

LAB. 07 (<http://diana.uceh.ualg.pt/Inst/lab07.pdf>)

Interface RS-232

Material

computador
compilador Borland C ou Microsoft QuickBasic
cabo série (RS-232)
Osciloscópio digital Tektronix TDS210

Objectivos

Pretende-se neste trabalho familiarizar o aluno com a interface RS-232 e a sua aplicação em aquisição de dados num ambiente laboratorial

Introdução

A interface RS-232 é um standard para ligação de terminais de dados a modems que define níveis lógicos e um conector para a transferência de dados. Na actualidade é utilizado indistintamente para ligar o mais variado tipo de aparelhos a PCs.

É uma interface série: os dados são enviados em série, bit a bit, a uma taxa de transmissão que pode ser fixada entre 110 e 19200 bits por segundo ("baud"). O standard especifica um conector de 25 pinos (DB-25), em que cada pino tem uma função bem definida. No entanto, na actualidade é bem mais vulgar os PCs utilizarem um conector de 9 pinos (DB-9).

Tambem a maior parte dos pinos utilizados na ligação a modems pode ser dispensados, sendo apenas fundamentais três pinos: (1) para transmitir (TxD), (2) para receber (RxD), e (3) um pino de *Ground*.

A interface RS-232 é normalmente utilizada para enviar dados de uma forma assíncrona. Isto quer dizer que a interface pode estar inerte ("idle") durante largos períodos e ficar activa subitamente (é o caso de alguém a bater texto num teclado). O sincronismo entre o receptor e o emissor dos dados é conseguida enviando sempre um start bit no início da transmissão e um stop bit no fim da transmissão.

Estas breves linhas servem apenas de introdução à interface RS-232, recomendando-se vivamente a leitura do Apêndice A, para a realização do trabalho.

Montagem Experimental

Neste trabalho pretende-se

- (a) familiarizar o aluno com o formato dos dados na transmissão assíncrona
- (b) enviar e receber mensagens para/do osciloscópio digital Tektronix TDS210

Transmissão assíncrona

Na primeira parte vamos utilizar um programa que coloca o PC como um simples terminal: o programa *HyperTerminal* (Encontra-se um atalho em Program > Accessories)

- Configura o programa para enviar dados pela porta COM2: com o seguinte formato:

9600 Baud, 7 bits de dados, 1 bit de paridade par, 1 stop bit, handshake por software

Ver
Figura 1.

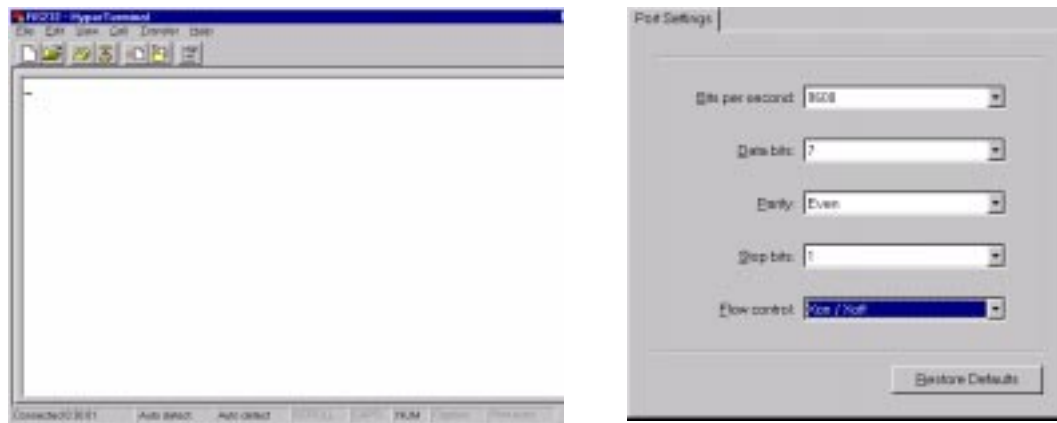


Figura 1: Configuração da porta COM2:

- Ligar o osciloscópio ao pino 2 da interface RS-232 e enviar um carácter à escolha ("A", "5", "J", etc.)

O osciloscópio pode ter dificuldades de sincronismo porque a taxa de repetição do teclado é demasiado baixa para se conseguir uma imagem parada no osciloscópio...

Para resolver este problema vamos fazer um pequeno programa em QuickBasic (c:\QB)

```
10 OPEN "com2:9600,p,7,1,CD0,CS0,DS0,OP0,RS" FOR OUTPUT
AS #1
20 PRINT #1, "J"
30 GOTO 20
```

ou em Borland C (c:\bc)

```
#include <stdio.h>
#include <windows.h>
main()
{
DCB far *dcb ;
int err, idComDev ;
FILE *fp ;
```

```
idComDev = OpenComm("COM2", 3, 3);  
err = BuildCommDCB("COM2:9600,p,7,1", dcb);  
err = SetCommState(dcb);  
while (1)  
{  
WriteComm(idComDev, "J", 1);  
// TransmitCommChar(idComDev, 0x4A);  
}  
}
```

que envia os caracteres desejados continuamente à máxima velocidade especificada (9600 bits/segundo)

Faz o teu programa em c:\users\999999 (onde 999999 é o teu número de aluno)

Depois de estares convencido que estás familiarizado com o formato dos dados, corre o programa `c:\users\jbastos\rs232.exe` e identifica o carácter que está a ser enviado por este programa:

- qual a taxa de transmissão
- qual o carácter
- qual é a paridade
- quantos stop bits

Comunicação com o osciloscópio digital Tektronix TDS210

Nesta segunda parte vais enviar mensagens e receber dados do osciloscópio digital Tektronix TDS210 através da porta série.

Nota que há alguma sobreposição com o lab08-Interface IEEE-488 (<http://diana.uceh.ualg.pt/Inst/lab08.pdf>)—o que aprenderes agora vai ser útil para o lab08 e vice-versa.

Pretende-se realizar um programa em Borland C ou em QuickBasic que através da interface RS-232

- configure remotamente o osciloscópio com
 - escala vertical no canal 1: 2V/divisão
 - escala temporal (horizontal): 250 μ s/divisão
- faça a aquisição de um sinal, por exemplo o próprio sinal de calibração que existe no osciloscópio (é uma onda quadrada)

O objectivo final é obter no computador sob a forma de tabela com duas colunas (tempo, sinal digitalizado) um *screen dump* do canal 1 do osciloscópio.

Alguns comandos que o osciloscópio aceita

Instrumentação - Lab. 07

curve[?]

exemplo: curve?

horizontal:scale [?, value]

exemplo: horizontal:scale 500e-6

ch<x>:scale [?, value]

exemplo: ch1:scale?

(quando o comando termina em "?" o osciloscópio responde...)

Para a realização deste trabalho é fundamental consultar os manuais de referência do osciloscópio TDS210!

NÃO LEVAR ESTES MANUAIS PARA FORA DO LABORATÓRIO DE INSTRUMENTAÇÃO

APÊNDICE A - Tutorial sobre a interface RS-232

Introdução

A interface RS-232 foi criada pela Electronics Industries Association (EIA) como um standard para ligação de terminais de dados a modems. É um standard de hardware que define níveis lógicos, e um conector para a transferência de dados. Mostra-se na Tabela 1 as principais características da interface.

| | |
|---|-----------------|
| <i>Data</i> | |
| Format | Serial |
| Timing | Asynchronous |
| Rate | 110–19,200 baud |
| <i>Connector</i> | |
| Type | DB–25 |
| Number Lines | 25 |
| Data Lines | 2 |
| <i>Signal Levels</i> | |
| Negative logic | Pins 2 & 3 |
| Positive logic | All other pins |
| High level | 5 to 25 V |
| Low level | –5 to –25V |
| <i>Line Naming Convention</i> | |
| Lines named with respect to DTE | |
| <i>Connector Gender</i> | |
| DTE | Male |
| DCE | Female |
| This convention is not adhered to in practice | |

Tabela 1: Características da interface RS232

Os dados são enviados em série, bit a bit, a uma taxa de transmissão entre 110 e 19200 bits por segundo ("baud"). É utilizado um conector de 25 contactos (DB-25), em que cada contacto tem uma função bem definida. Note-se que o standard não é seguido estritamente na prática. Por exemplo, a taxa de transmissão atinge 115200 baud e é utilizado um conector de 9 contactos (DB-9) nos PCs actuais.

Os níveis de tensão nas linhas de dados do standard RS-232 são os seguintes: um valor lógico 1, designado por "mark", é uma tensão entre -3V e -15V ; um valor lógico 0, designado por "space", é uma tensão em carga entre +3V e +15V (logo as linhas de dados funcionam em lógica negativa). Valores típicos usados são $\pm 10V$ e $\pm 12V$.

Para reduzir a interferência ("crosstalk") entre linhas adjacentes, os tempos de subida e descida dos sinais estão limitados a um máximo de $30V/\mu s$. O standard especifica ainda uma distância máxima de 15 metros para uma taxa de transmissão de 19200 bits por segundo. No entanto, é usual utilizarem-se interfaces RS-232 para distâncias até 1000 metros, enviando-se os dados a taxas de transmissão mais baixas.

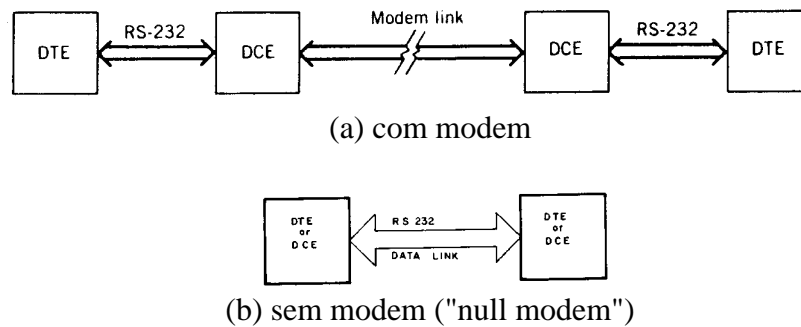


Figura 2: Comunicação série RS-232

Existe alguma confusão na aplicação do standard porque actualmente ele é utilizado para ligar uma grande variedade de aparelhos entre si, e não apenas a *modems* como foi inicialmente concebido. Mostra-se na Figura 2 uma comunicação standard entre dois aparelhos. Na Figura 2a os dois aparelhos estão ligados através de interfaces série RS-232 a modems, e estes estão ligados entre si através de uma linha telefónica. Os aparelhos (computadores, etc.) são designados por "data terminal equipment" (DTE). Os modems são designados por "data communications equipment" (DCE).

Os aparelhos DTE e DCE vêm uma ligação RS-232 de perspectiva diferente. Quando se interfaciam dois aparelhos entre si sem a presença de modems, configuração normalmente designada por "null modem" (ver Figura 2b), é importante saber se os aparelhos funcionam como DCE ou DTE.

As linhas de sinal e de controle

Mostra-se na Figura 3 um diagrama esquemático do conector standard da interface RS232 (DB-25). Apresenta-se também uma tabela com a função dos contactos do conector (as linhas são designadas relativamente a um aparelho DTE).

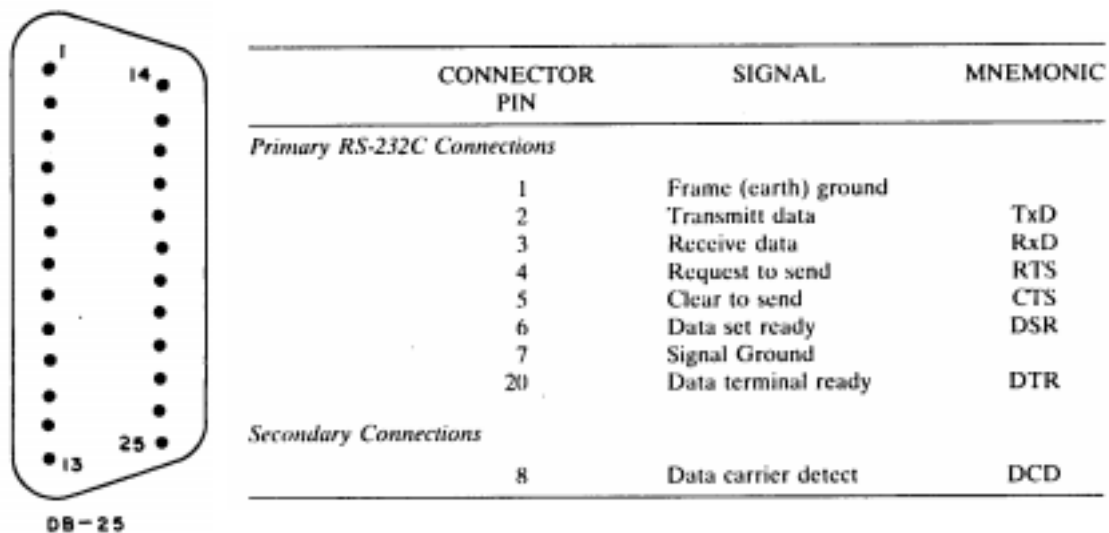


Figura 3: Conector standard RS-232

É importante dizer que nos PCs actuais é utilizado um conector de 9 pinos (DB-9) em vez do conector de 25 pinos (DB-25). Para facilitar a conversão mostra-se na Tabela 2 o significado dos pinos utilizados em ambos os conectores.

| PINOUT of the SERIAL PORT (--> direction is out of PC) | | | | | |
|---|--------|---------|---------------------|-----------|----------------------------|
| 9-pin | 25-pin | Acronym | Full-Name | Direction | What-it-May-Do/Mean |
| 3 | 2 | TxD | Transmit Data | --> | Transmits bytes out of PC |
| 2 | 3 | RxD | Receive Data | <-- | Receives bytes into PC |
| 7 | 4 | RTS | Request To Send | --> | RTS/CTS flow control |
| 8 | 5 | CTS | Clear To Send | <-- | RTS/CTS flow control |
| 6 | 6 | DSR | Data Set Ready | <-- | I'm ready to communicate |
| 4 | 20 | DTR | Data Terminal Ready | --> | I'm ready to communicate |
| 1 | 8 | DCD | Data Carrier Detect | <-- | Modem connected to another |
| 9 | 22 | RI | Ring Indicator | <-- | Telephone line ringing |
| 5 | 7 | | Signal Ground | | |

Tabela 2: Conversão entre os conectores DB-25 e DB-9

Apenas duas linhas da interface RS-232 (contactos 2 e 3 do conector) são utilizadas para transferência de dados. Estas são as únicas linhas que usam lógica negativa.

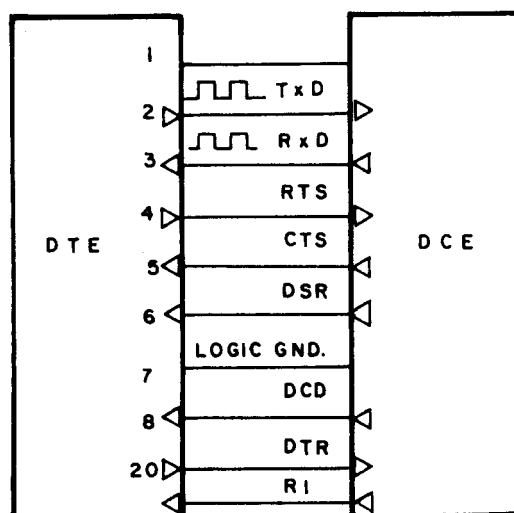


Figura 4: Ligação entre um aparelho DTE e um DCE

Mostra-se na Figura 4 um diagrama esquemático de uma ligação entre um equipamento DTE e outro do tipo DCE. Neste caso, existe uma correspondência contacto a contacto entre o DTE e o DCE. Na figura incluiu-se ainda setas, que indicam qual o aparelho que define o estado de cada linha.

Quando o DTE pretende enviar dados:

- o DTE activa as linhas DTR e RTS para indicar que se encontra pronto a enviar.
- o DCE responde activando as linhas de DSR e CTS indicando que se encontra pronto a receber.
- o DTE envia dados pela linha TxD
- no fim da transmissão, o DTE desactiva as linhas de RTS e DTR.

No caso de ser o DCE a enviar dados a sequência de sincronização é a seguinte:

- o DTE activa a linha de DTR para indicar que se encontra pronto a receber, mas não activa a linha de RTS
- as linhas de DSR e DCD são activadas pelo DCE
- o DTE recebe dados

- no fim da transmissão a linha de DTR é desactivada

Descreve-se, de seguida, a utilização da interface RS-232 em ligações directas entre computadores e terminais ("null modem"). Esta configuração é frequentemente utilizada em aquisição de dados, em que o computador comunica com um ou vários instrumentos através de interfaces RS-232.

Uma ligação com um cabo standard entre dois aparelhos DTE não resultaria (ambos tentariam enviar dados pela linha 2). A solução consiste em cruzar linhas, de forma a iludir ambos os aparelhos DTE de que estão a comunicar com aparelhos DCE.

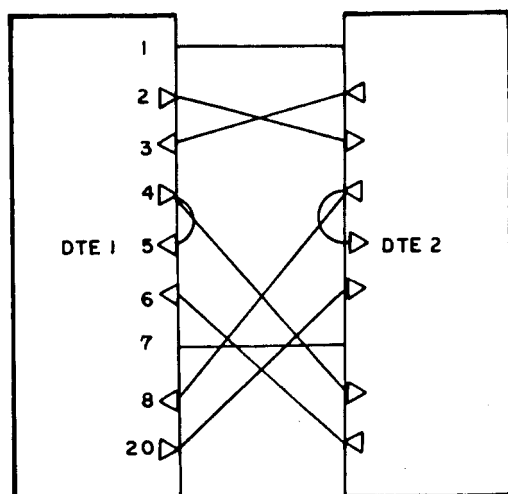
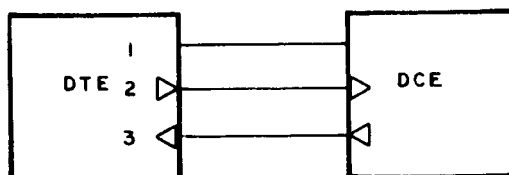


Figura 5: Configuração de uma ligação entre dois aparelhos DTE

Mostra-se na Figura 5 a configuração mais usual.

As linhas de controle são ligadas da seguinte forma: a linha de RTS é ligada à linha de CTS (desactivando o "handshake") e à linha de DCD do outro aparelho; a linha de DTR de um aparelho é ligada à linha de DSR do outro. O efeito do conjunto é que, quando cada aparelho activa as suas linhas de controle, a transmissão pode começar.



B.

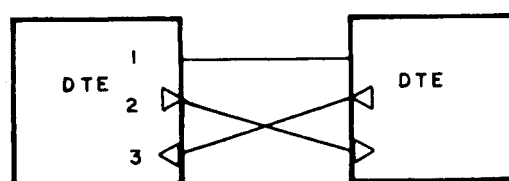


Figura 6: Ligações simples RS-232 (sem linhas de controle)

Mas na maior parte das situações apenas 3 linhas são necessárias: as duas linhas de sinal e a linha de terra de sinal. Mostra-se, na Figura 6 um esquema com estas ligações, em que se supõe que é ignorado pelos aparelhos o estado das linhas de controle.

Nesta situação o controlo do fluxo de informação é feito por *software*: é o modo Xon-Xoff de que falamos na secção seguinte.

Formato dos dados

Existe um considerável grau de flexibilidade no formato dos dados transmitidos através de uma ligação RS-232.

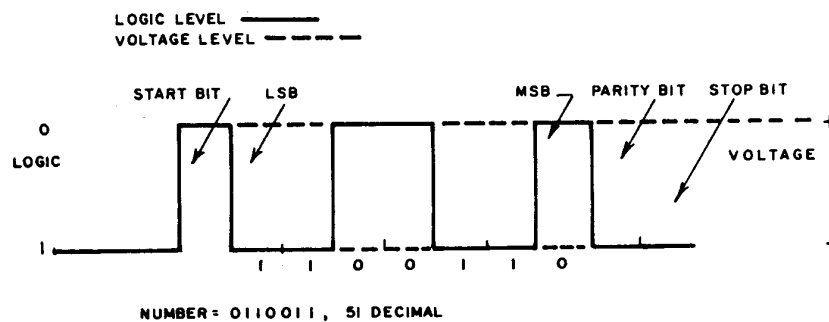


Figura 7: Sequência de bits que corresponde à transmissão de um byte

Mostra-se na Figura 7 a sequência de bits que corresponde ao envio de um byte (8 bits) de dados. Esta sequência é a transmissão de um "caracter" em código ASCII, em conjunto com um bit de "start", um número pré-definido de bits de "stop", e um bit de paridade.

O bit de "start" é um "0" lógico, e os bits de "stop" são "1"s lógicos. Este facto obriga a que haja sempre uma transição entre o fim de um caracter e o começo do próximo, que é utilizada pelo aparelho que recebe para se sincronizar com o aparelho que envia. Note-se ainda que os níveis de sinal nas linhas de dados numa interface RS232 são em lógica negativa, e a figura reflecte este facto.

Neste exemplo, envia-se um bit de "start", a representação em ASCII do caracter "3", um bit de paridade, e um "stop" bit. Notar que no diagrama temporal da figura o bit menos significativo (LSB) está à esquerda porque é enviado em primeiro lugar! Notar também que o bit de paridade é 1, a que corresponde paridade impar—o número de bits a 1 do caracter mais o bit de paridade é um numero impar. (Paridade par—o bit de paridade é tal que o número total de bits incluindo o bit de paridade é um número par.)

| Bits Per Character | Parity Bit | Stop Bits |
|--------------------|------------|-----------|
| 7 | +EVEN | +1 |
| 7 | +ODD | +1 |
| 8 | NONE | +2 |
| 8 | NONE | +1 |
| 8 | +EVEN | +1 |
| 8 | +ODD | +1 |
| 7 | +EVEN | +2 |
| 7 | +ODD | +2 |

Tabela 3: alguns formatos possíveis para a transmissão RS232

Mas há outros formatos vulgarmente utilizados, como se mostra Tabela 3. (Nesta tabela o bit de "start" não aparece explicitamente na tabela por ser comum a todos os formatos.)

O byte (8 bits) de dados é constituído por

- (a) 7 bits que representam um carácter ASCII e 1 bit de paridade (que pode ser ignorado)
- (b) 8 bits de dados que representam um número binário entre 0 e 255

Os stop bits são pelo menos 1, mas há casos (em interfaces mais antigas e mais lentas) que são necessários 2 stop bits. Note-se que o estado de repouso da linha de transmissão é o estado lógico "1" a que corresponde estar a enviar continuamente stop bits!

Um dos parâmetros importantes que deve ser definido numa interface RS-232 é a taxa de transmissão, i.e., o número de bits enviado por segundo ("baud rate"). Sendo, em regra, utilizados 10 bits para o envio de um carácter, o número de caracteres enviados por segundo é aproximadamente a taxa de transmissão dividida por 10.

O código ASCII é o código de dados mais utilizado em comunicações entre computadores. É um código de 7 bits e (por isso) pode representar 128 caracteres. Destes, os primeiros 32 são caracteres de controle (não imprimíveis) e os restantes 96 são imprimíveis. Mostra-se na Tabela 4 o código ASCII em representação hexadecimal.

| HEX-ASCII TABLE | | | | | | | | | |
|-----------------|-----------------------|----|----|----|---|----|------------|--|--|
| 00 | NUL | 21 | ! | 42 | B | 63 | c | | |
| 01 | SOH | 22 | " | 43 | C | 64 | d | | |
| 02 | STX | 23 | # | 44 | D | 65 | e | | |
| 03 | ETX | 24 | \$ | 45 | E | 66 | f | | |
| 04 | EOT | 25 | % | 46 | F | 67 | g | | |
| 05 | ENO | 26 | & | 47 | G | 68 | h | | |
| 06 | ACK | 27 | ' | 48 | H | 69 | i | | |
| 07 | BEL | 28 | (| 49 | I | 6A | j | | |
| 08 | BS | 29 |) | 4A | J | 6B | k | | |
| 09 | HT | 2A | * | 4B | K | 6C | l | | |
| 0A | LF | 2B | + | 4C | L | 6D | m | | |
| 0B | VT | 2C | , | 4D | M | 6E | n | | |
| 0C | FF | 2D | - | 4E | N | 6F | o | | |
| 0D | CR | 2E | . | 4F | O | 70 | p | | |
| 0E | SO | 2F | / | 50 | P | 71 | q | | |
| 0F | SI | 30 | 0 | 51 | Q | 72 | r | | |
| 10 | DLE | 31 | 1 | 52 | R | 73 | s | | |
| 11 | DC1 (X-ON) | 32 | 2 | 53 | S | 74 | t | | |
| 12 | DC2 (TAPE) | 33 | 3 | 54 | T | 75 | u | | |
| 13 | DC3 (X-OFF) | 34 | 4 | 55 | U | 76 | v | | |
| 14 | DC4 (TAPE) | 35 | 5 | 56 | V | 77 | w | | |
| 15 | NAK | 36 | 6 | 57 | W | 78 | x | | |
| 16 | SYN | 37 | 7 | 58 | X | 79 | y | | |
| 17 | ETB | 38 | 8 | 59 | Y | 7A | z | | |
| 18 | CAN | 39 | 9 | 5A | Z | 7B | { | | |
| 19 | EM | 3A | : | 5B | [| 7C |] | | |
| 1A | SUB | 3B | ; | 5C | \ | 7D | ^ | | |
| 1B | ESC | 3C | < | 5D |] | | (ALT MODE) | | |
| 1C | FS | 3D | = | 5E | ^ | 7E | > | | |
| 1D | GS | 3E | > | 5F | — | 7F | DEL | | |
| 1E | RS | 3F | ? | 60 | \ | | (RUB OUT) | | |
| 1F | US | 40 | " | 61 | a | | | | |
| 20 | SP | 41 | A | 62 | b | | | | |

Tabela 4: O código ASCII em representação hexadecimal

Dos caracteres não-imprimíveis é importante mencionar alguns já nossos conhecidos:

- BS (0x08) - Back Space
- HT (0x09) - Horizontal Tab
- LF (0x0A) - Line Feed
- VT (0x0B) - Vertical Tab
- FF (0x0C) - Form Feed
- CR (0x0D) - Carriage Return

Finalmente os caracteres XON (0x11) e XOFF (0x13) são muito importantes para sincronizar o receptor e o transmissor quando o handshake não é feito por hardware (com as linhas de controlo) mas sim por software. Quando o receptor não pode receber mais dados (porque tem o buffer de entrada quase cheio) envia o caracter XOFF informando o transmissor para parar de transmitir. Quando o receptor pode mais uma vez receber dados envia o caracter XON. Note-se que a interface é *full-duplex* (os dados circulam nos dois sentidos simultaneamente) e por isso o receptor pode a qualquer momento interromper o emissor.