Project 1 program, is a code designed with MPLAB X IDE v5.40, with the pic-as compiler v2.30, for the PIC18F452 microcontroller. It's configured to use a 4MHz crystal oscillator, to achieve a 1MHz instruction frequency. PORTB and PORTC were chosen because they are easier to configure, PORTA wasn't chosen because several of his pins are analog inputs and that means a little bit more code lines to achieve his correct configuration. PORTB is the output and bits 0 and 1 from PORTC are inputs. PORTC's bit 0 is to change the sequence and bit 1 is for reverse the direction. A 0.5 seconds delay is implemented on code using the Timer0 to make possible to see the sequences and the direction changes due to the program execution speed, it's too high and we need to introduce the delay to see changes on the output. This delay can be done with an iterative loop and some variables that can be decremented to wait until they become zero, introducing a delay meanwhile, but the delay with Timer0 is more accurate, that's why was chosen. The algorithm executed by code is as similar as the statement, start showing a pattern on PORTB, if a logical 1 is detected on bit 0 of PORTC, then shows the next pattern on PORTB and so on, if a logical 1 is detected on bit 1 of PORTC, the direction of the pattern is reversed. Before change the output the delay is executed.

The code effectiveness is good, it uses as less instructions as were possible, resulting in less execution time added to the fact that the objects of the task were achieved.

```
        #include "ECE332_assembly_includes_00.inc"
        #include <xc.inc>

        PSECT           udata_acs
        GLOBAL          flags
flags:  DS      1       ; place for value
                        ; flags(0): sequence change
                        ; flags(1): reverse direction
                        ; flags(2): left(0) or right(1) direction
                        ; flags(3): off pattern
                        ; flags(4): pattern1 in use
                        ; flags(5): pattern2 in use
                        ; flags(6): pattern3 in use
        PSECT resetVec,class=CODE,reloc=2
resetVec:       goto    main    ; goto entry
        PSECT           code
main:
    setf    BSR,a               ; select bank 14
    movlw   0b00000111          ; move 00000111 to w register
    movwf   T0CON,a             ; configures Timer0
                        ; Timer0 stopped for now, configured as a 16 bit timer
                        ; internal instruction cycle selected
                        ; prescaler assigned to Timer0, his clock input comes from prescaler output
    movlw   0xF8         ; move F8 to w register
    movwf   TMR0H,a          ; move F8 to TMR0H for a 0.5s delay
    movlw   0x5E         ; move 5E to w register
    movwf   TMR0L,a          ; move 5E to TMR0L register for a 0.5s delay
    clrf    TRISB,a      ; clear all TRISB bits, configuring all PORTB pins as outputs
    movlw   0x01         ; copy value 0x01 in w register
    movwf   PORTB,a          ; move the value on w register to PORTB
    bsf     flags,4,a    ; indicates pattern1 is in use
    bsf     T0CON,7,a        ; enables Timer0
    movlw   0b00000011       ; move 00000011 to w register
    movwf   TRISC,a      ; configures bits 0 and 1 as inputs of PORTC
                        ; bit 0 for change sequence, bit 1 for reverse direction
loop:call   delay       ; delay to make possible to see the sequences
    btfss   flags,1,a    ; if reverse direction flag is not 1
    goto    l3           ; go to l3 label to continue, else...
rev:bcf            flags,1,a    ; clear flag before reverse
    btfsc   flags,2,a    ; if sequence is not rotating to the right(1), skip next line
    goto    l5           ; if rotating to the right go to l5 label
    rrncf   PORTB,f,a    ; right rotate
```

```
    bsf      flags,2,a    ; set the flags(2) to indicate rotation to the  right(1)

    goto     l4           ; go to label l4 to continue

l5: rlncf    PORTB,f,a          ; rotate to left

    bcf      flags,2,a    ; clear flags(2) to indicate left(0) rotation

    goto     l4           ; go to l4 label to continue

l3: btfss    flags,2,a    ; if sequence is left(0) rotating

    rlncf    PORTB,f,a    ; keep left rotating

    btfsc    flags,2,a    ; if sequence is right(1) rotating

    rrncf    PORTB,f,a    ; keep rotating to the right


l4: btfss    flags,0,a    ; if sequence change flag is 0

    goto     loop         ; go to loop label, else:

    btfsc    flags,3,a    ; if off pattern is in use

    goto     l7           ; go to l7 label

    btfsc    flags,4,a    ; if pattern1 is in use

    goto     l8           ; go to l8 label

    btfsc    flags,5,a    ; if pattern2 is in use

    goto     l9           ; go to l9 label

    btfsc    flags,6,a    ; if pattern3 is in use

    goto     l10          ; go to l10 label

    goto     loop         ; repeat all again
```

```
l7: bcf     flags,3,a    ; if off pattern is in use, first clear his flag
    bsf     flags,4,a    ; set the pattern1 flag to indicate its been used
    bcf     flags,0,a    ; clear the sequence change flag
    movlw   0x01         ; copy value 0x01 in w register (pattern1)
    movwf   PORTB,a             ; move the value on w register to PORTB
    goto    loop         ; repeat all again
l8: bcf     flags,4,a    ; if pattern1 is in use, first clear his flag
    bsf     flags,5,a    ; set the pattern2 flag to indicate its been used
    bcf     flags,0,a    ; clear the sequence change flag
    rlncf   PORTB,w,a           ; creating the pattern2
    iorwf   PORTB,f,a           ; pattern2 created
    goto    loop         ; repeat all again
l9: bcf     flags,5,a    ; if pattern2 is in use, first clear his flag
    bsf     flags,6,a    ; set the pattern3 flag to indicate its been used
    bcf     flags,0,a    ; clear the sequence change flag
    rlncf   PORTB,w,a           ; creating the pattern3
    iorwf   PORTB,f,a           ; pattern3 created
    goto    loop         ; repeat all again
l10:bcf     flags,6,a    ; if pattern3 is in use, first clear his flag
    bsf     flags,3,a    ; set the off pattern flag to indicate its been used
    bcf      flags,0,a    ; clear the sequence change flag

    clrf    PORTB,a    ; off pattern

l6: goto loop             ; repeat all again

////////////////////////////////////////////////////////////////////////////////

delay:btfsc PORTC,0,a      ; if change sequence button is pressed

    bsf      flags,0,a    ; set flags(0) to indicate that sequence must be changed

    btfsc   PORTC,1,a          ; if reverse direction button is pressed

    bsf      flags,1,a    ; set flags(1) to indicate that direction must be reversed

    btfss INTCON,2,a          ; if the interruption flag of Timer0 is set, 0.5s delay is complete

    goto    delay        ; wait for Timer0 overflow

    bcf      INTCON,2,a        ; clear the Timer0 interrupt flag bit

    movlw   0xF8       ; move F8 to w register

    movwf   TMR0H,a          ; move F8 to TMR0H for a 0.5s delay

    movlw   0x5E       ; move 5E to w register

    movwf   TMR0L,a          ; move 5E to TMR0L register for a 0.5s delay

    return                  ; delay end

////////////////////////////////////////////////////////////////////////////////

; TMR0H and TMR0L values calculation

;          Fosc=4MHz, Fcpu=1MHz, prescaler 1:256,

;     Ftimer=1MHz/256=3906.25Hz

;     Ttimer=1/3906.25Hz=0.000256=256us
;    number of count for 0.5s delay: 0.5/0.000256=1953.125=7A1h

;    value for Timer0 16 bits register= FFFFh-07A1h=F85E

;    TMR0H=F8, TMR0L=5E


        END   resetVec        ; start address
```