

```
list P=PIC16F690
```

```
#include "p16F690.inc"
```

```
__config (_CP_OFF & _WDT_OFF & _PWRTE_ON & _HS_OSC)
```

```
;
; PORT DIAGRAM
;*****
; 1 +5V      VDD      VSS      GND      20
; 2 OSC1     PORT A5  PORT A0      19
; 3 OSC2     PORT A4  PORT A1
; 4          PORT A3  PORT A2      17
; 5          PORT C5  PORT C0  QueryLine 16
; 6          PORT C4  PORT C1  To XBeePic 15
; 7          PORT C3  PORT C2      14
; 8          PORT C6  PORT B4      13
; 9          PORT C7  PORT B5  EUSART-RX 12
; 10 EUSART-TX PORT B7  PORT B6      11
;*****

;
; variable definitions
;-

#define BAUD_RATE_GEN d'32' ;9600 baud rate with sync, brgh and brgl6
=0
#define PORTC_Dir b'00000000' ;direction register for PORTC
#define PORTB_Dir b'00100000' ;direction register for PORTB
#define Option_RInit b'11010111' ;Initialize Option Register for Timer0,
With internal clock and Presecale of 256
#define Start_Byte 0x02 ;Start byte received from Balloon Monitor
#define Balloon_Pop_CK 0x53 ;Check sum received if Balloon Popped

Counter equ 0x20 ;Count number of bytes received
Xr equ 0x21 ;Register to store Xored bytes
RECEIVED_MSG equ 0x22
W_TEMP equ 0x23
STATUS_TEMP equ 0x24
CK equ 0x25 ;Mihi- error while debug
; *****
; Program Space starts
; *****

org 0
goto Main
org 4
goto ISR ;ISR fired at PC = 4
org 5

; *****
; Start of Main Program
; *****

Main:
call Initialize

Loop:
btfsc INTCON,T0IF ;Check Timer 0 flag, set to flag at 13ms
interval for 20Mhz Fosc
```

```

goto    Timer0_Set
goto    Loop

```

```
;;-----Initialize various ports and subsystems-----
```

```
Initialize:
```

```

BANKSEL    ANSEL                ;clear ANSEL and ANSELH registers
CLRF      ANSEL
CLRF      ANSELH

BANKSEL    PORTC
CLRF      PORTC                ;clear PORTC

BANKSEL    TRISC

MOVLW     PORTC_Dir            ;set up TRIS registers for port C
MOVWF     TRISC
MOVLW     PORTB_Dir           ;set up TRIS registers for port B
MOVWF     TRISB

banksel   Counter
CLRF      Counter            ;Initialize Counter to 0
movlw     Start_Byte         ;Initialize Xr to 0x02 (Start Byte)
movwf     Xr

```

```
Init_Timer0:
```

```

banksel   TMR0
movlw     0x01                ;2s complement of FF
movwf     TMR0                ;Write max value in TMR0
movlw     Option_RInit
banksel   OPTION_REG
movwf     OPTION_REG         ;Initialize Timer with internal clock and
Presecale of 256

```

```
RXInit:
```

```

banksel   TXSTA
movlw     BAUD_RATE_GEN       ;Set baud rate to 9600
movwf     SPBRG
bcf       TXSTA,SYNC         ;disable SYNC bit, enable SPEN
banksel   RCSTA
bsf       RCSTA,SPEN
bsf       INTCON,GIE         ;Inable peripheral, global and recieve
interrupt
bsf       INTCON,PEIE
banksel   PIE1
bsf       PIE1,RCIE
banksel   RCSTA
bsf       RCSTA,CREN         ;enable continous receive enable
return

```

```
;;-----Interrupt service routine-----
```

```
ISR:
```

```
BeginISR:
```

```

movwf     W_TEMP              ;Interrupt begin routine
swapf    STATUS, W           ;save working and status registers

```

```

        movwf    STATUS_TEMP

ISRs:

        movf     RCREG,w           ;read receive register
        movwf    RECEIVED_MSG     ;store read value in RECIEVED_MSG
        incf     Counter,f        ;increment Counter
        btfsc   Counter,3        ;check if counter reached 8
        goto    Chk_Msg          ;if yes, Check Msg
        goto    Xoring           ;else, keep Xoring

EndISR:                                ;Interrupt end routine

        swapf   STATUS_TEMP,W     ;retrieve working and status registers
        movwf   STATUS
        swapf   W_TEMP,F
        swapf   W_TEMP,W
        retfie                                ;return from interrupt

;;-----Low Level Functions-----
Timer0_Set:

        bcf     INTCON,T0IF      ;Clear Timer 0 flag
        btfsc  PORTC,RC0        ;If RC0 is high, Pull it down
        goto   Pull_Line_Down
        goto   Pull_Line_Up     ;else, pull it up

Pull_Line_Down:

        bcf     PORTC,RC0       ;pull RC0 low
        clrf   Counter         ;Reset counter to 0
        movlw  Start_Byte      ;move 0x02 to Xr so that when it is Xored
        with 0x02, result is 0
        movwf  Xr
        goto   Loop

Pull_Line_Up:

        bsf    PORTC,RC0       ;pull RC0 high
        goto   Loop

Xoring:

        movf    RECEIVED_MSG,W
        movwf   CK              ;store msg received in CK (to be later
        checked against Balloon_PoP_CK)
        movf    RECEIVED_MSG,W
        xorwf   Xr,F            ;move msg received to W ; mihi
        ;Xor Byte recieved with value in Xr
        goto   EndISR          ;0x02 would xor with 0x02 to give 0, Xor
        with next byte would give the byte itself

Chk_Msg:

        movf    Xr,F            ;move Xr to itself
        btfsc  STATUS,Z        ;If value in Xr is 0, Z will be set
        goto   CheckSum_Matched ;if Z set then CheckSum matched
        goto   EndISR

CheckSum_Matched:

        movlw  Balloon_Pop_CK
        xorwf  CK,W             ;check CK against Balloon_PoP_CK ; mihi

```

```
    btfsc    STATUS,Z           ;if equal, balloon is popped
    goto    Balloon_Popped
    goto    EndISR
```

Balloon\_Popped:

```
    bsf     PORTC,RC1          ;Set RC1 line high to denote Balloon Popped
    bcf     INTCON,GIE         ;No further interrupts/communication
    required from balloon monitor
    goto    EndISR
```

End

```
;;-----END OF PROGRAM-----
```