

# RS232 ASCII String Generator

PROJECT

by Richard Grodzik

**R**S232 communications has been with us for over twenty five years and the most common baud rate (speed) is still 9600. Slow by modern day standards, but it is probably the most reliable of all computer communications to such an extent that complex crc error checking is not required and even simple parity checking is disabled. Most PC's/Laptops are equipped with a serial COM1/COM2 port identified by the 9 pin D-type connector. But with the progress of PC technology this RS232 port is gradually being omitted in favour of USB and firewire

communication ports, which offer megabit speed. Unfortunately the complexity of the communication protocol which uses packet data similar to networks precludes the hobbyist/amateur/semi-professional, at this point in time, from interfacing home-brewed projects to these ports.

RS232 is still however used universally in a diverse range of equipment including data loggers, electronic weighing scales, meters and of course Terminals that offer 100% reliability. I would not be surprised if the National Lottery computer network does not use RS232 terminal communications in some form. What prompted me

to design this project was my experimentation with real-time data (ASCII RS232) streaming to the Excel spreadsheet. As data arrives at the PC's COM port, it is immediately printed into the cells of the spreadsheet. So, I needed an RS232 source, so short of purchasing another PC, I decided to make a small unit which would allow me to generate RS232 data with a couple of buttons.

## Description.

The RS232 ASCII string generator is based on a LCD display which can be either a 1 line 16 character or 2 line display, since the same software will work on either - the 2 line display is preferred since it separated the input data from the ASCII String. Two simple press switches SW1 SW2 are used to produce any character found on a full size keyboard. Each switch is used to select the least significant (4 bits) and most significant nibble of the ASCII character. Pressing either switch (They also work together) with increment the data from 0 to F. Keeping the switch pressed with cause automatic incrementation at 0.5Hz. So by selecting 41h will print the ASCII

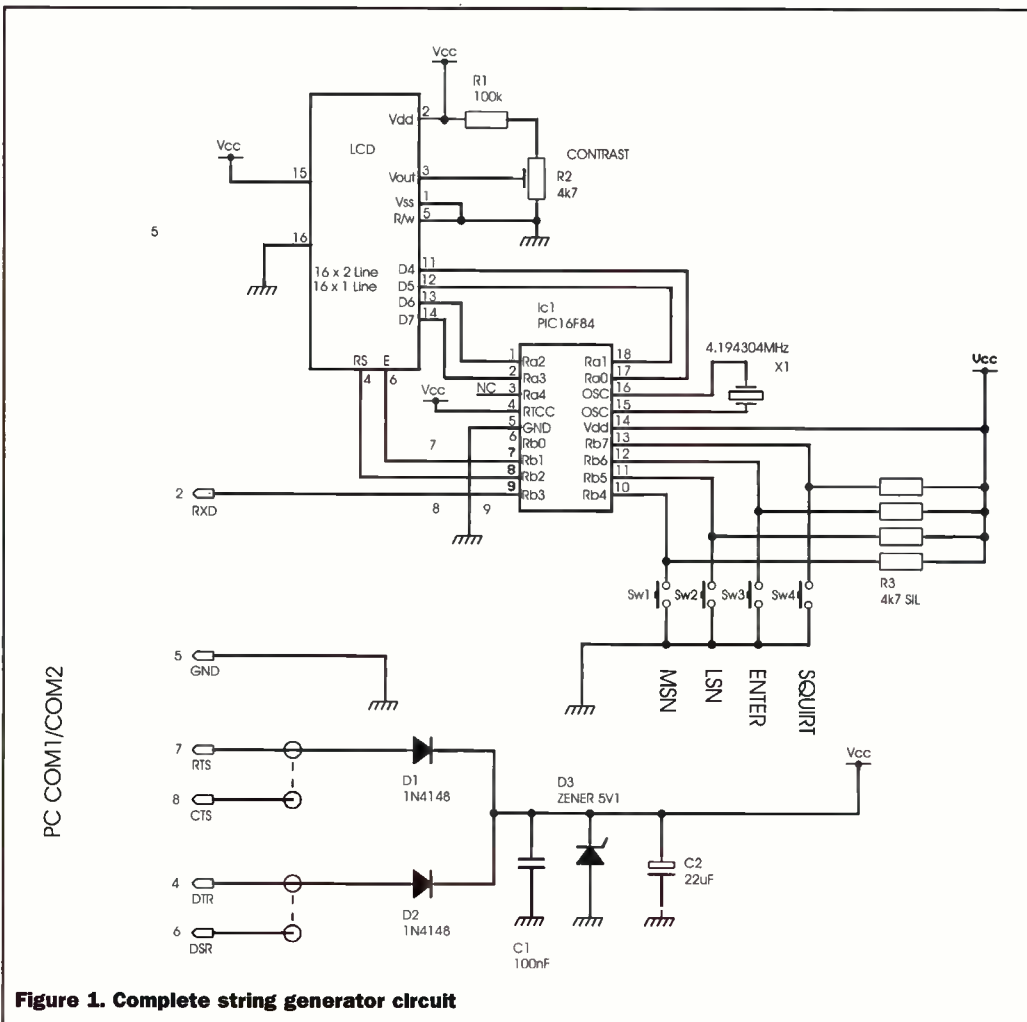
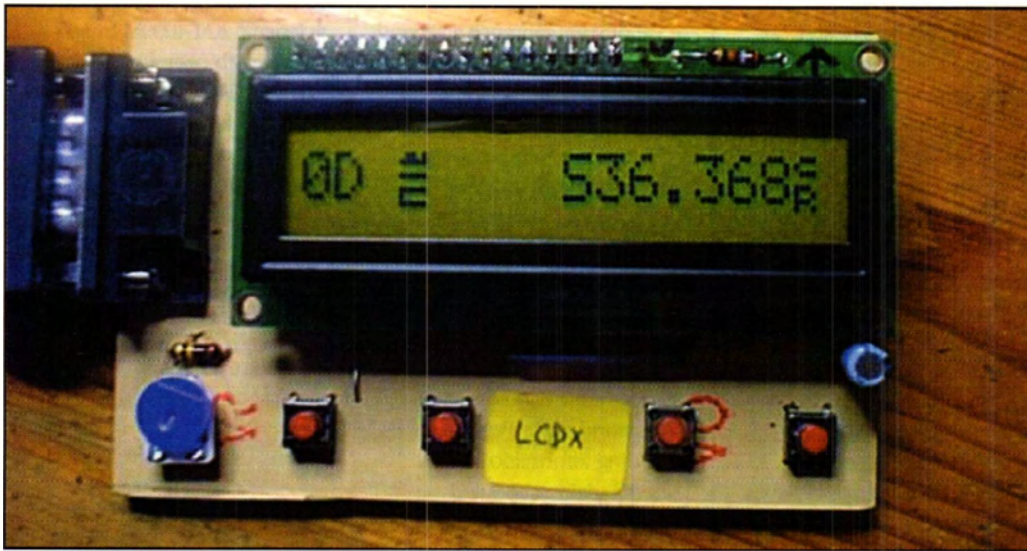


Figure 1. Complete string generator circuit

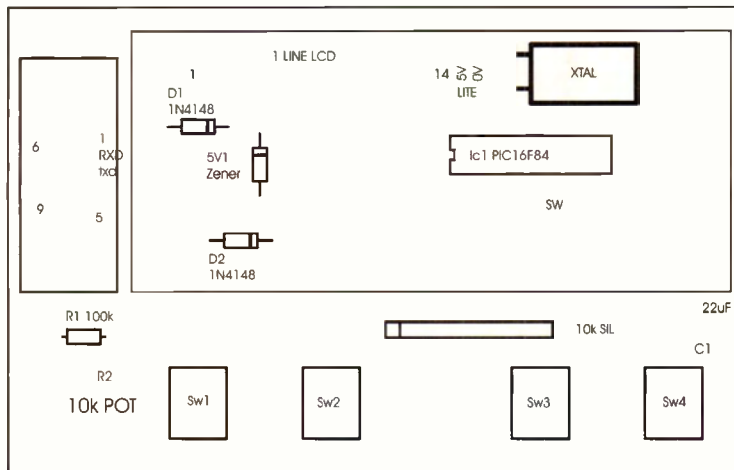


Figure 2. Component overlay

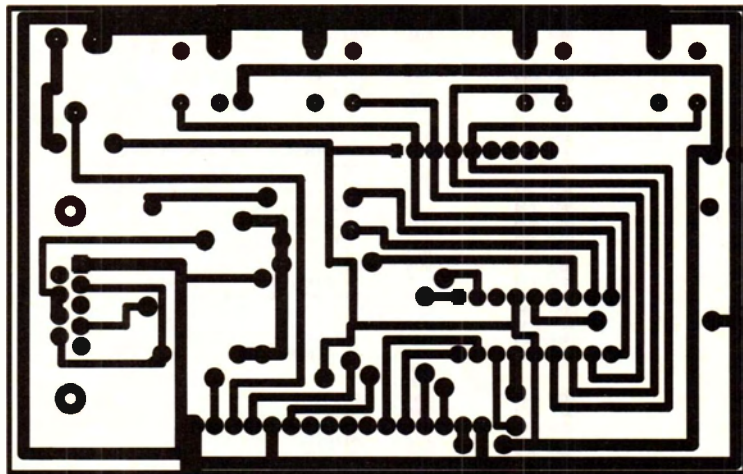


Figure 3. PCB track bottom side

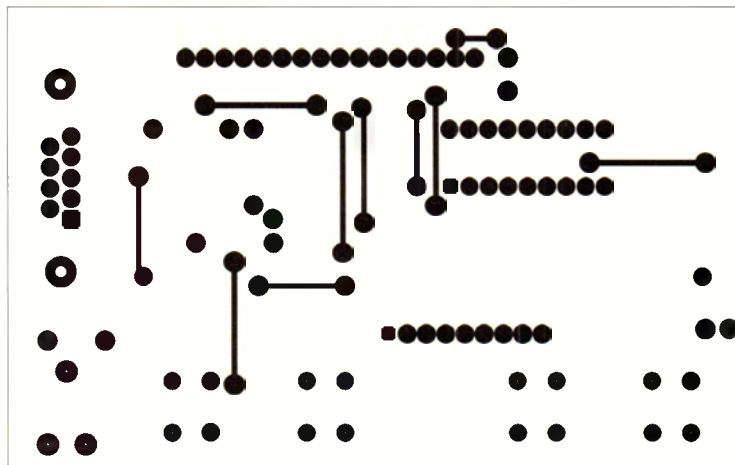


Figure 4. PCB track top side showing interconnections

character 'A' on the display. An ASCII table found in most publications is used to check the Byte required. I still use a table, as I still don't know all the codes! Of course control codes such as linefeed (0Ah) can be selected - a carriage return code (0Dh) will print cr on the display. Each character is selected by pressing the 'ENTER' key (SW3), and gradually the entire ASCII character string is built up (Maximum of 20 characters). Then simply pressing the SQUIRT key (SW4) will output the character string at 9600 baud at precise 1 second intervals.

The power supply for the unit is taken from the PC's serial port. The hardware handshake lines are rarely used and are an ideal source of voltage (9 volts at 20mA). The RTS/CTS and DTR/DSR pins are tied together and they feed the two diodes D1 and D2. These 2 diodes can be incorporated in the 9 pin hood of the connector. A simple 5V1 zener diode then supplies 5 volts to the circuitry. R2 is a contrast control, and provision is made on the PCB artwork for a backlit LCD (pins 15 and 16 supply the power for the light). When choosing a backlit display select a low current type. A 4.194304 crystal allows for precise timing for the 9600-baud rate and for the 1 second delay.

**Note:** A Pre-programmed PIC is available from the author for £15.00 pounds sterling, P&P inc. from R.Grodzik (Micros) 53 Chelmsford Road, Bradford BD3 8QN U.K.

Also see the authors web-site at: <http://members.netscapeonline.co.uk/dgrodzik>

## PARTS LIST

### Resistors

R1	100K
R2	4K7 Potentiometer
R3	4K7 SIL network

### Capacitors

C1	100nF
C2	22µF

### Semiconductors

D1,D2	1N4148
D3	5V1 zener
IC1	PIC16F84 (see note)

### Others

SW1-SW4 press to make button switch  
 9 pin D-Type connectors, LCD display 1 or 2 line 16 character display.  
 X1 crystal 4.194304MHz



```

0096 0814      0241  MOVF LCDDATA,0 ;LSNIBBLE
0097 0085      0242  MOVWF PORT_A
0098 1106      0243  BCF PORT_B,RS
0099 1486      0244  BSF PORT_B,E
009A 20C4      0245  CALL LDELAY
009B 1086      0246  BCF PORT_B,E
009C 3400      0247  RETLW 0
                0248
009D          0249  WRITE
009E 008D      0250  MOVWF A_N
009F 0E00      0251  SWAPF A_N,0
009F 0085      0252  MOVWF PORT_A
00A0 1506      0253  BCF PORT_B,RS
00A1 1486      0254  BSF PORT_B,E
00A2 20C8      0255  CALL DELAY_DATA
00A3 1086      0256  BCF PORT_B,E
00A4 080D      0257  MOVF A_N,0
00A5 0085      0258  MOVWF PORT_A
00A6 1506      0259  BSF PORT_B,RS
00A7 1486      0260  BSF PORT_B,E
00A8 20C8      0261  CALL DELAY_DATA
00A9 1086      0262  BCF PORT_B,E
00AA 3400      0263  RETLW 0
                0264
                0265

```

MPASM 01.02.05 Intermediate LCORG1.ASM 11-28-2000 18:28:22 PAGE 6

LOC OBJECT CODE LINE SOURCE TEXT  
VALUE

```

00A8          0266  TX0_DATA
00A8 1586      0267  BSF PORT_B,TXD
00AC 208E      0268  NEXT CALL DELAY
00AD 0C90      0269  RRF BUFFER,1
00AE 1803      0270  BTFSZ STATUS,0
00AF 1586      0271  BSF PORT_B,TXD
                0272
00B0 1C03      0273  BTFSZ STATUS,0
00B1 1186      0274  BCF PORT_B,TXD
00B2 088F      0275  DECFSZ COUNTER,1
00B3 28AC      0276  GOTO NEXT
00B4 208E      0277  CALL DELAY
00B5 1186      0278  BCF PORT_B,TXD
00B6 208E      0279  CALL DELAY
00B7 208E      0280  CALL DELAY
00B8 3400      0281  RETLW 0
                0282
00B9          0283  CONVERT
00B9 3AFF      0284  XORLW 0FFH
00BA 0090      0285  MOVWF BUFFER
00BB 3008      0286  MOVWL B
00BC 008F      0287  MOVWF COUNTER
00BD 3400      0288  RETLW 0
                0289
00BE          0290  DELAY
00BE 3021      0291  MOVWL BAUD9600
00BF 0095      0292  MOVWF DLYCNT
00C0 0895      0293  REDX DECFSZ DLYCNT,1
00C1 28C0      0294  GOTO REDX
00C2 0000      0295  NOP
00C3 3400      0296  RETLW 0
                0297
00C4          0298  LDELAY
00C4 3003      0299  MOVWL B'00000011'
00C5 0062      0300  OPTION
00C6 0181      0301  CLRF RTCC
00C7          0302  L_DELAY
00C7 0064      0303  CLRWD
00C8 1F81      0304  BTFSZ RTCC,7
00C9 28C7      0305  GOTO L_DELAY
00CA 3400      0306  RETLW 0
                0307
00CB          0308  DELAY_DATA
00CB 30FF      0309  MOVWL DEBOUNCE
00CC 0095      0310  MOVWF DLYCNT
00CD 0895      0311  REDB DECFSZ DLYCNT,1
00CE 28C0      0312  goto REDB
00CF 0064      0313  CLRWD
00D0 3400      0314  RETLW 0
                0315
                0316
00D1          0317  BINARY
00D1 0782      0318  ADOWF PC

```

MPASM 01.02.05 Intermediate LCORG1.ASM 11-28-2000 18:28:22 PAGE 7

LOC OBJECT CODE LINE SOURCE TEXT  
VALUE

```

00D2 3430      0319  RETLW '0'
00D3 3431      0320  RETLW '1'
00D4 3432      0321  RETLW '2'
00D5 3433      0322  RETLW '3'
00D6 3434      0323  RETLW '4'
00D7 3435      0324  RETLW '5'
00D8 3436      0325  RETLW '6'
00D9 3437      0326  RETLW '7'
00DA 3438      0327  RETLW '8'
00DB 3439      0328  RETLW '9'
00DC 3441      0329  RETLW 'A'
00DD 3442      0330  RETLW 'B'
00DE 3443      0331  RETLW 'C'
00DF 3444      0332  RETLW 'D'
00E0 3445      0333  RETLW 'E'
00E1 3446      0334  RETLW 'F'
                0335
                0336
                0337
00E2          0338  LONGTIME
00E2 3008      0339  ;1000.23 MS = 32
                0340
00E3 3008      0341  MOVWL B
00E4 3007      0342  MOVWL B'00000111'
00E5 0062      0343  OPTION
00E6 0181      0344  LONG CLR RTCC
00E7 0064      0345  LONG CLRWD
00E8 0000      0346  NOP
00E9 1F81      0347  BTFSZ RTCC,7
00EA 28E7      0348  GOTO LONG
00EB 0898      0349  DECFSZ SLOW,1
00EC 28E6      0350  GOTO LONGX
00ED 3400      0351  RETLW 0
                -0352
                0353
                0354
00EE          0355  SECOND
00EE 3020      0356  ;1000.23 MS = 32
00EF 0098      0357  MOVWL .32
00F0 3007      0358  MOVWF SLOW
00F1 0062      0359  MOVWL B'00000111'
00F2 0181      0360  OPTION
00F3 0064      0361  LONG CLR RTCC
00F4 0000      0362  LONG CLRWD
00F5 1F81      0363  NOP
00F6 28F3      0364  BTFSZ RTCC,7
                0365  GOTO LONG

```

```

00F7 0898      0366  DECFSZ SLOW,1
00F8 28F2      0367  GOTO LONGA
00F9 3400      0368  RETLW 0
                0369
00FA          0370  GRAPHICS
00FA 3040      0371  MOVWL 040H

```

MPASM 01.02.05 Intermediate LCORG1.ASM 11-28-2000 18:28:22 PAGE 8

LOC OBJECT CODE LINE SOURCE TEXT  
VALUE

```

00FB 208F      0372  CALL COMMAND
                0373
00FC 300E      0374  MOVWL 0EH
00FD 209D      0375  CALL WRITE
00FE 3010      0376  MOVWL 010H
00FF 209D      0377  CALL WRITE
0100 300E      0378  MOVWL 0EH
0101 2090      0379  CALL WRITE
0102 3000      0380  MOVWL 0
0103 2090      0381  CALL WRITE
0104 301C      0382  MOVWL 01CH
0105 2090      0383  CALL WRITE
0106 3012      0384  MOVWL 012H
0107 209D      0385  CALL WRITE
0108 301C      0386  MOVWL 01CH
0109 2090      0387  CALL WRITE
010A 3012      0388  MOVWL 012H
010B 2090      0389  CALL WRITE
010C 3400      0390  RETLW 0
                0391
                0392
                0393
                0394

```

MPASM 01.02.05 Intermediate LCORG1.ASM 11-28-2000 18:28:22 PAGE 9

SYMBOL TABLE

```

LABEL VALUE
AA 003C
ADDRESS 000E
ALLSTRING 0062
ASCII 0013
A N 000D
BAUD9600 0021
BB 0047
BINARY 0001
BUFFER 0010
CARIAGERETURN 002A
CHARS 001A
COMMAND 008F
CONVERT 00B9
COUNTER 000C
COUNTR 000F
CYCLEALL 0009
DATABYTE 0011
DEBOUNCE 00FF
DELAY 008E
DELAY_DATA 00CB
DLYCNT 0015
E 0001
ENTER 0006
FSR 0004
GETBYTE 0048
GETHIGH 0040
GETLOW 0035
GRAPHICS 00FA
HBYTE 0017
INC DATA 0019
INDF 0000
INIT_DISPLAY 0071
LBYTE 0016
LCDDATA 0014
LDELAY 00C4
LON 00F3
LONG 00E7
LONGA 00F2
LONGTIME 00E2
LONGX 00E6
LSN 0005
L_DELAY 00C7
MSN 0004
NEXT 00AC
NEXTDATA 000F
NUMBER 0019
PC 0002
PORT_A 0005
PORT_B 0006
PRINTBYTE 0022
REDB 00C0
REDX 00C0
MPASM 01.02.05 Intermediate LCORG1.ASM 11-28-2000 18:28:22 PAGE 10

```

SYMBOL TABLE - Continued

```

LABEL VALUE
RS 0002
RTCC 0001
SECOND 00EE
SELECTINIT 0080
SEND 005C
SENDZ 005E
SLOW 0018
SQUIRT 0007
STATUS 0003
STRING 0012
TXD 0003
TX0_DATA 00AB
VV 002D
WRITE 009D
WH 002F
_16C84 0001

```

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

```

0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX

```

```

0080 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
00C0 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX

```

```

0100 : XXXXXXXXXXXXXXXX--
0140 : -----

```

All other memory blocks unused.

Errors : 0  
Warnings : 0  
Messages : 0