

Micros Peripherals and Interfaces Report

by Daniel Frankland

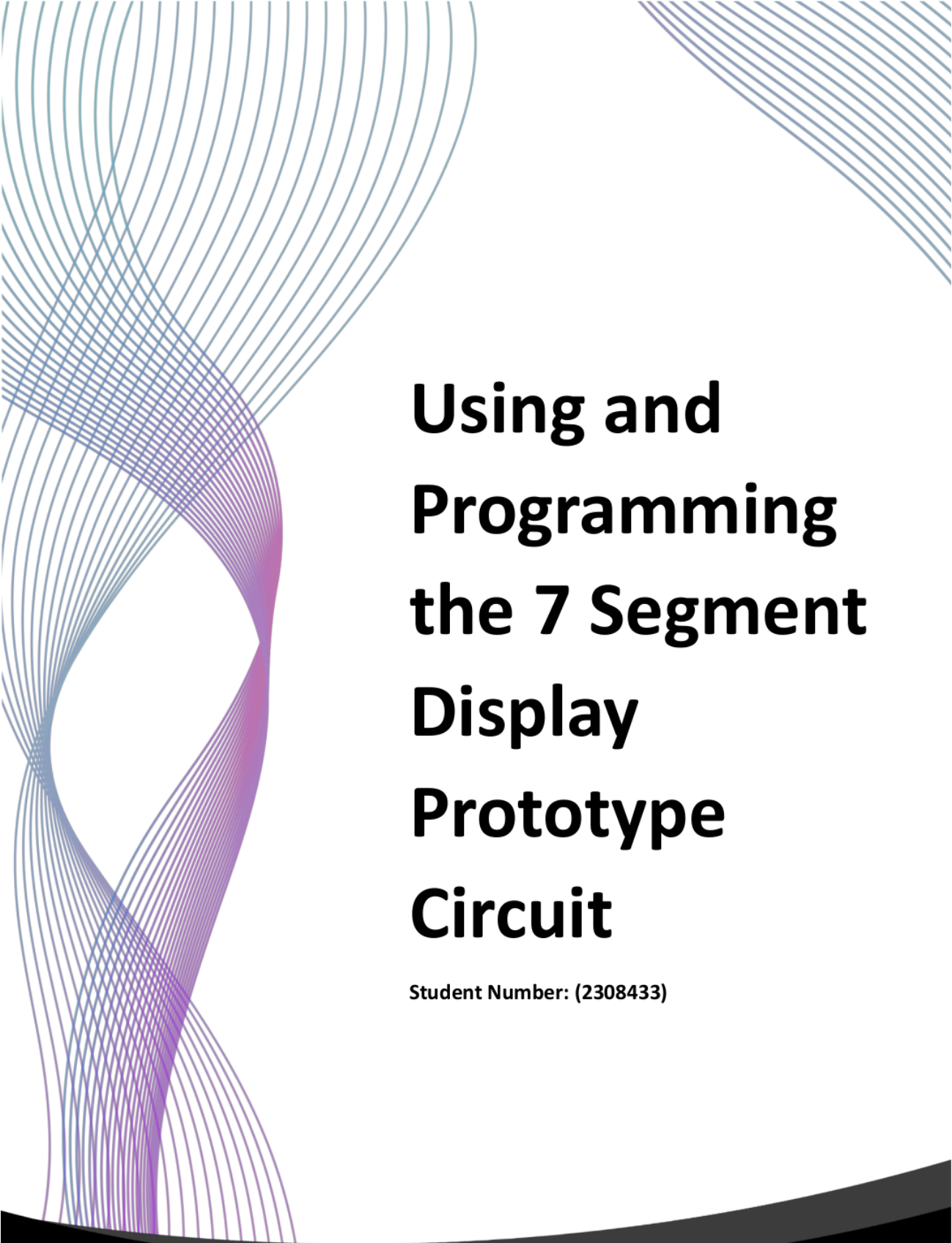
Submission date: 02-Jan-2025 08:10PM (UTC+0000)

Submission ID: 247039149

File name: 51461_Daniel_Frankland_Micros_Peripherals_and_Interfaces_Report_621295_1410493394.docx
(2.45M)

Word count: 1608

Character count: 8496



Using and Programming the 7 Segment Display Prototype Circuit

Student Number: (2308433)

2. Table of Contents

1. Title Page.....	1
2. Table of Contents	2
3. Table of Figures	3
4. The Circuit	4
4.1. Circuit Explanation	4
4.2. Component Explanations	7
4.2.1. Resistors	7
4.2.2. Capacitors.....	7
4.2.3. Transistors	7
4.2.4. LED's.....	7
5. Display 0 to F	8
5.1. Code	8
5.2. Flowchart	11
5.3. Code Operation	12
5.4. Binary Calculations.....	12
6. Display L with one button and R with the other	13
6.1. Code	13
6.2. Flowchart.....	15
7. Latching buttons to display all LED's with one button and none with the other.....	16
7.1. Code	16
7.2. Flowchart	18
8. Counter which starts on 5 and counts down to 0 or up to 9 with button presses	19
8.1. Code	19
8.2. Flowchart	26
9. Sound piezo when a button is pressed	27
9.1. Code	27
9.2. Flowchart	29
10. References.....	30

3. Table of Figures

Figure 1 The Circuit in Proteus	4
Figure 2 Circuit-7 segment display	4
Figure 3 Circuit- Resistor Bank	5
Figure 4 Circuit-Button and chip power	5
Figure 5 Circuit-Crystal Oscillator	6
Figure 6 Circuit-Sounder Circuit	6
Figure 7 Capacitor smoothing	7
Figure 8 Display 0-F Code-1.....	8
Figure 9 Display 0-F Code-2.....	9
Figure 10 Display 0-F Code-3.....	10
Figure 11 Display 0-F Code-4.....	10
Figure 12 7-segment led display calculations	12
Figure 13 Display L and R Code-1.....	13
Figure 14 Display L and R Code-2.....	14
Figure 15 Display L and R Flowchart	15
Figure 16 Display all LED's on and off Code-1.....	16
Figure 17 Display all LED's on and off Code-2	17
Figure 18 Display all LED's on and off Flowchart.....	18
Figure 19 Display 0-9 Code-1	19
Figure 20 Display 0-9 Code-2	20
Figure 21 Display 0-9 Code-3	21
Figure 22 Display 0-9 Code-4	22
Figure 23 Display 0-9 Code-5	23
Figure 24 Display 0-9 Code-6	24
Figure 25 Display 0-9 Code-7	25
Figure 26 Display 0-9 Flowchart	26
Figure 27 Sound piezo Code-1	27
Figure 28 Sound piezo Code-2.....	28
Table 1 Binary and Bitwise Calculations	12

4. The Circuit

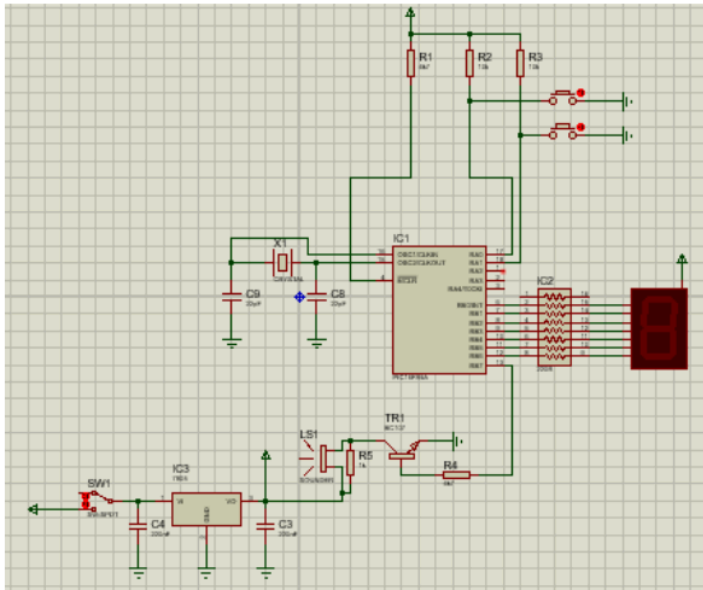


Figure 1 The Circuit in Proteus

4.1. Circuit Explanation

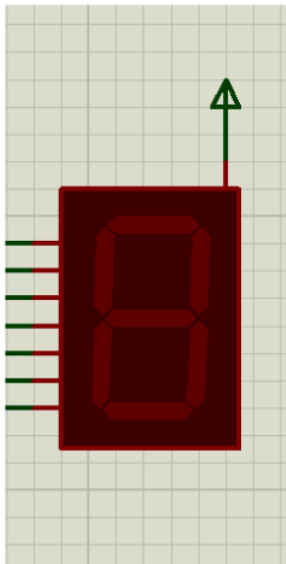


Figure 2 Circuit-7 segment display

Figure 2 is the 7-segment display which was used in the circuit. It has a total of 8 pins, one pin is for the common anode. The other 7 pins are used to control each individual LED of the display. These LED's can be in either the on or off state. These are controlled by individual pins on the pic microcontroller.

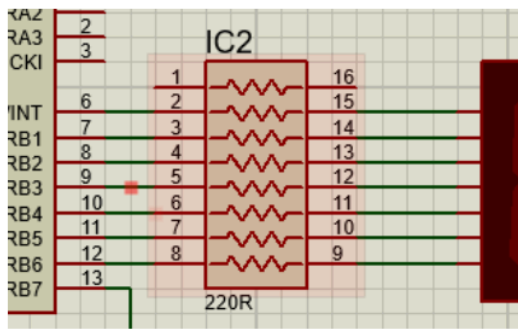


Figure 3 Circuit- Resistor Bank

The IC2 in Figure 3 is the 220-ohm resistor bank which is used to make sure that the LED's aren't over powered with too much current. This is connected directly to the pic and the display. "The series resistors limit the drive current." [1]

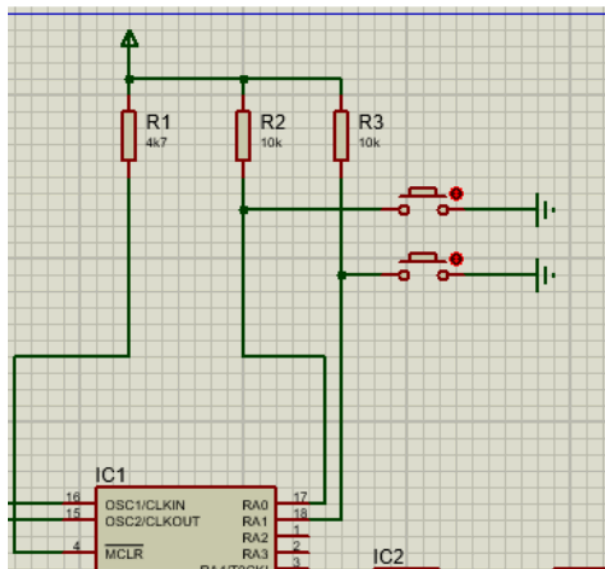


Figure 4 Circuit-Button and chip power

At the top of Figure 4 is the power that supplies both of the button inputs and power directly to the pic. The resistors are used to limit the current that is being provided into the pic. Each button has a separate 10k resistor which provides power directly into the pic. When the button is pressed the current will take the path of least resistance and will go straight to ground setting the pic's input to low.

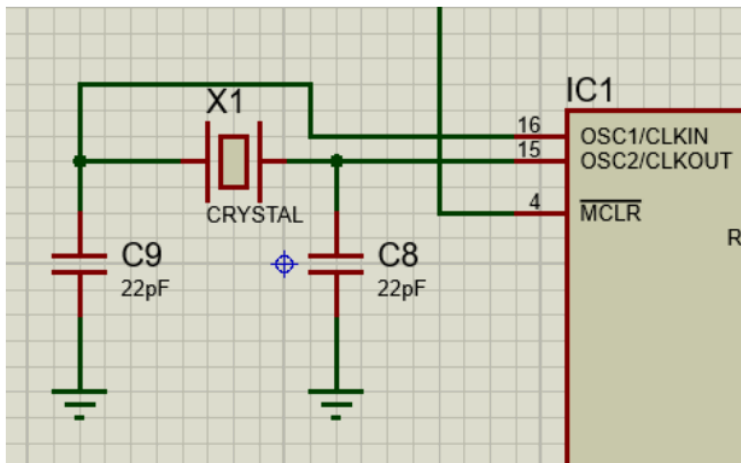


Figure 5 Circuit-Crystal Oscillator

Figure 5 shows the crystal oscillator circuit that is used to provide an accurate reference clock which makes sure that all the pic's instructions are completed on time. The capacitors are then used to smooth out the spikes of voltage in the circuit.

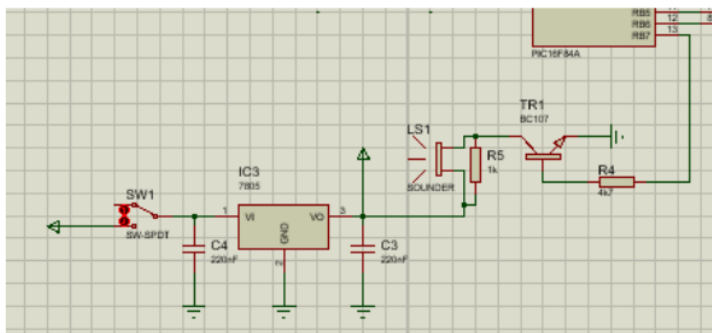


Figure 6 Circuit-Sounder Circuit

Figure 6 shows the sounder circuit. IC3 as seen on the image is a voltage regulator used to supply a set voltage to the components. The capacitors are used to smooth and stabilise voltage spikes. R5 (resistor) running in parallel with the sounder is used to limit the current going through the sounder. The transistor is then used as a switch to control the current flowing through the circuit. This will be active when the transistor is provided with a reference from the PIC, which is fed through a resistor to ensure correct operation. This is known as biasing resistors. Transistors and other devices need to have their AC and DC operating characteristics and gain values set up for correct operation. [2]

4.2. Component Explanations

4.2.1. Resistors

What is a resistor and why is it used in this circuit? A resistor is an electrical component that restricts the flow of electrical current in a circuit. [2]. There are multiple use cases for resistors and in this circuit only a few of its cases are used. These are current limiting and biasing.

4.2.2. Capacitors

Capacitors are similar to a battery where they store charge and release it. However, unlike the battery a capacitor temporarily stores electrical energy. It can take short time to charge up and can release all of its charge quickly. [3]

In the circuit these are used for smoothing out voltage spikes in the circuit. When the voltage spikes down the stored energy will be released to help smooth this out. An example of this can be seen below in Figure 7 where its used in AC-DC conversions with a rectifier.

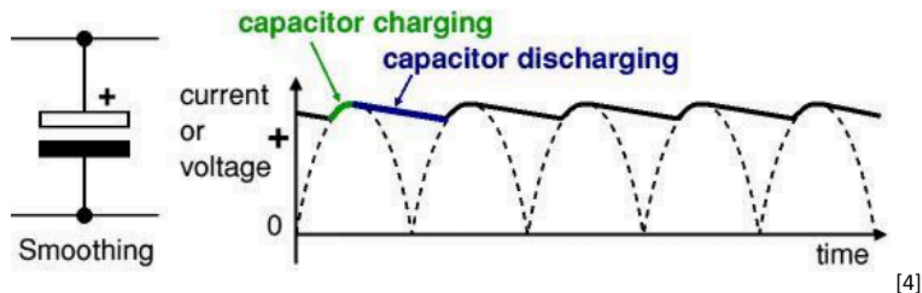


Figure 7 Capacitor smoothing

4.2.3. Transistors

A transistor in its simplest form is an electronic switch. It allows a small amount of current to switch on or off a much larger amount of current. [5]. On the specified circuit we are using the transistor exactly as described. Where we are using the output from the microcontroller to trigger the transistor overcoming its minimum biasing voltage, which will allow the secondary circuit to be powered and operate.

4.2.4. LED's

LED's (light emitting diodes) are a type of semiconductor light source that combine a p type semiconductor and a n type semiconductor. When a sufficient forward voltage is applied it will cause the electrons to recombine at the P-N junction releasing energy in the form of light [6]. In the case of the circuit being used we use multiple LED's inside of a 7-segment display with a common anode (positive pin).

5. Display 0 to F

5.1. Code

```
1 ;=====
2 ; Main.asm file generated by New Project wizard
3 ;
4 ; Created By: Daniel Frankland
5 ; Processor: PIC16F84A
6 ; Compiler: MPASM (Proteus)
7 ;=====
8
9 ;=====
10 ; DEFINITIONS
11 ;=====
12
13 #include p16f84a.inc ; Include register definition file
14
15 portB equ 0x06 ;port B address links to the data sheet
16 trisB equ 0x06 ;trisB address in register bank 1
17
18 status equ 0x03 ;status register
19 RP0 equ 5 ;bit 5 of status register
20 RP1 equ 6 ;bit 6 of status register
21
22 ;=====
23 ; VARIABLES
24 ;=====
25 count equ 0x0c ;counter 0 to 255
26 nCount equ 0x0d ;pause routine inner loop counter
27 mCount equ 0x0e ;pause routine outer loop counter
28 ;=====
29 ; RESET and INTERRUPT VECTORS
30 ;=====
31
32 ; Reset Vector
33 RST code 0x0
34 goto Start
35
36 ;=====
37 ; CODE SEGMENT
38 ;=====
39
40 PGM code
41 Start
42 bcf status,RP0 ; clear bank selection bits in the status register
43 bcf status,RP1
44
```

Figure 8 Display 0-F Code-1

```

45      bsf      status,RP0      ; select bank 1
46      movlw   0x00            ; load w with 0x00
47      movwf   trisb           ; load it into trisb to make portb all outputs
48      bcf      status,RP0      ; reselect bank 0
49
50      clrf     portB           ; clear all lines low
51      clrf     count           ; initialize counter to 0
52
53  get_count
54
55      movlw   b'01000000'      ;0
56      movwf   portB            ; move W to port B to update the display
57      call    pause            ; delay 200ms by subroutine
58
59      movlw   b'01111001'      ;1
60      movwf   portB            ; move W to port B to update the display
61      call    pause            ; delay 200ms by subroutine
62
63      movlw   b'00100100'      ;2
64      movwf   portB            ; move W to port B to update the display
65      call    pause            ; delay 200ms by subroutine
66
67      movlw   b'00110000'      ;3
68      movwf   portB            ; move W to port B to update the display
69      call    pause            ; delay 200ms by subroutine
70
71      movlw   b'00011001'      ;4
72      movwf   portB            ; move W to port B to update the display
73      call    pause            ; delay 200ms by subroutine
74
75      movlw   b'00010010'      ;5
76      movwf   portB            ; move W to port B to update the display
77      call    pause            ; delay 200ms by subroutine
78
79      movlw   b'00000011'      ;6
80      movwf   portB            ; move W to port B to update the display
81      call    pause            ; delay 200ms by subroutine
82
83      movlw   b'01111000'      ;7
84      movwf   portB            ; move W to port B to update the display
85      call    pause            ; delay 200ms by subroutine

```

Figure 9 Display 0-F Code-2

```

86
87     movlw b'00000000'    ;8
88     movwf portB          ; move W to port B to update the display
89     call  pause          ; delay 200ms by subroutine
90
91     movlw b'00010000'    ;9
92     movwf portB          ; move W to port B to update the display
93     call  pause          ; delay 200ms by subroutine
94
95     movlw b'00001000'    ;A
96     movwf portB          ; move W to port B to update the display
97     call  pause          ; delay 200ms by subroutine
98
99     movlw b'00000011'    ;B
100    movwf portB          ; move W to port B to update the display
101    call  pause          ; delay 200ms by subroutine
102
103    movlw b'01000110'    ;C
104    movwf portB          ; move W to port B to update the display
105    call  pause          ; delay 200ms by subroutine
106
107    movlw b'00100001'    ;D
108    movwf portB          ; move W to port B to update the display
109    call  pause          ; delay 200ms by subroutine
110
111    movlw b'00000110'    ;E
112    movwf portB          ; move W to port B to update the display
113    call  pause          ; delay 200ms by subroutine

```

Figure 10 Display 0-F Code-3

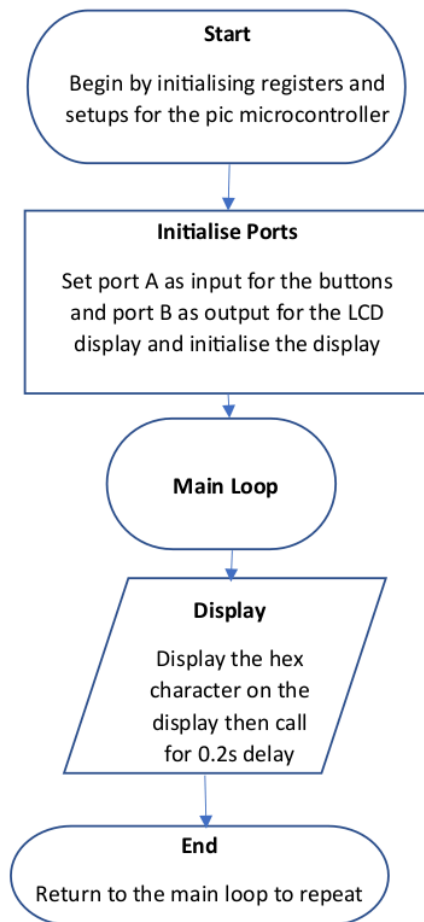
```

114
115     movlw b'00001110'    ;F
116     movwf portB          ; move W to port B to update the display
117     call  pause          ; delay 200ms by subroutine
118
119
120     goto  get_count      ; repeat forever
121
122 ;=====
123 ;subroutines
124 ;=====
125 pause
126     movlw 0xff           ; set w = 255 decimal
127     movwf mCount         ; mCount = w: outer loop counter
128 loadN
129     movlw 0xff           ; set w = 255 decimal
130     movwf nCount         ; nCount = w: inner loop counter
131 decN
132     decfsz nCount, f      ; nCount--
133     goto  decN            ; if nCount != 0 then repeat nCount--
134
135     decfsz mCount, f      ; else decrement Count
136     goto  loadN           ; if mCount != 0 then
137                         ; reload nCount to 255 and decrement
138     return               ; else exit subroutine
139
140
141 end

```

Figure 11 Display 0-F Code-4

5.2. Flowchart



5.3. Code Operation

This task was to display the hexadecimal characters 0-F on the display. First the initialisation of the ports is completed whilst loading in the selected banks. After that I progressed into the main function, each hexadecimal character is moved into the w register then moved to port b for the display, then the delay of 200ms is called. This continues for every selected character until it reaches the end and is then repeated infinitely whilst the pic is provided power.

5.4. Binary Calculations

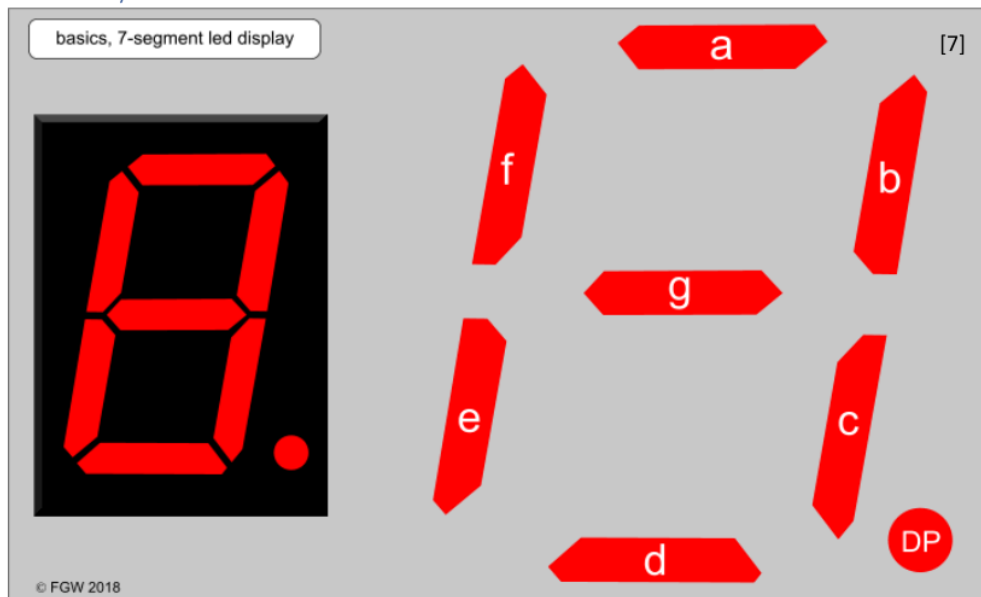


Figure 12 7-segment led display calculations

To work out the individual LED's which needed displaying I used Figure 12 to assist with calculating the binary values.

Due to the way the seven-segment display is powered for the individual LED's to be powered on the ports are being set as ground (0) rather the power (1). The first bit of all of the 8 bits is set as 0 to avoid sending a power to pin B7. The binary value appears backwards due to it being read from left to right with left most bit being the most significant bit and right most bit being the least significant bit.

Character to display	Binary value	Character to display	Binary value
0	01000000	8	00000000
1	01111001	9	00010000
2	00100100	A	00001000
3	00110000	B	00000011
4	00011001	C	01000110
5	00010010	D	00100001
6	00000011	E	00000110
7	01111000	F	00001110

Table 1 Binary and Bitwise Calculations

6. Display L with one button and R with the other

6.1. Code

```
1 ;=====
2 ; Main.asm file generated by New Project wizard
3 ;
4 ; Created by: Daniel Frankland
5 ; Processor: PIC16F84A
6 ; Compiler: MPASM (Proteus)
7 ;=====
8
9 ;=====
10 ; DEFINITIONS
11 ;=====
12
13 #include p16f84a.inc ; Include register definition file
14
15 portB equ 0x06 ;port B address links to the data sheet
16 trisB equ 0x06 ;trisB address in register bank 1
17
18 status equ 0x03 ;status register
19 RP0 equ 5 ;bit 5 of status register
20 RP1 equ 6 ;bit 6 of status register
21
22 portA equ 0x05 ;port A address links to the data sheet
23 trisA equ 0x05 ;Tris A adress in reigster bank
24
25 ;=====
26 ; VARIABLES
27 ;=====
28 count equ 0x0c ;counter 0 to 255
29 nCount equ 0x0d ;pause routine inner Loop counter
30 mCount equ 0x0e ;pause routine outer Loop counter
31
32
33
34 ;=====
35 ; CODE SEGMENT
36 ;=====
37 org 0
38 movlw 0x01 ;set the port A pins
39 movwf trisA ;set to input
40 bcf status,5 ;swithc back to bank 0
41 goto But1 ;go to but1 line
42
43 ;=====
44 ;MAIN LOOP
45 ;=====
46 Loop
47
```

Figure 13 Display L and R Code-1

```

48 ;=====
49 ;BUTTON1 LOOP
50 ;=====
51 But1
52 BTFSC portA,1 ;if portA bit 1 is 0 skip next insturction if 1 complete next insturction
53 goto But2 ;goes back to the top of the Loop
54 call pause ;delay
55 movlw 0x00
56 tris portB
57 movlw b'01001110' ; bit value to display r
58 movwf portB ;send to display
59
60 ;=====
61 ;BUTTON2 LOOP
62 ;=====
63 But2 BTFSC portA,0 ;if portA bit 1 is 0 skip next insturction if 1 complete next insturction
64 goto But1 ;goes back to the top of the Loop
65 call pause ;delay
66 movlw 0x00
67 tris portB
68 movlw b'01000111' ;bit value to display L
69 movwf portB ;send to display
70
71 goto Loop ;go back to loop
72
73 ;=====
74 ;SUBROUTINES
75 ;=====
76
77 pause
78     movlw 0xff ; set w = 255 decimal
79     movwf mCount ; mCount = w: outer Loop counter
80 loadN
81     movlw 0xff ; set w = 255 decimal
82     movwf nCount ; nCount = w: inner Loop counter
83 decN
84     decfsz nCount, f ; nCount--
85     goto decN ; if nCount != 0 then repeat nCount--
86
87     decfsz mCount, f ; else decrement Count
88     goto loadN ; if mCount != 0 then
89     return ; else exit subroutine
90
91
92 end

```

Figure 14 Display L and R Code-2

6.2. Flowchart

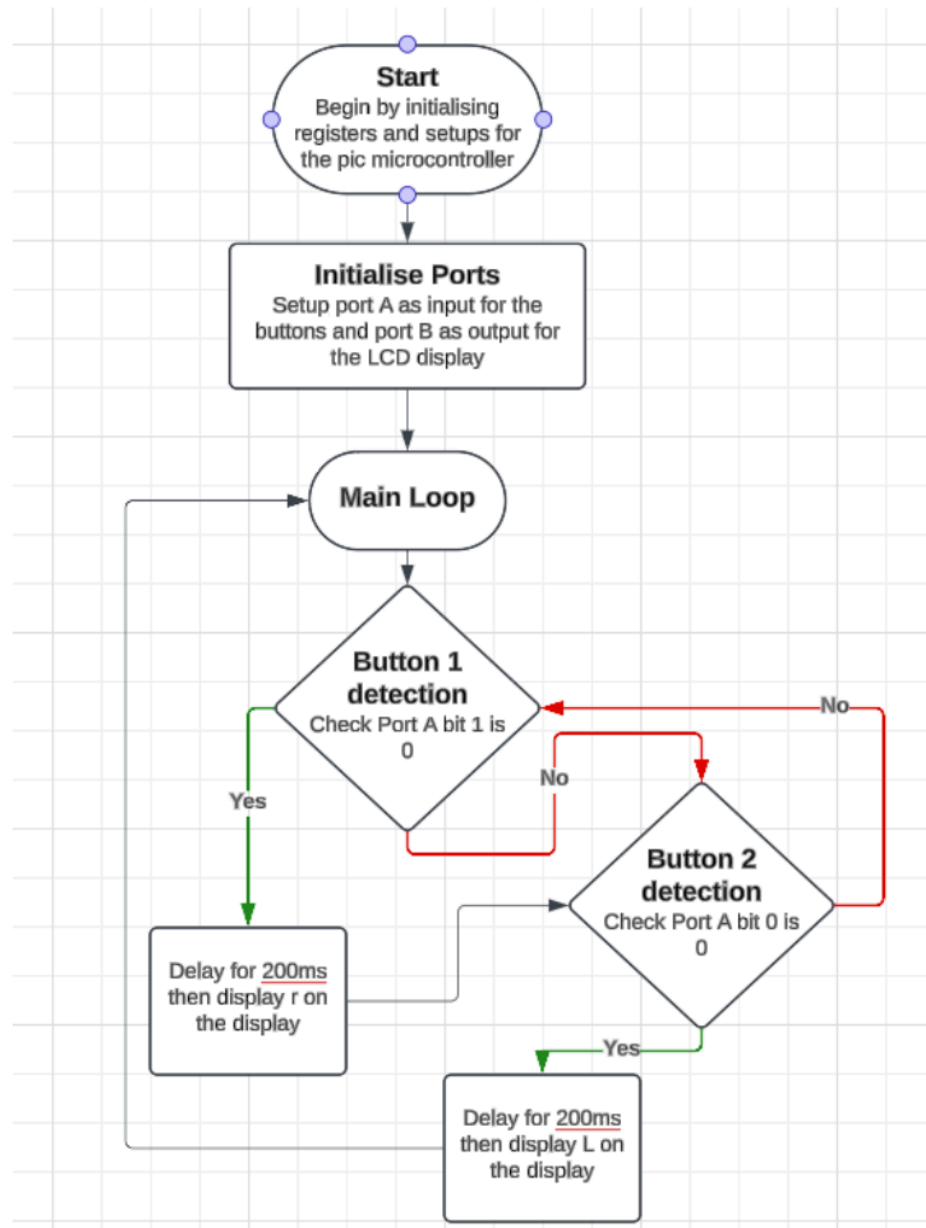


Figure 15 Display L and R Flowchart

7. Latching buttons to display all LED's with one button and none with the other

7.1. Code

```
1 ;=====
2 ; Main.asm file generated by New Project wizard
3 ;
4 ; Created by: Daniel Frankland
5 ; Processor: PIC16F84A
6 ; Compiler: MPASM (Proteus)
7 ;=====
8
9 ;=====
10 ; DEFINITIONS
11 ;=====
12
13 #include p16f84a.inc ; Include register definition file
14
15 portB equ 0x06 ;port B address Links to the data sheet
16 trisB equ 0x06 ;trisB address in register bank 1
17
18 status equ 0x03 ;status register
19 RP0 equ 5 ;bit 5 of status register
20 RP1 equ 6 ;bit 6 of status register
21
22 portA equ 0x05 ;port A address links to the data sheet
23 trisA equ 0x05 ;Tris A adress in reigster bank
24
25 ;=====
26 ; VARIABLES
27 ;=====
28 count equ 0x0c ;counter 0 to 255
29 nCount equ 0x0d ;pause routine inner loop counter
30 mCount equ 0x0e ;pause routine outer loop counter
31
32 ;=====
33 ; CODE SEGMENT
34 ;=====
35 org 0 ;programm memory location 0
36 movlw 0x00
37 tris portB
38 movlw 0x01 ;set the port A pins
39 movwf trisA ;set to input
40 bcf status,5 ;switch back to bank 0
41 goto But1 ;go to but1 line
42
43 ;=====
44 ;MAIN LOOP
45 ;=====
46 Loop
47 ;=====
```

Figure 16 Display all LED's on and off Code-1

```

48 ;BUTTON1 LOOP
49 ;=====
50 But1
51 BTFSC portA,1 ;if portA bit 1 is 0 skip next insturction if 1 complete next insturction
52 goto But2 ;goes back to the top of the loop
53 call pause ;delay
54 movlw b'00000000' ; bit value to set each segment of the LED's to on
55 movwf portB ;send to display
56
57 ;=====
58 ;BUTTON2 LOOP
59 ;=====
60 But2
61 BTFSC portA,0 ;if portA bit 0 is 0 skip next insturction if 1 complete next insturction
62 goto But1 ;goes back to the top of the loop
63 call pause ;delay
64 movlw b'01111111' ;bit value to set each segement of LED's to off
65 movwf portB ;send to display
66
67 goto Loop ;go back to loop
68
69 ;=====
70 ;SUBROUTINES
71 ;=====
72
73 pause
74 movlw 0xff ; set w = 255 decimal
75 movwf mCount ; mCount = w: outer loop counter
76 loadN
77 movlw 0xff ; set w = 255 decimal
78 movwf nCount ; nCount = w: inner loop counter
79 decN
80 decfsz nCount, f ; nCount--
81 goto decN ; if nCount != 0 then repeat nCount--
82
83 decfsz mCount, f ; else decrement Count
84 goto loadN ; if mCount != 0 then
85 ; reload nCount to 255 and decrement
86 ; else exit subroutine
87 return
88 end

```

Figure 17 Display all LED's on and off Code-2

7.2. Flowchart

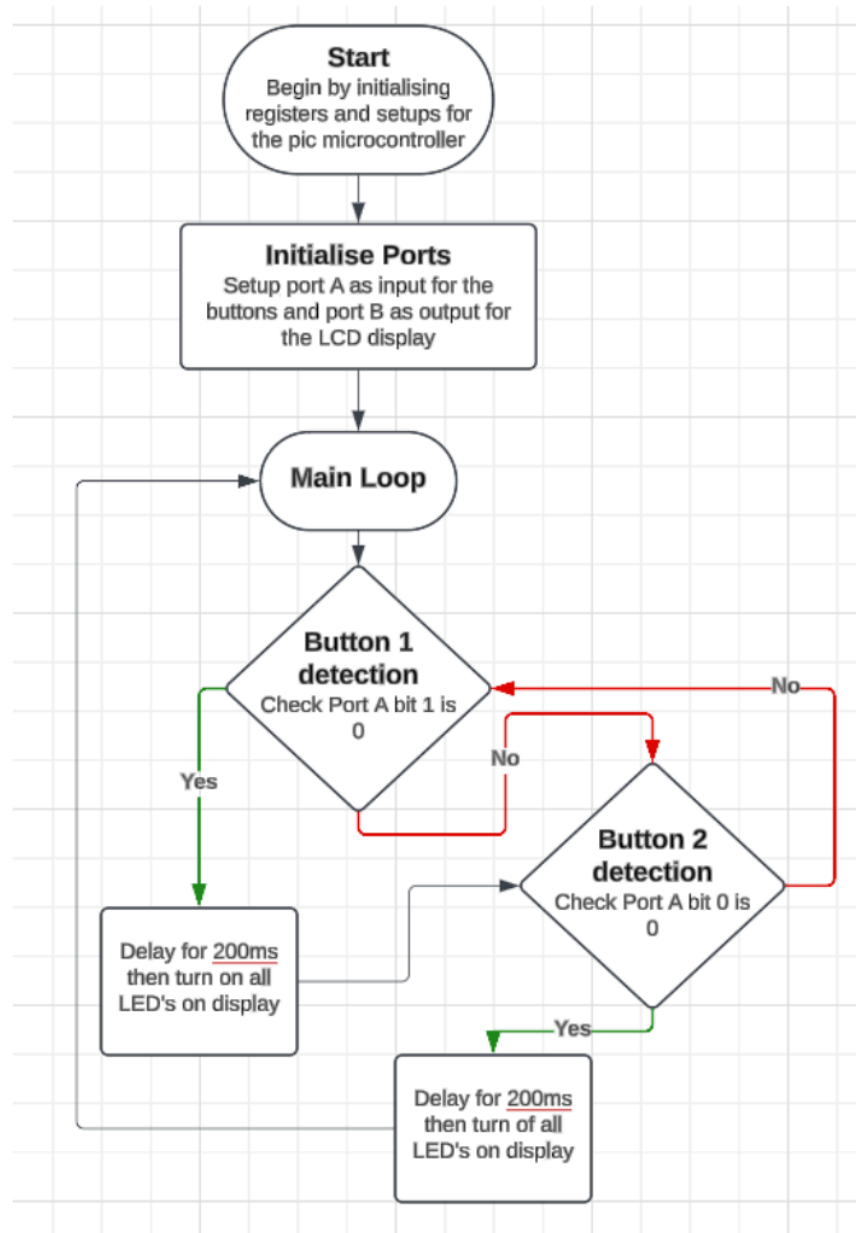


Figure 18 Display all LED's on and off Flowchart

8. Counter which starts on 5 and counts down to 0 or up to 9 with button presses

8.1. Code

```
1 ;=====
2 ; Main.asm file generated by New Project wizard
3 ;
4 ; Created by: Daniel Frankland
5 ; Processor: PIC16F84A
6 ; Compiler: MPASM (Proteus)
7 ;=====
8
9 ;=====
10 ; DEFINITIONS
11 ;=====
12
13 #include p16f84a.inc ; Include register definition file
14
15 portB equ 0x06 ;port B address links to the data sheet
16 trisB equ 0x86 ;trisB address in register bank 1
17
18 status equ 0x03 ;status register
19 RP0 equ 5 ;bit 5 of status register
20 RP1 equ 6 ;bit 6 of status register
21
22 portA equ 0x05 ;port A address links to the data sheet
23 trisA equ 0x85 ;Tris A address in register bank
24
25 ;=====
26 ; VARIABLES
27 ;=====
28 count equ 0x0C ; Variable to hold the count
29 nCount equ 0x0D ; Pause routine inner loop counter
30 prevStateBut1 equ 0x0E ; Previous state for button 1
31 prevStateBut2 equ 0x0F ; Previous state for button 2
32
33 ;=====
34 ; CODE SEGMENT
35 ;=====
36 org 0
37 goto Start
38
39 Start
40 ; Setting PORTA as inputs for the buttons
41 bsf status, RP0 ; Select bank 1
42 movlw 0xFF
43 movwf trisA ; Set PORTA as input
44
45 ; Set PORTB as output for the display
46 clrf trisB ; Set PORTB pins as output
47 bcf status, RP0 ; Switch back to bank 0
```

Figure 19 Display 0-9 Code-1

```

48
49 clrfs prevStateBut1    ; Initialize previous button state to 0
50 clrfs prevStateBut2    ; Initialize previous button state to 0
51
52 ;=====
53 ; Initialize display with '5' and start count at 5
54 ;=====
55 movlw b'00010010'      ; Binary for displaying 5
56 movwf portB             ; Send initial display to PORTB
57 movlw 0x05              ; Initialize count at 5
58 movwf count
59
60 mainLoop
61 ;=====
62 ; Button 1 detection (count up)
63 ;=====
64 BTFSS portA, 0          ; Check if button 1 is pressed
65 goto checkReleaseBut1   ; If pressed check for release
66 clrfs prevStateBut1     ; Clear previous state if not pressed
67 call pause
68
69 ;=====
70 ; Button 2 detection (count down)
71 ;=====
72 BTFSS portA, 1          ; Check if button 2 is pressed
73 goto checkReleaseBut2   ; If pressed check for release
74 clrfs prevStateBut2     ; Clear previous state if not pressed
75 call pause
76
77 goto mainLoop           ; Repeat the Loop
78
79 ;=====
80 ; Button 1 release detection (count up)
81 ;=====
82 checkReleaseBut1
83 movf prevStateBut1, w    ; Load previous state for button 1
84 btfsc status, Z          ; If previous state was 0 (button not pressed) continue
85 call count_up            ; Call count_up routine
86 movlw 0x01
87 movwf prevStateBut1     ; Update previous state to 1 (button pressed)
88
89 goto mainLoop
90

```

Figure 20 Display 0-9 Code-2

```

91 ;=====
92 ; Button 2 release detection (count down)
93 ;=====
94 checkReleaseBut2
95 movf prevStateBut2, w ; Load previous state for button 2
96 btfsc status, Z ; If previous state was 0 (button not pressed) continue
97 call count_down ; Call count down routine
98 movlw 0x01
99 movwf prevStateBut2 ; Update previous state to 1 (button pressed)
100
101 goto mainLoop
102
103 ;=====
104 ;Increment count up
105 ;=====
106 count_up
107 movf count, w ; Load current count into W
108 sublw 0x09 ; Check if count is 9
109 btfsc status, Z ; If count is 9 skip increment
110 return
111 incf count, f ; Increment count
112 call display ; Update display
113 return
114
115 ;=====
116 ;Increment count down
117 ;=====
118 count_down
119 movf count, w ; Load current count into W
120 btfsc status, Z ; If count is 0 skip decrement
121 return
122 decf count, f ; Decrement count
123 call display ; Update display
124 return
125
126 ;=====
127 ;Display the value loop
128 ;=====
129 display
130 movf count, w ; Move count value to W
131 movwf portB ; Send to display on PORTB
132 call pause
133 call check ;calls the check loop to display the correct number
134 return

```

Figure 21 Display 0-9 Code-3

```

135
136 ;=====
137 ;check if count is 0
138 ;=====
139 check
140 movf 0x0C, w      ;memory location of count to load into w
141 movlw 0x00         ;load w with 0
142 subwf 0x0C, w      ;subtract 0 from count
143 btfss status, Z    ;if the zero flag is set skip next line else continue
144 goto check_1       ;check next value
145 movlw b'01000000' ;load correct value for display
146 movwf portB        ;send to display
147 return             ;return
148
149 ;=====
150 ;check if count is 1
151 ;=====
152 check_1
153 movlw 0x01         ;load w with 1
154 subwf 0x0C, w      ;subtract 1 from count
155 btfss status, Z    ;if the zero flag is set skip next line else continue
156 goto check_2       ;check next value
157 movlw b'01111001' ;load correct value for display
158 movwf portB        ;send to display
159 return             ;return
160
161 ;=====
162 ;check if count is 2
163 ;=====
164 check_2
165 movlw 0x02         ;load w with 2
166 subwf 0x0C, w      ;subtract 2 from count
167 btfss status, Z    ;if the zero flag is set skip next line else continue
168 goto check_3       ;check next value
169 movlw b'00100100' ;load correct value for display
170 movwf portB        ;send to display
171 return             ;return
172

```

Figure 22 Display 0-9 Code-4

```

173 ;=====
174 ;check if count is 3
175 ;=====
176 check_3
177 movlw 0x03          ;load w with 3
178 subwf 0x0C, w       ;subtract 3 from count
179 btfss status, Z     ;if the zero flag is set skip next line else continue
180 goto check_4        ;check next value
181 movlw b'00110000'   ;load correct value for display
182 movwf portB         ;send to display
183 return              ;return
184
185 ;=====
186 ;check if count is 4
187 ;=====
188 check_4
189 movlw 0x04          ;load w with 4
190 subwf 0x0C, w       ;subtract 4 from count
191 btfss status, Z     ;if the zero flag is set skip next line else continue
192 goto check_5        ;check next value
193 movlw b'00011001'   ;load correct value for display
194 movwf portB         ;send to display
195 return              ;return
196
197 ;=====
198 ;check if count is 5
199 ;=====
200 check_5
201 movlw 0x05          ;load w with 5
202 subwf 0x0C, w       ;subtract 5 from count
203 btfss status, Z     ;if the zero flag is set skip next line else continue
204 goto check_6        ;check next value
205 movlw b'00010010'   ;load correct value for display
206 movwf portB         ;send to display
207 return              ;return
208

```

Figure 23 Display 0-9 Code-5

```

209 ;=====
210 ;check if count is 6
211 ;=====
212 check_6
213 movlw 0x06          ;load w with 6
214 subwf 0x0C, w       ;subtract 6 from count
215 btfss status, Z     ;if the zero flag is set skip next line else continue
216 goto check_7        ;check next value
217 movlw b'00000011'   ;load correct value for display
218 movwf portB         ;send to display
219 return              ;return
220
221 ;=====
222 ;check if count is 7
223 ;=====
224 check_7
225 movlw 0x07          ;load w with 7
226 subwf 0x0C, w       ;subtract 7 from count
227 btfss status, Z     ;if the zero flag is set skip next line else continue
228 goto check_8        ;check next value
229 movlw b'01111000'   ;load correct value for display
230 movwf portB         ;send to display
231 return              ;return
232
233 ;=====
234 ;check if count is 8
235 ;=====
236 check_8
237 movlw 0x08          ;load w with 8
238 subwf 0x0C, w       ;subtract 8 from count
239 btfss status, Z     ;if the zero flag is set skip next line else continue
240 goto check_9        ;check next value
241 movlw b'00000000'   ;load correct value for display
242 movwf portB         ;send to display
243 return              ;return
244

```

Figure 24 Display 0-9 Code-6

```

245 ;=====
246 ;check if count is 9
247 ;=====
248 check_9
249 movlw 0x09           ;load w with 9
250 subwf 0x0C, w        ;subtract 9 from count
251 btfss status, Z      ;if the zero flag is set skip next line else continue
252 return               ;check next value
253 movlw b'00010000'    ;load correct value for display
254 movwf portB          ;send to display
255 return               ;return
256
257 ;=====
258 ;pause
259 ;=====
260 pause
261 movlw 0xFF           ; Load W with 255 for outer loop
262 movwf nCount         ; Set outer loop counter
263
264 delayLoop
265 decfsz nCount, f      ; Decrement counter
266 goto delayLoop       ; Repeat until nCount reaches 0
267 return               ; Exit delay
268
269 end

```

Figure 25 Display 0-9 Code-7

8.2. Flowchart

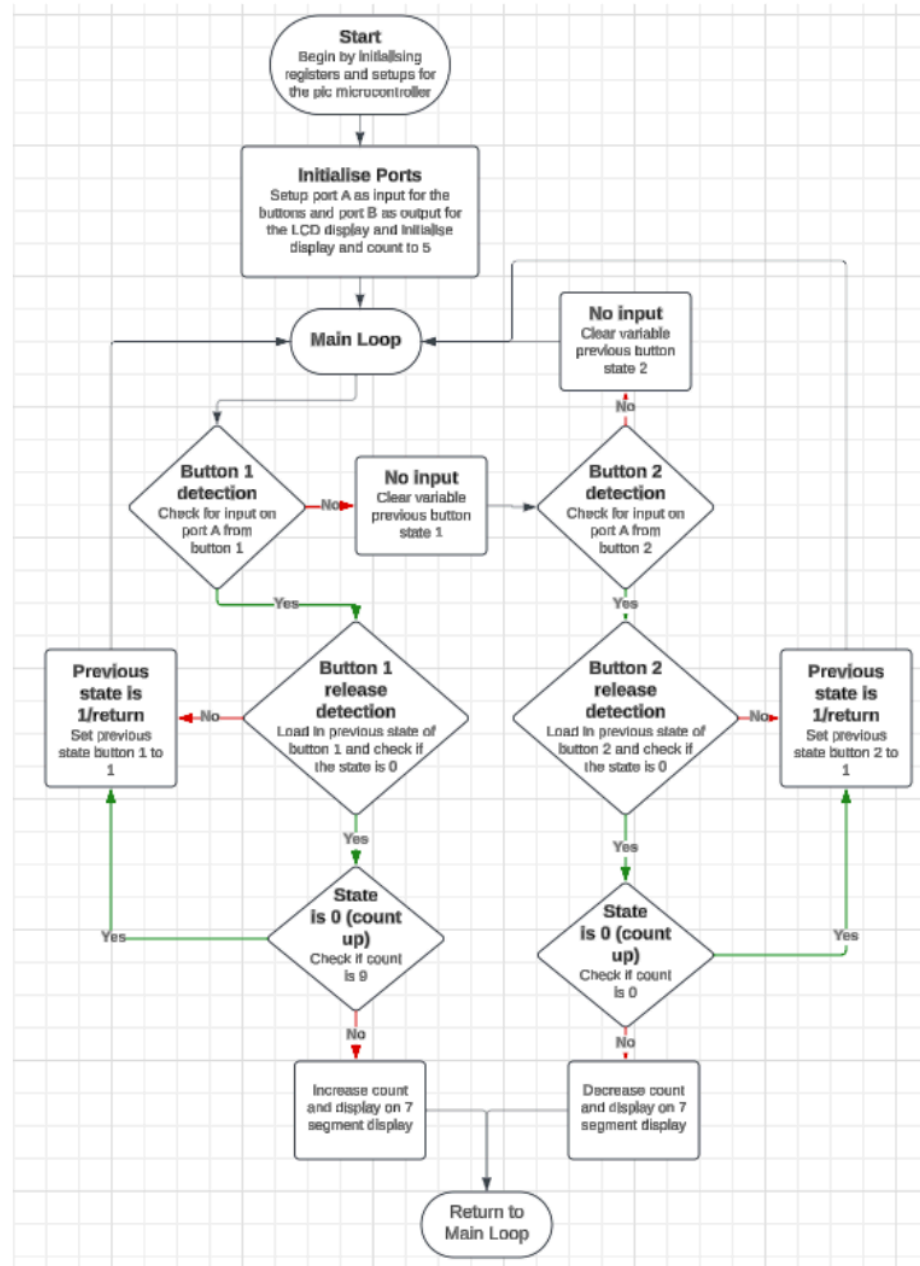


Figure 26 Display 0-9 Flowchart

9. Sound piezo when a button is pressed

9.1. Code

```
1  ;=====
2  ; Created by: Daniel Frankland
3  ; Processor: PIC16F84A
4  ; Compiler: MPASM (Proteus)
5  ;=====
6
7  ;=====
8  ; DEFINITIONS
9  ;=====
10
11 #include p16f84a.inc      ; Include register definition file
12
13 portB    equ 0x06         ;port B address  links to the data sheet
14 trisB    equ 0x86         ;trisB address in register bank 1
15
16 status   equ 0x03         ;status register
17 RP0      equ 5            ;bit 5 of status register
18 RP1      equ 6            ;bit 6 of status register
19
20 portA     equ 0x05         ;port A address links to the data sheet
21 trisA     equ 0x85         ;Tris A address in register bank
22
23 ;=====
24 ;Variables
25 ;=====
26 Delay EQU    0x20 ; Delay counter
27
28 ;=====
29 ;code segment
30 ;=====
31 org 0
32 goto Start    ; Jump to start
33
34 Start
35 bsf        status,RP0     ; Select Bank 1
36 movlw      0xFF           ; Load w with 1s
37 movwf      trisA          ; Set portA as input
38
39 ; Set PORTB as output for the spekaer and the display
40 clrf trisB                ; Set portB pins as output
41 bcf status, RP0           ; Switch back to bank 0
42
```

Figure 27 Sound piezo Code-1

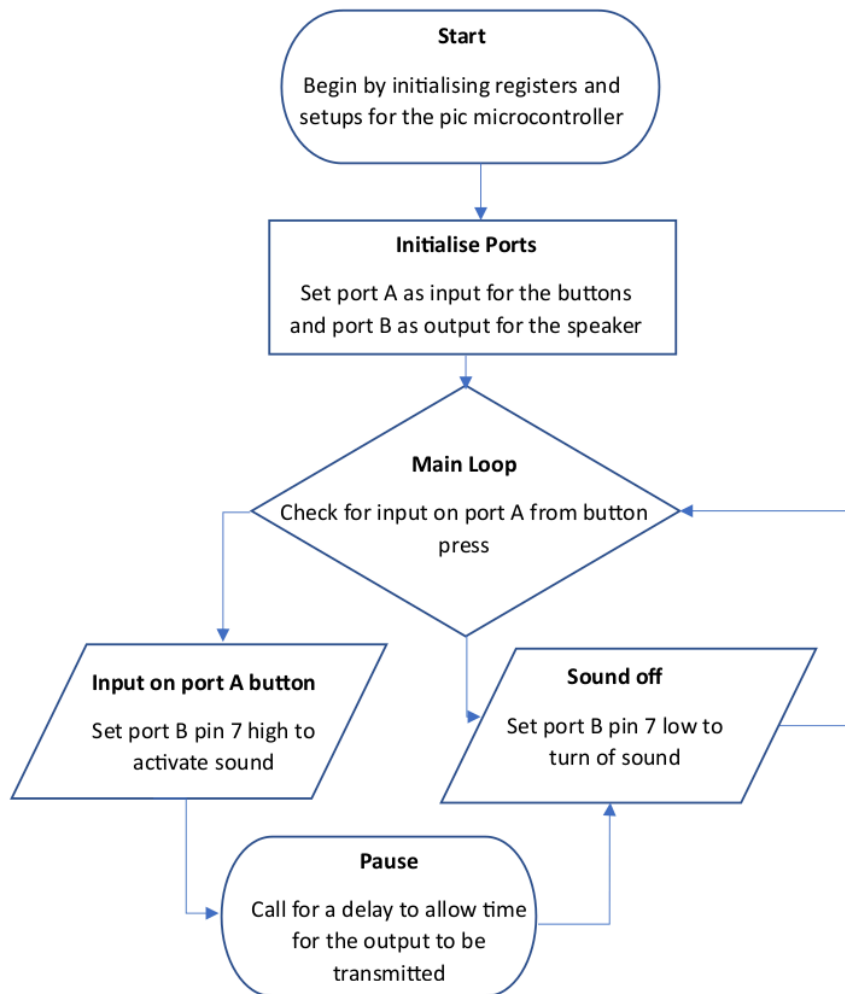
```

43 ;=====
44 ;main loop
45 ;=====
46 main_loop
47 movlw b'01111111' ;clear display
48 movwf portB ; move W to port B to update the display
49 btfsc portA,0 ; check if button is pressed
50 goto sound_off ; if button not pressed turn sound off
51 call sound_on ; if button pressed turn sound on
52 goto main_loop ; repeat loop
53
54 sound_on
55 bsf portB,7 ; Set speaker pin high
56 call pause ;call puase for a delay
57 bcf portB,7 ; Set speaker pin low
58 call pause ; Call pause for a delay
59 return
60
61 sound_off
62 bcf portB,7 ; turn speaker off
63 goto main_loop ; goes to main loop
64
65 ;=====
66 ;SUBROUTINES
67 ;=====
68 pause
69 movlw 0xFF ; Load delay counter
70 movwf Delay
71 delay_loop
72 decfsz Delay,F ; Decrement delay counter
73 goto delay_loop ; Repeat until zero
74 return
75
76 end
77

```

Figure 28 Sound piezo Code-2

9.2. Flowchart



10. References

- [1 W. H. Paul Horowitz, "Optoelectronics: emitters," in *The Art of Electronics*, 3rd ed., Cambridge University Press, 2015, p. 838.
- [2 "RS Components," 5 August 2025. [Online]. Available: https://uk.rs-online.com/web/content/discovery/ideas-and-advice/resistors-guide?srsId=AfmBOoobtJJCQARIKDCIBXs0Dm6vIEET6-Tnrg_OcJhNhxaSENXEQM-7. [Accessed 30 December 2024].
- [3 University of Birmingham, "Physics A level revision resource: Introduction to capacitors," [Online]. Available: <https://www.birmingham.ac.uk/study/undergraduate/schools-and-colleges/post-16/a-level-stem-resources/capacitors#:~:text=Unlike%20the%20battery%2C%20a%20capacitor,all%20the%20energy%20very%20quickly.> [Accessed 30 December 2024].
- [4 Z. Peterson, "Causes of Ripple Voltage: Power Supply Noise and Transients in Your PDN," North West Engineering Solutions, 29 October 2019. [Online]. Available: <https://www.nwengineeringllc.com/article/causes-of-ripple-voltage-power-supply-noise-and-transients-in-your-pdn.php>. [Accessed 30 December 2024].
- [5 M. Donnison, "How A Transistor Works," Kitronik, 15 January 2014. [Online]. Available: https://kitronik.co.uk/blogs/resources/how-a-transistor-works?srsId=AfmBOoqdAj870sGZ_uQkQG-fKUAQMqk3_CAUH52yMnmN52p7OX_yh-Vk. [Accessed 30 December 2024].
- [6 ROHM Semiconductor, "Light Emitting Diode. What are LED's and How Do They Work?," [Online]. Available: <https://www.rohm.com/electronics-basics/leds/what-are-leds#:~:text=Compared%20with%20conventional%20light%20sources,generation%20with%20little%20wasted%20electricity.> [Accessed 30 December 2024].
- [7 F. Wouterlood, "thesolaruniverse," 03 27 2020. [Online]. Available: <https://thesolaruniverse.wordpress.com/2020/03/27/eight-digit-7-segment-led-display-for-arduino-controlled-by-a-max7219-chip/>. [Accessed 11 December 2024].
- [8 TemplateLab, "TemplateLab," [Online]. Available: <https://templatelab.com/cover-page-templates/>. [Accessed 01 November 2023].

Micros Peripherals and Interfaces Report

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

70/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

GRADING FORM: GJ_2019

DANIEL FRANKLAND

348

ENGAGEMENT WITH LITERATURE



Reasonable level of EWL. Could have included even more sources for more marks here.

70

KNOWLEDGE AND UNDERSTANDING

 Good level of K&U demonstrated from the research. Very well explained through the use of diagrams. **75**

COGNITIVE AND INTELLECTUAL

 Rather a 'brute force' approach used for solving the programming, but adequate for a L4 module. **55**

PRACTICAL APPLICATION

 Electronics well implemented - all included. A reasonable attempt at the majority of the software as well, albeit as mentioned previously, some of the solutions are a bit 'clunky' and there seem to be a few logical errors in a small number of places (e.g, in 6.1 - no tris function on the display? Does the buzzer routine actually buzz as it should?), **68**

TRANSFERABLE SKILLS

 Very well written and presented. Excellent use of diagrams. **80**