```
; **** USART Module : Polling Transmission
;=================================================
; DEFINITIONS
;=================================================
        PROCESSOR PIC16F628
        #include <P16F628.INC>
        __CONFIG    _CP_OFF & _MCLRE_ON & INTRC_OSC_NOCLKOUT & _LVP_OFF
        & _WDT_OFF
#define NL  0x0A    ; New Line
#define FF  0x0C    ; Form feed
#define CR  0x0D    ; Carriage return

;=================================================
; VARAIBLES
;=================================================
        cblock  0x20
            count
            count0
            count1
            count2
            temp
            buffer
            index
        endc
;=================================================
; RESET and INTERRUPT VECTORS
;=================================================
        ORG     0x00
        goto    main
        ORG     0x04
        retfie

Main:   call    init
Message_loop:
        movlw   0x00
        movwf   index
        call    send_message1

        movlw   .250
        call    DelaymS

RX_loop:
        btfss   PIR1, RCIF
        goto    RX_loop

        movf    RCREG,w     ; Save the recieved character in 'temp'
        movwf   temp

        ; Send the reply message back to the computer
        movlw   0x00
        movwf   index
        call    send_message2

        movf    temp,w
        movwf   TXREG
        btfss   PIR1,TXIF
        goto    $-1
        movlw   CR
        movwf   TXREG
        btfss   PIR1,TXIF
        goto    $-1
        goto    RX_loop

; this subroutine sends a string of many characters from a look-up table
'message'
send_message1:
        movf    index,w
        call    message1        ; get character from the message table
```

```
67          movwf   buffer
68          movlw   0xFF
69          subwf   buffer,w    ; check if we are at the end of message?
70          btfsc   STATUS,Z
71          return
72          incf    index,f
73  TX1:    btfss   PIR1,TXIF   ; TXIF='1' if the TXREG is empty
74          goto    TX1
75          movf    buffer,w
76          movwf   TXREG
77          goto    send_message1 ; go back and do it again
78
79  send_message2:
80          movf    index,w
81          call    message2        ; get character from the message table
82          movwf   buffer
83          movlw   0xFF
84          subwf   buffer,w    ; check if we are at the end of message?
85          btfsc   STATUS,Z
86          return
87          incf    index,f
88  TX2:    btfss   PIR1,TXIF   ; TXIF='1' if the TXREG is empty
89          goto    TX2
90          movf    buffer,w
91          movwf   TXREG
92          goto    send_message2 ; go back and do it again
93
94  message1:
95          addwf   PCL,f
96          DT      NL,"Press any keys to get response from
            PIC16F628A",NL,CR,0xFF
97
98  message2:
99          addwf   PCL,f
100         DT      NL,"You press: ",0xFF
101
102 init:
103         ;Initialization of USART module
104         banksel CMCON
105         movlw   .7
106         movwf   CMCON   ; disable analog comparator
107         banksel TRISA
108         movlw   0x00
109         movwf   TRISA
110 ; Bits 1 and 2 of Port B are multiplexed as TX/CK and RX/DT for USART
    operation.
111 ; These bits must be set to input in the TRISB register.
112         bcf     TRISB,2 ; RB2 as output (Tx pin)
113         bsf     TRISB,1 ; RB1 as input (Rx pin)
114
115 ; The asynchronous baud rate(ABR) is calculated as follows:
116 ;   ABR = Fosc/{S*(x+1)}
117 ; x is value in SPBRG register.
118 ; S is 64 if high baud rate select bit(BRGH) in the TXSTA control register
    is clear (slow-speed baud rate).
119 ; S is 16 if the  BRGH bit is set (high-speed baud rate).
120 ; For setting to 9600 baud rate using a 4MHz oscillator at a high-speed
    baud rate the formula is:
121 ; At high speed (BRGH=1)
122 ; 9600 = 4,000,000/{16*(x+1)}
123 ; x = 25.041 -> use x=25
124 ; calculate baud rate from x=25 to find %error
125 ; baud rate = 4,000,000/{16*(25+1)} = 9,615 (0.16% error)
126 ; At slow speed (BRGH=0)
127 ; baud rate = 4,000,000/{64*(25+1)} = 2,403.85 (0.16% error)
128 ;
129
130         banksel SPBRG
```

```
131           movlw   .25     ; 2400 baud rate at 4MHz crystal
132           movwf   SPBRG   ; Place in baud rate generator
133   ; TXSTA (Transmit Status and Control Register) bit map:
134   ;   7   6   5   4   3   2   1   0   <== bits
135   ;   |   |   |   |   |   |   |   |______ Tx9D 9nth data bit on
136   ;   |   |   |   |   |   |   |           (used for parity)
137   ;   |   |   |   |   |   |   |___________ TRMT Transmit Shift Register
138   ;   |   |   |   |   |   |               1 = TSR empty
139   ;   |   |   |   |   |   |             * 0 = TSR full
140   ;   |   |   |   |   |   |_______________ BRGH High Speed Baud Rate
141   ;   |   |   |   |   |                   (Asynchronous mode only)
142   ;   |   |   |   |   |                   1 = high speed (*4)
143   ;   |   |   |   |   |                 * 0 = low speed
144   ;   |   |   |   |   |_________________ NOT USED
145   ;   |   |   |   |_____________________ SYNC USART Mode Select
146   ;   |   |   |                         1 = synchronous mode
147   ;   |   |   |                       * 0 = asynchronous mode
148   ;   |   |   |_________________________ TXEN Transmit Enable
149   ;   |   |                           * 1 = transmit enable
150   ;   |   |                             0 = transmit disable
151   ;   |   |_____________________________ TX9 Enable 9-bit Transmit
152   ;   |                                 1 = 9-bit transmission mode
153   ;   |                               * 0 = 8-bit mode
154   ;   |_______________________________ CSRC Clock Source Select
155   ;                                     Not used in asynchronous mode
156   ;                                     Synchronous mode:
157   ;                                        1 = Master Mode (internal clock)
158   ;                                      * 0 = Slave Mode (external clock)
159   ; Setup value = 0010 0000 = 0x20
160           banksel TXSTA   ; TXEN = '1', BRGH = '0'
161           movlw   0x20    ; Enable transmission and low speed baud rate
162           movwf   TXSTA
163   ; RCSTA (Receive Status and Control Register) bit map:
164   ;   7   6   5   4   3   2   1   0   <== bits
165   ;   |   |   |   |   |   |   |   |______ RX9D 9th data bit received
166   ;   |   |   |   |   |   |   |           (can be parity parity)
167   ;   |   |   |   |   |   |   |___________ OERR Overrun error
168   ;   |   |   |   |   |   |                1 = error clear by software
169   ;   |   |   |   |   |   |_______________ FERR Framing error
170   ;   |   |   |   |   |                    1 = error
171   ;   |   |   |   |   |_________________ NOT USED
172   ;   |   |   |   |_____________________ CREN Continuous Receive Enable
173   ;   |   |   |                           Asynchronous mode
174   ;   |   |   |                            * 1 = Enable continuous receive
175   ;   |   |   |                              0 = Disable continuous receive
176   ;   |   |   |                           Synchronous mode
177   ;   |   |   |                              1 = Enables until CREN cleared
178   ;   |   |   |                              0 = Disables continuous receive
179   ;   |   |   |_________________________ SREN Single Receiver Enable
180   ;   |   |                              Asynchronous mode = don't care
181   ;   |   |                              Synchronous mode
182   ;   |   |                                 1 = Enables single receiver
183   ;   |   |                                 0 = Disables single receiver
184   ;   |   |_____________________________ RX9  9-bit Receive Enable
185   ;   |                                   1 = 9-bit reception
186   ;   |                                 * 0 = 8-bit reception
187   ;   |_______________________________ SPEN Serial Port Enable
188   ;                                     * 1 = RX/DT and TX/CK are serial pins
189   ;                                       0 = Serial port disable
190   ; Setput value: 1001 0000 = 0x90
191           banksel RCSTA   ; Bank 0
192           movlw   0x90    ; Enable Serial port and continous reception
193           movwf   RCSTA
194
195           return
196
197   DelaymS:
198           movwf   count2
```

```
199             incf    count2,f
200             decfsz  count2,f
201             goto    $+2
202             goto    $+3
203             call    Delay1mS
204             goto    $-4
205             return
206
207     Delay1mS:
208             movlw   .50             ; 1 cyc
209             movwf   count1          ; 1 cyc
210     outterloop:
211             movlw   .5              ; 1 cyc * count1
212             nop                     ; 1 cyc * count1
213             movwf   count0          ; 1 cyc * count1
214     innerloop:
215             decfsz  count0,F        ; 1 cyc * count1 * count0
216             goto    innerloop       ; 2 cyc * count1 * count0
217             decfsz  count1,F        ; 1 cyc * count1
218             goto    outterloop      ; 2 cyc * count1
219             return                  ; 1 cyc
220             ; total = 3 + (6+3.count0).count1
221             ; count0 = 5 , count1 = 50, total = 1053 cyc ??
222     ;================================================
223             END
```