



CSE 3104.2 / CSE 3134.2

MICROPROCESSORS AND EMBEDDED SYSTEMS

SPRING 2024

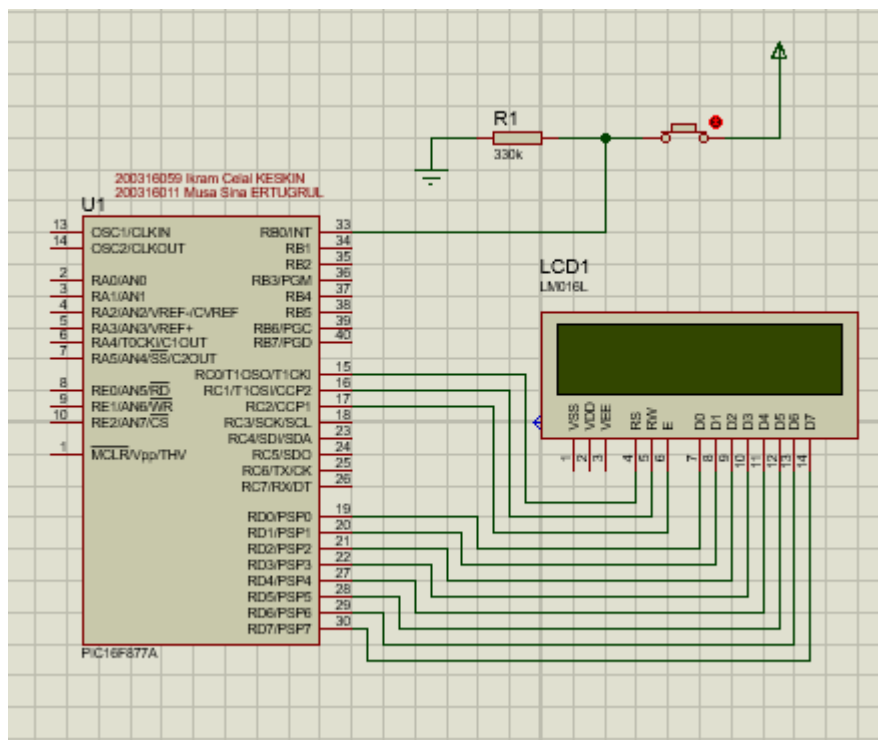
Lab Project Report

Ikram Celal KESKIN – 200316059

Musa Sina ERTUGRUL – 200316011

Submission Date: 31 May 2024

LAB 1



```

1 ;200316059 Ikram Celal ESKIN
2 ;200316011 Musa Sina ERTUGRUL
3
4 LIST P = PIC16F877A
5 #INCLUDE <P16F877A.INC>
6 __CONFIG H'3F31'
7
8 #define RS PORTC,0 ; Lcd Data/Command
9 #define RW PORTC,1 ; Read/Write Data
10 #define EN PORTC,2 ; LCD Enable
11
12 S1 EQU 0X20 ; Delay var
13 S2 EQU 0X21 ; Delay var
14 TEMP EQU 0X22 ; For Working Reg Back-up
15 CHAR EQU 0X23 ; For Counter Display next char;
16 CHAR2 EQU 0X24
17
18 ;First Address-----
19 ORG 0x00
20 START
21
22 CALL REG_INIT ; Call function to initialize registers
23 CALL LCD_INIT ; Call function to initialize LCD
24
25 MOVLW 0x80
26 CALL LCD_COMMAND ; Move cursor to the start of the first line
27 MOVLW 0X01
28 CALL LCD_COMMAND ; Clear LCD screen
29
30 POLL
31 BTFSS PORTB,0 ; Polling for a condition on PORTB, pin 0
32 GOTO POLL
33 CALL WRITE_COUNTER ; Call function to write counter value to LCD
34 CALL PRINT ; Call function to print counter value
35 MOVLW 0X02

```

```

36 CALL    LCD_COMMAND    ; Move cursor to the start of the second line
37 GOTO    POLL           ; Continue polling
38
39 WRITE_COUNTER
40 MOVLW    "C"
41 CALL    LCD_DATA        ; Write character 'C' to LCD
42 CALL    SHIFT_RIGHT     ; Shift cursor to the right
43 MOVLW    "o"
44 CALL    LCD_DATA        ; Write character 'o' to LCD
45 CALL    SHIFT_RIGHT     ; Shift cursor to the right
46 MOVLW    "u"
47 CALL    LCD_DATA        ; Write character 'u' to LCD
48 CALL    SHIFT_RIGHT     ; Shift cursor to the right
49 MOVLW    "n"
50 CALL    LCD_DATA        ; Write character 'n' to LCD
51 CALL    SHIFT_RIGHT     ; Shift cursor to the right
52 MOVLW    "t"
53 CALL    LCD_DATA        ; Write character 't' to LCD
54 CALL    SHIFT_RIGHT     ; Shift cursor to the right
55 MOVLW    "e"
56 CALL    LCD_DATA        ; Write character 'e' to LCD
57 CALL    SHIFT_RIGHT     ; Shift cursor to the right
58 MOVLW    "r"
59 CALL    LCD_DATA        ; Write character 'r' to LCD
60 CALL    SHIFT_RIGHT     ; Shift cursor to the right
61 MOVLW    ":"
62 CALL    LCD_DATA        ; Write character ':' to LCD
63 CALL    SHIFT_RIGHT     ; Shift cursor to the right
64 RETURN
65
66 SHIFT_LEFT
67 MOVLW    0X04
68 CALL    LCD_COMMAND     ; Command to shift cursor left
69 RETURN
70

```

```

71 SHIFT_RIGHT
72 MOVLW    0X06
73 CALL    LCD_COMMAND     ; Command to shift cursor right
74 RETURN
75
76 ;TRIS REG Initialization for PORT A B C -----
77 REG_INIT
78 BANKSEL TRISC
79 MOVLW    B'00000001'
80 MOVWF    TRISB          ; Set TRISB to configure as input
81 CLRF     TRISC          ; Clear TRISC
82 CLRF     TRISD          ; Clear TRISD
83 BANKSEL PORTD
84 CLRF     PORTD          ; Clear PORTD
85 CLRF     PORTB          ; Clear PORTB
86 CLRF     CHAR           ; Clear CHAR variable
87 RETURN
88
89 ;For Printing Char on LCD Display-----
90 PRINT
91 BANKSEL CHAR
92 BCF      STATUS,C
93 MOVLW    D'10'
94 SUBWF    CHAR2,W
95 BTFSS    STATUS,C
96 GOTO     PRINT2
97 CLRF     CHAR2
98 PRINT2
99 MOVLW    "0"
100 ADDWF    CHAR,0
101 CALL    LCD_DATA        ; Print character to LCD
102 CALL    SHIFT_RIGHT     ; Shift cursor to the right
103 MOVLW    "0"
104 ADDWF    CHAR2,0
105 CALL    LCD_DATA        ; Print character to LCD

```

```

106 CALL    SHIFT_RIGHT    ; Shift cursor to the right
107 INCF    CHAR2
108 BCF     STATUS,C
109 MOVLW   D'10'
110 SUBWF   CHAR2,W
111 BTFSS   STATUS,C
112 RETURN
113 INCF    CHAR
114 RETURN
115
116 ;For LCD Display Initialization-----
117 LCD_INIT
118 MOVLW   0X38            ; Function set: 8-bit mode, 2-line display, 5x7 font
119 CALL    LCD_COMMAND     ; Send command to LCD
120 MOVLW   0X06            ; Entry mode set: Increment cursor, no display shift
121 CALL    LCD_COMMAND     ; Send command to LCD
122 MOVLW   0X0E            ; Display control: Display on, cursor on, blink cursor off
123 CALL    LCD_COMMAND     ; Send command to LCD
124 MOVLW   0X01            ; Clear display
125 CALL    LCD_COMMAND     ; Send command to LCD
126 RETURN
127
128 ;-----
129 LCD_COMMAND
130 BCF     RS              ; Set RS pin low (command mode)
131 BCF     RW              ; Set RW pin low (write mode)
132 BSF     EN              ; Set EN pin high
133 MOVWF   TEMP            ; Move command to TEMP register
134 CALL    DELAY           ; Delay function
135 MOVFW   TEMP
136 BANKSEL PORTD
137 MOVWF   PORTD           ; Send command to PORTD
138 BCF     EN              ; Set EN pin low
139 RETURN
140

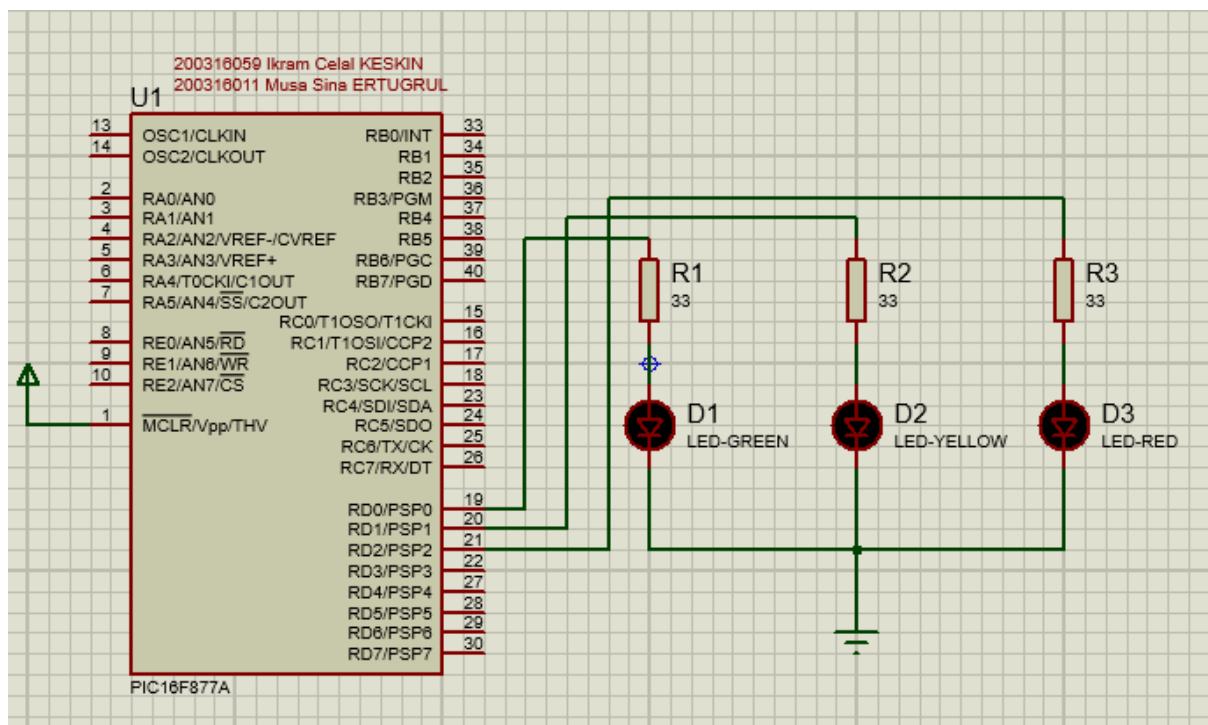
```

```

141 ;-----
142 LCD_DATA
143 BSF     RS              ; Set RS pin high (data mode)
144 BCF     RW              ; Set RW pin low (write mode)
145 BSF     EN              ; Set EN pin high
146 MOVWF   TEMP            ; Move data to TEMP register
147 CALL    DELAY           ; Delay function
148 MOVFW   TEMP
149 BANKSEL PORTD
150 MOVWF   PORTD           ; Send data to PORTD
151 BCF     EN              ; Set EN pin low
152 RETURN
153
154 ;-----FOR DELAY-----
155 DELAY
156 MOVLW   0XFF
157 MOVWF   S1
158 L1
159     MOVLW 0XFF
160     MOVWF S2
161     L2
162     DECFSZ S2
163     GOTO  L2
164     DECFSZ S1
165     GOTO  L1
166 RETURN
167
168 END

```

LAB 2



```

1 ;200316059 Ikram Celal KESKIN
2 ;200316011 Musa Sina ERTUGRUL
3
4
5 LIST P = PIC16F877A
6 #INCLUDE <P16F877A.INC>
7 ;__CONFIG H'3F31'
8
9 ;Var Def-----
10 T0Counter EQU 0X25
11 Itr2 EQU 0X26
12 Itr1 EQU 0X27
13 Itr3 EQU 0X28
14
15 ;First Address-----
16 ORG 0x00
17 GOTO start
18
19 ;Interrupt Address-----
20 ORG 0x04
21 BTFSS INTCON,TMR0IE
22 GOTO intRet
23 BTFSS INTCON,TMR0IF
24 GOTO intRet
25 MOVLW D'6' ;256-6 = 250 Step
26 MOVWF TMR0
27 BCF INTCON,TMR0IF ; INTCON Timer Flag Cleared
28 INCF T0Counter
29
30 intStart
31 MOVLW D'250' ;250 x 4000 us = 1000 000 us
32 SUBWF T0Counter,W
33 BTFSS STATUS,C ;T0Counter >= 250 ? IF then skip below instruction.
34 GOTO intRet ;Else Retrun from Interrupt
35 CLRF T0Counter ;IF T0 > 250 -> T0 = 0

```

```

36         BTFSS    PORTD,2
37         GOTO     threeSec
38         BTFSS    PORTD,1
39         GOTO     twoSec
40         GOTO     oneSec
41     threeSec
42         MOVLW    D'3'
43         INCF     Itr2
44         BCF      STATUS,C
45         SUBWF    Itr2,W
46         BTFSS    STATUS,C
47         GOTO     intRet
48         GOTO     checkOthers
49     twoSec
50         MOVLW    D'2'
51         INCF     Itr2
52         BCF      STATUS,C
53         SUBWF    Itr2,W
54         BTFSS    STATUS,C
55         GOTO     threeSec
56         BTFSS    PORTD,1
57         GOTO     setLedYellow
58         GOTO     checkOthers
59     oneSec
60         MOVLW    D'1'
61         INCF     Itr2
62         BCF      STATUS,C
63         SUBWF    Itr2,W
64         BTFSS    STATUS,C
65         GOTO     threeSec
66         GOTO     setLedGreen
67     checkOthers
68         MOVLW    D'1'
69         BCF      STATUS,C
70         SUBWF    Itr3,W

```

```

70         SUBWF    Itr3,W
71         BTFSS    STATUS,C
72         GOTO     setLedGreen
73         MOVLW    D'2'
74         BCF      STATUS,C
75         SUBWF    Itr3,W
76         BTFSS    STATUS,C           ;If 1. LED = 1 -> cont. to turn out LED1 &
77         GOTO     setLedYellow
78         MOVLW    D'3'
79         BCF      STATUS,C
80         SUBWF    Itr3,W
81         BTFSS    STATUS,C
82         GOTO     setLedRed
83         GOTO     intRet
84
85         ;
86     setLedGreen
87         BCF      PORTD,2
88         BCF      PORTD,1           ;Swap LED Conditions...
89         BSF      PORTD,0
90         CLRF     Itr2
91         CLRF     Itr1
92         INCF     Itr3
93         GOTO     intRet
94     setLedYellow
95         BCF      PORTD,0           ;Swap LED Conditions...
96         BSF      PORTD,1
97         CLRF     Itr2
98         CLRF     Itr1
99         BTFSS    PORTD,2
100        INCF     Itr3
101        GOTO     intRet
102     setLedRed
103        BCF      PORTD,0
104        BCF      PORTD,1           ;Swap LED Conditions...

```

```

105          BSF      PORTD,2
106          CLRF     Itr2
107          CLRF     Itr1
108          CLRF     Itr3
109          BCF      STATUS,C
110          GOTO     intRet
111
112 intRet
113          RETFIE
114
115 ;Main Programm-----
116 start    CLRF      T0Counter
117          BANKSEL  TRISD
118          CLRF     TRISD
119          BANKSEL  OPTION_REG
120          MOVLW    B'11010011'      ;OPT_REG 1:16  1 x 250 x 16 = 4000 us
121          MOVWF    OPTION_REG
122          BANKSEL  PORTD
123          CLRF     PORTD
124          MOVLW    D'6'
125          MOVWF    TMR0
126          BSF      INTCON,TMR0IE
127          BSF      INTCON,GIE
128          CLRF     Itr3
129          MOVLW    D'1'
130          MOVWF    Itr3
131          CLRF     Itr2
132          MOVLW    D'0'
133          MOVWF    Itr2
134          CLRF     Itr1
135          MOVLW    D'0'
136          MOVWF    Itr1
137          BSF      PORTD,0
138          BCF      PORTD,1
139          BCF      PORTD,2

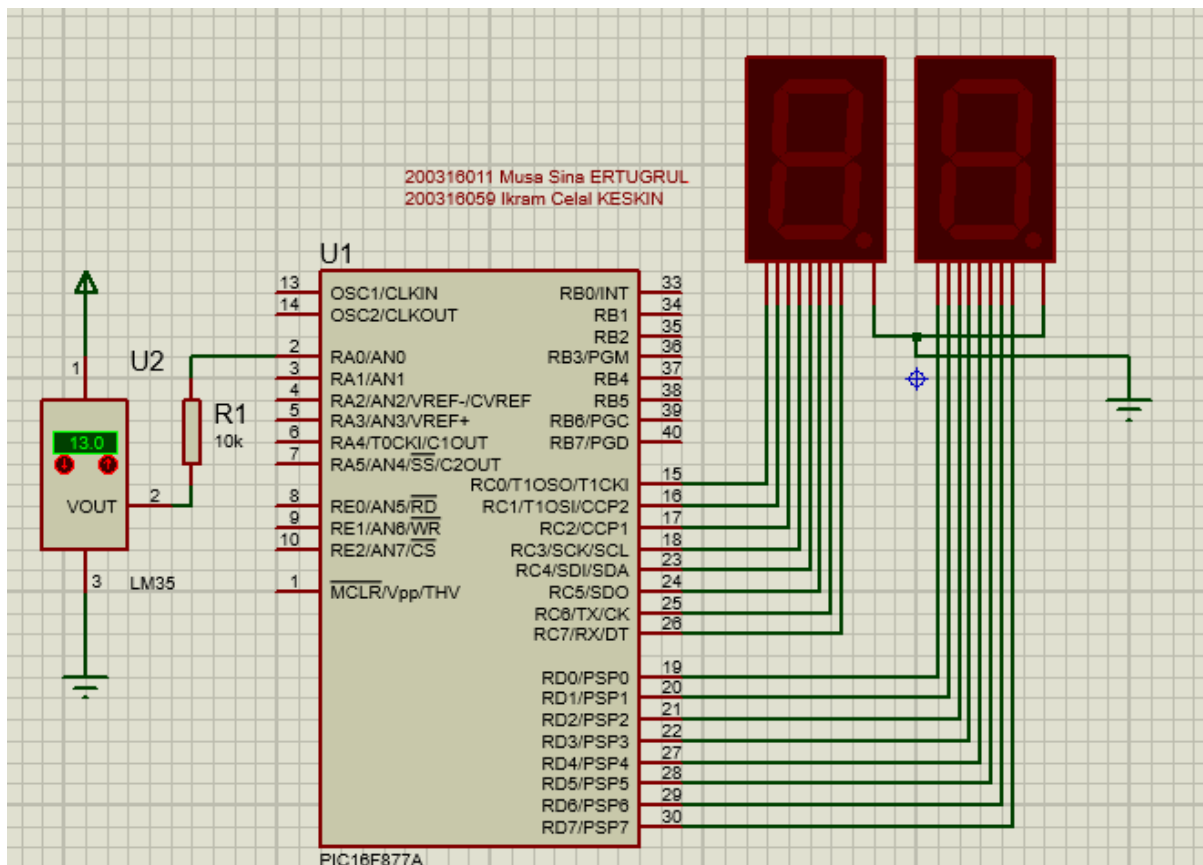
```

```

134          CLRF     Itr1
135          MOVLW    D'0'
136          MOVWF    Itr1
137          BSF      PORTD,0
138          BCF      PORTD,1
139          BCF      PORTD,2
140
141 loop     GOTO     loop
142
143 END

```

LAB3



```

1 ;200316059 Ikram Celal KESKIN
2 ;200316011 Musa Sina ERTUGRUL
3 LIST P=16F877A           ; Define target microcontroller
4 #INCLUDE <P16F877A.INC>  ; Include necessary header file
5 __CONFIG H'3F31'         ; Set configuration bits
6
7 DEG      EQU 0x20        ; Define memory location for degrees
8 decimal  EQU 0x21        ; Define memory location for decimal value
9 tens     EQU 0x22        ; Define memory location for tens place
10 ones    EQU 0x23        ; Define memory location for ones place
11
12 ORG      0x00            ; Program start address
13
14 GOTO     start           ; Jump to start of program
15
16 start:
17     MOVLW 0xFF           ; Load W register with 0xFF
18     BANKSEL TRISA        ; Select bank for TRISA register
19     MOVWF TRISA          ; Set all PORTA pins as input (ADC input)
20     CLRF  TRISB          ; Set all PORTB pins as output
21     CLRF  TRISC          ; Set all PORTC pins as output
22     CLRF  TRISD          ; Set all PORTD pins as output
23     BANKSEL PORTA        ; Select bank for PORTA register
24     CLRF  PORTB          ; Clear PORTB
25     CLRF  PORTC          ; Clear PORTC
26     CLRF  PORTD          ; Clear PORTD
27     MOVLW B'01000001'    ; Load W register with binary value 01000001
28     MOVWF ADCON0         ; Configure ADCON0: ADC Clock Fosc/8, ADON=1
29     BANKSEL ADCON1       ; Select bank for ADCON1 register
30     MOVLW B'10000000'    ; Load W register with binary value 10000000
31     MOVWF ADCON1         ; Configure ADCON1: Right justified, VREF=VDD
32
33 loop:
34     CALL  ReadADC        ; Call subroutine to read ADC
35     BANKSEL PORTB        ; Select bank for PORTB register

```



```

35     BANKSEL PORTB      ; Select bank for PORTB register
36     MOVFW  ADRESH      ; Move high byte of ADC result to W
37     MOVWF  PORTB       ; Move ADC result to PORTB
38     CALL   Degree      ; Call subroutine to process degree
39     GOTO   loop        ; Repeat loop
40
41 ;-----
42
43 ReadADC:
44     BANKSEL ADCON0      ; Select bank for ADCON0 register
45     BSF     ADCON0, GO   ; Start ADC conversion
46 read:
47     BTFSC  ADCON0, GO   ; Wait for ADC conversion to complete
48     GOTO   read
49
50     BANKSEL ADRESH      ; Select bank for ADRESH register
51     MOVFW  ADRESH      ; Move high byte of ADC result to W
52     RETURN              ; Return from subroutine
53
54 ;-----
55 Degree:
56     MOVFW  ADRESH      ; Move high byte of ADC result to W
57     MOVWF  DEG         ; Store in DEG memory location
58     MOVLW  0x0A        ; Load W register with value 0x0A (10 in decimal)
59     CALL   TensOnesDeclare ; Call subroutine to process tens and ones place
60     RETURN              ; Return from subroutine
61
62 ;-----
63 TensOnesDeclare:
64     BANKSEL DEG         ; Select bank for DEG register
65     MOVF   DEG, W       ; Move value of DEG to W
66     MOVWF  decimal      ; Store in decimal memory location
67     CLRF   tens         ; Clear tens place
68     CLRF   ones         ; Clear ones place
69     RETURN
70
71 countTens:
72     SUBLW  10           ; Subtract 10 from W
73     BTFSS  STATUS, C    ; If result is negative, end loop
74     GOTO   doneTens
75     INCF   tens, F      ; Increment tens place
76     MOVWF  decimal      ; Update decimal value
77     GOTO   countTens    ; Repeat loop
78     RETURN
79
80 doneTens:
81     MOVF   decimal, W   ; Move value of decimal to W
82     MOVWF  ones         ; Store in ones place
83     RETURN
84
85 showSevenSeg:
86     MOVF   tens, W      ; Move value of tens to W
87     CALL   table        ; Lookup seven segment display table
88     BANKSEL PORTC       ; Select bank for PORTC register
89     MOVWF  PORTC        ; Display tens on PORTC
90     MOVF   ones, W      ; Move value of ones to W
91     CALL   table        ; Lookup seven segment display table
92     BANKSEL PORTD       ; Select bank for PORTD register
93     MOVWF  PORTD        ; Display ones on PORTD
94     RETURN              ; Return from subroutine
95
96 ;-----
97 table:
98     addwf  PCL           ; Add W register to PCL
99     retlw  B'11111100'   ; Lookup table entry for 0
100    retlw  B'01100000'   ; Lookup table entry for 1
101    retlw  B'11011010'   ; Lookup table entry for 2
102    retlw  B'11110010'   ; Lookup table entry for 3
103    retlw  B'01100110'   ; Lookup table entry for 4
104    retlw  B'10110110'   ; Lookup table entry for 5
105    retlw  B'10111110'   ; Lookup table entry for 6

```

```

96 ;-----
97 table:
98     addwf    PCL           ; Add W register to PCL
99     retlw    B'11111100'   ; Lookup table entry for 0
100    retlw    B'01100000'   ; Lookup table entry for 1
101    retlw    B'11011010'   ; Lookup table entry for 2
102    retlw    B'11110010'   ; Lookup table entry for 3
103    retlw    B'01100110'   ; Lookup table entry for 4
104    retlw    B'10110110'   ; Lookup table entry for 5
105    retlw    B'10111110'   ; Lookup table entry for 6
106    retlw    B'11100000'   ; Lookup table entry for 7
107    retlw    B'11111110'   ; Lookup table entry for 8
108    retlw    B'11110110'   ; Lookup table entry for 9
109
110    END                    ; End of program
111
112 ; Read ADC Value
113 ; Store Adc value on DEG
114 ; Subtract 10 and Increase the 'tens' variable by 1 until it throws the negative flag
115 ; Writes the remaining value to the ones variable
116 ; Shows the related value in 7 segments from the related port (using table)

```