University of Jordan
School of Engineering
Department of Mechatronics Engineering
Microprocessor and Microcontroller Laboratory
0908432
Exp. 9: Serial Communication

Objectives

1- To become familiar with the use of serial communications through the USART.
2- To demonstrate methods of remote control using serial communications.
3- To use the debugging facility of the MPLAB IDE to fix program bugs.

Pre-lab Preparation:

1- Review the sections in the book regarding the USART.
2- Read the PIC16F877A data sheet especially chapter 10.
3- Review the instruction set of the PIC 16F877A.
4- Read the assembly programs **carefully** and try to understand the operation and the settings used.

Procedure:

we are going to use the USART of the PIC to receive a character from the PC and return (send) next character to PC again. The communication is done using the RS232 protocol by utilizing the TTL to RS232 converter IC MAX202 on the board. You will need to use a communications program on the PC to monitor the data sent by the PIC.

Exercise: -

-Modify the code to receive a number and show the next one on 7-Segments and sent it to the PC again.

-Modify the code to classify the characters into 9 groups (1-9) as following and show the number of group on 7-Segments

Group	1	2	3	4	5	6	7	8	9
Character	A, B, C	D, E, F	G, H, I	J, K, L	M, N, O	P, Q, R	S, T, U	V, W, X	Y, Z

```
;*************************************************************
; This program to receive a character from the PC and return (send) next character to PC again.
;*************************************************************

        include "p16f877A.inc"

    __CONFIG    _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_OFF & _XT_OSC

;*************************************************************
; User-defined variables

        cblock 0x20
                WTemp                   ; Must be reserved in all banks
                StatusTemp
                Counter
        endc

        cblock 0x0A0            ; bank 1 assignnments
                WTemp1                  ; bank 1 WTemp
        endc

        cblock 0x120            ; bank 2 assignnments
                WTemp2                  ; bank 2 WTemp
        endc

        cblock 0x1A0            ; bank 3 assignnments
                WTemp3                  ; bank 3 WTemp
        endc
;*************************************************************
; Macro Assignments

push    macro
        movwf       WTemp       ;WTemp must be reserved in all banks
        swapf       STATUS,W    ;store in W without affecting status bits
        banksel     StatusTemp  ;select StatusTemp bank
        movwf       StatusTemp  ;save STATUS
        endm

pop     macro
        banksel     StatusTemp          ;point to StatusTemp bank
        swapf       StatusTemp,W        ;unswap STATUS nybbles into W
        movwf       STATUS              ;restore STATUS (which points to where W was stored)
        swapf       WTemp,F             ;unswap W nybbles
        swapf       WTemp,W             ;restore W without affecting STATUS
        endm
```

```
;*****************************************************************
; Start of executable code

        org     0x00            ; Reset vector
        nop
        goto    Main

;*****************************************************************
; Interrupt vector

        org     0x04            ; interrupt vector
        goto    IntService

;*****************************************************************
; Main program

Main
        call    Initial         ; Initialize everything
MainLoop
        nop
        nop
        goto    MainLoop    ; Do it again

;*****************************************************************
; Initial Routine

Initial
        movlw       D'25'           ; This sets the baud rate to 9600
        banksel     SPBRG               ; assuming BRGH=1 and Fosc=4.000 MHz
        movwf       SPBRG

        banksel     RCSTA
        bsf         RCSTA,SPEN      ; Enable the serial port
        bcf         RCSTA,RX9 ; Disable9-bit Receive
        bsf         RCSTA,CREN      ; Enable continuous receive

        banksel     TXSTA
        bcf         TXSTA,SYNC      ; Set up the port for asynchronous operation
        bsf         TXSTA,TXEN      ; Transmit enabled
        bsf         TXSTA,BRGH      ; High baud rate
        bcf         TXSTA,TX9 ; Disable9-bit send

        banksel     PIE1            ; Enable the Timer2 interrupt
        bsf     PIE1, RCIE
        bcf     TRISC,RC6           ; Set RC6 to output Send Pin
```

```
        bsf     TRISC,RC7           ; Set RC7 to input Receive Pin

        banksel     INTCON          ; Enable global and peripheral interrupts
        bsf     INTCON, GIE
        bsf     INTCON, PEIE

        banksel     Counter
        clrf    Counter

        return

;*****************************************************************
; Interrupt Service Routine
; This routine is called whenever we get an interrupt.

IntService
        push
        btfsc   PIR1, RCIF  ; Check for a Timer2 interrupt
        call    RECEIVE
;       btfsc   ...         ; Check for another interrupt
;       call    ...
;       btfsc   ...         ; Check for another interrupt
;       call    ...
        pop
        retfie

;*****************************************************************
RECEIVE

        movf    RCREG,w
        movf    RCREG,w
        movwf       Counter
        incf    Counter,w

        banksel     TXREG
        movwf       TXREG           ; Send a next character out the serial port
        banksel     TXSTA
L1      btfss   TXSTA,TRMT
        goto    L1

        return

        end
```