

# PIC Exercises 2014 Autumn

## Exercise 1:

To set the 4 lower pins of PORTA into input mode we use the Binary value B'00001111' or 0x0F. We use and ANDLW command to discard the most significant bits.

```
list p=16f877A
#include <p16f877A.inc>
__CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_ON & _HS_OSC &
_WRT_OFF & _LVP_OFF & _CPD_OFF

ORG 0x00
RESET GOTO      START

ORG 0x05
START BSF       STATUS,RP0 ; Move to Bank 1
      MOVLW     0x07        ; Load W with 0x07
      MOVWF     ADCON1      ; Set PORTA into digital mode
      MOVLW     B'00001111' ; Load W with a value
      MOVWF     TRISA       ; Set PORTA to input mode
      CLRF      TRISB       ; Set PORTB to output mode
      BCF       STATUS,RP0 ; Move to Bank 0

      MOVF      PORTA,W     ; Copy PORTA to W
      ADDLW     .5          ; Add 5 to W
      ANDLW     B'00001111' ; Discard the four most significant bits
      MOVWF     PORTB       ; Write W to PORTB

      SLEEP
      GOTO      $-1

END
```

## Exercise 2:

We use the C bit of the STATUS register to determine whether 5 is larger than 10.

on SUBLW, C will be 0 (borrowed) if the subtraction resulted in a negative number or a 1 (not borrowed) if the subtraction resulted in a positive number.

```

list p=16f877A
#include <p16f877A.inc>
__CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_ON & _HS_OSC &
_WRT_OFF & _LVP_OFF & _CPD_OFF

RESET      ORG 0x00
           GOTO    START

START      ORG 0x05
           CLRF    PORTB      ; Turn off all LEDs attached to PORTB
           BSF     STATUS,RP0  ; Move to Bank 1
           CLRF    TRISB      ; Set PORTB to output mode
           BCF     STATUS,RP0  ; Move to Bank 0

           MOVLW   .10        ; Load W with 10
           SUBLW   .5          ; Subtract W from 5
           BTFSC   STATUS,C    ; Skip on borrow
           GOTO    FINISH      ; Jump to FINISH
           BSF     PORTB,RB0   ; Turn on LED attached to RB0

FINISH     SLEEP
           GOTO    $-1

           END

```

## Exercise 3:

The initial values for CountLow and CountHigh are '237' and '65' respectively.

The Mainloop section of the program takes 6 cycles, one for each instruction before reaching delay.

$$\text{MainloopCycles} = 6$$

The first run of 'Delay' takes three cycles (1xDECFSZ+2xGOTO), and is run 236 times, (237-1). On the 237<sup>th</sup> run of the loop, DECFSZ skips, taking 2 cycles.

$$\text{CountLowFirstCycles} = (237 - 1) \times 3 + 2 = 710$$

CountHigh is decremented and then a GOTO causes us to start decrementing CountLow again, this takes 3 cycles. Now, CountLow has a value of 0, DECFSZ will cause it to overflow, meaning the number of times it will be decremented before skipping is 255.

$$\text{CountLowCycles} = ((256 - 1) \times 3 + 2) + 3 = 770$$

This will run 64 times, on the 65<sup>th</sup> run, the CountHigh loop will skip, returning to Mainloop, this skip takes 4 cycles (2xDECFSZ and 2xGOTO).

$$\text{CountHighCycles} = 770 \times 64 + 4 = 49584$$

The final number of cycles taken is:

$$\text{Cycles} = 710 + 49284 + 6 = 50000$$

Each cycle takes 200us.

$$50000 \times 200 \text{ us} = 10 \text{ ms}$$

Which gives us an LED which flashes every 10 milliseconds.

```

list p=16f877A
#include <p16f877A.inc>
__CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_ON & _HS_OSC &
_WRT_OFF & _LVP_OFF & _CPD_OFF

; Define Memory Locations Names
CountHigh EQU 0x70
CountLow  EQU 0x71

RESET_V    ORG 0x00
           GOTO    START

START      ORG 0x05
           CLRF    PORTB      ; Clear PORTB
           BSF     STATUS,RP0 ; Move to Bank 1
           CLRF    TRISB      ; Set PORTB to output
           BCF     STATUS,RP0 ; Move to Bank 0

MainLoop   MOVLW    .1         ; Load W with decimal 1
           XORWF   PORTB      ; Toggle PORTB, RB0

           MOVLW    .237       ; Load W with a value for CountLow
           MOVWF   CountLow
           MOVLW    .65        ; Load W with a value for CountHigh
           MOVWF   CountHigh

Delay      DECFSZ   CountLow    ; Decrease CountLow, skip if zero
           GOTO    Delay
           DECFSZ   CountHigh  ; Decrease CountHigh, skip if zero
           GOTO    Delay
           GOTO    MainLoop

END

```

## Exercise 4:

A PCL lookup table lets us branch execution depending on the value of the working register. This is useful when we need to do very different operations for each state of a particular input.

```

list p=16f877A
#include <p16f877A.inc>
__CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_ON & _HS_OSC &
_WRT_OFF & _LVP_OFF & _CPD_OFF

RESET      ORG 0x00
           GOTO      START

START      ORG 0x05
           CLRF      PORTB      ; Turn all PORTB LEDs off
           BSF      STATUS,RP0  ; Move to Bank 1
           CLRF      TRISB      ; Set PORTB to output mode
           BCF      STATUS,RP0  ; Move to Bank 0

           MOVLW     .3         ; Move 3 into W register
           ADDWF     PCL,F      ; Put W+PCL in PCL
           GOTO      Case0     ; Jump to Case0
           GOTO      Case1     ; Jump to Case1
           GOTO      Case2     ; Jump to Case2
           GOTO      Case3     ; Jump to Case3

Case0      MOVLW     B'00000000' ; Light 0 LEDs
           GOTO      FINISH

Case1      MOVLW     B'00000001' ; Light 1 LED
           GOTO      FINISH

Case2      MOVLW     B'00000011' ; Light 2 LEDs
           GOTO      FINISH

Case3      MOVLW     B'00000111' ; Light 3 LEDs
           GOTO      FINISH

FINISH     MOVWF     PORTB      ; Write the number in W to PORTB

           SLEEP
           GOTO      $-1

           END

```

## Exercise 5:

We should be very familiar with the setup code for the PIC.

There are two ways to return from a subroutine. We can simply use RETURN and the value in W will remain when the CALL ends, or we can use RETLW, which will load a value into W and end the CALL in one operation.

Here I have used CLRF and COMF to set PORTA to 0xFF (really, 0x3F as the two most significant bits of PORTA are unused and always 0). You can just as easily use MOVLW 0xFF, MOVWF PORTA.

```

list p=16f877A
#include <p16f877A.inc>
__CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_ON & _HS_OSC &
_WRT_OFF & _LVP_OFF & _CPD_OFF

RESET_V      ORG 0x00
              GOTO  START

START        ORG 0x05
              CALL      Setup      ; Call the Setup Routine

              MOVLW     .5         ; Put Decimal 5 in W
              CALL      Aplus      ; Call Aplus

              MOVLW     .7         ; Put Decimal 7 in W
              CALL      Aplus      ; Call Aplus

              SLEEP
              GOTO      $-1

;SUBROUTINE: Setup
Setup        BSF        STATUS,RP0 ; Goto bank1
              MOVLW     0x07       ; Put 0x07 in W
              MOVWF     ADCON1     ; Set PORTA to Digital I/O Mode
              CLRF      TRISA      ; Set all PORTA Pins to input
              COMF      TRISA,F    ; Set all PORTA Pins to input
              CLRF      TRISB      ; Set all PORTB Pins to Output
              BCF        STATUS,RP0 ; Goto Bank0
              RETURN              ; Return to where CALL occurred

;SUBROUTINE: Aplus
Aplus        ADDWF      PORTA,W    ; Add PORTA to W
              MOVWF     PORTB      ; Write W to PORTB
              RETURN              ; Return to where CALL occurred

END

```