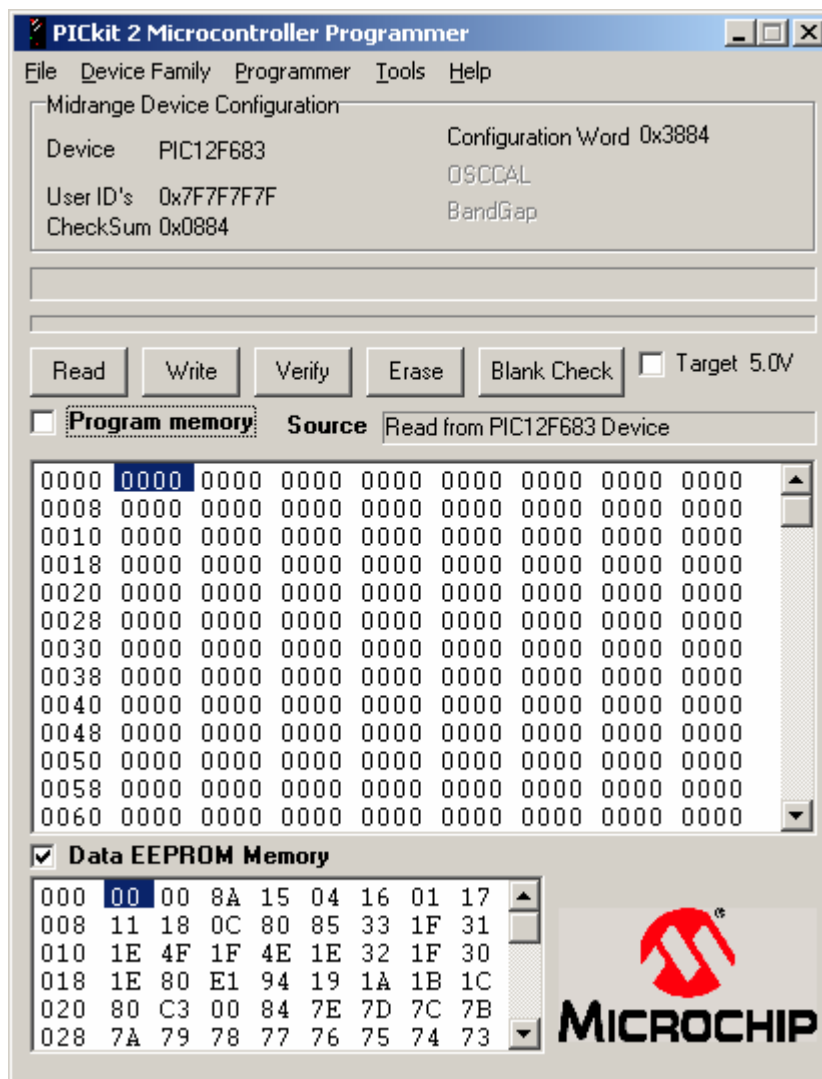


Programming the Programmable Christmas Star

Although the PIC source code will not be released, the command codes which define the display patterns are held in unprotected EEPROM, which you can update, independently of the protected code held in flash memory, with a suitable PIC programmer.

An excellent low-cost programmer is Microchip's PICkit2, available for around \$95. It comes with software that allows the PIC's EEPROM to be updated without affecting the program code in flash memory. The new command codes can be typed directly into the PICkit2 EEPROM window and loaded to the microcontroller. But it's very important to uncheck "Program memory", so that the program code is not overwritten. See the screenshot below.



Pattern sequence command codes

Code	Command	Description
0	Pause	All LEDs off. Use for a short pause between pattern sequences
1 - 91	Pattern	Display a pre-defined pattern of up to four LEDs which are on "at once". Refer to the following table for a definition of each pattern
92 - 126	Twinkle	LEDs are lit, one at a time, in pseudo-random order, in quick succession to create an overall "twinkling" effect. Twinkle rate = (code-91)ms between changes. If the code value = 92, a different LED is lit every 1ms – you may think too fast for the eye to see. But due to imperfections in the "random" number generation, you'll still see a shimmer at this maximum twinkle rate. For code = 126, the twinkling is at its slowest, around 29Hz.
127	End of sequence	Marks the end of the programmed sequence. Not necessary if your display codes fill the whole EEPROM, as the interpreter will restart at the beginning if the end of the EEPROM is reached.
128	End loop	Go back to first pattern in current loop – see below.
129 - 191	Start loop	Use this to create loops, to avoid having to fill the EEPROM with repeated sequences of codes to create a repeating effect. Instead, place a "start loop" instruction at the start of the sequence, and an "end loop" (128) instruction at the end. Repeat count = code-128 E.g. to repeat a sequence of patterns four times, you would place a code of 132 (= 128 + 4) before the first pattern code, and a code of 128 after the last. Note that nested loops are not supported. An "end loop" code will always return to the most recent "start loop".
192 - 255	Set Speed	Sets the display rate, i.e. the time spent displaying each pattern before moving to the next in sequence. It allows you to vary the speed of the display in different parts of the presentation. Freq = $1000000/[8192(256-pattern)]$ Hz Max. freq. (code = 255) is 122Hz. Min. freq (code = 192) is 1.9Hz. The default display rate, if you don't set your own speed, is 6.8Hz

Pattern Definitions

Code	Description	LED 1	LED 2	LED 3	LED 4
	<i>Individual LEDs</i>				
1	1 only	1			
2	2 only	2			
3	3 only	3			
4	4 only	4			
5	5 only	5			
6	6 only	6			
7	7 only	7			
8	8 only	8			
9	9 only	9			
10	10 only	10			
11	11 only	11			
12	12 only	12			
13	13 only	13			
14	14 only	14			
15	15 only	15			
16	16 only	16			
17	17 only	17			
18	18 only	18			
19	19 only	19			
20	20 only	20			
	<i>Arms</i>				
21	SE arm	8	10	7	3
22	NE arm	14	13	19	20
23	NW arm	15	16	5	18
24	SW arm	2	6	11	9
	<i>Rings</i>				
25	ring 1 - inner	3	14	15	2
26	ring 2 - inner mid	7	13	16	6
27	ring 3 - outer mid	10	19	5	11
28	ring 4 - outer	8	20	18	9
29	Small points	12	4	1	17
	<i>Complimentary pairs</i>				
30	N S	1	12		
31	E W	4	17		
32	SE1 NW1	3	15		
33	SE2 NW2	7	16		
34	SE3 NW3	10	5		
35	SE4 NW4	8	18		
36	SW1 NE1	2	14		
37	SW2 NE2	6	13		
38	SW3 NE3	11	19		
39	SW4 NE4	9	20		
	<i>Half arms</i>				
40	SE inner	3	7		
41	SE outer	10	8		
42	NE inner	14	13		
43	NE outer	19	20		
44	NW inner	15	16		
45	NW outer	5	18		
46	SW inner	2	6		

47	SW outer	11	9		
	<i>Complimentary halves</i>				
48	SE NW inner	3	7	15	16
49	SE NW outer	10	8	5	18
50	SW NE inner	2	6	14	13
51	SW NE outer	11	9	19	20
	<i>Alternate LEDs - 2 per arm</i>				
52	SE 1 3	3	10		
53	SE 2 4	7	8		
54	NE 1 3	14	19		
55	NE 2 4	13	20		
56	NW 1 3	15	5		
57	NW 2 4	16	18		
58	SW 1 3	2	11		
59	SW 2 4	6	9		
	<i>Alternate LEDS - 4 per diagonal</i>				
60	SE 1 3 NW 1 3	3	10	15	5
61	SE 2 4 NW 2 4	7	8	16	18
62	NE 1 3 SW 1 3	14	19	2	11
63	NE 2 4 SW 2 4	13	20	6	9
64	SE 1 3 NW 2 4	3	10	16	18
65	SE 2 4 NW 1 3	7	8	15	5
66	NE 1 3 SW 2 4	14	19	6	9
67	NE 2 4 SW 1 3	13	20	2	11
	<i>Inner and outer - 2 per arm</i>				
68	SE arm	3	8		
69	NE arm	14	20		
70	NW arm	15	18		
71	SW arm	2	9		
	<i>Inner and outer - 4 per diagonal</i>				
72	SE NW	3	8	15	18
73	NE SW	14	20	2	9
	<i>Middle LEDS - 2 per arm</i>				
74	SE arm	7	10		
75	NE arm	13	19		
76	NW arm	16	5		
77	SW arm	6	11		
	<i>Middle LEDS - 4 per diagonal</i>				
78	SE NW	7	10	16	5
79	NE SW	13	19	6	11
	<i>Inner and outer half arms - opposites on diagonal</i>				
80	SE inner NW outer	3	7	5	18
81	SE outer NW inner	10	8	15	16
82	NE inner SW outer	14	13	11	9
83	NE outer SW inner	19	20	2	6
	<i>Inner and outer half arms - perpendicular opposites</i>				
84	SE inner NE outer	3	7	19	20
85	SE inner SW outer	3	7	11	9
86	NE inner SE outer	14	13	10	8
87	NE inner NW outer	14	13	5	18
88	NW inner NE outer	15	16	19	20
89	NW inner SW outer	15	16	11	9
90	SW inner SE outer	2	6	10	8
91	SW inner NW outer	2	6	5	18

Example

As an example of how to put it all together, here's some code to twinkle at a moderate rate for 10s, then turn off (pause) for 1s, then repeat:

Code	Comment
195	display speed = 2.0Hz
148	repeat following patterns 20 times (128+20=148)
101	twinkle at 101-91=10ms per change (100Hz)
128	end loop
0	pause (all off)
0	pause again – at 2Hz we need 2 pauses to make 1 second
127	end sequence (repeat from beginning)

The corresponding hexadecimal code, as you would type it into the PICkit 2's EEPROM window, is:

C3 94 65 80 00 00 7F