



Future Technology Devices International Ltd.

Technical Note

TN_132 Adding FTDI Devices VCP Driver Support to Android

Document Reference No.: FT_000491

Version 1.1

Issue Date: 2011-08-25

This document describes the steps necessary to include the FTDI VCP driver for FTxxxx devices in an Android operating system build.

Future Technology Devices International Limited (FTDI)

Unit 1, 2 Seaward Place, Glasgow G41 1HH, United Kingdom
Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758
E-Mail (Support): support1@ftdichip.com Web: <http://www.ftdichip.com>

Copyright © 2011 Future Technology Devices International Limited

Table of Contents

1	Introduction	2
2	Building A Kernel With FTDI VCP Driver Support Included	3
2.1	Prerequisites.....	3
2.2	Configuring the Kernel	3
2.3	Building the Kernel	5
3	Modifying A Pre-Compiled Kernel To Include The FTDI VCP Driver Kernel Module	6
3.1	Prerequisites.....	6
3.2	Obtaining The Correct Kernel Source	6
3.3	Configuring The Kernel	8
3.4	Building And Inserting The Kernel Module	10
4	Permissions.....	12
5	Accessing The COM Port Via Java APIs.....	13
6	Contact Information.....	14
7	Appendix A – References.....	16
	Document References	16
	Acronyms and Abbreviations	16
8	Appendix B – List of Tables & Figures	17
9	Appendix C – Revision History.....	18

1 Introduction

The Android OS shares its kernel with the Linux OS. Linux supports serial ports, including USB serial ports, via kernel modules. Since the Linux kernel includes a kernel module to communicate with FTDI's FTxxxx devices via a serial port interface it is possible to build that same support into a kernel for the Android OS.

As the driver forms part of the kernel image for an embedded system, it is not a straightforward case of copying a driver file to a handset to enable this functionality. The developer needs to be in control of the OS at the source code level in order to include this functionality.

This document presents two alternative methods of including VCP support for FTDI devices in the Android OS; the first considers building a complete kernel image including the FTDI VCP driver module, the second addresses building a compatible FTDI kernel module and inserting it in to a functional kernel.

Customers wishing to support the FT232H device should note that support for this IC is included in kernel 2.6.39 and later. All other devices are supported from kernel 2.6.31.

2 Building A Kernel With FTDI VCP Driver Support Included

This section describes how to build FTDI VCP driver support in to a kernel image when building the entire OS from source.

2.1 Prerequisites

In order to include the FTDI VCP driver in an Android image and use it successfully, the following are required:

- A hardware platform including a USB host device supported by the Android/Linux kernel.
 - FTDI testing was conducted using a [BeagleBoard-xM Rev C](#).
- An FTDI based device for testing with
 - FTDI testing was conducted with an FT232R based US232R cable.
- A development system with the Android build tools installed and correctly configured.
 - FTDI development was conducted on a Linux machine running [Ubuntu 10.04](#) with the default kernel (2.6.32).
- Android OS source code, including kernel source, which has been configured and known to build a working Android OS for the chosen development platform.
 - FTDI testing was based on Android 2.3 with a 2.6.32 kernel as provided in the [TI Gingerbread development kit for OMAP devices](#), although some modifications were required to make it work with the BeagleBoard-xM Rev C platform. This kit also includes the required build tools.
 - The default kernel configuration was generated with the following command in the Android kernel source directory:

```
make config omap3_beagle_defconfig
```

2.2 Configuring the Kernel

First, the kernel source configuration file must be modified to include support for the appropriate modules. In particular, the modules required are USB Host support, USB Serial support and the FTDI SIO kernel module. The simplest and fastest way to add these is to modify the kernel config file using the menu to modify the default configuration via the following command (assuming that the current directory is the Android kernel source directory)

```
make menuconfig ARCH=arm CROSS_COMPILE=arm-eabi-
```

This will display a menu from which the appropriate modules can be selected. Select Device Drivers -> USB Support.

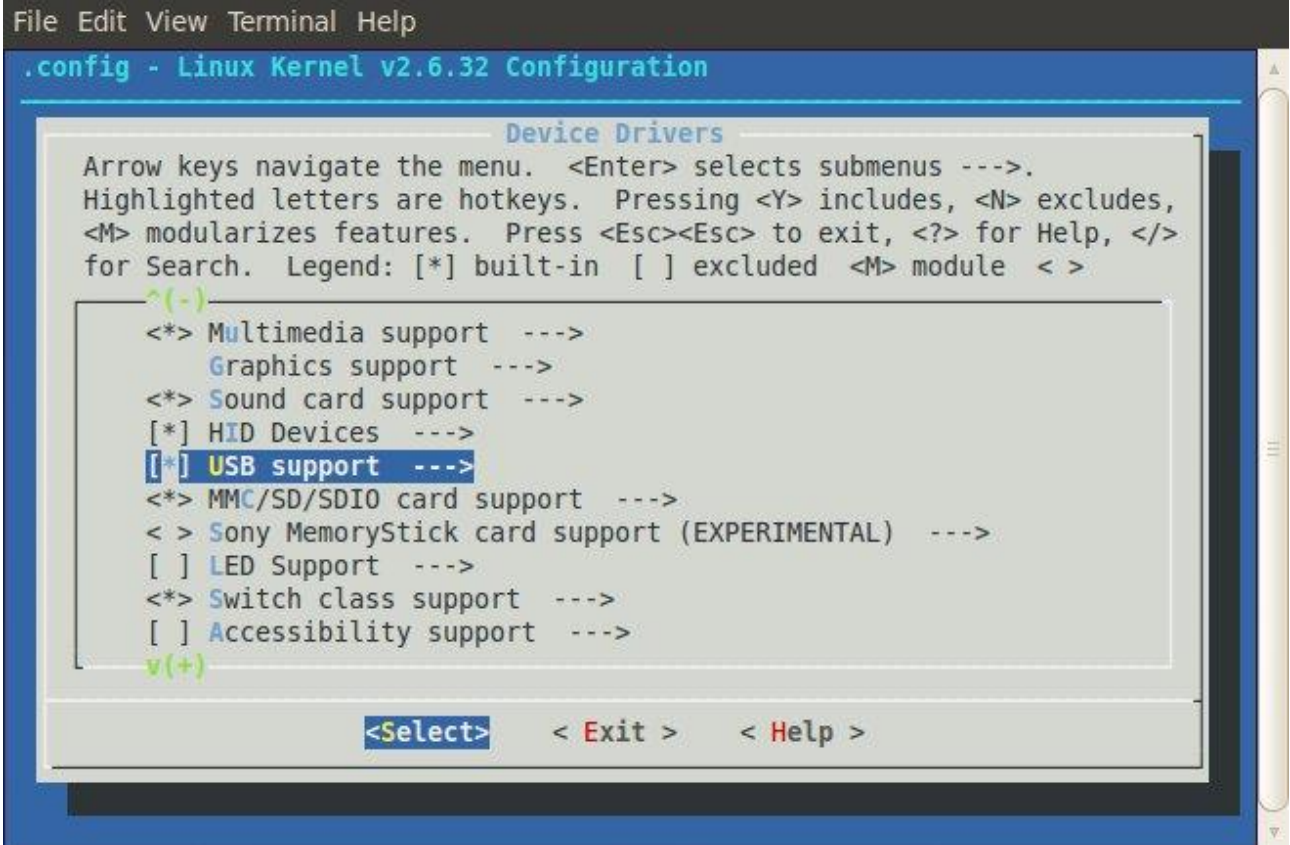


Figure 1: USB support in the kernel configuration menu

"Support for Host-side USB" should be marked as built-in; if it is not, it should be added. Select "USB Serial Converter Support" to display a new menu, then highlight "USB FTDI Single Port Serial Driver".

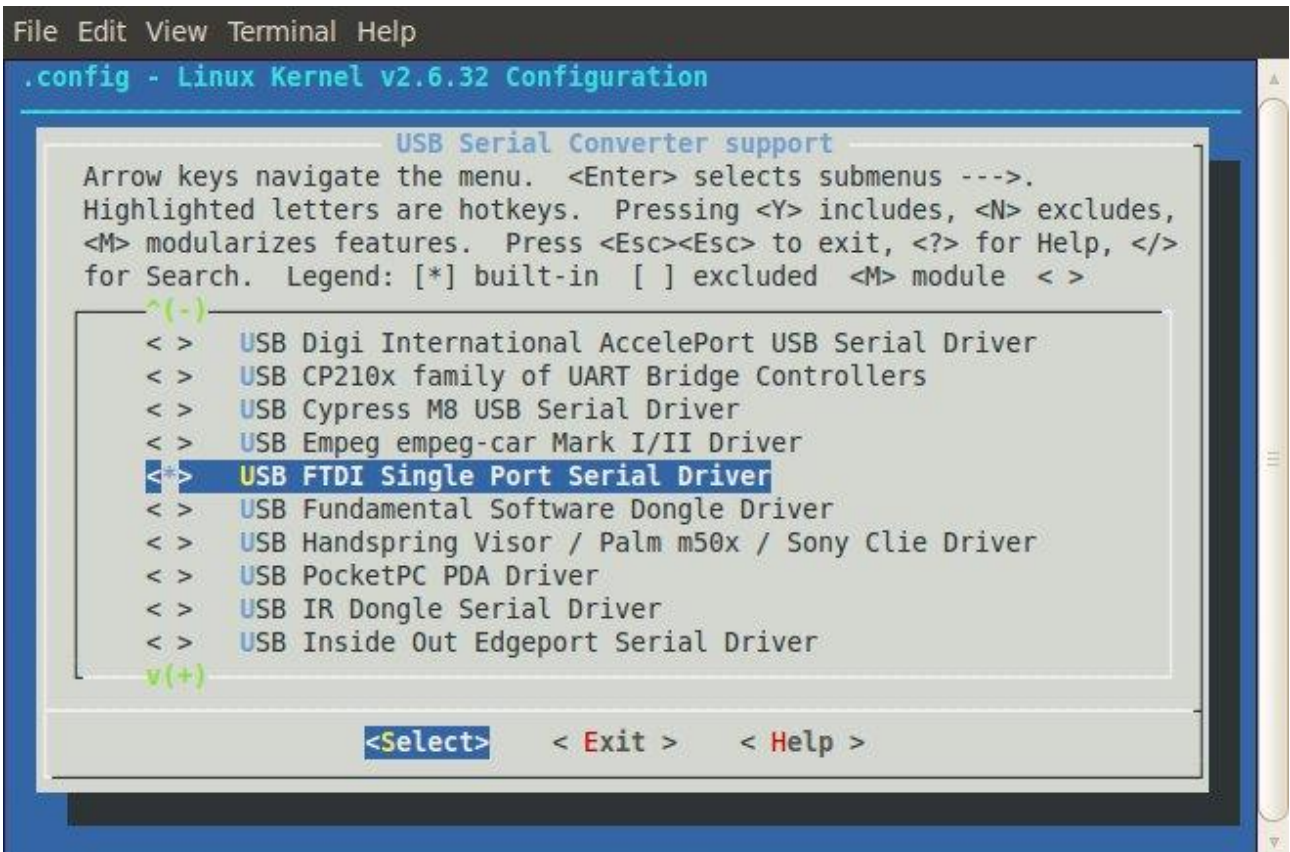


Figure 2: including FTDI VCP support in the kernel

Press "Y" to include support for this module in the kernel build. Please note that although the menu lists the driver as only supporting FTDI's single port devices, the driver included with the 2.6.32 kernel does in fact support all of the FTxxxx devices except the FT232H. To include support for the FT232H as well, kernel 2.6.39 or later should be used..

Exit the menu and select to save when prompted. At this point, the kernel can be rebuilt and the resulting file will include VCP support for FTxxxx devices.

2.3 Building the Kernel

Now that the kernel has been configured to include support for the VCP driver, the kernel can be rebuilt to include the kernel modules. To rebuild the kernel as an image file, execute this command:

```
make ARCH=arm CROSS_COMPILE=arm-eabi- uImage
```

This process may take some time depending on the host build system. Once the build is complete, the resulting kernel will contain the modules necessary to communicate with an FTxxxx device over USB.

Booting Android with the modified kernel and connecting an FTDI device to an available USB host port should now result in the VCP driver being loaded and the creation of an entry in the /dev directory of the format ttyUSB* where * is a number.

3 Modifying A Pre-Compiled Kernel To Include The FTDI VCP Driver Kernel Module

This section describes how to insert the FTDI VCP driver kernel module in to a working Android image. The directions provided here are based on directions provided on <http://android.serverbox.ch/?p=285> which were originally provided by Manuel Di Cerbo.

3.1 Prerequisites

In order to build the FTDI VCP driver kernel module and insert it in to an Android image, the following are required:

- A hardware platform including a USB host device supported by the Android/Linux kernel.
 - Note that root access must be available on the device to accomplish the insertion of the kernel module. Obtaining root access on a production Android device can be non-trivial and is outwith the scope of this document.
- An FTDI based device for testing with
 - FTDI testing was conducted with an FT232R based US232R cable.
- A development system with the Android build tools installed and correctly configured.
 - FTDI development was conducted on a Linux machine running [Ubuntu 10.04](#) with the default kernel (2.6.32).
 - The Android SDK and NDK packages must be installed.

3.2 Obtaining The Correct Kernel Source

First, determine the kernel version running on the Android device. This is important as a kernel module built for a different kernel version will most likely not work as intended.

This can be done in two ways. The kernel version is displayed on the Android device Settings screen under About Tablet

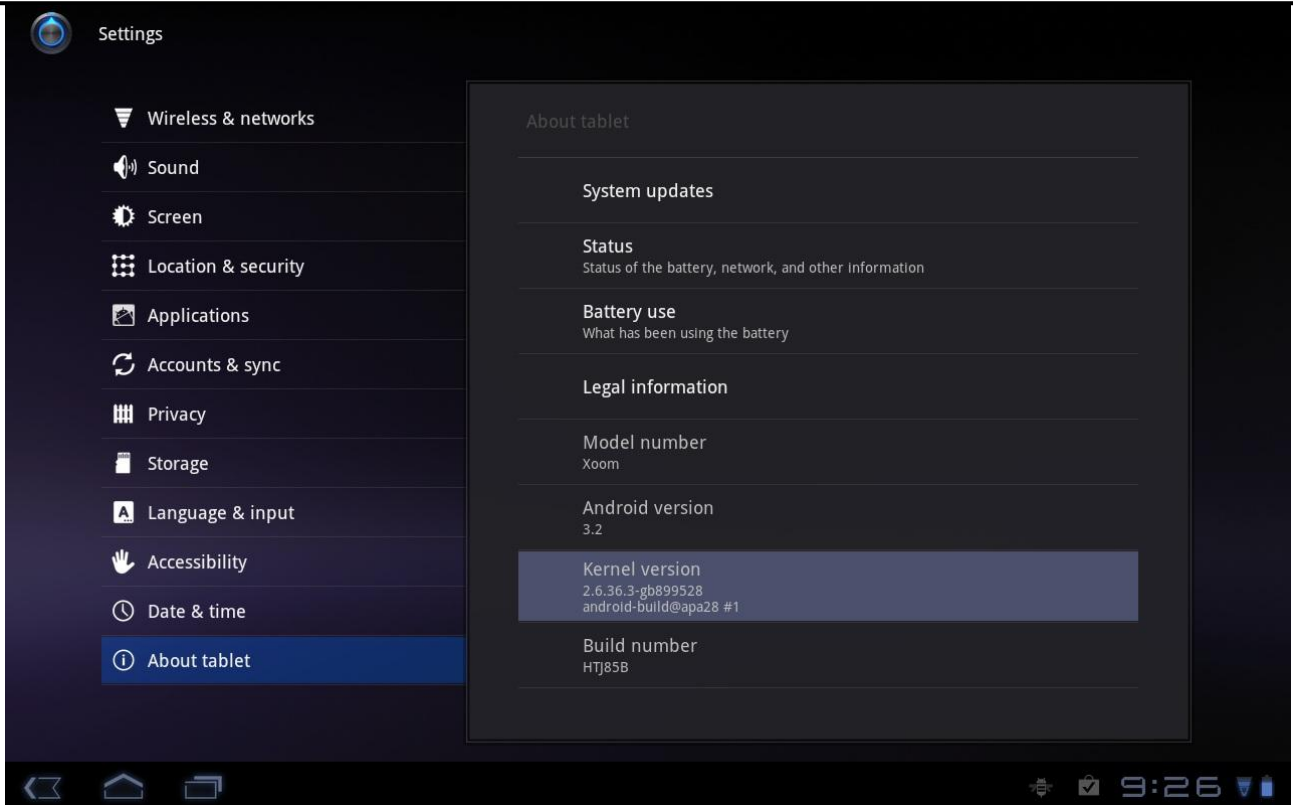


Figure 3: Android device Settings page showing the kernel version

Alternatively, the kernel version can be determined from a command prompt (either via a terminal emulator program or via an ADB shell session) with the command:

```
cat /proc/version
```

For the tablet matching the above screen shot, the output from this command is

```
Linux version 2.6.36.3-gb899528 (android-build@apa28.mtv.corp.google.com)
(gcc version 4.4.3 (GCC) ) #1 SMP PREEMPT Wed Jun 15 12:28:16 PDT 2011
```

It should be noted that the tablet used here does not have a USB host enabled and the information provided is for illustration only.

The obtained kernel version number can then be used to obtain the correct kernel sources from <http://android.git.kernel.org/>. Note that the processor type must also be known to find the correct area of the site for the required kernel source. In the example provided here, the processor is an NVIDIA Tegra 2 and the corresponding URL for the 2.6.36 kernel source is

<http://android.git.kernel.org/?p=kernel/tegra.git;a=tree;h=refs/heads/android-tegra-2.6.36;hb=refs/heads/android-tegra-2.6.36>

From here, a snapshot of the kernel can be downloaded. Once downloaded, extract the files to a new directory at ~/android-kernel-source. At the time of writing this document the file available from here is tegra-android-tegra-2.6.36-9001adc.tar.gz

3.3 Configuring The Kernel

The existing kernel configuration can be extracted from the Android device via ADB with the command

```
adb pull /proc/config.gz config.gz
```

The existing Android kernel configuration can then be applied to the snapshot obtained above. In order to do this, execute the following command on the development machine (assuming that the kernel snapshot was extracted to the ~/android-kernel-source/ directory):

```
cat config.gz > ~/android-kernel-source/tegra-android-tegra-2.6.36-9001adc/.config
```

Once the existing configuration is applied, it can then be modified to include support for the FTDI VCP kernel module. To do this, use menuconfig from the Android NDK. Change to the Android kernel source directory:

```
cd ~/android-kernel-source/tegra-android-tegra-2.6.36-9001adc
```

Assuming that the Android NDK is installed in the /opt/ directory, add the NDK directory to the PATH as follows:

```
export PATH=/opt/android-ndk-r5/toolchains/arm-eabi-4.4.0/prebuilt/linux-x86/bin/:$PATH
```

This allows us to use the Android NDK build tools to build all or part of the Android kernel from the source directory. Execute the following command to display a kernel configuration menu:

```
make menuconfig ARCH=arm CROSS_COMPILE=arm-eabi-
```

This will display a menu from which the appropriate modules can be selected. Note that in order to be able to build the FTDI VCP driver as a module, loadable module support must be enabled:

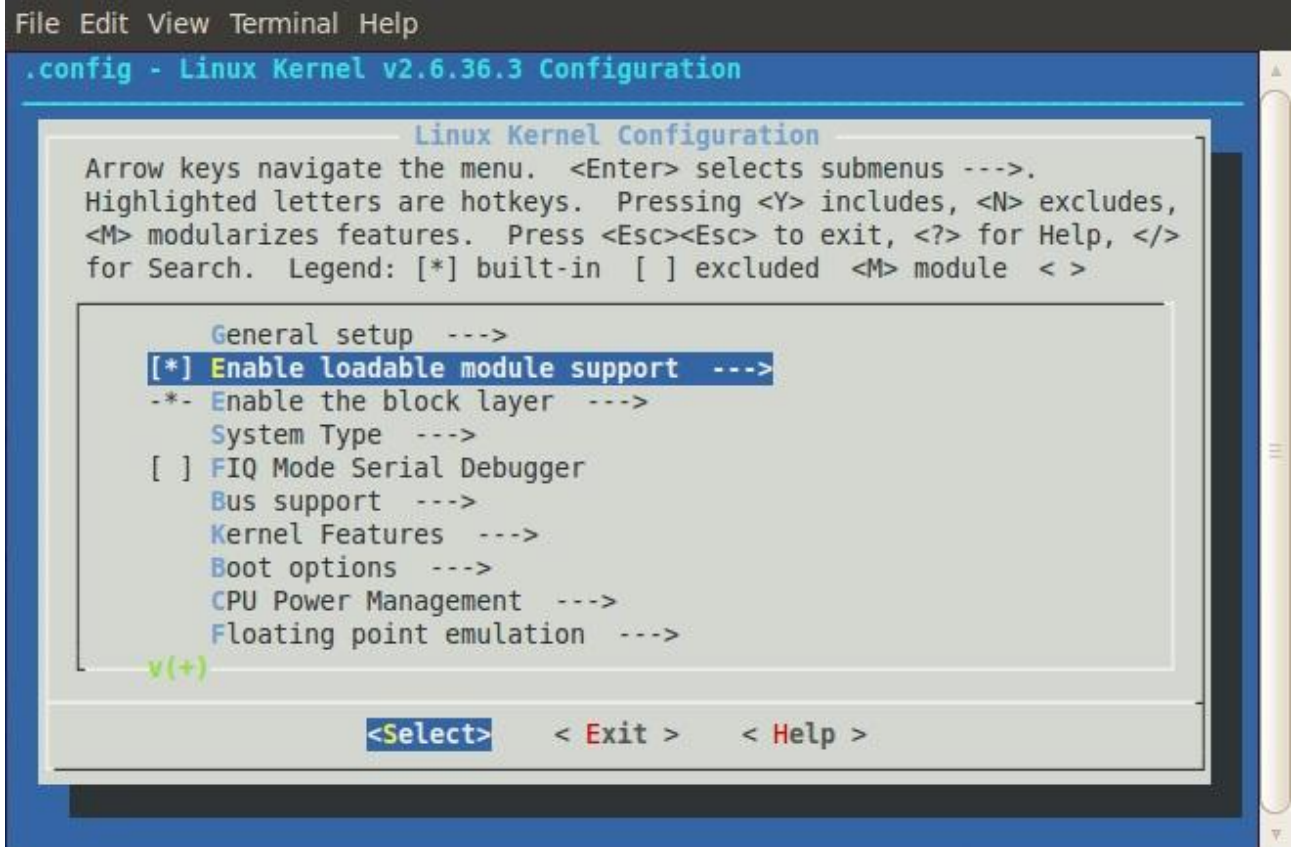


Figure 4: menuconfig showing Loadable Module Support

Select Device Drivers -> USB Support. Select "USB Serial Converter Support". Press "M" to include this support as a kernel module and enable a sub-menu, then highlight "USB FTDI Single Port Serial Driver". Press "M" to select the FTDI VCP driver as a kernel module.

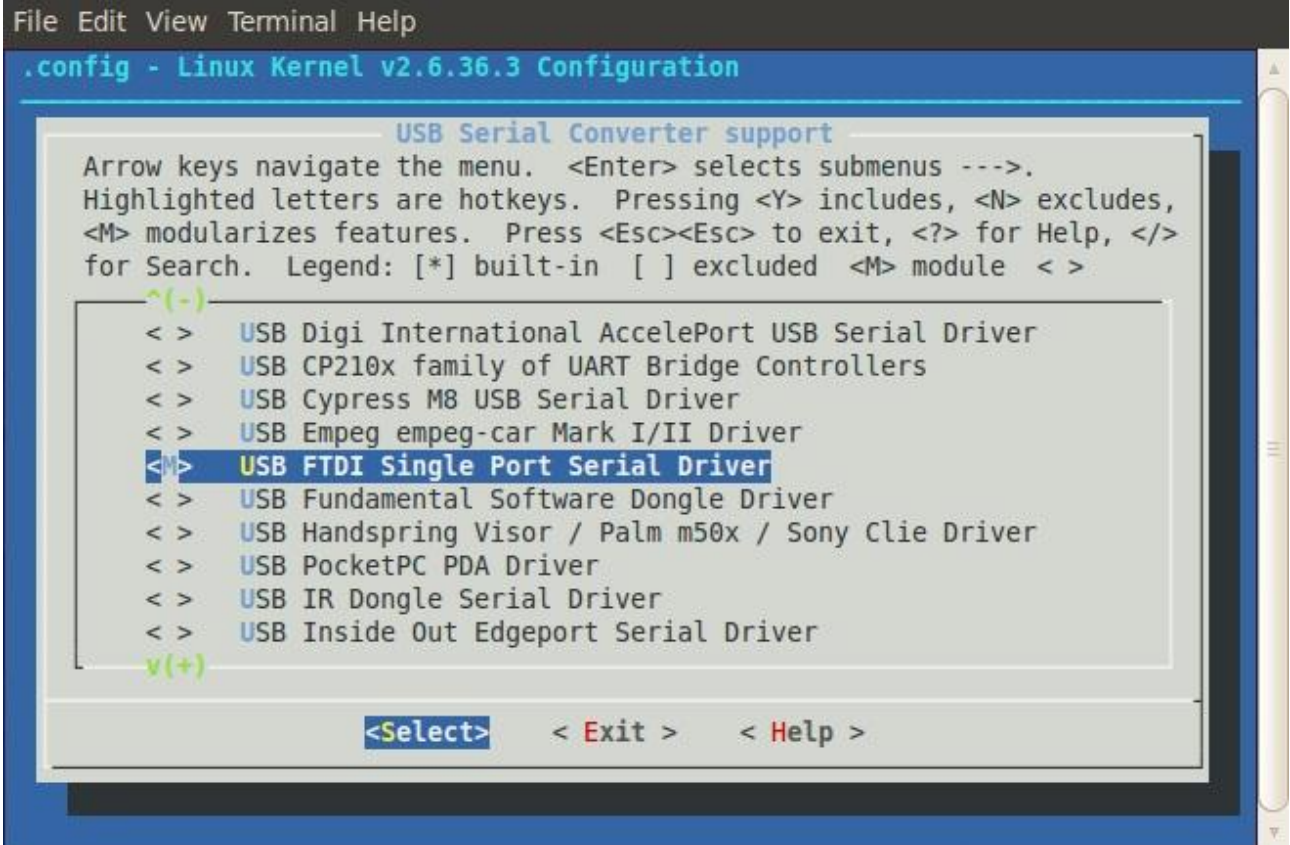


Figure 5: enabling FTDI VCP driver support as a kernel module

Select Exit and choose to save the configuration when exiting.

To complete the kernel configuration, the Makefile should be edited to match the extraversion number of the extracted kernel with the sources downloaded from <http://android.git.kernel.org>; this is listed at the top of the Makefile.

3.4 Building And Inserting The Kernel Module

Ensure that the Android NDK is on the PATH. Assuming that the Android NDK is installed in the /opt/ directory, add the NDK directory to the PATH as follows if this has not already been done:

```
export PATH=/opt/android-ndk-r5/toolchains/arm-eabi-4.4.0/prebuilt/linux-x86/bin/:$PATH
```

To build kernel modules, execute the following command in the Android kernel sources directory:

```
make modules ARCH=arm CROSS_COMPILE=arm-eabi-
```

Once complete, the ftdi_sio.ko kernel module will be present in the drivers/usb/serial directory.

Next, copy the kernel module from the development machine to the Android device. This can be done via ADB with the following command:

```
adb push drivers/usb/serial/ftdi_sio.ko /sdcard
```

The module must then be inserted in to the kernel. This must be done via the adb shell and requires root permissions. If root permissions are not available, the operation will fail. Start an ADB session with this command on the development machine:

```
adb shell
```

Once the ADB shell session is running, execute the following commands to insert the kernel module:

```
su  
cd /sdcard  
insmod ftdi_sio.ko
```

At this point the process is complete and the FTDI VCP driver is available on the Android device.

4 Permissions

Although the kernel module will now load for an FTxxxx device when it is enumerated and create a device in the /dev directory, the default permissions for the device will not allow the default Android user to access the port.

To resolve this issue, a modification is required to the ueventd.rc file in the Android root file system (rootfs) to set the permissions appropriately. Appending the following line to the end of the ueventd.rc file should resolve the issue:

```
/dev/ttyUSB* 0666 root root
```

Once this line is included, the serial port permissions will allow read and write access for all users and native applications should be able to access the serial port at this point. For example, commands at the Android terminal emulator such as

```
echo "Hello serial port" > /dev/ttyUSB0
```

and

```
cat /dev/ttyUSB0
```

should work as expected.

5 Accessing The COM Port Via Java APIs

Despite native applications now having access to the serial port device, applications developed using the Android SDK will not be able to access the serial port since the Android platform does not include an API to access serial ports from Java applications. This is a serious limitation for Android application developers wishing to access FTxxxx devices via the VCP driver.

An open source project has been created to provide the missing serial port Java APIs for Android (<http://code.google.com/p/android-serialport-api/>). This project allows access to serial ports, USB or otherwise, from a Java application on the Android platform. It has been tested on FTDI's BeagleBoard-xM Rev C development platform and has been shown to work after being recompiled from source to remove the O_DIRECT flag from line 96 of the SerialPort.c source file; the resulting library (libserial_port.so) should then be copied to /rootfs/system/lib and the application should be copied to /rootfs/system/app. After copying these files, the SerialPort application will appear on the main Android application screen and will communicate with the USB serial port.

Please note however, that this is an open-source community project and is not maintained or supported by FTDI.

6 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com
Web Site URL <http://www.ftdichip.com>
Web Shop URL <http://www.ftdichip.com>

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan, R.O.C.
Tel: +886 (0) 2 8791 3570
Fax: +886 (0) 2 8791 3576

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com
Web Site URL <http://www.ftdichip.com>

Branch Office – Hillsboro, Oregon, USA

Future Technology Devices International Limited (USA)
7235 NW Evergreen Parkway, Suite 600
Hillsboro, OR 97123-5803
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) us.admin@ftdichip.com
Web Site URL <http://www.ftdichip.com>

Branch Office – Shanghai, China

Future Technology Devices International Limited (China)
Room 408, 317 Xianxia Road,
Shanghai, 200051
China
Tel: +86 21 62351596
Fax: +86 21 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com
Web Site URL <http://www.ftdichip.com>

Distributor and Sales Representatives

Please visit the Sales Network page of the [FTDI Web site](#) for the contact details of our distributor(s) and sales representative(s) in your country.

System and equipment manufacturers and designers are responsible to ensure that their systems, and any Future Technology Devices International Ltd (FTDI) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested FTDI devices and other materials) is provided for reference only. While FTDI has taken care to assure it is accurate, this information is subject to customer confirmation, and FTDI disclaims all liability for system designs and for any applications assistance provided by FTDI. Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, United Kingdom. Scotland Registered Company Number: SC136640

7 Appendix A – References

Document References

<http://developer.android.com>

<http://beagleboard.org/hardware-xM>

http://processors.wiki.ti.com/index.php/TI-Android-GingerBread-2.3-DevKit-1.0_ReleaseNotes

<http://www.ubuntu.com/>

<http://www.kernel.org/>

<http://code.google.com/p/android-serialport-api/>

<http://android.serverbox.ch/?p=285>

Acronyms and Abbreviations

Terms	Description
OS	Operating System
USB	Universal Serial Bus
VCP	Virtual COM Port
SDK	Software Development Kit
ADB	Android Debug Bridge
NDK	Native Development Kit

8 Appendix B – List of Tables & Figures

List of Tables

No table of figures entries found.

List of Figures

Figure 1: USB support in the kernel configuration menu	4
Figure 2: including FTDI VCP support in the kernel	4
Figure 3: Android device Settings page showing the kernel version	7
Figure 4: menuconfig showing Loadable Module Support	9
Figure 5: enabling FTDI VCP driver support as a kernel module	10

9 Appendix C – Revision History

Revision	Changes	Date
1.0	Initial Release for beta test	2011-06-20
1.0	Initial release to the FTDI web	2011-06-29
1.1	Added directions for building only the kernel module. Clarified details for SerialPort API.	2011-08-25