TRC102 Range Test at 433.92MHz

By Scott Shoaf, RFIC Product Engineer, RF Monolithics, Dallas, TX

Disclaimer: This test is intended to provide a benchmark for range performance of the TRC102. Each operating environment will differ with each having unique obstacles for RF propagation to contend with.

The TRC102 has +7dBm (5mW) of output power and the best receive sensitivity of our RFIC line of short range radios. The performance at 433.92 MHz is used as a baseline for range test characterization between 315MHz and 916MHz. From this, one can gain an idea of how much line-of-sight range to expect depending upon the frequency of operation and the operating environment.

In the U.S., FCC part 15 limits transmission output power to 0dBm (1mW) for short range, unlicensed radio applications. The output power of the TRC102 is adjustable which allows for testing and operation at other power levels. Since the FCC limit is 0dBm, the output power on the TRC102 was adjusted down -6dB from the peak power level.

The setup parameters for the Transmitter, using the RFDA, were as follows:

Freq – 433.92MHz
Oscillator Enabled
Synthesizer Enabled
Clock Output Disabled
Pin 8 – Data Detector Output
PLL Dithering On
Crystal Load – 8.5pF
FSK Deviation – 15kHz
Pout - -6dB (0 dBm)
Polarity of Modulation – Fo+df
Clock Buffer Slew - >5MHz
Data Rate – 2400 (19200)
  -Prescaler Enabled (Disabled)
  -R=17

For the above Transmitter settings, the respective register values are as follows:

Configuration - 0x8010
Frequency setting – 0xA620
Power Management – 0x8219 (Transmitter Off), 0x8239 (Transmitter On)
Receiver Setting – N/A
Transmitter Setting – 9800
Synch Character – 0xCEE2
PLL Command – 0xCC06
AFC Command – 0xC4E7
Data Rate Command – 0xC691
Data Filter Command – N/A
FIFO Buffer Cmd – N/A

The setup parameters for the Receiver, using the RFDA, were as follows:

Freq – 433.92MHz
Oscillator Enabled
Synthesizer Enabled
Clock Output Disabled
Pin 8 – Data Detector Output
PLL Dithering On
Crystal Load – 8.5pF
LNA Gain – Max
DRSSI - -103dBm
Baseband BW – 67kHz
Valid Data Detector – Medium
Synch Charac Byte – E2 (Programmable)
Clock Buffer Slew - >5MHz
AFA Enabled
 -Fine Mode Enabled
 -Mode – Auto, keep offset
 -Tuning - +7/-8 Fres
 -Output Enabled
Data Rate – 2400 (19200)
 -Prescaler Enabled (Disabled)
 -R=17
Data Filter
 -Clock Recovery – Slow
 -Filter Type – Digital LPF
 -DQD – 4
FIFO Buffer
 -Enable Synch Latch
 -Disable Sensitive Reset
 -FIFO Fill Start – Synch Pattern
 -FIFO IT level – 8

For the above Receiver settings, the respective register values are as follows:

Configuration - 0x8010
Frequency setting – 0xA620
Power Management – 0x82D9 (Receive and Baseband On)
Receiver Setting – 0x95C0
Transmitter Setting – N/A
Synch Character – 0xCEE2
PLL Command – 0xCC06
AFC Command – 0xC4E7
Data Rate Command – 0xC691
Data Filter Command – 0xC22C
FIFO Buffer Cmd – 0xCA83

All tests were conducted using the antenna soldered to a DR-TRC102-433 evaluation board. The data rate was configured to 2.4kbps. A data payload of 35 bytes was transmitted, including a 2 byte preamble.

The first test uses a simple λ/4 monopole at 433.92MHz on both transmitter and receiver. The transmitter was mounted at a height of 3m (10ft). The receiver was held at 1.5m high as the distance between the two were increased. A line-of-sight range of 675m (2217ft) was achieved using the monopole antenna.

The second test uses Murata's loaded monopole at 433.92MHz on both transmitter and receiver. This is the same antenna used on the DR-1300A-DK ASH Development Kit. The transmitter was mounted at a height of 3m (10ft). The receiver was also held at 1.5m high as the distance between the two were increased. A line-of-sight range of 725m (2376ft) was achieved using the loaded monopole antenna.

This data shows that operating the TRC102 at max power (+7dBm), a range of >1000m is achievable.

The evaluation boards were not specifically designed to optimize use with an antenna. The evaluation board does not provide a "balanced" ground plane for use with a monopole antenna structure, thus, a PCB design that is optimized for a monopole, using a λ/4 radiating element centered in a ground plane, would theoretically give an additional range of 15-30m (50-100ft).

Range	λ/4 Monopole	End Loaded Monopole
Max	>1000m	>1000m
+0dBm	675m(2271ft)	725m (2376ft)

Receiver Range Test Assembly Code for C8051F330:

```
$include (c8051f330.inc)

;-- Bit Addressable
RXFLG       EQU         00H         ;RX FLAG AT REG 20H, BIT 0
STATFLG     EQU         01H         ;STATUS READ FLAG AT REG 20H, BIT 1

;-- Byte Addressable
HIGHBYTE EQU            21H         ;HIGH BYTE OF IC WORD
LOWBYTE  EQU            22H         ;LOW BYTE OF IC WORD
TBLOFF      EQU         23H         ;Table offset value
CNT         EQU         24H         ;byte count
CHKSMH      EQU         25H         ;CHECKSUM HIGH BYTE
RXBUF       EQU         26H         ;RX BUFFER AREA TO 46h

;PORT 0------------------------
RXLED       EQU         P0.0    ;O LED      1 P/P   0
ACKLED      EQU         P0.1    ;O LED      1 P/P   0
SLPLED      EQU         P0.2    ;O LED      1 P/P   0
TXLED       EQU         P0.3    ;O LED      1 P/P   0

RS232TX     EQU         P0.4    ;O          1 P/P   1
RS232RX     EQU         P0.5    ;I          0 O/D   1
SCK         EQU         P0.6    ;O          1 P/P   0
SDO         EQU         P0.7    ;I          0 O/D   1

;PORT 1------------------------
SDI     EQU         P1.0        ;O          1 P/P   0
SEL         EQU         P1.1    ;O          1 P/P   1
IRQ     EQU         P1.2        ;I          0 O/D   1
FSEL        EQU         P1.3    ;O          1P/P    0
;LED        EQU         P1.3    ;O LED
FFULL       EQU         P1.4    ;I          0 O/D   1
RSSI        EQU         P1.5    ;I          0 O/D   1
DDET        EQU         P1.6    ;I (or INT) 0 O/D   1
RNGTST      EQU         P1.7    ;I          0 O/D   1

    ORG         00h
    LJMP    MAIN
    ORG         0FFh

;------ Initialization functions -----------------
MAIN:

  mov  PCA0MD,    #00h

Port_IO_Init:

mov  P0MDOUT,   #05Fh     ;0101 1111
mov  P1MDOUT,   #0Bh;0000 1011
mov  P0SKIP,    #00Fh
mov         P0,             #0B0h   ;1011 0000
mov         P1,             #0F6h   ;1111 0110
mov         P2,             #000h
mov  XBR0,     #003h   ;0000 0011
mov  XBR1,     #040h   ;0100 0000

Timer1_Init:
    mov  TMOD,      #021h       ;TMR1 Mode 2(2 8-bit), TMR0 Mode 1(16-bit)
    mov  TH1,      #0CBh        ;UART Reload value for 19.2Baud
    MOV      TL1,               #0CBH       ;INIT TMR1
    mov  CKCON,     #00h        ;SYSCLK/12

UART_Init:
  mov  SCON0,     #030h         ;RXEN,RX INT active on stop bit

SPI_Init:
  mov  SPI0CFG,   #047h
```

```
   mov  SPI0CN,    #0Fh
   mov  SPI0CKR,   #000H        ;CLK = 3.06/2 = 1.5 MHz

Oscillator_Init:
   mov  OSCICN,    #80h         ;SYSCLK = 24.5 MHz/8 = 3.0MHz


Interrupts_Init:
   mov      PCA0MD,#00h
   MOV      EIE1,#80h
   SETB     PSPI0
   SETB     EA



;Main Code Section***********************************

     mov         R0,#00h         ;CLEAR REGISTERS
     mov         R1,#00h         ;
     mov         R2,#00h         ;
     mov         R3,#00h         ;
     mov         R4,#00h         ;
     mov         R5,#00h         ;
     mov         R6,#00h         ;
     mov         R7,#00h         ;

     CLR         STATFLG
     CLR         RXFLG

;Flash LED's ON STARTUP

     SETB        RXLED           ;TURN ON LED
     SETB        TR0                 ;TMR0 ENABLED
NXT:
     JNB         TF0,NXT         ;WAIT TIL TMR OVERFLOW THEN JUMP
     CLR         TF0             ;CLEAR OVERFLOW FLAG
     SETB        TXLED           ;TURN ON LED
NXT1:
     JNB         TF0,NXT1        ;WAIT TIL TMR OVERFLOW THEN JUMP
     CLR         TF0             ;CLEAR OVERFLOW FLAG
     SETB        ACKLED              ;TURN ON LED
NXT2:
     JNB         TF0,NXT2        ;WAIT TIL TMR OVERFLOW THEN JUMP
     CLR         TF0             ;CLEAR OVERFLOW FLAG
     SETB        SLPLED              ;TURN ON LED
NXT3:
     JNB         TF0,NXT3        ;WAIT TIL TMR OVERFLOW THEN JUMP
     CLR         TF0             ;CLEAR OVERFLOW FLAG
     CLR         RXLED
     CLR         TXLED
     CLR         ACKLED
     CLR         SLPLED              ;TURN OFF LED
     CLR         TR0                     ;TMR0 DISABLED
;-----------------------------------------------------------------

;TEST IF JUMPER INSTALLED FOR DATA RATE
     JNB         RNGTST,CFG2

;Configure Device FOR 2.4KBPS
CFG1:
     mov         XBR1,#0C0h              ;DIS WEAK PULLUPS
     mov         CNT,#0Dh                ;LOAD BYTE COUNT
     mov         DPTR,#RXSETUP2400           ;load table pointer
     mov         TBLOFF,#0               ;set offset value
A1:  mov         A,TBLOFF                ;load offset value
     mov         R1,#HIGHBYTE            ;load buffer with HIGHBYTE ADDRESS
     movc        A,@A+DPTR      ;load table byte
     mov         @R1,A                       ;...into buffer
     inc         TBLOFF                      ;incr offset
     inc         R1                          ;incr buffer to LOWBYTE ADDRESS
```

```
      mov        A,TBLOFF                 ;load offset value
      movc       A,@A+DPTR    ;load table byte
      mov        @R1,A                         ;...into buffer
      inc        TBLOFF
      MOV        SPI0CN,#009h  ;SET nSEL LOW (CHIP SELECT)
      ACALL      SPISEND                  ;DO ACTUAL SPI TRANSACTION
      MOV        SPI0CN,#00Dh ;SET nSEL HIGH (DESELECT CHIP)
      djnz       CNT,A1                 ;DECREMENT BYTE COUNTER
      SJMP       START
;-----------------------------------------------------------------

;Configure Device 19.2KBPS
CFG2:
         mov        XBR1,#0C0h               ;DIS WEAK PULLUPS
         mov        CNT,#0Dh                 ;LOAD BYTE COUNT
         mov        DPTR,#RXSETUP19200          ;load table pointer
         mov        TBLOFF,#0                ;set offset value
RA1:     mov        A,TBLOFF            ;load offset value
         mov        R1,#HIGHBYTE             ;load buffer with HIGHBYTE ADDRESS
         movc       A,@A+DPTR    ;load table byte
         mov        @R1,A                         ;into buffer
         inc        TBLOFF                        ;incr offset
         inc        R1                            ;incr buffer to LOWBYTE ADDRESS
         mov        A,TBLOFF                 ;load offset value
         movc       A,@A+DPTR    ;load table byte
         mov        @R1,A                         ;into buffer
         inc        TBLOFF
         MOV        SPI0CN,#009h  ;SET nSEL LOW (CHIP SELECT)
         ACALL      SPISEND                  ;DO ACTUAL SPI TRANSACTION
         MOV        SPI0CN,#00Dh ;SET nSEL HIGH (DESELECT CHIP)
         djnz       CNT,RA1                  ;DECREMENT BYTE COUNTER
;-----------------------------------------------------------------

;**********************************************************
;******************  MAIN LOOP   **********************
;**********************************************************
START:
         CLR        A                                   ;CLEAR ACC
         CLR        CHKSMH                        ;CLEAR CHECKSUM HIGH BYTE
         MOV        CNT,#1FH                 ;LOAD BYTE COUNT
         MOV        R1,#RXBUF                     ;LOAD ADDR OF FIRST BUFFER LOC
XX1:     JNB        DDET,XX1                 ;WAIT FOR VALID DATA
Z1:      JNB        FFULL,Z1                 ;TEST IF DATA IN FIFO READY

;--------------------------------------------------------------
;BEGIN DATA RX
;**REFER TO RECOMMENDED READ PROCESS IN DATASHEET
;  1-PULL nCS "HIGH"
;  2-PULL FSEL "LOW"
;  3-WAIT FOR FINT TO GO "HIGH" INDICATING RX DATA RDY
;  4-WRITE A DUMMY BYTE TO THE SPI AND READ FIFO DATA BACK
;--------------------------------------------------------------
UNO:
         SETB       ACKLED
         MOV        SPI0DAT,#00H        ;WRITE DUMMY BYTE TO SPI
WAIT3:
         JNB        SPIF, WAIT3         ;WAIT FOR SPI DONE
         CLR        SPIF                     ;RESET FLAG
         MOV        @R1,SPI0DAT         ;WRITE BYTE TO RX BUFFER LOC
         INC        R1
         DJNZ       CNT,Z1              ;DECREMENT COUNT. BAIL IF ALL BYTES RX
         LJMP       COMPARE                  ;IF ALL BYTES READ THEN COMPARE
;--------------------------------------------------------------
;COMPARE THE RX DATA TO DATA IN MEMORY
;--------------------------------------------------------------
COMPARE:                                 ;COMPARE READ VALUES TO THOSE IN MEM
         mov        CNT,#1Eh                 ;LOAD RX DATA COUNTER
         mov        R1,#RXBUF                     ;load table pointer
         mov        DPTR,#TXDATA             ;load table pointer
         CLR        A
```

```
HERE:
        MOVC    A,@A+DPTR
        mov     B,@R1
        CJNE    A,B,RESTART
        INC     DPTR
        INC     R1
        CLR     A
        DJNZ    CNT,HERE

;---------------------------------------------------------------
        ;FLASH GREEN LED IF DATA GOOD
;---------------------------------------------------------------
        SETB    SLPLED              ;TURN ON LED
        MOV     TL0,#00H
        MOV     TH0,#0E0H
        SETB    TR0                 ;TMR0 ENABLED
;CLEAR BUFFER
        MOV     CNT,#20h            ;LOAD BYTE COUNT
        MOV     R1,#RXBUF               ;LOAD ADDR OF FIRST BUFFER LOC
X2: MOV         @R1,#00h
        INC     R1
        DJNZ    CNT,X2
;--- THIS RESETS THE SYNCH CHARAC RECOGNITION -----
        mov     HIGHBYTE,#0CAH
        MOV     LOWBYTE,#81H     ;LOAD FIFO/RESET CONFIG REG
        MOV     SPI0CN,#009h  ;SET nSEL LOW (CHIP SELECT)
        ACALL   SPISEND                 ;CLEAR SYNCH CHAR RECOG
        MOV     SPI0CN,#00Dh ;SET nSEL HIGH (DESELECT CHIP)
        MOV     LOWBYTE,#83H     ;LOAD FIFO/RESET CONFIG REG
        MOV     SPI0CN,#009h  ;SET nSEL LOW (CHIP SELECT)
        ACALL   SPISEND                 ;RESET FIFO FILL ON SYNCH CHAR
        MOV     SPI0CN,#00Dh ;SET nSEL HIGH (DESELECT CHIP)
;--------------------------------------------------------
WT:     JNB     TF0,WT              ;WAIT TIL TMR OVERFLOW THEN JUMP
        CLR     TF0                 ;CLEAR OVERFLOW FLAG
        CLR     SLPLED              ;TURN OFF LED
        CLR     TR0
YR1:    JB      DDET,YR1        ;WAIT FOR VALID DATA INACTIVE
        CLR     ACKLED              ;TURN OFF LED
        LJMP    START
;---------------------------------------------------------------

;---------------------------------------------------------------
;           FLASH RED LED IF BAD DATA
;---------------------------------------------------------------
RESTART:                        ;CLEAR BUFFER AND FLASH ERR LED
        SETB    TXLED           ;TURN ERR LED ON
        MOV     TL0,#00H
        MOV     TH0,#0C0H
        SETB    TR0                     ;TMR0 ENABLED
;CLEAR BUFFER
RST:
        MOV     CNT,#20h            ;LOAD BYTE COUNT
        MOV     R1,#RXBUF               ;LOAD ADDR OF FIRST BUFFER LOC
X1:     MOV     @R1,#00h
        INC     R1
        DJNZ    CNT,X1

;--- THIS RESETS THE SYNCH CHARAC RECOGNITION -----
    mov     HIGHBYTE,#0CAH
    MOV     LOWBYTE,#81H     ;LOAD FIFO/RESET CONFIG REG
    MOV     SPI0CN,#009h  ;SET nSEL LOW (CHIP SELECT)
    ACALL   SPISEND                 ;CLEAR SYNCH CHAR RECOG
    MOV     SPI0CN,#00Dh ;SET nSEL HIGH (DESELECT CHIP)
    MOV     LOWBYTE,#83H     ;LOAD FIFO/RESET CONFIG REG
    MOV     SPI0CN,#009h  ;SET nSEL LOW (CHIP SELECT)
    ACALL   SPISEND                 ;RESET FIFO FILL ON SYNCH CHAR
    MOV     SPI0CN,#00Dh ;SET nSEL HIGH (DESELECT CHIP)
;--------------------------------------------------------
WT2:
        JNB     TF0,WT2             ;WAIT TIL TMR OVERFLOW THEN JUMP
        CLR     TF0                 ;CLEAR OVERFLOW FLAG
        CLR     TXLED               ;TURN OFF ERR LED
        CLR     TR0
XR1:    JB      DDET,XR1      ;WAIT FOR VALID DATA INACTIVE
        CLR     ACKLED
        LJMP    START
;------------------------------------------------------------


; ************************ SPI SEND *******************************************
SPISEND:
        ;(Chip Select already LOW)
        MOV     SPI0DAT,HIGHBYTE        ;WRITE HIGH BYTE TO SPI
WAIT4:
        JNB     SPIF,WAIT4              ;WAIT FOR SPI TO FINISH FIRST XFER,LOOP IF STILL BUSY
        CLR     SPIF                        ;CLEAR SPI INT FLAG TO PROCEED
        MOV     SPI0DAT,LOWBYTE         ;WRITE LOW BYTE TO SPI
WAIT5:
        JNB     SPIF,WAIT5                  ;WAIT FOR SPI TO FINISH 2ND XFER,LOOP IF STILL BUSY
        CLR     SPIF                        ;CLEAR SPI INT FLAG TO PROCEED
RTRN:
        CLR     ACKLED
        RET
; ******************************************************************************

RXSETUP2400:
        DB  80h
        DB  67h     ;config reg

        DB  0A6h
        DB  40h     ;Freq set

        DB  96H
        DB  0a0H    ;RX set

        DB  98h
        DB  00h     ;TX set

        DB  0CEh
        DB  0D4H    ;Synch Char

        DB  0CCh
        DB  06h     ;PLL cmd

        DB  0C6h
        DB  91h     ;Data Rate 2.4Kbps

        DB  0C4h
        DB  0D7h    ;AFA

        DB  0CAh
        DB  83h     ;RX FIFO (FILL ALWAYS 87H) (SYNCH CHAR 83H)

        DB  0C2h
        DB  2Ch     ;Baseband Filter


        DB  82h
        DB  0D9h    ;Pwr mng      ;TURN ON RX

        DB  82h
        DB  049h    ;Pwr mng      ;TURN OFF SYNTH (CALIBRATE)

        DB  82h
        DB  0D9h    ;Pwr mng      ;TURN ON SYNTH (CALIBRATE)


RXSETUP19200:
        DB  80h
```

```
          DB  67h       ;config reg

          DB  0A6h
          DB  40h       ;Freq set

          DB  96H
          DB  0C0H      ;RX set

          DB  98h
          DB  00h       ;TX set

          DB  0CEh
          DB  0D4H      ;Synch Char

          DB  0CCh
          DB  06h       ;PLL cmd

          DB  0C6h
          DB  11h       ;Data Rate 19.2Kbps

          DB  0C4h
          DB  0D7h      ;AFA

          DB  0CAh
          DB  83h       ;RX FIFO (FILL ALWAYS 87H) (SYNCH CHAR 83H)

          DB  0C2h
          DB  2Ch       ;Baseband Filter

          DB  82h
          DB  0D9h      ;Pwr mng

          DB  82h
          DB  0C9h      ;Pwr mng            ;TURN OFF SYNTH (CALIBRATE)

          DB  82h
          DB  0D9h      ;Pwr mng            ;TURN ON SYNTH (CALIBRATE)


TXDATA:
          DB  ' '
          DB  'R'
          DB  'F'
          DB  'M'
          DB  ' '
          DB  'R'
          DB  'F'
          DB  'I'
          DB  'C'
          DB  ' '
          DB  'R'
          DB  'A'
          DB  'N'
          DB  'G'
          DB  'E'
          DB  ' '
          DB  'T'
          DB  'E'
          DB  'S'
          DB  'T'
          DB  ' '
          DB  '4'
          DB  '3'
          DB  '3'
          DB  '.'
          DB  '9'
          DB  '2'
          DB  ' '
          DB  'M'
          DB  'H'
          DB  'z'

    END
```

Transmitter Range Test Assembly Code for C8051F330:

```
$include (c8051f330.inc)
;-- Bit Addressable
RXFLG       EQU     00H         ;RX FLAG AT REG 20H, BIT 0
STATFLG     EQU     01H         ;STATUS READ FLAG AT REG 20H, BIT 1

;-- Byte Addressable
HIGHBYTE    EQU     21H         ;HIGH BYTE OF IC WORD
LOWBYTE     EQU     22H         ;LOW BYTE OF IC WORD
TBLOFF      EQU     23H         ;Table offset value
CNT         EQU     24H         ;byte count

;PORT 0------------------------
RXLED       EQU     P0.0  ;O LED          1 P/P      0
ACKLED      EQU     P0.1  ;O LED          1 P/P   0
SLPLED      EQU     P0.2  ;O LED       1 P/P      0
TXLED       EQU     P0.3  ;O LED       1 P/P      0

RS232TX     EQU     P0.4    ;O                 1 P/P      1
RS232RX     EQU     P0.5    ;I                 0 O/D      1
SCK         EQU     P0.6    ;O                 1 P/P      0
SDO         EQU     P0.7    ;I                 0 O/D      1

;PORT 1------------------------
SDI         EQU     P1.0    ;O                 1 P/P      0
SEL         EQU     P1.1    ;O                 1 P/P      1
IRQ         EQU     P1.2    ;I                 0 O/D      1
FSEL        EQU     P1.3    ;O                 1   P/P    1
;LED        EQU     P1.3    ;O LED
DCLK        EQU     P1.4    ;I                 0 O/D      1
RSSI        EQU     P1.5    ;I                 0 O/D      1
VDDET       EQU     P1.6    ;I (or INT)   0 O/D      1
RNGTST      EQU     P1.7    ;I                 0 O/D      1

    ORG         00h
    LJMP    MAIN

    ORG         73H             ;TMR3 interrupt
    LJMP    INTT

    ORG         0FFh
;**** UART ISR *******************
;  THE TRANSMIT IS PERFORMED ON A TIMER INTERRUPT.
INTT:
    mov         TMR3CN,#00h     ;TMR3 OFF

;Transmit Packet
;Turn on Transmitter and begin TX preamble while loading other data
    mov         HIGHBYTE,#82h       ;load SPI address
    mov         LOWBYTE,#39h        ;Load SPI data, Turn on TX
    MOV         SPI0CN,#009h        ;SET nSEL LOW (CHIP SELECT)
    ACALL       SPISEND
    SETB        TXLED               ;Turn on LED

;Begin loading data payload
    mov         HIGHBYTE,#0B8h      ;load address of TX reg
    MOV         SPI0DAT,HIGHBYTE    ;ADDRESS THE TX REG
W2:JNB          SPIF,W2             ;WAIT FOR SPI TO FINISH
    CLR         SPIF                ;CLEAR SPI INT FLAG
    mov         CNT,#25h            ;load byte count(35)
    mov         DPTR,#TXDATA        ;load table pointer
    mov         TBLOFF,#0           ;set offset value
Z1: mov         A,TBLOFF            ;load offset value
    mov         R1,#LOWBYTE         ;load buffer pointer
    movc        A,@A+DPTR           ;load table byte
    mov         @R1,A               ;..into LOWBYTE buffer
    inc         TBLOFF              ;incr offset
    MOV         SPI0DAT,LOWBYTE         ;WRITE DATA BYTE TO SPI
```

```
W3:
     JNB        SPIF,W3                     ;WAIT FOR SPI TO FINISH
     CLR        SPIF                        ;CLEAR SPI INT FLAG
LP:
     JNB        SDO,LP                      ;loop until next byte load
     djnz       CNT,Z1
     MOV         SPI0CN,#00Dh               ;SET nSEL HIGH (DESELECT CHIP) TO WRITE TO NEW REGISTER
     mov        HIGHBYTE,#82h               ;load address
     mov        LOWBYTE,#19h                ;Load data, Turn OFF TX
     MOV        SPI0CN,#009h                ;SET nSEL LOW (CHIP SELECT)
     ACALL      SPISEND
     CLR        TXLED                       ;Turn off LED
DN:
     mov        TMR3CN,#04h                 ;TMR3 en
     MOV        SPI0CN,#00Dh                ;SET nSEL HIGH (DESELECT CHIP)

     RETI                                   ;RETURN
;*****************************************************************************************************

;------ Initialization functions -----------------
MAIN:

     mov        PCA0MD,    #000h
     mov        P0MDOUT,   #05Fh       ;0101 1111
     mov        P1MDOUT,   #0Bh        ;0000 1011
     mov        P0SKIP,    #00Fh
     mov        P0, #0B0h
     mov        P1, #0FEh              ;1111 1110
     mov        P2, #000h
     mov        XBR0,   #003h
     mov        XBR1,   #040h

Timer1_Init:
     mov        TMOD,      #021h       ;TMR1 Mode 2(2 8-bit), TMR0 Mode 1(16-bit)
     mov        TH1,       #0CBh       ;UART Reload value for 19.2Baud
     MOV        TL1,      #0CBH        ;INIT TMR1
     mov        CKCON,     #00h        ;SYSCLK/12
     mov        TCON,  #040h           ;TMR1 En

Timer3_Init:
     MOV        TMR3CN,    #00h        ;TMR3 dis, TMR3 clk = 255.208kHz,TMR3 MODE 16-BIT AUTORELD
     MOV        TMR3RLH,  #090h
     MOV        TMR3RLL,  #00h

UART_Init:
     mov        SCON0,     #030h       ;RXEN,RX INT active on stop bit

SPI_Init:
     mov         SPI0CFG,   #040h
     mov        SPI0CN,    #00Dh
     mov        SPI0CKR,   #00H

Oscillator_Init:
     mov        OSCICN,    #80h        ;SYSCLK = 24.5 MHz/8 = 3.06 MHz

Interrupts_Init:
     mov        PCA0MD,#00h
     MOV        EIE1,#80h
     SETB       PSPI0
          SETB  EA

;Main Code Section***********************************

          mov        R0,#00h      ;CLEAR REGISTERS
          mov        R1,#00h      ;
          mov        R2,#00h      ;
          mov        R3,#00h      ;
          mov        R4,#00h      ;
          mov        R5,#00h      ;
          mov        R6,#00h      ;
```

```
        mov     R7,#00h         ;
        CLR     STATFLG
        CLR     RXFLG

;Flash LED's

        SETB    RXLED           ;TURN ON LED
        SETB    TR0             ;TMR0 ENABLED
NXT:
        JNB     TF0,NXT         ;WAIT TIL TMR OVERFLOW THEN JUMP
        CLR     TF0             ;CLEAR OVERFLOW FLAG
        SETB    TXLED           ;TURN ON LED
NXT1:
        JNB     TF0,NXT1        ;WAIT TIL TMR OVERFLOW THEN JUMP
        CLR     TF0             ;CLEAR OVERFLOW FLAG
        SETB    ACKLED          ;TURN ON LED
NXT2:
        JNB     TF0,NXT2        ;WAIT TIL TMR OVERFLOW THEN JUMP
        CLR     TF0             ;CLEAR OVERFLOW FLAG
        SETB    SLPLED          ;TURN ON LED
NXT3:
        JNB     TF0,NXT3        ;WAIT TIL TMR OVERFLOW THEN JUMP
        CLR     TF0             ;CLEAR OVERFLOW FLAG
        CLR     RXLED
        CLR     TXLED
        CLR     ACKLED
        CLR     TR0             ;TMR0 DISABLED
;---------------------------------------------------------------

        mov     CKCON,#000h       ;SYSCLK/12

;*********************************************************
;******************  MAIN LOOP   **********************
;*********************************************************

        JNB     RNGTST,CFG2

;Configure Device

CFG1:
        mov     XBR1,#0C0h              ;disable port pullups
        mov     CNT,#09h                ;load byte counter
        mov     DPTR,#TXSETUP_PMAX          ;load table pointer
        mov     TBLOFF,#0               ;set offset value
A1:     mov     A,TBLOFF                ;load offset value
        mov     R1,#HIGHBYTE            ;load buffer pointer
        movc    A,@A+DPTR                ;load table byte
        mov     @R1,A                   ;...into buffer
        inc     TBLOFF                  ;incr offset
        inc     R1                      ;incr buffer
        mov     A,TBLOFF                ;load offset value
        movc    A,@A+DPTR                ;load table byte
        mov     @R1,A                   ;...into buffer
        inc     TBLOFF
        MOV     SPI0CN,#009h            ;SET nSEL LOW (CHIP SELECT)
        ACALL   SPISEND

        MOV     SPI0CN,#00Dh       ;SET nSEL HIGH (DESELECT CHIP)
        djnz    CNT,A1             ;decrement byte counter
        SJMP    START
CFG2:
        mov     XBR1,#0C0h              ;disable port pullups
        mov     CNT,#09h                ;load byte counter
        mov     DPTR,#TXSETUP_0dBm          ;load table pointer
        mov     TBLOFF,#0               ;set offset value
YA1:    mov     A,TBLOFF                ;load offset value
        mov     R1,#HIGHBYTE            ;load buffer pointer
        movc    A,@A+DPTR               ;load table byte
        mov     @R1,A                   ;...into buffer
        inc     TBLOFF                  ;incr offset
        inc     R1                  ;incr buffer
        mov     A,TBLOFF            ;load offset value
        movc    A,@A+DPTR           ;load table byte
        mov     @R1,A               ;...into buffer
        inc     TBLOFF
        MOV     SPI0CN,#009h        ;SET nSEL LOW (CHIP SELECT)
        ACALL   SPISEND
        MOV   SPI0CN,#00Dh          ;SET nSEL HIGH (DESELECT CHIP)
        djnz    CNT,YA1             ;decrement byte counter

;*******  Start Timer3 and LOOP til next TX  ******************
START:
        mov     TMR3CN,#04h         ;TMR3 en
IL:     NOP
        SJMP    IL

;***** READ CHIP STATUS *****
STATGO:
        SETB        TXLED
        CLR         STATFLG         ;IF SET, CLEAR FLAG FIRST
        MOV         SPI0CN,#009h    ;SET nSEL LOW (CHIP SELECT)
        MOV         SPI0DAT,#00H    ;WRITE DUMMY BYTE TO SPI
WAITSS:
        JNB         SPIF, WAITSS
        CLR         SPIF
        MOV         R0,SPI0DAT      ;WRITE HIGH BYTE TO BUFFER
        MOV         SPI0DAT,#00H    ;WRITE DUMMY BYTE TO SPI
WAITZZ:
        JNB         SPIF, WAITZZ
        CLR         SPIF
        MOV         R1,SPI0DAT      ;WRITE LOW BYTE TO BUFFER

; *** SPI SEND ***********************************************
SPISEND:
        ;(Chip Select already LOW)
        MOV         SPI0DAT,HIGHBYTE        ;WRITE HIGH BYTE TO SPI
WAIT4:
        JNB         SPIF,WAIT4          ;WAIT FOR SPI TO FINISH FIRST XFER,LOOP IF STILL BUSY
        CLR         SPIF                ;CLEAR SPI INT FLAG TO PROCEED
        MOV         SPI0DAT,LOWBYTE         ;WRITE LOW BYTE TO SPI
WAIT5:
        JNB         SPIF,WAIT5          ;WAIT FOR SPI TO FINISH 2ND XFER,LOOP IF STILL BUSY
        CLR         SPIF                ;CLEAR SPI INT FLAG TO PROCEED
RTRN:
        CLR         ACKLED
        RET

TXSETUP_PMAX: ;PMAX
        DB 80h
        DB 0A7h         ;config reg

        DB 0A6h
        DB 40h      ;Freq set

        DB 82h
        DB 19h      ;Pwr mng

        DB 98h
        DB 10h      ;TX set

        DB 0CEh
        DB 0E2h     ;Synch Char

        DB 0CCh
        DB 06h      ;PLL cmd

        DB 0C6h
        DB 91h      ;Data Rate 2400

        DB 0CAh
        DB  81h     ;Dis RESET

        DB  0C4H
        DB  0D7H    ;AFA

TXSETUP_0dBm: ;0dBm
        DB  80h
        DB  0A7h    ;config reg

        DB  0A6h
        DB  40h     ;Freq set

        DB  82h
        DB  19h     ;Pwr mng

        DB  98h
        DB  10h     ;TX set

        DB  0CEh
        DB  0E2h    ;Synch Char

        DB  0CCh
        DB  06h     ;PLL cmd

        DB  0C6h
        DB  91H     ;Data Rate 2400

        DB  0CAh
        DB  81h     ;Dis RESET

        DB  0C4H
        DB  0D7H    ;AFA

TXDATA:
        DB  0AAh        ;0
        DB  0AAh        ;0
        DB  0AAh        ;0
        DB  0AAh        ;1
        DB  2Dh         ;2
        DB  0D4h        ;3   'FOR TRC101 AND RXC101'
        DB  ' '         ;4
        DB  'R'         ;5
        DB  'F'         ;6
        DB  'M'         ;7
        DB  ' '         ;8
        DB  'R'         ;9
        DB  'F'         ;A
        DB  'I'         ;B
        DB  'C'         ;C
        DB  ' '         ;D
        DB  'R'         ;E
        DB  'A'         ;F
        DB  'N'         ;10
        DB  'G'         ;11
        DB  'E'         ;12
        DB  ' '         ;13
        DB  'T'         ;14
        DB  'E'         ;15
        DB  'S'         ;16
        DB  'T'         ;17
        DB  ' '         ;18
        DB  '4'         ;19
        DB  '3'         ;1A
        DB  '3'         ;1B
        DB  '.'         ;1C
        DB  '9'         ;1D
        DB  '2'         ;1E
        DB  ' '         ;1F
        DB  'M'         ;20
        DB  'H'         ;21
```

```
          DB  'z'          ;22
          DB  0DH          ;23
          DB  07H          ;23
          DB  98H          ;24
          DB  00H          ;25

     END
```