```
    1: ;;;
    2: ;;; 485 tutorial
    3: ;;; r <raffi@media.mit.edu>
    4: ;;; 2/11/03
    5: ;;;
    6:
    7:         LIST    p = 16f876
    8:
    9: #include p16f876.inc
   10:
   11:         __config        _HS_OSC & _WDT_OFF & _PWRTE_ON & _BODEN_OFF & _CP_OFF & _LVP
_OFF
   12:
   13: LED_GREEN       EQU     0
   14: LED_RED         EQU     1
   15:
   16: PIN_SRX         EQU     7
   17: PIN_STX         EQU     6
   18: PIN_SDE         EQU     5
   19:
   20: TIMER0          EQU     0x20    ; use this for a "clock"
   21:
   22:
   23: ;;;
   24: ;;; MACROS
   25: ;;;
   26:
   27: #define BIT(i) (1<<(i))
   28:
   29:
   30: ;;; a macro to move a literal into a memory address
   31: MVL     MACRO   l, dest
   32:         MOVLW   l
   33:         MOVWF   dest
   34:         ENDM
   35:
   36:
   37: ;;; a macro to transmit out a literla
   38: TXL     MACRO   l
   39:         MOVLW   l
   40:         CALL    tx
   41:         ENDM
   42:
   43:
   44:
   45: ;;;
   46: ;;; PROGRAM CODE
   47: ;;;
   48:
   49:         ORG     0x0000
   50:         GOTO    init
   51:         ORG     0x0004
   52:         RETFIE
   53:
   54: init:
   55:         CLRF    STATUS
   56:         CLRF    PORTC
   57:
   58:         CLRF    T1CON           ; setup to use TIMER1
   59:         CLRF    TMR1H
   60:         CLRF    TMR1L
   61:         CLRF    INTCON
   62:
   63:         BSF     STATUS, RP0     ; bank 1
   64:         MVL     BIT(PIN_SRX), TRISC ; port c outputs except rx pin
   65:
   66:         CLRF    PIE1            ; disable peripheral irrupts
   67:
   68:         ;; setup USART at 38.4kbps
   69:         MVL     d'32', SPBRG
   70:         MVL     BIT(TXEN) | BIT(BRGH), TXSTA
   71:         BCF     STATUS, RP0     ; bank 0
   72:         MVL     BIT(SPEN) | BIT(CREN), RCSTA
   73:         BCF     RCSTA, CREN     ; reset the USART to be in receive
   74:         BSF     RCSTA, CREN
   75:
   76:         ;; start TIMER1 and lets start keeping track of time
   77:         CLRF    PIR1
   78:         MVL     BIT(T1CKPS1) | BIT(T1CKPS0) | BIT(TMR1ON), T1CON
   79:         MVL     d'10', TIMER0
   80:
   81:         BSF     PORTC, LED_GREEN; turn on the LEDs
   82:         BSF     PORTC, LED_RED
   83:
   84: main:
   85:         BTFSC   PIR1, TMR1IF    ; see if the clock has "ticked"
   86:         GOTO    timer
   87: _main_after_timer:
   88:         BTFSC   RCSTA, OERR     ; if there is a serial error, then reset
   89:         GOTO    rx_reset
   90:         BTFSC   PIR1, RCIF      ; if there is a received byte, then deal
   91:         GOTO    rx
   92: _main_after_serial:
   93:         GOTO    main            ; restart the main loop
   94:
   95:
   96: ;;; handle the timer tick -- whenever it does tick (approx once a second then
   97: ;;; we are going to transmit out Hi!
   98: timer:
   99:         BCF     PIR1, TMR1IF    ; reset the timer
  100:         MVL     d'10', TIMER0
  101:         TXL     d'72'           ; H
  102:         TXL     d'105'          ; i
  103:         TXL     d'33'           ; !
  104:         TXL     d'13'           ; /CR
  105:         TXL     d'10'           ; /LF
  106:         GOTO    _main_after_timer
  107:
  108:
  109: ;;; handle the incoming bytes -- we just flash the LED when that happens
  110: rx:
  111:         MOVF    RCREG, W
  112:         BTFSC   RCSTA, FERR
  113:         GOTO    rx_reset
  114:
  115:         MOVLW   BIT(LED_RED)    ; flash the red light
  116:         XORWF   PORTC, F
  117:         GOTO    _main_after_serial
  118:
  119:
  120: ;;; reset the serial port
  121: rx_reset:
  122:         BCF     RCSTA, CREN
  123:         BSF     RCSTA, CREN
  124:         GOTO    _main_after_serial
  125:
  126:
  127: ;;; the transmit character
  128: tx:
  129:         BCF     RCSTA, CREN     ; turn off receive on serial
  130:         BSF     PORTC, PIN_SDE  ; enable transmit
  131:         MOVWF   TXREG           ; move W -> TXREG
```

```
  132:         BSF     STATUS, RP0     ; bank 1
  133: _tx_wait:
  134:         BTFSS   TXSTA, TRMT     ; wait until TRMT is high
  135:         GOTO    _tx_wait
  136:         BCF     STATUS, RP0     ; bank 0
  137:         BCF     PORTC, PIN_SDE  ; disable transmit
  138:         BSF     RCSTA, CREN     ; enable receive on serial
  139:         MOVLW   BIT(LED_GREEN)  ; flash the green light
  140:         XORWF   PORTC, F
  141:         RETURN
  142:
  143:         END
  144:
```