

```

1  ; ***** USART Module : Polling Transmission
2  ;=====
3  ; DEFINITIONS
4  ;=====
5      PROCESSOR PIC16F628
6      #include <P16F628.INC>
7      _CONFIG    _CP_OFF & _MCLRE_ON & INTRC_OSC_NOCLKOUT & _LVP_OFF
8      & _WDT_OFF
9  #define NL      0x0A      ; New Line
10 #define FF      0x0C      ; Form feed
11 #define CR      0x0D      ; Carriage return
12 ;=====
13 ; VARAIBLES
14 ;=====
15      cblock      0x20
16          count
17          count0
18          count1
19          count2
20          temp
21          buffer
22          index
23      endc
24 ;=====
25 ; RESET and INTERRUPT VECTORS
26 ;=====
27      ORG          0x00
28      goto         main
29      ORG          0x04
30      goto         usart_RX_ISR
31
32 main:      call     init
33      movlw        0x00
34      movwf        index
35      call         send_message
36 inf_loop:
37      nop
38      goto         inf_loop
39
40 ;Serial Comm Rx interrupt ISR
41 usart_RX_ISR:
42 EXT_INT:  btfss    INTCON,INTF
43      goto         serial_ISR
44      bcf          INTCON,INTF
45      movlw        B'00000100'
46      xorwf        PORTA,f
47      goto         ISR_exit
48 serial_ISR:
49      btfss        PIR1,RCIF
50      goto         ISR_exit      ; get out if not the receive interrupt
51      movf         RCREG,w      ; RCIF flag is cleared if RCREG is read
52      movwf        buffer
53 LED0:      movlw        '0'
54      subwf        buffer,w
55      btfss        STATUS,Z
56      goto         LED1
57      movlw        0x00
58      movwf        PORTA
59      bsf          PORTA,0
60      goto         ISR_exit
61 LED1:      movlw        '1'
62      subwf        buffer,w
63      btfss        STATUS,Z
64      goto         LED2
65      movlw        0x00
66      movwf        PORTA
67      bsf          PORTA,1

```

```

68      goto      ISR_exit
69  LED2:  movlw    '2'
70      subwf     buffer,w
71      btfss     STATUS,Z
72      goto      LED3
73      movlw     0x00
74      movwf     PORTA
75      bsf       PORTA,2
76      goto      ISR_exit
77  LED3:  movlw    '3'
78      subwf     buffer,w
79      btfss     STATUS,Z
80      goto      ISR_exit
81      movlw     0x00
82      movwf     PORTA
83      bsf       PORTA,3
84  ISR_exit:
85      retfie
86
87  send_message:
88      movf      index,w
89      call      message      ; get character from the message table
90      movwf     buffer
91
92      movlw     .100
93      call      DelaymS
94
95      movlw     0xFF
96      subwf     buffer,w      ; check if we are at the end of message?
97      btfsc     STATUS,Z
98      return
99      incf      index,f
100  TX:    btfss   PIR1,TXIF      ; TXIF='1' if the TXREG is empty
101      goto      TX
102      movf      buffer,w
103      movwf     TXREG
104      goto      send_message ; go back and do it again
105
106  message:
107      addwf     PCL,feed
108      DT        FF,NL,"EGEE 380 Serial Communication Example",NL,CR
109      DT        "at 2400 baud rate speed .....:-)",NL,CR
110      DT        "Press keys [0,1,2,3] to turn-on LEDs",NL,CR,0xFF
111
112  init:
113      ;Initialization of USART module
114      banksel   CMCON
115      movlw     .7
116      movwf     CMCON      ; disable analog comparator
117      banksel   TRISA
118      movlw     0x00
119      movwf     TRISA
120      ;+++++
121      banksel   INTCON
122      bsf       INTCON,INTE
123      bcf       INTCON,INTF ; External interrupt clear
124
125      banksel   OPTION_REG
126      bsf       OPTION_REG,INTEDG
127
128      ; Bits 1 and 2 of Port B are multiplexed as TX/CK and RX/DT for USART
      operation.
129      ; These bits must be set to input in the TRISB register.
130      bsf       TRISB,2 ; RB2 as input (Tx pin)
131      bsf       TRISB,1 ; RB1 as input (Rx pin)
132
133      ; The asynchronous baud rate(ABR) is calculated as follows:
134      ;   ABR = Fosc/{S*(x+1)}

```

```

135 ; x is value in SPBRG register.
136 ; S is 64 if high baud rate select bit(BRGH) in the TXSTA control register
    is clear (slow-speed baud rate).
137 ; S is 16 if the BRGH bit is set (high-speed baud rate).
138 ; For setting to 9600 baud rate using a 4MHz oscillator at a high-speed
    baud rate the formula is:
139 ; At high speed (BRGH=1)
140 ;  $9600 = 4,000,000 / \{16 * (x+1)\}$ 
141 ;  $x = 25.041 \rightarrow$  use  $x=25$ 
142 ; calculate baud rate from  $x=25$  to find %error
143 ;  $\text{baud rate} = 4,000,000 / \{16 * (25+1)\} = 9,615$  (0.16% error)
144 ; At slow speed (BRGH=0)
145 ;  $\text{baud rate} = 4,000,000 / \{64 * (25+1)\} = 2,403.85$  (0.16% error)
146 ;
147
148         banksel SPBRG
149         movlw    .25      ; 2400 baud rate at 4MHz crystal
150         movwf   SPBRG    ; Place in baud rate generator
151 ; TXSTA (Transmit Status and Control Register) bit map:
152 ;   7   6   5   4   3   2   1   0   <== bits
153 ;   |   |   |   |   |   |   |   |   Tx9D 9th data bit on
154 ;   |   |   |   |   |   |   |   |   (used for parity)
155 ;   |   |   |   |   |   |   |   |   TRMT Transmit Shift Register
156 ;   |   |   |   |   |   |   |   |   1 = TSR empty
157 ;   |   |   |   |   |   |   |   |   * 0 = TSR full
158 ;   |   |   |   |   |   |   |   |   BRGH High Speed Baud Rate
159 ;   |   |   |   |   |   |   |   |   (Asynchronous mode only)
160 ;   |   |   |   |   |   |   |   |   1 = high speed (*4)
161 ;   |   |   |   |   |   |   |   |   * 0 = low speed
162 ;   |   |   |   |   |   |   |   |   NOT USED
163 ;   |   |   |   |   |   |   |   |   SYNC USART Mode Select
164 ;   |   |   |   |   |   |   |   |   1 = synchronous mode
165 ;   |   |   |   |   |   |   |   |   * 0 = asynchronous mode
166 ;   |   |   |   |   |   |   |   |   TXEN Transmit Enable
167 ;   |   |   |   |   |   |   |   |   * 1 = transmit enable
168 ;   |   |   |   |   |   |   |   |   0 = transmit disable
169 ;   |   |   |   |   |   |   |   |   TX9 Enable 9-bit Transmit
170 ;   |   |   |   |   |   |   |   |   1 = 9-bit transmission mode
171 ;   |   |   |   |   |   |   |   |   * 0 = 8-bit mode
172 ;   |   |   |   |   |   |   |   |   CSRC Clock Source Select
173 ;   |   |   |   |   |   |   |   |   Not used in asynchronous mode
174 ;   |   |   |   |   |   |   |   |   Synchronous mode:
175 ;   |   |   |   |   |   |   |   |   1 = Master Mode (internal clock)
176 ;   |   |   |   |   |   |   |   |   * 0 = Slave Mode (external clock)
177 ; Setup value = 0010 0000 = 0x20
178         banksel TXSTA    ; TXEN = '1', BRGH = '0'
179         movlw    0x20    ; Enable transmission and low speed baud rate
180         movwf   TXSTA
181 ; RCSTA (Receive Status and Control Register) bit map:
182 ;   7   6   5   4   3   2   1   0   <== bits
183 ;   |   |   |   |   |   |   |   |   RX9D 9th data bit received
184 ;   |   |   |   |   |   |   |   |   (can be parity parity)
185 ;   |   |   |   |   |   |   |   |   OERR Overrun error
186 ;   |   |   |   |   |   |   |   |   1 = error clear by software
187 ;   |   |   |   |   |   |   |   |   FERR Framing error
188 ;   |   |   |   |   |   |   |   |   1 = error
189 ;   |   |   |   |   |   |   |   |   NOT USED
190 ;   |   |   |   |   |   |   |   |   CREN Continuous Receive Enable
191 ;   |   |   |   |   |   |   |   |   Asynchronous mode
192 ;   |   |   |   |   |   |   |   |   * 1 = Enable continuous receive
193 ;   |   |   |   |   |   |   |   |   0 = Disable continuous receive
194 ;   |   |   |   |   |   |   |   |   Synchronous mode
195 ;   |   |   |   |   |   |   |   |   1 = Enables until CREN cleared
196 ;   |   |   |   |   |   |   |   |   0 = Disables continuous receive
197 ;   |   |   |   |   |   |   |   |   SREN Single Receiver Enable
198 ;   |   |   |   |   |   |   |   |   Asynchronous mode = don't care
199 ;   |   |   |   |   |   |   |   |   Synchronous mode
200 ;   |   |   |   |   |   |   |   |   1 = Enables single receiver

```

```

201 ; | | 0 = Disables single receiver
202 ; | | _____ RX9 9-bit Receive Enable
203 ; | | 1 = 9-bit reception
204 ; | | * 0 = 8-bit reception
205 ; | _____ SPEN Serial Port Enable
206 ; | * 1 = RX/DT and TX/CK are serial pins
207 ; | 0 = Serial port disable
208 ; Setput value: 1001 0000 = 0x90
209     banksel RCSTA ; Bank 0
210     movlw 0x90 ; Enable Serial port and continous reception
211     movwf RCSTA
212 ; Setting up interrupt control registers
213     banksel INTCON
214     bsf INTCON,PEIE ; Enable peripheral interrupt (serial port)
215     bsf INTCON,GIE ; Enable global interrupt
216
217     banksel PIE1
218     bsf PIE1,RCIE ; Enable only receiving interrupt
219
220     banksel PIR1 ; Bank 0
221     bcf PIR1,RCIF ; Clear receive interrupt flag
222
223     movlw 0x00
224     movwf PORTA ; Turn-off all LEDs
225     return
226
227 DelaymS:
228     movwf count2
229     incf count2,f
230     decfsz count2,f
231     goto $+2
232     goto $+3
233     call Delay1mS
234     goto $-4
235     return
236
237 Delay1mS:
238     movlw .50 ; 1 cyc
239     movwf count1 ; 1 cyc
240 outterloop:
241     movlw .5 ; 1 cyc * count1
242     nop ; 1 cyc * count1
243     movwf count0 ; 1 cyc * count1
244 innerloop:
245     decfsz count0,F ; 1 cyc * count1 * count0
246     goto innerloop ; 2 cyc * count1 * count0
247     decfsz count1,F ; 1 cyc * count1
248     goto outterloop ; 2 cyc * count1
249     return ; 1 cyc
250 ; total = 3 + (6+3.count0).count1
251 ; count0 = 5 , count1 = 50, total = 1053 cyc ??
252 ;=====
253     END

```