

Informática II

Puerto serie en la PC

Gonzalo F. Perez Paina



Universidad Tecnológica Nacional
Facultad Regional Córdoba
UTN-FRC

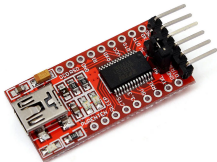
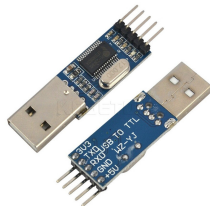
– 2024 –

Introducción

- ▶ El USB (Universal Serial Bus) es el puerto de comunicaciones más utilizado para la conexión de periféricos en la actualidad.
- ▶ Existen diferentes modelos de adaptadores USB-serie.
- ▶ Cuando se habla de **puerto serie** se hace referencia la **comunicación serial asíncrona** tipo RS-232 o UART-TTL.

Introducción

- ▶ El USB (Universal Serial Bus) es el puerto de comunicaciones más utilizado para la conexión de periféricos en la actualidad.
- ▶ Existen diferentes modelos de adaptadores USB-serie.
- ▶ Cuando se habla de **puerto serie** se hace referencia la **comunicación serial asíncrona** tipo RS-232 o UART-TTL.



- ▶ Adaptador USB-RS232.
- ▶ Adaptador USB-UART con/sin cable y con/sin bits de control (handshake).

Puerto serie en la PC – Acceso a dispositivo

En GNU/Linux se tiene acceso a los dispositivos (puertos series) mediante archivos de dispositivos, los cuales se encuentran en el directorio `/dev`.

Puerto serie en la PC – Acceso a dispositivo

En GNU/Linux se tiene acceso a los dispositivos (puertos series) mediante archivos de dispositivos, los cuales se encuentran en el directorio `/dev`.

Ejemplo de archivo de dispositivos

```
crw-rw---- 1 root dialout 188, 0 jul 20 14:02 /dev/ttyUSB0
crw-rw-rw- 1 root dialout 166, 0 jul 20 15:57 /dev/ttyACM0
brw-rw---- 1 root disk      8, 0 jul 20 14:02 /dev/sda
```

('c': dispositivos de caracteres – 'b' dispositivos de bloques)

Puerto serie en la PC – Acceso a dispositivo

En GNU/Linux se tiene acceso a los dispositivos (puertos serie) mediante archivos de dispositivos, los cuales se encuentran en el directorio `/dev`.

Ejemplo de archivo de dispositivos

```
crw-rw---- 1 root dialout 188, 0 jul 20 14:02 /dev/ttyUSB0
crw-rw-rw- 1 root dialout 166, 0 jul 20 15:57 /dev/ttyACM0
brw-rw---- 1 root disk      8, 0 jul 20 14:02 /dev/sda
```

('c': dispositivos de caracteres – 'b' dispositivos de bloques)

- Los archivos de dispositivos para los puertos serie suelen ser: `/dev/ttyS0`, `/dev/ttyS1`, `/dev/ttyUSB0`, `/dev/ttyACM0`, etc.

Puerto serie en la PC – Acceso a dispositivo

En GNU/Linux se tiene acceso a los dispositivos (puertos series) mediante archivos de dispositivos, los cuales se encuentran en el directorio `/dev`.

Ejemplo de archivo de dispositivos

```
crw-rw---- 1 root dialout 188, 0 jul 20 14:02 /dev/ttyUSB0
crw-rw-rw- 1 root dialout 166, 0 jul 20 15:57 /dev/ttyACM0
brw-rw---- 1 root disk      8, 0 jul 20 14:02 /dev/sda
```

('c': dispositivos de caracteres – 'b' dispositivos de bloques)

- ▶ Los archivos de dispositivos para los puertos serie suelen ser: `/dev/ttyS0`, `/dev/ttyS1`, `/dev/ttyUSB0`, `/dev/ttyACM0`, etc.
- ▶ Para ver el archivo de dispositivo creado al conectar el hardware (Arduino o adaptador USB-serie) utilizando el comando `dmesg`.

Puerto serie en la PC – Acceso a dispositivo

Ejemplo de salida de `dmesg` con placa Arduino UNO

```
usb 3-1: new full-speed USB device number 15 using xhci_hcd
usb 3-1: New USB device found, idVendor=1a86, idProduct=7523
usb 3-1: New USB device strings: Mfr=0, Product=2, SerialNumber=0
usb 3-1: Product: USB2.0-Serial
ch341 3-1:1.0: ch341-uart converter detected
usb 3-1: ch341-uart converter now attached to ttyUSB0
```


Puerto serie en la PC – Acceso a dispositivo

Ejemplo de salida de `dmesg` con placa Arduino UNO

```
usb 3-1: new full-speed USB device number 15 using xhci_hcd
usb 3-1: New USB device found, idVendor=1a86, idProduct=7523
usb 3-1: New USB device strings: Mfr=0, Product=2, SerialNumber=0
usb 3-1: Product: USB2.0-Serial
ch341 3-1:1.0: ch341-uart converter detected
usb 3-1: ch341-uart converter now attached to ttyUSB0
```

Ejemplo de salida de `dmesg` con adaptador USB-serie

```
usb 3-2: new full-speed USB device number 17 using xhci_hcd
usb 3-2: New USB device found, idVendor=067b, idProduct=2303
usb 3-2: New USB device strings: Mfr=1, Product=2, SerialNumber=0
usb 3-2: Product: USB-Serial Controller
usb 3-2: Manufacturer: Prolific Technology Inc.
pl2303 3-2:1.0: pl2303 converter detected
usb 3-2: pl2303 converter now attached to ttyUSB1
```

Puerto serie en la PC – Acceso a dispositivo

- ▶ Para modificar y ver la configuración actual de un puerto serie (archivo de dispositivo en `/dev`) se puede utilizar el comando `stty`.
 - ▶ Para ver la velocidad de comunicación actual

```
$> stty -F /dev/ttyUSB0  
speed 9600 baud; line = 0;
```

- ▶ Para modificar la velocidad de comunicación

```
$> stty -F /dev/ttyUSB1 115200  
$> stty -F /dev/ttyUSB1  
speed 115200 baud; line = 0;
```

Puerto serie en la PC – Acceso a dispositivo

- ▶ Para modificar y ver la configuración actual de un puerto serie (archivo de dispositivo en `/dev`) se puede utilizar el comando `stty`.

- ▶ Para ver la velocidad de comunicación actual

```
$> stty -F /dev/ttyUSB0  
speed 9600 baud; line = 0;
```

- ▶ Para modificar la velocidad de comunicación

```
$> stty -F /dev/ttyUSB1 115200  
$> stty -F /dev/ttyUSB1  
speed 115200 baud; line = 0;
```

- ▶ Ejemplo con Arduino:

- ▶ Abrir IDE Arduino y grabar el ejemplo `AnalogReadSerial`
 - ▶ Ver valores enviados desde el Monitor Serial (IDE)
 - ▶ Ver valores enviados desde `cutecom`

Puerto serie en la PC – Acceso a dispositivo

- ▶ Para modificar y ver la configuración actual de un puerto serie (archivo de dispositivo en `/dev`) se puede utilizar el comando `stty`.

- ▶ Para ver la velocidad de comunicación actual

```
$> stty -F /dev/ttyUSB0  
speed 9600 baud; line = 0;
```

- ▶ Para modificar la velocidad de comunicación

```
$> stty -F /dev/ttyUSB1 115200  
$> stty -F /dev/ttyUSB1  
speed 115200 baud; line = 0;
```

- ▶ Ejemplo con Arduino:

- ▶ Abrir IDE Arduino y grabar el ejemplo `AnalogReadSerial`
 - ▶ Ver valores enviados desde el Monitor Serial (IDE)
 - ▶ Ver valores enviados desde `cutecom`

- ▶ Algunas terminales: `screen`, `cutecom`, `minicom`, `PuTTY`, etc.

Puerto serie en la PC – Acceso a dispositivo

Puerto serie en la PC – Acceso a dispositivo

- ▶ Uso de `socat` para generación de puertos “virtuales”
`$> socat PTY,link=/tmp/ttyS0 PTY,link=/tmp/ttyS1`
- ▶ Mostrar en terminal lo recibido por el puerto serie
`$> cat /tmp/ttyS1`
- ▶ Enviar una cadeba a un puerto serie
`$> echo "Hola" > /tmp/ttyS0`
- ▶ Enviar una cadena desde `cutecom`

Programación – Abrir y cerrar el puerto serie

```
1 #include <stdio.h> /* Standard input/output definitions */
2 #include <fcntl.h> /* File control definitions */
3 #include <unistd.h> /* UNIX standard function definitions */
4
5 int main(void) {
6     int fd; /* File descriptor for the port */
7
8     fd = open("/dev/ttyUSB0", O_RDWR | O_NOCTTY | O_NDELAY);
9
10    if(fd == -1)
11        /* ERROR */
12
13    close(fd);
14    return 0;
15 }
```

Flags:

- ▶ O_RDWR: Lectura/Escritura
- ▶ O_NOCTTY: Para que no sea una terminal de control (ps -a)
- ▶ O_NDELAY: Ignora la señal DCD (o el proceso duerme hasta activarse DCD)

Programación – Escribir y leer datos

```
1 #include <stdio.h> /* Standard input/output definitions */
2 #include <fcntl.h> /* File control definitions */
3 #include <unistd.h> /* UNIX standard function definitions */
4
5 int main(void) {
6     int n, fd;
7     char* data;
8
9     fd = open("/dev/ttyUSB0", O_RDWR | O_NOCTTY | O_NDELAY);
10    if(fd == -1) {} /* ERROR */
11
12    /* Writing data to the port */
13    n = write(fd, "Hello\n", 6);
14    if(n < 0)
15        /* ERROR */
16
17    /* Reading data from the port */
18    /* n = read(fid, data, 4); */
19    /* if(n < 4) */
20
21    close(fd);
22    return 0;
23 }
```

Configuración del puerto serie (termios)

Terminales en sistemas POSIX (Portable Operating System Interface (UNIX))

- ▶ `termios.h`: estructura de control de terminal y funciones de control.
- ▶ Cambiar parámetros tales como velocidad de comunicación, tamaño trama, etc.

Las dos funciones más importantes son: `tcgetattr()`, `tcsetattr()`

Miembros de la [estructura termios](#):

- ▶ `c_cflags`: Opciones de control
- ▶ `c_lflags`: Opciones de línea
- ▶ `c_iflags`: Opciones de entrada
- ▶ `c_oflags`: Opciones de salida
- ▶ `c_cc`: Caracteres de control
- ▶ `c_ispeed`: Baudrate de entrada (interfaz nueva)
- ▶ `c_ospeed`: Baudrate de salida (interfaz nueva)

Configuración del puerto serie (termios)

```
struct termios options;
/* Get the current options for the port */
tcgetattr(fd, &options);

/* Set the baud rates to 19200 */
cfsetispeed(&options, B19200);
cfsetospeed(&options, B19200);

/* Enable the receiver and set local mode */
options.c_cflag |= (CLOCAL | CREAD);

/* No parity 8N1 */
options.c_cflag &= ~PARENB; /* No parity */
options.c_cflag &= ~CSTOPB; /* 1 stop bit */
options.c_cflag &= ~CSIZE; /* Mask the character size bits */
options.c_cflag |= CS8;

/* Set the new options for the port */
tcsetattr(fd, TCSANOW, &options);
```
