Driver - Description

devu-ftdix32-speed

FT232 – FT2232 specific usb-uart serial communications manager for QNX Neutrino
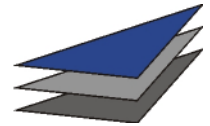
👉 You must be root to start this driver.

Syntax:

devu-ftdix32-speed [[*options*]

      [*u deviceNr*][,*hostNr*][.*tierPort*][.*interfaceNr*]]]... &

Runs on:

x86 with an FTDI FT232BM of FTDI FT2232BM -compatible UART and a working USB host controller (io-usb –dehci –dohci).

Specification:

Supported USB-Device (basic uart functionality):

    VID: 0403 PID: 6001 - FT8U2XX Device
    VID: 0403 PID: 6010 - FT2232

Serial Parameters

- Baudrates: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600 [Baud]
- Databits: 7, 8
- Stopbits: 1, 2
- Parity: none, odd, even
- Hardware flow control: none, Hardware (RTS/CTS)
- Software flow control: none, xon/xoff
- **Default values**: 115200, 8, 1, none with hardware flow control disabled

Options:

The options are position-dependent and affect the subsequent ports.

-V *vendor ID*

        use carefully!!!

        Define additional customer specific USB-Vendor ID.

-P *product ID*

        use carefully!!!

        Define additional customer specific USB-Product ID.

-b *number*

        The initial baud rate (default 115200).

-C *size*

        The size of the canonical buffer in bytes (default 256).

-E

        Start in raw mode (the default). Software flow control is disabled by default.

-e

        Start in edited mode (default raw). Software flow control is enabled by default.

-F

        Disable hardware flow control (hardware flow control is disabled by default). Hardware flow control is not supported in edited mode.

-f

        Enable hardware flow control. Hardware flow control is not supported in edited mode.

-I *number*

        The size of the interrupt input buffer in bytes (default 2048).

-O *number*

        The size of the interrupt output buffer in bytes (default 2048).

-S|s

        Disable / enable software flow control. The default depends on the mode: in raw mode (**-E**, the default), it's disabled; in edited mode (**-e**), it's enabled.

        The order in which you specify the **-E** or **-e**, and **-S** or **-s** options matters:

Options	Mode	Software flow control
-e	Edited	Enabled
-S -e	Edited	Enabled
-e -S	Edited	Disabled
-E	Raw	Disabled
-s -E	Raw	Disabled
-E -s	Raw	Enabled

-u *deviceNr,hostNr.tierPort.interfaceNr*

        deviceNr can be used to specified the device index. The device name has to been unique and it must be handled by the user.

hostNr specified the usb host channel on which the device will be connected
tierPort describes the physically usb topology which will be used by the device
interfaceNr defines the used device interface.
The tool usb-port, (written by Triadem) shows the hostNr.tierPort of all connected devices.

Description:

The devu-ftdix32-speed manager is a usb-uart device manager for QNX Neutrino. It supports multiple simultaneous USB-to-serial devices.

The serial driver's priority floats to the priority of the client. All internal events are processed at priority 24 (inherited from the internal pulse). The event handling priority is hard coded and isn't configurable by any of the options listed. (The driver's main.c program would need modification in order to change the priority). When the driver talks to a client application, it's running at the priority of the client. All other processing takes place either at priority 24r or at interrupt time.

Each device is given a name in the pathname space of /dev/ser*n*, where *n* starts at 100 (unless changed via the **-u** option) and increases.

☞ If your application uses /dev/console, you should create a link from it to one of /dev/ser1, /dev/ser2, ... by adding a line like this to the buildfile used by mkifs:

[type=link] /dev/console = /dev/ser1

All devices are support by default standard hardware flow control on input and output (RTS/CTS). This can be disabled by the -**F** option.

☞ Hardware flow control is not supported in edited mode.

A read request by default returns when at least 1 character is available. To increase efficiency, you can control three parameters to control when a read is satisfied:

Time
    Return after a specified amount of time has elapsed.
Min
    Return when this number of characters are in the input buffer.
Char
    Return if this forwarding character is in the input buffer.

☞ If the Min value is greater than the size of the input buffer, the Min value is clipped to the size of the buffer. To avoid this, the size of the input buffer can be changed with the **-l** option.

These parameters are set using library routines (see *tcgetattr()*, *tcsetattr()*, *readcond()* and *TimerTimeout()* in the *Library Reference*).

The devu-ftdix32-speed manager supports both raw and edited modes, making it a real tty device.

The following fields and flags are supported in the termios structure:

Field	Supported flags
c_cc	All characters
c_iflag	BRKINT ICRNL IGNBRK IXON
c_oflag	OPOST
c_cflag	CLOCAL CSIZE CSTOPB PARENB PARODD
c_lflag	ECHO ECHOE ECHOK ECHONL ICANON IEXTEN ISIG NOFLSH

Examples:

- Start devu-ftdix32-speed, defaulting for all USB-UART:

      devu-ftdix32-speed &

          result: /dev/ser100   /*with baud=115200 ...

                             /dev/ser101   /*with baud=115200 ...

- Start devu-ftdix32-speed, defaulting for all USB-UART, but change baud rate to 38400 from 115200 default:

      devu-ftdix32-speed -b38400 &

          result: /dev/ser100   /*with baud=38400 ...

                             /dev/ser101   /*with baud=38400 ...

- You can specify multiple **-F** and **-f** options; they're position-dependent, and affect the next serial port:

      devu-ftdix32-speed -b38400 –F –u200 –b57600 –f -u50 &

          result: /dev/ser200   /*with baud=38400 ...

                             /dev/ser50   /*with baud=57600 ...

                             /dev/ser51   /*with baud=57600 ...

- You can configure a specific serial port selected by the usb-port address:

      devu-ftdix32-speed –b460800 –f –u50,3.0 &

          result: /dev/ser50   /*with baud=460800 ...

- Start the driver for a new "not official" supported device:

      devu-ftdi-speed -V0x0403 -P0x6014

See also:

Character I/O drivers (devc-*) in the Utilities Summary

Connecting Hardware in the Neutrino *User's Guide*