, if a bit warmer, it lights four LEDs (0,1,2 and 3.)

This is more ambitious, but the process is exactly the same as in the last program:

- Create the flowchart.
- Convert it into a Flowcode flowchart.
- Configure the component and command icons.
- Simulate the program.
- Compile it to the PIC chip and run it.

Flowchart for program 2

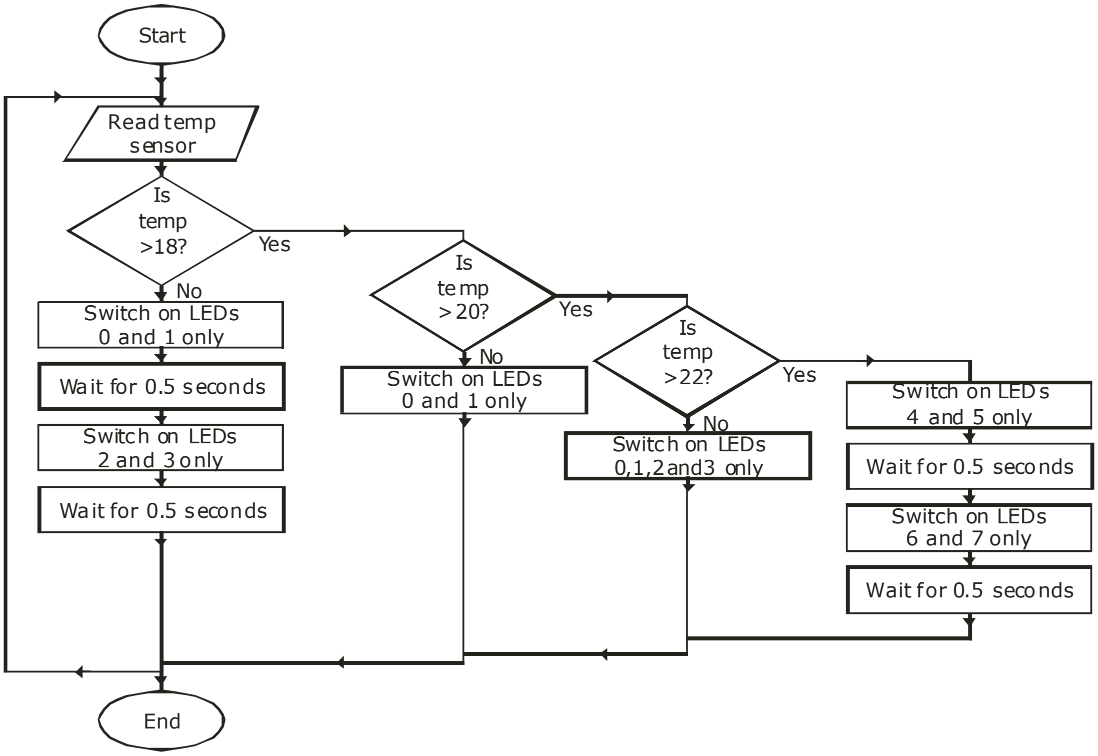

Start
Read temp sensor
Is temp >18?
Yes
No
Switch on LEDs 0 and 1 only
Wait for 0.5 seconds
Switch on LEDs 2 and 3 only
Wait for 0.5 seconds
Is temp >20?
Yes
No
Switch on LEDs 0 and 1 only
Is temp >22?
Yes
No
Switch on LEDs 0,1,2and3 only
Switch on LEDs 4 and 5 only
Wait for 0.5 seconds
Switch on LEDs 6 and 7 only
Wait for 0.5 seconds
End

Figure 15

Flowcode for Program 2

The Flowcode flowchart is shown in the next picture. Most of the icons have already been discussed in the previous program. Open a new workspace in Flowcode and then click and drag icons to create a copy of this flowchart.

There are two new command icons, and one new component. The command icons are both known as Component Macros. You find them in the Icons Toolbar running down the left-hand edge of the screen. (Be careful to choose Component Macros, and not just Macros! Component macros have hatched outer bands on the icon.)

The new component is an ADC, found in the Components toolbar.

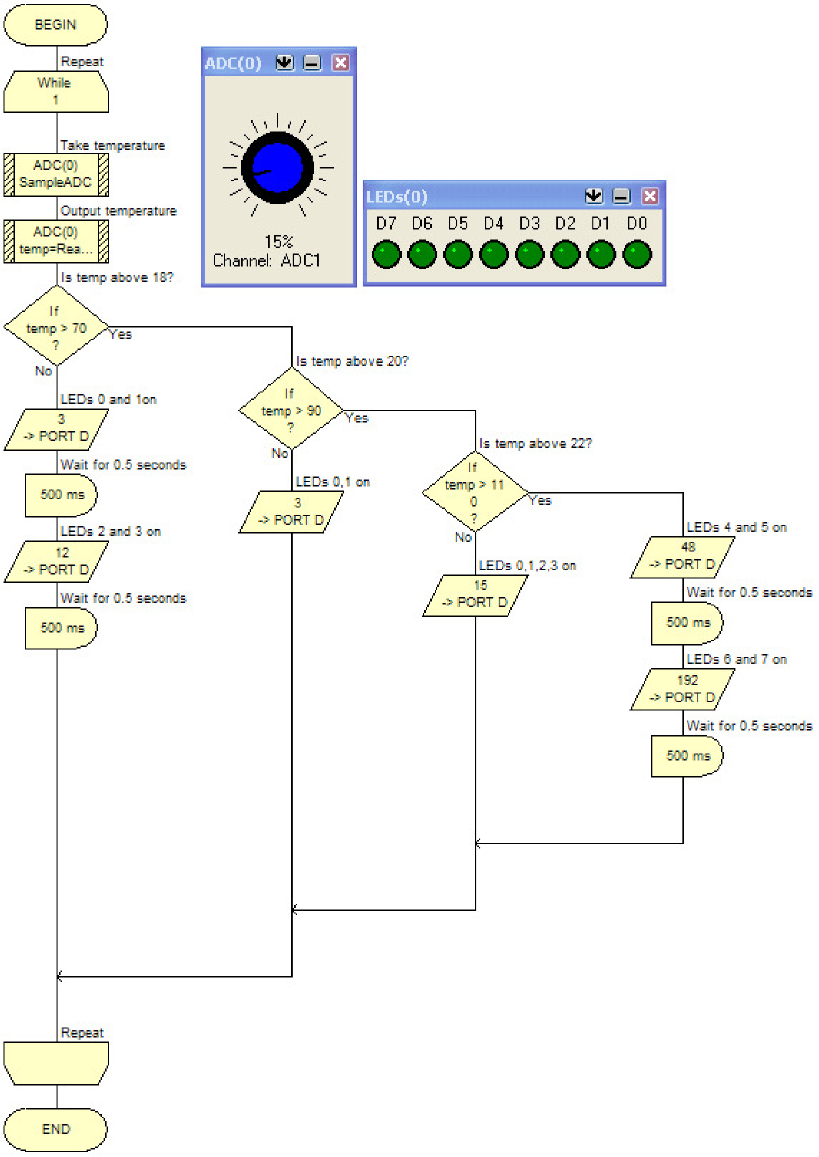
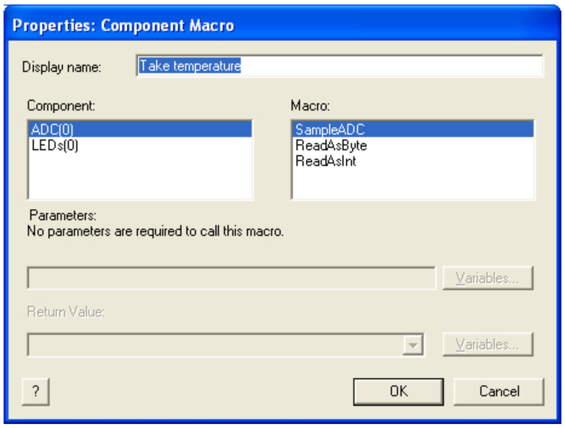


Figure 16

Most of the icons are configured in the same way as in the first program. Use the text descriptions and the text inside the icons as a guide, when configuring them. The values you use in the Decisions boxes may need to be adjusted in the light of experience. They depend on exactly which thermistor (and resistor) you use in your temperature sensor.

Component macros are pre-defined macros supplied with the components and are available for use only with that particular component. Some component macros require parameters, and a return value. Parameters can be numerical values or existing variables. If the component macro has a return value, then you can assign that value to an existing variable for use later in the flowchart.

Double-click on the first component macro, to open its Properties window. You will find that there are two components listed in the 'Component' window, ADC(0) and LEDs(0). Click on the ADC(0) component to select it. The Component Macros available for it are then listed in the 'Macro' window. Select the SampleADC macro. Notice that it requires no parameters and has no return value. This properties window is shown below.


Properties: Component Macro

Display name: Take temperature

Component: ADC(0), LEDs(0)

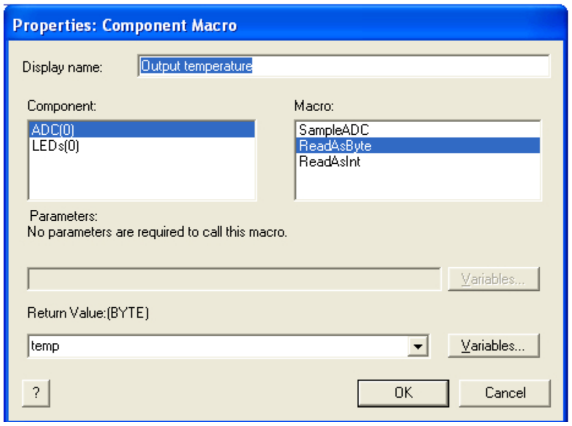
Macro: SampleADC, ReadAsByte, ReadAsInt

Parameters:
No parameters are required to call this macro.

Return Value:

Figure 17

Next, open the properties window of the second Component Macro. Select the ADC(0) component, but this time select the ReadAsByte macro. This has a return value, and you should create and use a variable called 'temp', by clicking on the 'Variables' button, and then on the 'Add New Variable' button. This properties window is shown below.


Properties: Component Macro

Display name: Output temperature

Component: ADC(0), LEDs(0)

Macro: SampleADC, ReadAsByte, ReadAsInt

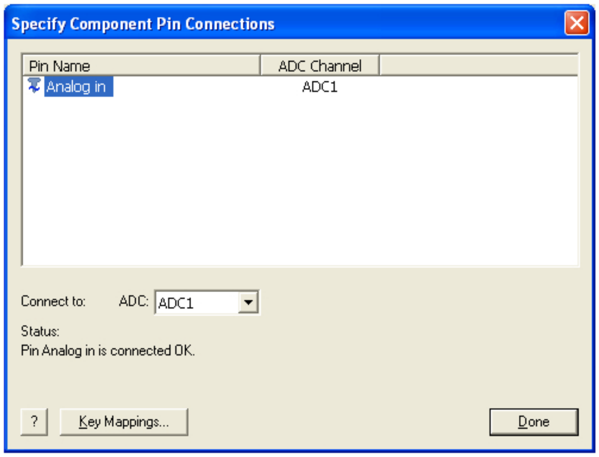
Parameters:
No parameters are required to call this macro.

Return Value:(BYTE)
temp

Figure 18

The next step is to configure the ADC(0) component. The (0) in its title signifies that it the first ADC component added to the flowchart. If you add a second one, it will be called ADC(1).

Click on the down arrow in the title bar of the ADC component, and select the 'Component Connections...' option. The 'Specify Component Pin Connections' window opens. The decision to make here is which analogue input on the PIC chip to use. The Demo Board already has an analogue component, the pot, connected to Port A bit 0, or as we should call it AN0, as we are dealing with an analogue input. There are fourteen such analogue inputs on the PIC16F887, spread across ports A, B and E. For our present purpose, we will use ADC channel 1, so select that from the pull-down menu next to the 'Connect to: ADC' label. The Properties window for this component allows you to select either a linear or a rotary pot for the on-screen representation of the device, and to change its colour. We will stick to default values. The 'Specify Component Pin Connections' window now looks like the following:



Pin Name	ADC Channel
Analog in	ADC1

Connect to: ADC: ADC1

Status:
Pin Analog in is connected OK.

Figure 19

Simulate program 2

Your Flowcode program is now complete. Test it by running the on-screen simulation.
If it works:

- When the knob of the on-screen pot is set to a minimum, LEDs 0 and 1 flash on and off alternately with LEDs 2 and 3, as a 'low temperature alarm'.
- As you rotate the knob, using the mouse cursor, this behaviour stops, and LEDs 0 and 1 switch on and stay on.
- A little further, and LEDs 2 and 3 come on as well.
- Turn the pot further, and LEDs 4 and 5 flash on and off alternately with LEDs 6 and 7, as a 'high temperature alarm'.

It’s soldering time!

The project needs a temperature sensor. This consists of a thermistor (the prototype used one with a resistance of 15kΩ at 25^{0}C) and a resistor (10kΩ in the prototype.) These values can be varied, in which case the thresholds in the Decision boxes in the Flowcode flowchart will have to be adjusted. Here is the circuit diagram for the temperature sensor:

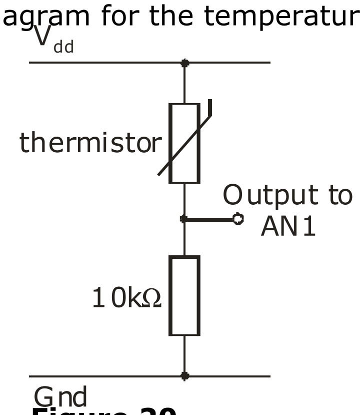


Figure 20

The sensor is constructed in the prototyping area at the bottom of the 44-pin Demo Board.

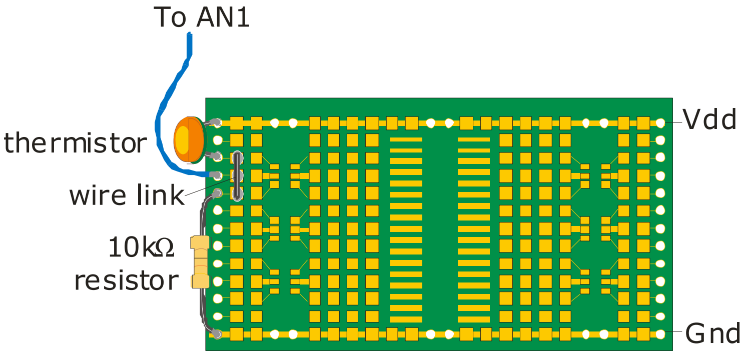


Figure 21

The prototype used leaded components, not surface mounted, and so the plated holes down the side of the board were used to make connections. A wire link was soldered to join the three pads connected to the thermistor, the connecting wire and the resistor. The connecting wire is then soldered to one of the plated holes on the input labelled RA1 (which is actually AN1 in this case.) Here is a photograph of the prototype.

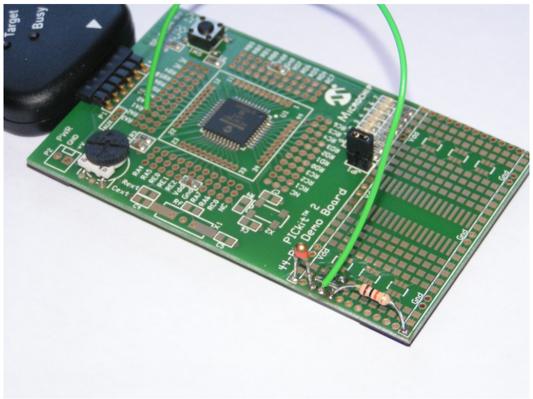


Figure 22

Programming the PIC chip

The procedure is the same as before.

- Connect the hardware to the USB port of your computer.
- Click on the 'Chip \ Compile to Chip...' option.
- Close the 'Complier Messages' window, when the transfer is complete.

The system can be tested by warming the thermistor between thumb and finger, or by blowing gently on the thermistor to warm it up. By 'tuning' the thresholds in the Decision boxes to match the temperature sensor, the system can be made extremely sensitive to temperature change, or can monitor a wide range of temperature.

And Finally!

Although this article has shown in some detail how to build and configure Flowcode flowcharts, much of the functionality of Flowcode has not been mentioned. The ability to perform calculations, manipulate string variables, create customised macros and inject sections of C programming has not been mentioned, along with a host of other powerful features of the program. Fortunately, the program comes equipped with a set of tutorials, accessed via the Help menu. Matrix Multimedia also offers a variety of supporting resources, available via their website www.matrixmultimedai.com.