

README.txt

Instructions for installing the D2XX shared lib

As Linux distributions vary these instructions are a guide to installation and use.

This setup works with Mandrake 9.2 but may require some investigation on other distributions.

This library has been tested using kernel 2.6.32.

D2XX documentation is available for the Windows DLL - some variations may occur between Linux and Windows implementation. We have endeavoured to make the APIs appear the same on both platforms however some issues may slip and we would appreciate that you contact support if you observe this.

D2XX for Linux was primarily developed to aid porting windows applications written with D2XX to Linux.

Unfortunately the source code for D2XX is not freely available - however if you prefer to have the source and are starting a project from scratch you can try libftdi from Thomas Jarosch.

libftd2xx uses an unmodified version of libusb (<http://libusb.wiki.sourceforge.net/>). Source code for libusb is included in the driver distribution in the libusb-1.0.8 directory.

Installation:

1. unzip and untar the file given to a suitable directory

```
gunzip libftd2xx1.0.4.tar.gz
```

```
tar -xvf libftd2xx1.0.4.tar
```

2. Change directory to the required architecture subdirectory, build/i386 for 32-bit or build/x86_64 for 64-bit.

3. As root user copy the following files to /usr/local/lib

```
cp libftd2xx.so.1.0.4 /usr/local/lib
```

3. Change directory to /usr/local/lib

```
cd /usr/local/lib
```

4. make symbolic links to these files using the following commands:

```
ln -s libftd2xx.so.1.0.4 libftd2xx.so
```

5. Change directory to /usr/lib

```
cd /usr/lib
```

6. make symbolic links to these files using the following commands:

```
ln -s /usr/local/lib/libftd2xx.so.1.0.4 libftd2xx.so
```

If you have problems with this check with usbview (search on the internet for application

or it can be sent to you by ftdi) or lsusb (command line tool) to check the usb file system is mounted properly.

Other problems will be related to the ftdi_sio driver loading -

1. you must unload this driver (and usbserial) if it is attached to your device ("rmmod ftdi_sio" and "rmmod usbserial" as root user).

2. Your PID/VID has not been included in the distribution. A PID of 0x6006 and VID of

0x0403 should work as a temporary workaround.

Release 1.0.4

Added support for FT232H devices.

Release 1.0.3

README.txt

Resolved issue in FT_ListDevices that could cause a crash.

Release 1.0.2

Eliminated dependence on GLIBC 2.7.

Eliminated dependence on GLIBC 2.8.

Fixed issues where invalid device serial numbers or descriptions could open channels on multi-interface devices.

Release 1.0.1

Resolved issue where attempting to unload kernel driver can allow multiple D2XX processes to access a single device concurrently.

Eliminated dependence on GLIBC 2.4.

Release 1.0.0

Major upgrade.

Now uses libusb 1.0.8 - requires udev (present on all 2.6 kernels).

Numerous bug fixes.

Now supports location IDs.

Release 0.4.14

Enhancement to enable library use in SuSe 10.2 default kernel.

Correct FT_GetDeviceInfoList behaviour.

Release 0.4.13

Software flow control fix (strip out flow control characters)

Added FT_SetDeadmanTimeout API call

Release 0.4.12

Configuration file (see Config.txt for further details) option added to perform a usb reset on device open.

Release 0.4.11

FT_SetUSBParameters enhanced - checks for invalid parameters

New driver version functions added.

Altered behaviour of a usb reset - only performed on unopened device (2.6 kernels only)

Improved multithreaded access.

Release 0.4.10

W32 functions and samples added (for the extra event handling)

FT232R EEPROM support added.

Some minor bug fixes regarding FT_OpenEx.

More robust checking of passed in arguments. Will fail on an invalid handle and NULL pointers instead of causing Segmentation Faults.

Release 0.4.9

Additional OpenEx bug fix - please report any future bugs as alternate distributions

may behave in a different manner.

100% CPU count fix when using FT_SetBitMode and not using FT_Read to clear internal buffers.

Release 0.4.8

Open bug fix - related to the detach of ftdi_sio kernel driver.

Release 0.4.7

Read timeout bug fix.

Release 0.4.6 (known as 0.4.6 on the web)

Bug fix to allow user area of FT2232 chip to be programmed.

Release 0.4.5 (known as 0.4.5 on the web)

Small change to allow detach of ftdi_sio kernel driver on device access.

Release 0.4.4 (known as 0.4.4 on the web)

Added FT_CreateDeviceInfoList, FT_GetDeviceInfoList, FT_GetDeviceInfoDetail,

README.txt

FT_EE_ReadEx and
FT_EE_ProgramEx.

Release 0.4.3 (known as 0.4.3 on the web)
Bug fix for 2.6 kernels with bulk writes - library now requires
"/proc/sys/kernel/osrelease" file to be present in system

Release 0.4.2 (known as 0.4.2 on the web)
Bug fix for the lib_table functionality. Now works with both static and dynamic
library versions.

Release 0.4.1 (known as 1.1.0.1 on the web)
Minor bug fix relating to modem event notification

Release 0.4.0 (known as 1.1.0.0 on the web)
User space driver compiled with gcc version 3.3.1
Major restructure of internals - it is now a pure c library so no c++ extensions
required.
Should help with some c++ linkage problems. Upped version to 0.4.0 for this
reason. PLEASE NOTE
due to the conversion from c++ to c there is an unlikely chance you will
experience problems.
If you do a recompile of your application may solve this. If not please contact
support. Users of the
PenScope and Dualscope so will experience problems related to this. Please
contact support for a suitable update.
FT_GetBitMode Bug Fix
FT_Read on a timeout Bug Fix
FT_GetStatus Bug fix
static library available (see static folder on how to compile)
static linkage to libusb to avoid installation issues

Release 0.3.0 (known as 1.0.0.0 on the web)
Timeouts added.
EEPROM reading/writing.
Event handling added (See example on how to use)
More examples to help with development
FT_SetVIDPID added - to allow setting of VID and PID - see example

Known issues: