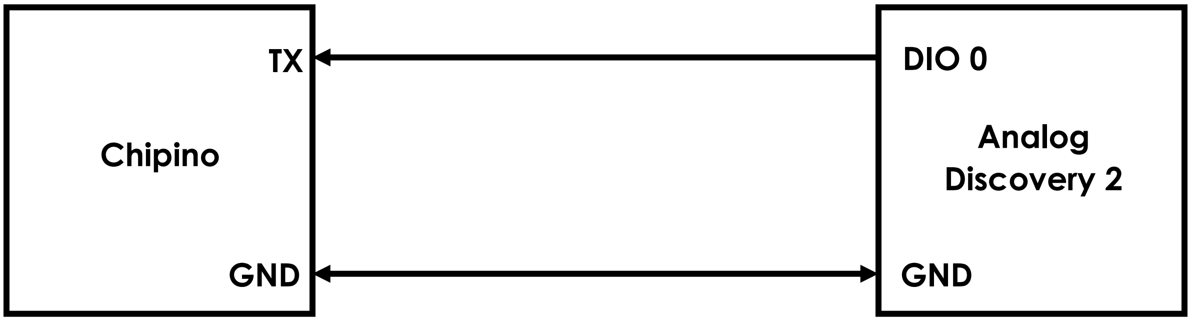
QUESTION 1

```
;/----------------------PROGRAM DESCRIPTION----------------------\;
       ;CONTINUOUSLY SEND THE MESSAGE "Trix are for kids!"
    ;9600 BPS, 8 DATA BITS, 1 START BIT, 1 STOP BIT, NO PARITY
                    ;TTL SERIAL FORMAT
;/------------------------------LIST------------------------------\;
LIST P=18F2420, MM=OFF, R=HEX, ST=OFF, X=OFF
;/--------------------------CONFIG BITS---------------------------\;
; CONFIG1H
  CONFIG  OSC = HS              ; Oscillator Selection bits (HS oscillator)
  CONFIG  FCMEN = OFF           ; Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor
disabled)
  CONFIG  IESO = OFF            ; Internal/External Oscillator Switchover bit (Oscillator
Switchover mode disabled)


; CONFIG2L
  CONFIG  PWRT = OFF            ; Power-up Timer Enable bit (PWRT disabled)
  CONFIG  BOREN = OFF           ; Brown-out Reset Enable bits (Brown-out Reset disabled in hardware
and software)
  CONFIG  BORV = 3              ; Brown Out Reset Voltage bits (Minimum setting)


; CONFIG2H
  CONFIG  WDT = OFF             ; Watchdog Timer Enable bit (WDT disabled (control is placed on the
SWDTEN bit))
  CONFIG  WDTPS = 32768         ; Watchdog Timer Postscale Select bits (1:32768)


; CONFIG3H
  CONFIG  CCP2MX = PORTC        ; CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
  CONFIG  PBADEN = OFF          ; PORTB A/D Enable bit (PORTB<4:0> pins are configured as digital
I/O on Reset)
  CONFIG  LPT1OSC = OFF         ; Low-Power Timer1 Oscillator Enable bit (Timer1 configured for
higher power operation)
  CONFIG  MCLRE = ON            ; MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)
; CONFIG4L
  CONFIG  STVREN = OFF          ; Stack Full/Underflow Reset Enable bit (Stack full/underflow will
not cause Reset)
  CONFIG  LVP = OFF             ; Single-Supply ICSP Enable bit (Single-Supply ICSP disabled)
  CONFIG  XINST = OFF           ; Extended Instruction Set Enable bit (Instruction set extension
and Indexed Addressing mode disabled (Legacy mode))
```

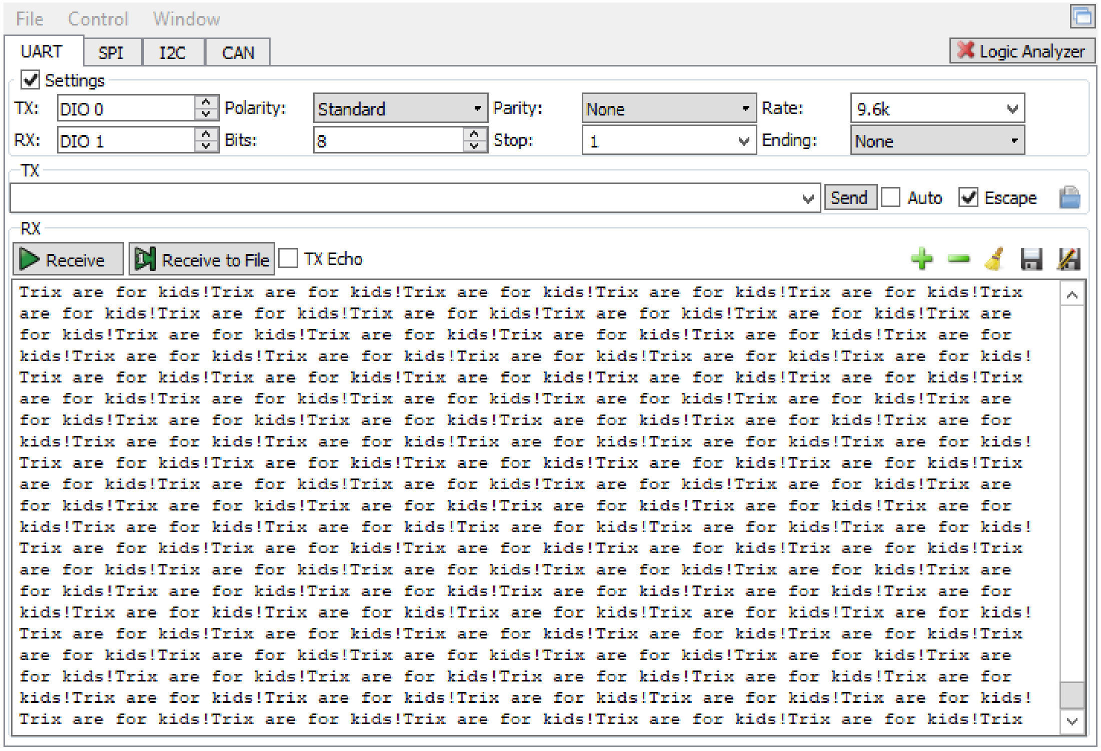
```
; CONFIG5L
  CONFIG  CP0 = OFF             ; Code Protection bit (Block 0 (000800-001FFFh) not code-protected)
  CONFIG  CP1 = OFF             ; Code Protection bit (Block 1 (002000-003FFFh) not code-protected)
; CONFIG5H
  CONFIG  CPB = OFF             ; Boot Block Code Protection bit (Boot block (000000-0007FFh) not
code-protected)
  CONFIG  CPD = OFF             ; Data EEPROM Code Protection bit (Data EEPROM not code-protected)


; CONFIG6L
  CONFIG  WRT0 = OFF            ; Write Protection bit (Block 0 (000800-001FFFh) not write-
protected)
  CONFIG  WRT1 = OFF            ; Write Protection bit (Block 1 (002000-003FFFh) not write-
protected)


; CONFIG6H
  CONFIG  WRTC = OFF            ; Configuration Register Write Protection bit (Configuration
registers (300000-3000FFh) not write-protected)
  CONFIG  WRTB = OFF            ; Boot Block Write Protection bit (Boot block (000000-0007FFh) not
write-protected)
  CONFIG  WRTD = OFF            ; Data EEPROM Write Protection bit (Data EEPROM not write-
protected)


; CONFIG7L
  CONFIG  EBTR0 = OFF           ; Table Read Protection bit (Block 0 (000800-001FFFh) not protected
from table reads executed in other blocks)
  CONFIG  EBTR1 = OFF           ; Table Read Protection bit (Block 1 (002000-003FFFh) not protected
from table reads executed in other blocks)


; CONFIG7H
  CONFIG  EBTRB = OFF           ; Boot Block Table Read Protection bit (Boot block (000000-0007FFh)
not protected from table reads executed in other blocks)
;/-----------------INCLUDE LIBRARY FOR PIC18F2420----------------\;
#include <p18f2420.inc>
;/---------------SETUP FOR POWER UP AND INTERRUPTS---------------\;
	ORG 0x00
	GOTO START	    ;Go to beginning of program
	ORG 0x08
	RETFIE
	ORG 0x18
	RETFIE
```

```
;/--------------------START OF ACTUAL PROGRAM--------------------\;
START
;/SETUP
       ;/Configure TXSTA register
       MOVLW B'00100000'   ;Enable transmit, 8-bit transmission
       MOVWF TXSTA         ;Asynchronous mode

       ;/Configure baud rate settings
       MOVLW D'25'         ;9600 bps. [(16 MHz / 64) / 9600] – 1 = 25.04 -> 25
       MOVWF SPBRG

       ;/Make TX pin an output pin
       BCF TRISC, TX

       ;/Enable Serial Port
       BSF RCSTA, SPEN
;/------------------------------MAIN-----------------------------\;
;/Continuously send “Trix are for kids!”
       ;/Load table pointer with address where MESSAGE is stored
INIT   MOVLW upper(MESSAGE)
       MOVWF TBLPTRU
       MOVLW high(MESSAGE)
       MOVWF TBLPTRH
       MOVLW low(MESSAGE)
       MOVWF TBLPTRL

       ;/Read from table, increment pointer, then send character
NEXT   TBLRD*+
       MOVF TABLAT, W             ;WREG = TABLAT
       BZ INIT                    ;Reinitialize TBLPTR when WREG = null
       RCALL SEND                 ;Transmit character
       BRA NEXT                   ;Repeat until null
;/--------------------------SUBROUTINES--------------------------\;
SEND
L1     BTFSS PIR1, TXIF           ;Make sure the last bit of the previous frame has been sent
       BRA L1
       MOVWF TXREG                ;Transmit character
       RETURN
```

```
;/------------------------DECLARATIONS------------------------\;
;*This is intentionally placed at the bottom as opposed to the top
;to let the assembler optimize program memory space and allow us
;to load the TBLPTR using the "upper", "high", and "low" commands
;on lines 82, 84, and 86 without having to know an actual memory address value
MESSAGE     DB "Trix are for kids!", 0
END
```



Wiring Block Diagram



☑ Settings

TX: DIO 0 Polarity: Standard Parity: None Rate: 9.6k

RX: DIO 1 Bits: 8 Stop: 1 Ending: None

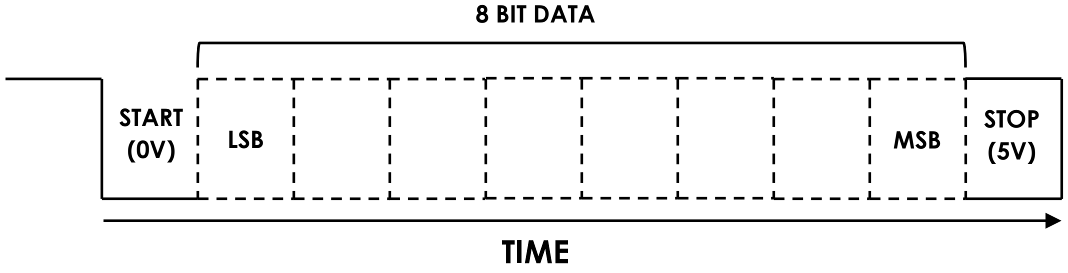
☐ Auto ☑ Escape

☐ TX Echo

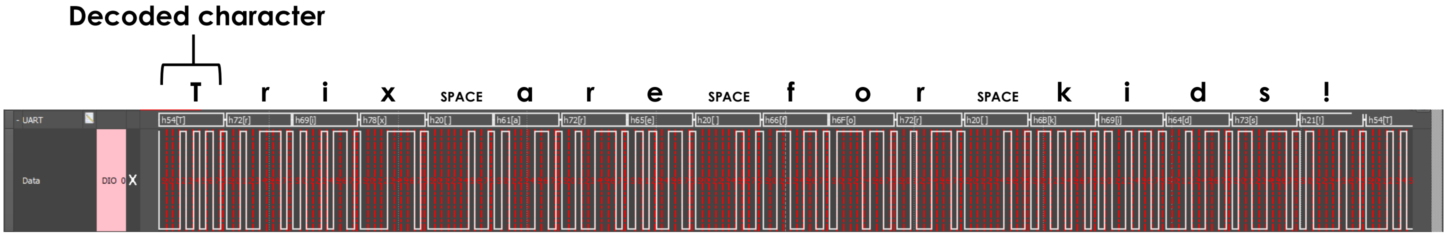
```
Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix
are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are
for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for
kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!
Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix
are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are
for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for
kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!
Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix
are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are
for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for
kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!
Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix
are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are
for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for
kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!
Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix
are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are
for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for
kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!
Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix are for kids!Trix
```

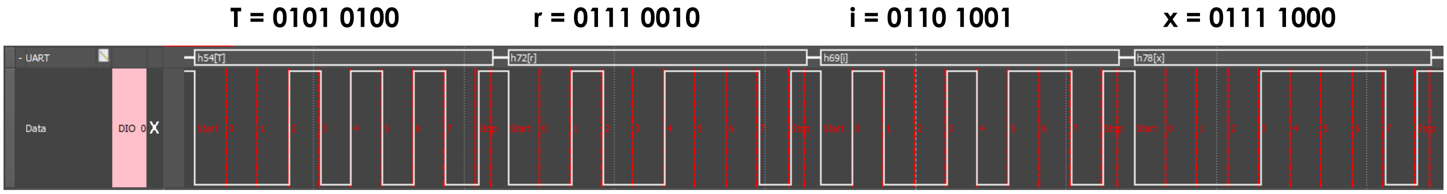
Measurement using the Protocol function in WaveForms

TTL RS232 FORMAT



Measurements using the Logic Analyzer function in WaveForms





QUESTION 2

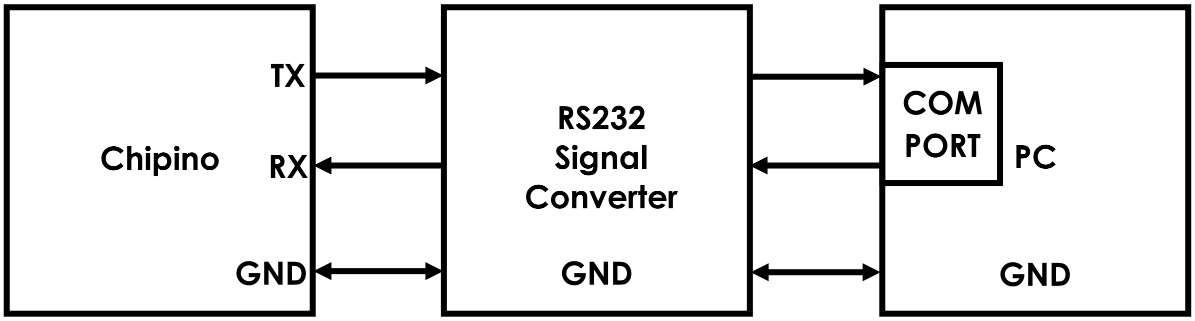
```
;/----------------------PROGRAM DESCRIPTION----------------------\;

              ;RECEIVE DATA FROM PC AND RETURN IT

    ;2400 BPS, 8 DATA BITS, 1 START BIT, 1 STOP BIT, NO PARITY

                  ;USES THE POLLING METHOD

;/------------------------------LIST------------------------------\;

LIST P=18F2420, MM=OFF, R=HEX, ST=OFF, X=OFF

;/--------------------------CONFIG BITS---------------------------\;

; CONFIG1H

  CONFIG  OSC = HS              ; Oscillator Selection bits (HS oscillator)

  CONFIG  FCMEN = OFF           ; Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor
disabled)

  CONFIG  IESO = OFF            ; Internal/External Oscillator Switchover bit (Oscillator
Switchover mode disabled)


; CONFIG2L

  CONFIG  PWRT = OFF            ; Power-up Timer Enable bit (PWRT disabled)

  CONFIG  BOREN = OFF           ; Brown-out Reset Enable bits (Brown-out Reset disabled in hardware
and software)

  CONFIG  BORV = 3              ; Brown Out Reset Voltage bits (Minimum setting)


; CONFIG2H

  CONFIG  WDT = OFF             ; Watchdog Timer Enable bit (WDT disabled (control is placed on the
SWDTEN bit))

  CONFIG  WDTPS = 32768         ; Watchdog Timer Postscale Select bits (1:32768)


; CONFIG3H

  CONFIG  CCP2MX = PORTC        ; CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)

  CONFIG  PBADEN = OFF          ; PORTB A/D Enable bit (PORTB<4:0> pins are configured as digital
I/O on Reset)

  CONFIG  LPT1OSC = OFF         ; Low-Power Timer1 Oscillator Enable bit (Timer1 configured for
higher power operation)

  CONFIG  MCLRE = ON            ; MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

; CONFIG4L

  CONFIG  STVREN = OFF          ; Stack Full/Underflow Reset Enable bit (Stack full/underflow will
not cause Reset)

  CONFIG  LVP = OFF             ; Single-Supply ICSP Enable bit (Single-Supply ICSP disabled)

  CONFIG  XINST = OFF           ; Extended Instruction Set Enable bit (Instruction set extension
and Indexed Addressing mode disabled (Legacy mode))
```

```
; CONFIG5L
  CONFIG  CP0 = OFF             ; Code Protection bit (Block 0 (000800-001FFFh) not code-protected)
  CONFIG  CP1 = OFF             ; Code Protection bit (Block 1 (002000-003FFFh) not code-protected)
; CONFIG5H
  CONFIG  CPB = OFF             ; Boot Block Code Protection bit (Boot block (000000-0007FFh) not
code-protected)
  CONFIG  CPD = OFF             ; Data EEPROM Code Protection bit (Data EEPROM not code-protected)


; CONFIG6L
  CONFIG  WRT0 = OFF            ; Write Protection bit (Block 0 (000800-001FFFh) not write-
protected)
  CONFIG  WRT1 = OFF            ; Write Protection bit (Block 1 (002000-003FFFh) not write-
protected)


; CONFIG6H
  CONFIG  WRTC = OFF            ; Configuration Register Write Protection bit (Configuration
registers (300000-3000FFh) not write-protected)
  CONFIG  WRTB = OFF            ; Boot Block Write Protection bit (Boot block (000000-0007FFh) not
write-protected)
  CONFIG  WRTD = OFF            ; Data EEPROM Write Protection bit (Data EEPROM not write-
protected)


; CONFIG7L
  CONFIG  EBTR0 = OFF           ; Table Read Protection bit (Block 0 (000800-001FFFh) not protected
from table reads executed in other blocks)
  CONFIG  EBTR1 = OFF           ; Table Read Protection bit (Block 1 (002000-003FFFh) not protected
from table reads executed in other blocks)


; CONFIG7H
  CONFIG  EBTRB = OFF           ; Boot Block Table Read Protection bit (Boot block (000000-0007FFh)
not protected from table reads executed in other blocks)
;/-----------------INCLUDE LIBRARY FOR PIC18F2420----------------\;
#include <p18f2420.inc>
;/-------------------------DECLARATIONS-------------------------\;
VAR    EQU 0x0A                 ;RAM location to copy received data to. VAR for ‘variable’
;/---------------SETUP FOR POWER UP AND INTERRUPTS---------------\;
       ORG 0x00
       GOTO START               ;Go to beginning of program
       ORG 0x08
       RETFIE
       ORG 0x18
       RETFIE
```

```
;/--------------------START OF ACTUAL PROGRAM--------------------\;

START

;/SETUP

       ;/Configure TXSTA register

       MOVLW B'00100000'          ;Enable transmit, 8-bit transmission

       MOVWF TXSTA                ;Asynchronous mode


       ;/Configure RCSTA register

       MOVLW B'10010000'          ;Enable serial port, continuously receive 8 bit data, no framing
                                   error bit, no overrun error bit

       MOVWF RCSTA


       ;/Configure baud rate settings

       MOVLW D'103'               ;2400 bps. [(16 MHz / 64) / 2400] - 1 = 103.16 -> 103

       MOVWF SPBRG


       ;/Make TX pin an output pin

       BCF TRISC, TX


       ;/Make RX pin an input pin

       BSF TRISC, RX

;/------------------------------MAIN------------------------------\;

;/Wait to receive data

RX1    BTFSS PIR1, RCIF           ;Wait to receieve data. Move on until entire data packet has been
                                   received

       BRA RX1

       MOVFF RCREG, VAR           ;Copy receieved data to RAM


;/Print "You typed in: " with received character

       RCALL PRINTCHAR


;/Go back to polling to receive another character

       BRA RX1
```

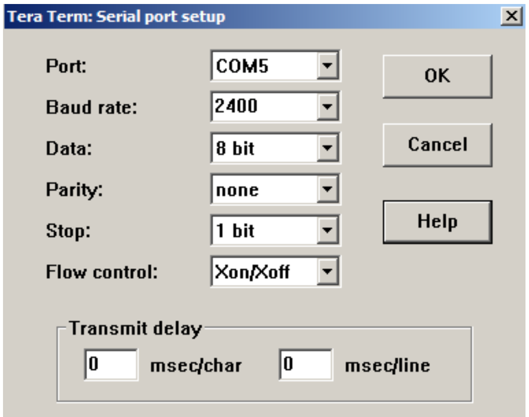
```
    ;/--------------------------SUBROUTINES--------------------------\;
    PRINTCHAR                        ;PRINTCHAR for “print character”
    ;/Print "You typed in: " with received character
           ;/Load table pointer with address where MESSAGE is stored
106        MOVLW upper(MESSAGE)
           MOVWF TBLPTRU
108        MOVLW high(MESSAGE)
           MOVWF TBLPTRH
110        MOVLW low(MESSAGE)
           MOVWF TBLPTRL

           ;/Read from table, increment pointer, then send character
    NEXT   TBLRD*+
           MOVF TABLAT, W            ;WREG = TABLAT
           BZ TX1                    ;Go to TX1 when WREG = null
           RCALL SEND                ;Transmit character
           BRA NEXT                  ;Read until null

    ;/Transmit received character
    TX1    MOVF VAR, WREG            ;Copy received character to WREG
           RCALL SEND                ;Send character
           RCALL LFCR                ;Send Newline and Carriage Return
           RETURN                    ;Exit HIGHLIGHT subroutine. Go to line 102
    ;/---------------------------------------------------------------\;
    SEND
    ;/Transmit subroutine
    L1     BTFSS PIR1, TXIF          ;Make sure the last bit of the previous frame has been sent
           BRA L1
           MOVWF TXREG               ;Send character
           RETURN
    ;/---------------------------------------------------------------\;
    LFCR                             ;LFCR for “Line Feed/Carriage Return”
    ;/Transmit Newline and Carriage Return
           MOVLW H'D'                ;Transmit Newline/Line Feed
           RCALL SEND
           MOVLW H'A'                ;Transmit Carriage Return
           RCALL SEND
           RETURN
```

```
;/------------------------DEFINITIONS-------------------------\;
;*This is intentionally placed at the bottom as opposed to the top
;to let the assembler optimize program memory space and allow us
;to load the TBLPTR using the "upper", "high", and "low" commands
;on lines 106, 108, and 110 without having to know an actual memory address value
MESSAGE      DB "You typed in: ", 0
END
```



Wiring Block Diagram

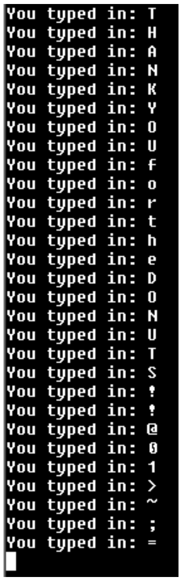
Screenshots from Tera Term terminal emulator



Port: COM5
Baud rate: 2400
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: Xon/Xoff

Transmit delay
0 msec/char
0 msec/line

Serial COM settings



```
You typed in: T
You typed in: H
You typed in: A
You typed in: N
You typed in: K
You typed in: Y
You typed in: O
You typed in: U
You typed in: f
You typed in: o
You typed in: r
You typed in: t
You typed in: h
You typed in: e
You typed in: D
You typed in: O
You typed in: N
You typed in: U
You typed in: T
You typed in: S
You typed in: !
You typed in: !
You typed in: @
You typed in: 0
You typed in: 1
You typed in: >
You typed in: ~
You typed in: ;
You typed in: =
█
```

Sample Output