NAME

termiox – extended general terminal interface

DESCRIPTION

The extended general terminal interface supplements the **termio**(7I) general terminal interface by adding support for asynchronous hardware flow control, isochronous flow control and clock modes, and local implementations of additional asynchronous features. Some systems may not support all of these capabilities because of either hardware or software limitations. Other systems may not permit certain functions to be disabled. In these cases the appropriate bits will be ignored. See **<sys/termiox.h>** for your system to find out which capabilities are supported.

Hardware Flow Control Modes

Hardware flow control supplements the **termio**(7I) **IXON**, **IXOFF**, and **IXANY** character flow control. Character flow control occurs when one device controls the data transfer of another device by the insertion of control characters in the data stream between devices. Hardware flow control occurs when one device controls the data transfer of another device using electrical control signals on wires (circuits) of the asynchronous interface. Isochronous hardware flow control occurs when one device controls the data transfer of another device by asserting or removing the transmit clock signals of that device. Character flow control and hardware flow control may be simultaneously set.

In asynchronous, full duplex applications, the use of the Electronic Industries Association's EIA-232-D Request To Send (RTS) and Clear To Send (CTS) circuits is the preferred method of hardware flow control. An interface to other hardware flow control methods is included to provide a standard interface to these existing methods.

The EIA-232-D standard specified only uni-directional hardware flow control - the Data Circuit-terminating Equipment or Data Communications Equipment (DCE) indicates to the Data Terminal Equipment (DTE) to stop transmitting data. The **termiox** interface allows both uni-directional and bi-directional hardware flow control; when bi-directional flow control is enabled, either the DCE or DTE can indicate to each other to stop transmitting data across the interface. Note: It is assumed that the asynchronous port is configured as a DTE. If the connected device is also a DTE and not a DCE, then DTE to DTE (for example, terminal or printer connected to computer) hardware flow control is possible by using a null modem to interconnect the appropriate data and control circuits.

Clock Modes

Isochronous communication is a variation of asynchronous communication whereby two communicating devices may provide transmit and/or receive clock to each other. Incoming clock signals can be taken from the baud rate generator on the local isochronous port controller, from CCITT V.24 circuit 114, Transmitter Signal Element Timing - DCE source (EIA-232-D pin 15), or from CCITT V.24 circuit 115, Receiver Signal Element Timing - DCE source (EIA-232-D pin 17). Outgoing clock signals can be sent on CCITT V.24 circuit 113, Transmitter Signal Element Timing - DTE source (EIA-232-D pin 24), on CCITT V.24 circuit 128, Receiver Signal Element Timing - DTE source (no EIA-232-D pin), or not sent at all.

In terms of clock modes, traditional asynchronous communication is implemented simply by using the local baud rate generator as the incoming transmit and receive clock source and not outputting any clock signals.

Terminal Parameters

The parameters that control the behavior of devices providing the **termiox** interface are specified by the **termiox** structure, defined in the **<sys/termiox.h>** header. Several **ioctl**(2) system calls that fetch or change these parameters use this structure:

```
#define         NFF         5
struct termiox      {
        unsigned short   x_hflag;        /* hardware flow control modes */
        unsigned short   x_cflag;        /* clock modes */
        unsigned short   x_rflag[NFF];   /* reserved modes */
        unsigned short   x_sflag;        /* spare local modes */
};
```

The **x_hflag** field describes hardware flow control modes:

RTSXOFF	**0000001**	Enable RTS hardware flow control on input.
CTSXON	**0000002**	Enable CTS hardware flow control on output.
DTRXOFF	**0000004**	Enable DTR hardware flow control on input.
CDXON	**0000010**	Enable CD hardware flow control on output.
ISXOFF	**0000020**	Enable isochronous hardware flow control on input.

The EIA-232-D DTR and CD circuits are used to establish a connection between two systems. The RTS circuit is also used to establish a connection with a modem. Thus, both DTR and RTS are activated when an asynchronous port is opened. If DTR is used for hardware flow control, then RTS must be used for connectivity. If CD is used for hardware flow control, then CTS must be used for connectivity. Thus, RTS and DTR (or CTS and CD) cannot both be used for hardware flow control at the same time. Other mutual exclusions may apply, such as the simultaneous setting of the **termio**(7I) **HUPCL** and the **termiox** DTRXOFF bits, which use the DTE ready line for different functions.

Variations of different hardware flow control methods may be selected by setting the the appropriate bits. For example, bi-directional RTS/CTS flow control is selected by setting both the **RTSXOFF** and **CTSXON** bits and bi-directional DTR/CTS flow control is selected by setting both the **DTRXOFF** and **CTSXON**. Modem control or uni-directional CTS hardware flow control is selected by setting only the **CTSXON** bit.

As previously mentioned, it is assumed that the local asynchronous port (for example, computer) is configured as a DTE. If the connected device (for example, printer) is also a DTE, it is assumed that the device is connected to the computer's asynchronous port using a null modem that swaps control circuits (typically RTS and CTS). The connected DTE drives RTS and the null modem swaps RTS and CTS so that the remote RTS is received as CTS by the local DTE. In the case that **CTSXON** is set for hardware flow control, printer's lowering of its RTS would cause CTS seen by the computer to be lowered. Output to the printer is suspended until the printer's raising of its RTS, which would cause CTS seen by the computer to be raised.

If **RTSXOFF** is set, the Request To Send (RTS) circuit (line) will be raised, and if the asynchronous port needs to have its input stopped, it will lower the Request To Send (RTS) line. If the RTS line is lowered, it is assumed that the connected device will stop its output until RTS is raised.

If **CTSXON** is set, output will occur only if the Clear To Send (CTS) circuit (line) is raised by the connected device. If the CTS line is lowered by the connected device, output is suspended until CTS is raised.

If **DTRXOFF** is set, the DTE Ready (DTR) circuit (line) will be raised, and if the asynchronous port needs to have its input stopped, it will lower the DTE Ready (DTR) line. If the DTR line is lowered, it is assumed that the connected device will stop its output until DTR is raised.

If **CDXON** is set, output will occur only if the Received Line Signal Detector (CD) circuit (line) is raised by the connected device. If the CD line is lowered by the connected device, output is suspended until CD is raised.

If **ISXOFF** is set, and if the isochronous port needs to have its input stopped, it will stop the outgoing clock signal. It is assumed that the connected device is using this clock signal to create its output. Transit and receive clock sources are programmed using the **x_cflag** fields. If the port is not programmed for external clock generation, **ISXOFF** is ignored. Output isochronous flow control is supported by appropriate clock source programming using the **x_cflag** field and enabled at the remote connected device.

The **x_cflag** field specifies the system treatment of clock modes.

XMTCLK	0000007	Transmit clock source:
XCIBRG	0000000	Get transmit clock from internal baud rate generator.
XCTSET	0000001	Get transmit clock from transmitter signal element timing (DCE source) lead, CCITT V.24 circuit 114, EIA-232-D pin 15.
XCRSET	0000002	Get transmit clock from receiver signal element timing (DCE source) lead, CCITT V.24 circuit 115, EIA-232-D pin 17.
RCVCLK	0000070	Receive clock source:
RCIBRG	0000000	Get receive clock from internal baud rate generator.
RCTSET	0000010	Get receive clock from transmitter signal element timing (DCE source) lead, CCITT V.24 circuit 114, EIA-232-D pin 15.
RCRSET	0000020	Get receive clock from receiver signal element timing (DCE source) lead, CCITT V.24 circuit 115, EIA-232-D pin 17.
TSETCLK	0000700	Transmitter signal element timing (DTE source) lead, CCITT V.24 circuit 113, EIA-232-D pin 24, clock source:
TSETCOFF	0000000	TSET clock not provided.
TSETCRBRG	0000100	Output receive baud rate generator on circuit 113.

TSETCTBRG	0000200	Output transmit baud rate generator on circuit 113.
TSETCTSET	0000300	Output transmitter signal element timing (DCE source) on circuit 113.
TSETCRSET	0000400	Output receiver signal element timing (DCE source) on circuit 113.
RSETCLK	0007000	Receiver signal element timing (DTE source) lead, CCITT V.24 circuit 128, no EIA-232-D pin, clock source:
RSETCOFF	0000000	RSET clock not provided.
RSETCRBRG	0001000	Output receive baud rate generator on circuit 128.
RSETCTBRG	0002000	Output transmit baud rate generator on circuit 128.
RSETCTSET	0003000	Output transmitter signal element timing (DCE source) on circuit 128.
RSETCRSET	0004000	Output receiver signal element timing (DCE) on circuit 128.

If the **XMTCLK** field has a value of **XCIBRG** the transmit clock is taken from the hardware internal baud rate generator, as in normal asynchronous transmission. If **XMTCLK** = **XCTSET** the transmit clock is taken from the Transmitter Signal Element Timing (DCE source) circuit. If **XMTCLK** = **XCRSET** the transmit clock is taken from the Receiver Signal Element Timing (DCE source) circuit.

If the **RCVCLK** field has a value of **RCIBRG** the receive clock is taken from the hardware Internal Baud Rate Generator, as in normal asynchronous transmission. If **RCVCLK** = **RCTSET** the receive clock is taken from the Transmitter Signal Element Timing (DCE source) circuit. If **RCVCLK** = **RCRSET** the receive clock is taken from the Receiver Signal Element Timing (DCE source) circuit.

If the **TSETCLK** field has a value of **TSETCOFF** the Transmitter Signal Element Timing (DTE source) circuit is not driven. If **TSETCLK** = **TSETCRBRG** the Transmitter Signal Element Timing (DTE source) circuit is driven by the Receive Baud Rate Generator. If **TSETCLK** = **TSETCTBRG** the Transmitter Signal Element Timing (DTE source) circuit is driven by the Transmit Baud Rate Generator. If **TSETCLK** = **TSETCTSET** the Transmitter Signal Element Timing (DTE source) circuit is driven by the Transmitter Signal Element Timing (DCE source). If **TSETCLK** = **TSETCRBRG** the Transmitter Signal Element Timing (DTE source) circuit is driven by the Receiver Signal Element Timing (DCE source).

If the **RSETCLK** field has a value of **RSETCOFF** the Receiver Signal Element Timing (DTE source) circuit is not driven. If **RSETCLK** = **RSETCRBRG** the Receiver Signal Element Timing (DTE source) circuit is driven by the Receive Baud Rate Generator. If **RSETCLK** = **RSETCTBRG** the Receiver Signal Element Timing (DTE source) circuit is driven by the Transmit Baud Rate Generator. If **RSETCLK** = **RSETCTSET** the Receiver Signal Element Timing (DTE source) circuit is driven by the Transmitter Signal Element Timing (DCE source). If **RSETCLK** = **RSETCRBRG** the Receiver Signal Element Timing (DTE source) circuit is driven by the Receiver Signal Element Timing (DCE source).

The **x_rflag** is reserved for future interface definitions and should not be used by any implementations. The **x_sflag** may be used by local implementations wishing to customize their terminal interface using the **termiox** ioctl system calls.

IOCTLS

The **ioctl**(2) system calls have the form:

ioctl (*fildes, command, arg*)
struct termiox **arg*;

The commands using this form are:

TCGETX
The argument is a pointer to a **termiox** structure. The current terminal parameters are fetched and stored into that structure.

TCSETX
The argument is a pointer to a **termiox** structure. The current terminal parameters are set from the values stored in that structure. The change is immediate.

TCSETXW
The argument is a pointer to a **termiox** structure. The current terminal parameters are set from the values stored in that structure. The change occurs after all characters queued for output have been transmitted. This form should be used when changing parameters that will affect output.

TCSETXF
The argument is a pointer to a **termiox** structure. The current terminal parameters are set from the values stored in that structure. The change occurs after all characters queued for output have been transmitted; all characters queued for input are discarded and then the change occurs.

FILES

/dev/*

SEE ALSO

stty(1), **ioctl**(2), **termio**(7I)