PhyFLEX-i.MX6 RDK Linux Quickstart PD12.0.3

1 About this Quickstart

This document describes how to install and work with the Linux Board Support Package (BSP) for the phyFLEX-i.MX6 platform. This BSP provides a fundamental software platform for development, deployment and execution on the phyFLEX-i.MX6.

The Quickstart contains instructions for:

- Host Setup
- Board Setup
- Building a BSP (Platform, Kernel, Root Filesystem)
- Flashing images (Barebox and Kernel)

2 Host Setup

The phyFLEX-i.MX6 (PFL-A-02) has been developed and tested with Ubuntu 10.04 LTS Lucid Lynx [Installation Guide]. Although it is possible to use a different OS, some setup information will contain OS-specific commands and paths for settings.

Update repositories and upgrade installed packages:

```
sudo apt-get update
sudo apt-get upgrade
```

2.1 Server Setup

Support for installing and setting up TFTP, NFS, and Samba server settings to enable communication between multiple systems and the target.

2.1.1 TFTP

TFTP allows files to be downloaded from one machine to another. With most embedded Linux devices, TFTP is an efficient way to boot the kernel during development so that the user does not have to flash a new kernel every time it is modified. It is also helpful when updating images in flash from Barebox.

First, start by installing the TFTP server.

```
sudo apt-get install tftpd-hpa
```

Next, files can be accessed from another machine on the same network by simply using the IP address of the host. Specify a folder where the files will reside on the host by replacing the folder path for TFTP_DIRECTORY with whatever folder you wish to use as your TFTP file storage location, or leave the folder as the default.

```
sudo gedit /etc/default/tftpd-hpa

# /etc/default/tftpd-hpa

TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/var/lib/tftpboot"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure"
```

If you made any changes to the settings of the TFTP server, you need to restart it for them to take effect.

```
sudo restart tftpd-hpa
```

Lastly, if you would like to grant every user on the system permission to place files in the TFTP directory, use the following command, replacing <TFTP_DIRECTORY> with your chosen location.

```
sudo chmod ugo+rwx <TFTP_DIRECTORY>
```

2.1.2 NFS

A network file-system (NFS) server gives other systems the ability to mount a file-system stored on the host and exported over the network. In embedded development, this is essential for quickly testing applications and evaluating different file-system setups.

To begin the installation process use the following command:

```
sudo apt-get install nfs-kernel-server
```

Exported filesystems are designated in the "/etc/exports" file and allow you to choose both the directory to be exported and many settings for accessing the exports. Below is an example for exporting a folder called "nfs_export-ex" located in a user's home directory.

```
sudo gedit /etc/exports

# /etc/exports

/home/<user>/nfs_export-ex  *(rw,sync,no_root_squash,no_subtree_check)
```

The options (rw, sync, no_root_squash, no_subtree_check) for this folder are essential in setting up the NFS export correctly. For more information on additional options, refer to the man page for 'exports'.

- rw enables

read and write access when the directory is mounted

- sync

tells the file-system to handle local access calls before remote access

- no_root_squash

allows root access when mounting the file-system

- no_subtree_check

reduces the number of checks the server must make to ensure that an exported sub-directory is within an exported tree and also enables access to root files in conjunction with no_root_squash

After modifying this file, in order to mount the directories as an NFS, you must force the NFS server to export all of the directories listed in "/etc/exports".

```
sudo /usr/sbin/exportfs -va
```

2.1.3 Samba

Samba servers are an excellent way to access a Linux file-system on a Windows machine via a network connection. Using a Samba server, it is quick and easy to transfer files between systems. To install a Samba server, use the following command:

```
sudo apt-get install samba
```

Before the Samba share can be mounted on another machine it's necessary to modify the configuration file to allow write access and access to home directories. Start by editing the "/etc/samba/smb.conf" file.

```
sudo gedit /etc/samba/smb.conf
```

Inside this file there are four specific things that need to be uncommented (remove the ';' at the beginning of the line) to enable the sharing of home folders and write access. Below is the section that must be modified:

```
#====================== Share Definitions ======================

# Un-comment the following (and tweak the other settings below to suit)
# to enable the default home directory shares. This will share each
# user's home directory as \\server\username
;[homes]
;   comment = Home Directories
;   browseable = yes

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
;   read only = no
```

The outcomes after the changes are made follow:

```
#====================== Share Definitions ======================

# Un-comment the following (and tweak the other settings below to suit)
# to enable the default home directory shares. This will share each
# user's home directory as \\server\username
[homes]
   comment = Home Directories
   browseable = yes

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
   read only = no
```

NOTE: It might also be necessary to change the "workgroup = " line to match the workgroup for your machine.

To apply the changes, the next step is to restart all Samba-related processes.

```
sudo restart smbd
sudo restart nmbd
```

Lastly, each user needs to have a password enabled to be able to use the Samba server. There are no rules for this password. The simplest method for choosing this password is to make it the same as the UNIX user's password, but it is not a requirement. After typing in the command below, you will be prompted to enter the password for the specified user.

```
sudo smbpasswd -a <user>
```

As mentioned in the configuration file, the samba share can be connected by accessing "\\<host machine ip>\\<user>" by either mounting a network share or using Windows explorer to navigate to it.

2.2 PTXdist

2.2.1 General Information

PTXdist is a set of tools created by Pengutronix to help manage all of the BSP sources, from x-loader to the filesystem and its applications. It makes it easier for a user to manage everything for specific platforms and toolchains without manually repeating relatively complex commands every time it is necessary to build a new image. It is important to note the PTXdist version because more than one may be necessary for building the toolchain and BSP.

2.2.2 Extracting Sources

Visit the phyFLEX-i.MX6's BSP page for information on the current versions of BSP tools.

Potentially, two versions of PTXdist will need to be downloaded:

1. ptxdist-2011.11.0 for installing/constructing the cross-compile toolchain (Optional if using a pre-existing environment)
2. ptxdist-2012.03.0 for building the BSP with the toolchain mentioned above.

The following instructions correspond to downloading PTXdist for building the BSP, *ptxdist-2012.03.0*. Therefore, to download the PTXdist software for building the toolchain, *ptxdist-2011.11.0*, repeat the instructions replacing *2012.03.0* with *2011.11.0*. In order to be built, extract the archive containing the PTXdist software:

```
tar -jxvf ptxdist-2012.03.0.tar.bz2
cd ptxdist-2012.03.0
```

Now that the source has been extracted, the next step is to configure it for building.

2.2.3 Pre-Requisites

PTXdist checks for specific packages that must be installed before it can be successfully built. From the PTXdist source directory, *ptxdist-2012.03.0*, execute the command to begin a script that uses GNU autotools to help set up the environment for building the distribution.

```
./configure
```

This command automatically stops if it is missing a package and states why and what package to install to continue with the initial setup. After successfully running the configure script, build and install PTXdist.

From the PTXdist source directory, *ptxdist-2012.03.0*:

```
make
sudo make install
```

The install location is "/usr/local" by default, which is why the **make install** command must be run with root privileges. If another directory is preferred for the install, use the --prefix option when installing, but be sure to add the "bin/" directory of the installed tools in this new folder to $PATH by modifying and sourcing ~/.bashrc.

```
source ~/.bashrc
```

Now that the install is complete, the PTXdist folder can be removed, as well as the original archive.

Optional:

```
cd ..
rm  -rf ptxdist-2012.03.0*
```

2.2.4 Using PTXdist

PTXdist is a console command tool and different functions are run by extending parameters to the **ptxdist** base command.

```
ptxdist <parameter>
```

To generate a full list of parameters and a description of the function, use the **help** command:

```
ptxdist help
```

Since PTXdist versions can be installed in parallel it may be useful to view the version number corresponding to the **ptxdist** command:

```
ptxdist --version
```

If the output of this command does not correlate to the desired version of PTXdist, affix the **ptxdist** command with the version number. For example, for the phyFLEX-i.MX6 BSP, affix all **ptxdist** commands with **2012.03.0** resulting in **ptxdist-2012.03.0**:

```
ptxdist-2012.03.0 <parameter>
```

Alternatively, to correlate the **ptxdist** command with a specific version create a symbolic link. For example, for **ptxdist** to correlate to **ptxdist-2012.03.0**:

```
sudo ln -fs /usr/local/lib/ptxdist-2012.03.0/bin/ptxdist /usr/local/bin/ptxdist
```

Potential Issues:

- Wrong PTXdist Version

It is important to use the correct version of PTXdist when building the Toolchain or BSP. If the incorrect version is used, the following will result:

error: The ptxconfig file version and ptxdist version do not match:configfile version: 2012.03.0ptxdist version: 2011.11.0

You can either migrate from an older ptxdist release with:'ptxdist migrate'

or, to ignore this error, add '--force'to ptxdist's parameters, e.g.:'ptxdist --force go'

The version of PTXdist used is noted as *ptxdist version* where the one required is specified as *configfile version*. Therefore, rerun the command with the correct version either appended to **ptxdist** or create a symbolic link. If the *merge* or *force* options were used on the Toolchain or BSP they will need to be removed and reinstalled to be built with the correct PTXdist version. Please note that the above is a generic example and may not apply directly to your BSP.

2.2.4.1 Setup

The first time PTXdist is used, there are some setup properties that have to be configured. To run PTXdist's setup, use the following command:

```
ptxdist setup
```

Once in the ptxdist setup, the only settings that should be modified are the *User->Name* and *User->eMail*. This is mainly for general logging purposes. If you wish to modify any other settings, please refer to the Getting Help section for a link to PTXdist documentation.

Since PTXdist works with sources only, it needs to grab source archives from the web using wget as it advances through its setup if they do not exist already. Therefore, an internet connection is required.

2.3 Toolchains

In order to build images or applications for an embedded device, it is necessary to have a cross toolchain that will perform the necessary operations to compile code for a specified processor and system setup.

Each toolchain will have a modified GNU Compiler Collection (gcc) designed for the desired setup. The phyFLEX-i.MX6 PD12.0.3 uses the arm-cortexa9 toolchain which can be built from the OSELAS.Toolchain-2011.11.1 and ptxdist-2011.11.0 sources.

2.3.1 Existing Toolchains

If a working toolchain is already installed for a given architecture, it is possible to use this instead of building an OSELAS Toolchain. Do note that since external toolchains have not been tested it is possible that they may not work properly across all environments.

An extra step needs to be taken if the plan is to use an external toolchain. By default, the software package will be set to check for the vendor toolchain that it was compiled with. In order to change this, use the following command inside the root directory of the BSP:

```
ptxdist platformconfig
    architecture --->
        toolchain --->
            () check for specific toolchain vendor
```

2.3.2 Building OSELAS Toolchains

An OSELAS toolchain, managed by PTXdist, can be easily built for a target architecture.

For the phyFLEX-i.MX6, the arm-cortexa9 architecture based toolchain is built from OSELAS.Toolchain-2011.11.1 [Here] and PTXdist 2011.11.0 [Here]. See Section 2.2 for information regarding the installation of PTXdist sources.

```
tar -jxvf OSELAS.Toolchain-2011.11.1.tar.bz2
cd OSELAS.Toolchain-2011.11.1
```

Be sure to use the correct ptxdist version for the toolchain by affixing all **ptxdist** commands with **2011.11.0** resulting in **ptxdist-2011.11.0** or see Section 2.2.4 for information on creating a symbolic link between **ptxdist** and **ptxdist-2011.11.0**.

```
ptxdist select ptxconfigs/arm-cortexa9-linux-gnueabi_gcc-4.6.2_glibc-2.14.1_binutils-2.21.1a_kernel-2.6.39-
sanitized.ptxconfig
ptxdist go
```

The toolchain is now built and installed in */opt/OSELAS.Toolchain-2011.11.1/arm-cortexa9-linux-gnueabi* and ready to be used for building the BSP.

If you wish to build applications outside of the BSP directory, add the toolchain location to your PATH. Use the following from the command line or permanently add to your PATH by including it in .bashrc:

```
PATH=/opt/OSELAS.Toolchain-<toolchain version>/arm-<processor>-linux-gnueabi_gcc-linaro-<version>_glibc-
<version>_binutils-<version>_kernel-<version>-sanitized/bin/:$PATH
```

Following a successful build, the OSELAS.Toolchain-2011.11.1 folder can be removed, as well as the original archive.

Optional:

```
cd ..
rm  -rf OSELAS.Toolchain-2011.11.1*
```

2.3.2.1 Protecting Toolchains

It is recommended, although optional, to set the /opt/OSELAS.Toolchain-2011.11.1/arm-cortexa9-linux-gnueabi directory and its contents as read-only to prevent accidental modifications to the toolchain.

3 Board Setup-phyFLEX-i.MX6

The phyFLEX-i.MX6 comes pre-flashed with barebox, the Linux kernel, and root filesystem. After the device is out of the box and setup, simply applying power will boot the pre-installed images from NOR flash.

3.1 Connections

Power and host-PC connections must be made to the target device. The hardware manual, included with the Rapid Development Kit, may be referred to for specific connection information.

3.1.1 Power

The primary input power for the phyFLEX-i.MX6 Carrier Board comes from the wall adapter jack, X12 (+12 V). Upon application of power, LED D100 and D102 should light up (green) and initial serial data will be sent by UART0. The Carrier Board provides options for a warm reset or system power ON/OFF without the removal of the power source through push buttons S1 and S2, respectfully.

3.1.2 Serial

A serial connection is used as system communication for boot-up interaction throughout start-up and as a monitoring/debugging interface. This connection is made between the Host and UART0 on the phyFLEX-i.MX6.

The following provides a summary of the serial settings required to allow console access over the serial port in a communications program on the host such as Minicom:

Setting	Value
Bits per second	115200 bsp
Data bits	8-bits
Stop-bit	1
Flow Control	None

3.1.2.1 Minicom

Minicom is the recommended communications program on the host for serial communication to the device.
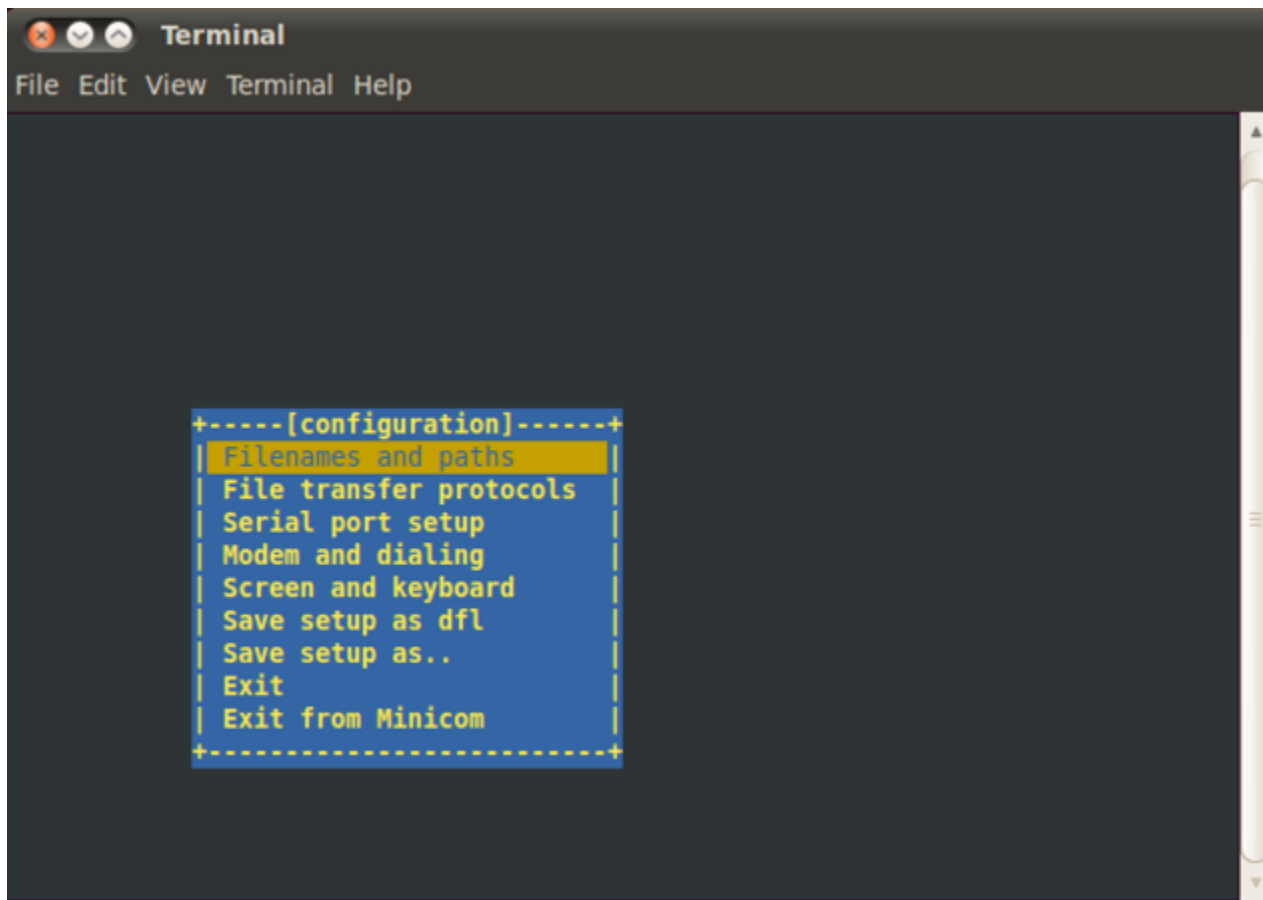
To install Minicom, execute the following in a terminal on the host:

```
/* if Minicom is not installed */
sudo apt-get install minicom
```

Start minicom from the terminal in the following way:

```
minicom -c on -s
```

Minicom will be executed and the main menu will be displayed in the terminal:

Navigate to "Serial port setup" in the Minicom main menu and modify line *A - Serial Device :* to read /dev/ttyS0 and line *E - Bps/Par/Bits :* to have a speed of 115200 and 8-N-1 (8N1) for the stop bits:

```
+------------------------------------------------------------------+
| A -     Serial Device        : /dev/ttyS0                        |
| B - Lockfile Location        : /var/lock                         |
| C -     Callin Program       :                                   |
| D -   Callout Program        :                                   |
| E -       Bps/Par/Bits       : 115200 8N1                        |
| F - Hardware Flow Control : Yes                                  |
| G - Software Flow Control : No                                  |
|                                                                  |
|    Change which setting?                                        |
+------------------------------------------------------------------+
        | Screen and keyboard    |
        | Save setup as dfl      |
        | Save setup as..        |
        | Exit                   |
        | Exit from Minicom      |
        +------------------------+
```



```
+------------------+---------[Comm Parameters]----------+----------------+
| A -     Serial De|                                    |                |
| B - Lockfile Loc|      Current: 115200 8N1           |                |
| C -    Callin Pro| Speed            Parity      Data  |                |
| D -   Callout Pro| A: <next>        L: None     S: 5  |                |
| E -      Bps/Par/B| B: <prev>        M: Even     T: 6  |                |
| F - Hardware Flo| C:   9600        N: Odd      U: 7  |                |
| G - Software Flo| D:  38400        O: Mark     V: 8  |                |
|                  | E: 115200        P: Space         |                |
|    Change which  |                                    |                |
+------------------| Stopbits                          |----------------+
        | Screen a| W: 1             Q: 8-N-1          |
        | Save set| X: 2             R: 7-E-1          |
        | Save set|                                    |
        | Exit    |                                    |
        | Exit fro| Choice, or <Enter> to exit?        |
        +---------+----------------------------------+
```

Note: *The serial device is dependent on what COM port you are connected to on your system, so /dev/ttyS0 is merely an example.*

Return to the main menu of minicom and select **Save setup as dfl** to make this the default setup anytime Minicom is loaded, meaning minicom -c on is all that needs to be done in the future for this machine to be able to communicate with the kit. Be sure that permissions allow writing to *minirc.dfl* by:

```
sudo chmod ugo+rwx /etc/minicom
```

3.1.3 Ethernet

The Ethernet connection is used for flashing, downloading, and debugging images and applications. Connect the cross-over Ethernet cable to the Ethernet connector on the target, X28, and appropriate network card on the host. LINK (green) and SPEED (yellow) indicator LEDs are integrated on the RJ-45 connector.

3.2 Image Format

The bootloader, kernel, and root filesystem specific to the phyFLEX-i.MX6 can be provided over a wide variety of sources such as the PHYTEC FTP, preloaded to NOR Flash, SD Card, TFTP Server, or NFS Server.

3.2.1 PHYTEC FTP

If for any reason it is necessary to re-flash the example images, they are located on the PHYTEC America FTP [here].

File	Description
barebox-image	Primary bootloader
barebox-default-environment	Default configuration environment for Barebox
uImage-pfla02	Linux kernel image
root-pfla02.ubi	Root filesystem for NAND and NOR Flash
root-pfla02.tgz	Root filesystem for SD/MMC

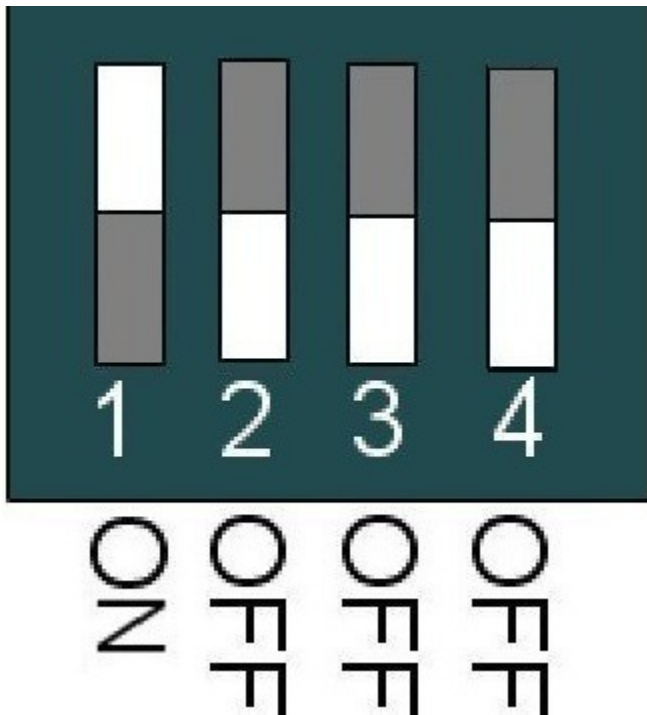Refer to the Flashing Images section for information on how to flash these images.

3.3 Booting Configurations

The bootloader, one of the key software components included in the BSP, completes the required hardware initializations to download and run operating system images. The boot mode, selected from the S3 dipswitch on the Carrier Board, determines the location of the primary bootloader, barebox-image. In this release, NOR Flash is the only supported boot mode, the S3 switch settings to specify the boot mode is given by the following:

3.3.1 SPI NOR

To boot from SPI NOR, use the following switch settings:

S3-1 ONS3-2, S3-3, S3-4 OFF

3.4 Working with Barebox

The phyFLEX-i.MX6 uses Barebox as its bootloader, which is provided with a predefined environment setup to support the target device. After connecting the board to a terminal emulator via a serial connection, apply power to start booting the target. Approximately three seconds are allotted to hit any key which will halt autoboot and enter Barebox:

A full list of available barebox commands along with their usage is available by the **help** command:

```
help
```

Barebox can be customized with environment variables and scripts to support a wide variety of booting and flashing options.

3.4.1 Environment Variables

The settings most necessary for operation are environment variables in Barebox. To obtain a list of current environment variables, use the **printenv** command:

```
printenv
```

3.4.2 Configuration File

The configuration file provided by Barebox is located in /env/config, this file allows the user among many things, to modify environment variables and setup networking parameters. To open and make edits to the /env/config file, do the following in Barebox:

```
edit /env/config
```

Use the configuration file for editing Barebox environment variables to specify network settings; image names, types, and default locations; and display selection.

To quit editing the configuration file, return to the Barebox prompt by pressing **CTL+D** to apply changes or **CTL+C** to cancel.

During start up Barebox loads a persistent environment including files and directories from /dev/env0, target's flash, to /env. Therefore, to store any changes made to the files in /env, such as the configuration file, use the **saveenv** command to save to /dev/env0:

```
saveenv
```

3.4.2.1 Network Settings

Set up the network configuration by editing the following lines in the configuration file:

```
eth0.ipaddr=###.###.###.###
eth0.netmask=###.###.###.###
eth0.serverip=###.###.###.###
```

3.4.2.2 NFS Root Directory

If intending to mount the root filesystem by NFS it is required to specify the path. The file, rootfs.tgz, should have been extracted to a location specific to the NFS server, such as */home/<user>/phyFLEX-i.MX6/NFS/*. In the configuration file, search for *nfsroot="$eth0.serverip:/path/to/root"* and modify to read the proper path:

```
nfsroot="$eth0.serverip:/home/<user>/phyFLEX-i.MX6/NFS/"
```

3.4.2.3 Image Files

The default location of the Linux kernel and root filesystem as well as the file type can be changed in the configuration file.

Kernel Image File

The Linux kernel image file is configurable in terms of the default location to load and execute the file, search for the *kernel_loc* environment variable and replace with one of the supported locations:

Supported Location	Configuration File
NOR	`kernel_loc = nor`
TFTP	`kernel_loc = tftp`

Root Filesystem Image

The root filesystem is configurable in terms of default location and the file type. The file type of the root filesystem is dependent on the location, see Section 3.2 for more information. Search for the *rootfs_loc* and *rootfs_type* environment variables and modify based on the following:

Location	File Location	File Type
NAND	`rootfs_loc = nand`	`rootfs_type = ubifs`
NFS	`rootfs_loc = net`	`rootfs_type = ext3`

3.4.3 Restore to Default

If the barebox environment variables need to be restored for any reason, simply delete the parameter save location in nor and the defaults will be restored with the next boot.

```
erase /dev/nor0.bareboxenv
```

3.4.4 Booting Options

From Barebox, the boot process continues by loading the kernel which then mounts the root filesystem. Both the kernel and root filesystem locations are determined from the Barebox environment. The locations classify booting Linux stand-alone or remote, where the components required are provided in the onboard media (NAND, SPI NOR, SD/MMC Card) or via a network (TFTP or NFS). For the phyFLEX-i.MX6 PD12.0.3 support for booting is provided:

File	Supported Boot Location
Barebox	NOR
Linux Kernel	NOR TFTP
Root Filesystem	NAND NFS

3.4.4.1 Boot Command

Selection of the location to continue booting is provided as a **boot** command in the current Barebox environment. This command allows the user to specify the mode over which it will boot with respect to the kernel, root filesystem, ip, and oftree options. The generic usage of this command is described by the following:

```
boot [-m <mode>] [-k <kernel_option>] [-r <rootfs_option>] [-i <ip_mode>] [-o <oftree_option>]
```

By typing _**boot_help** in the barebox prompt, a summary of the syntax, options, and parameters are provided.

3.4.4.2 NAND Booting

NAND Booting is not fully supported in this BSP release, however, by default the kit comes setup to load the root filesystem from NAND. The boot command can be used to specifically state NAND as the location of the filesystem, if this is the case be sure that the rootfs_type environment is set to*ubifs* as described in Section 3.2.4.3.

Description	Barebox
Boot kernel from default and mount rootfs from NAND Flash	`boot -r nand`

3.4.4.3 SPI NOR Booting

By default, the kit comes setup for a standalone boot from NOR Flash with the exception that the root filesystem is loaded from NAND. Therefore, without modification to environment variables simple type **boot** from Barebox. Alternatively, executing commands given by the following, tells the system to boot using the Linux kernel from NOR Flash, which is useful in the case where the boot mode on the Carrier Board is set in a configuration other than NOR boot:

Description	Barebox
Boot kernel from SPI NOR Flash and mount rootfs from default	`boot -k nor`

3.4.4.4 Remote Booting

For development it may be beneficial to modify the boot settings to allow the kernel to be loaded from TFTP, and/or mount a network filesystem hosted on the NFS, as specified in Sections 3.2.5 and 3.2.6, respectfully. Refer to Sections 3.4.2.1 and 3.4.2.2 for information on Configuration File settings as a part of setup for these options.

Examples of the wide variety of remote booting options the Barebox **boot** command supports is given by the following:

Description	Barebox
Boot kernel from TFTP and mount rootfs from default	`boot -k tftp`
Boot kernel from default and mount rootfs from NFS	`boot -r net`

| Boot kernel from TFTP and mount rootfs from NFS | `boot -k tftp -r net` |
| | `boot -m tftp` |

3.5 Booting the Target

After selecting the boot source, the target starts booting. When prompted, after the target has finished loading the system, the default login of **root** can be used:

3.5.1 Powering off the Target

Before removing power, it is recommended to execute the **poweroff** command to close out the filesystem, shut down the system, and power off. This is recommended to maintain system integrity by preserving bad blocks:

```
poweroff
```

The system can also be restarted:

```
reboot
```

4 Modifying the BSP

The BSP provided can be modified through the source code in the board files or through configuration management.

4.1 Board Files

All source code is located in the *BSP-Phytec-phyFLEX-i.MX6-PD12.0.3/platform-mx6/build-target* directory. To help integrate and modify features on the system for both driver development and general settings or Carrier Board design, it is necessary to know about the files summarized by the following:

- linux-3.0.35/arch/arm/mach-mx6/board-mx6q_phyflex.c
- barebox-2012.02.0/arch/arm/boards/phyflex-imx6/board.c
- barebox-2012.02.0/arch/arm/boards/phyflex-imx6/env/*
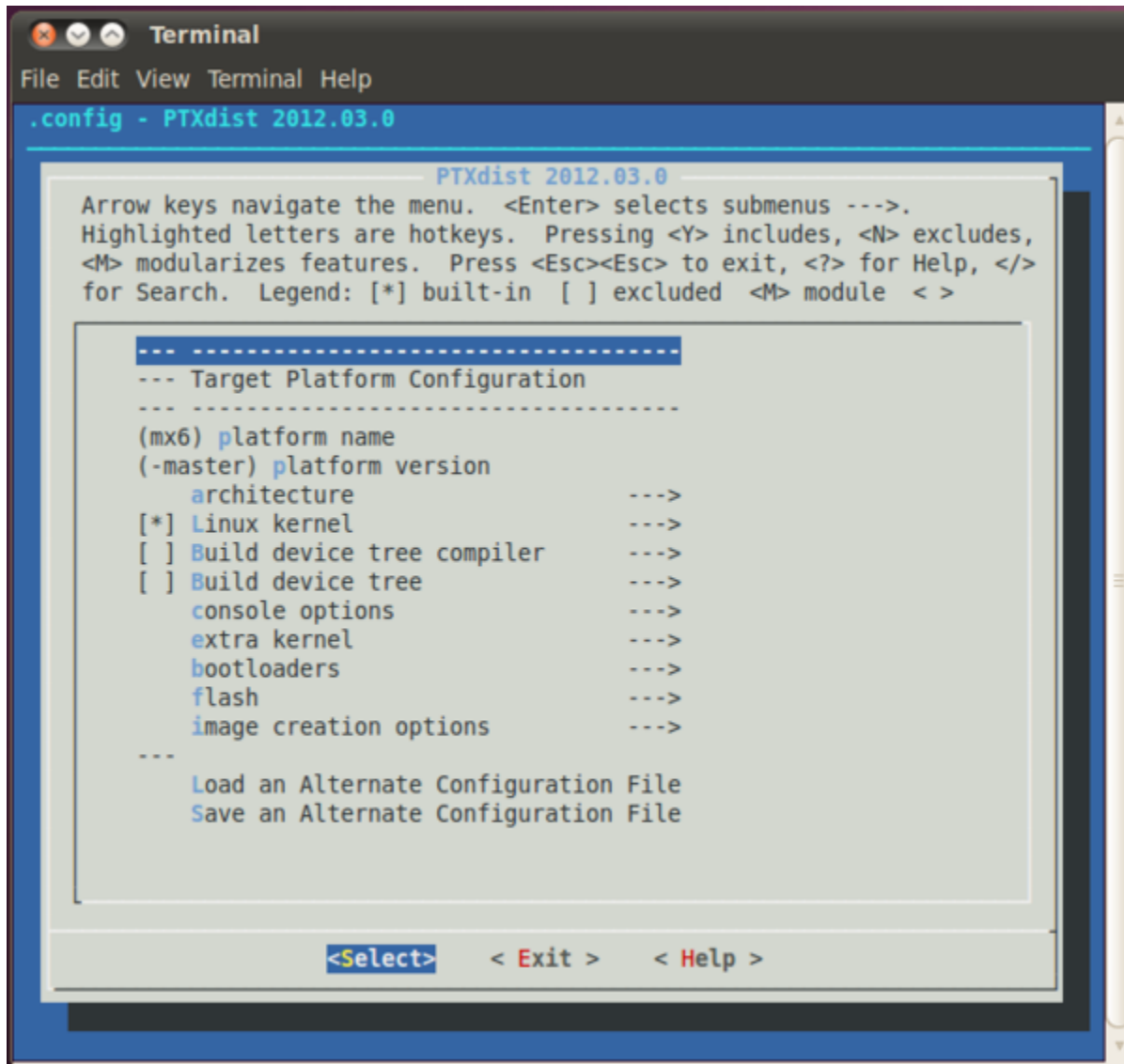
4.2 Managing Configurations

PTXdist uses the kernel configuration, KConfig, files present throughout the BSP for straightforward user configuration of individual settings and drivers. The platform, kernel, and project's root filesystem configuration menus will be the most beneficial.

The phyFLEX-iMX6 PD12.0.3 BSP uses PTXdist version 2012.03.0, see Section 2.2 for additional information on using PTXdist. If a symbolic link was not created between **ptxdist** and **ptxdist-2012.03.0** as described in Section 2.2.4, affix all **ptxdist** commands with *2012.03.0* resulting in **ptxdist-2012.03.0**.

4.2.1 Platform

The platform configuration menu contains the default settings for each platform including what bootloaders, kernel, and filesystem images are to be built and located. The settings for the platform are modified using the following command:
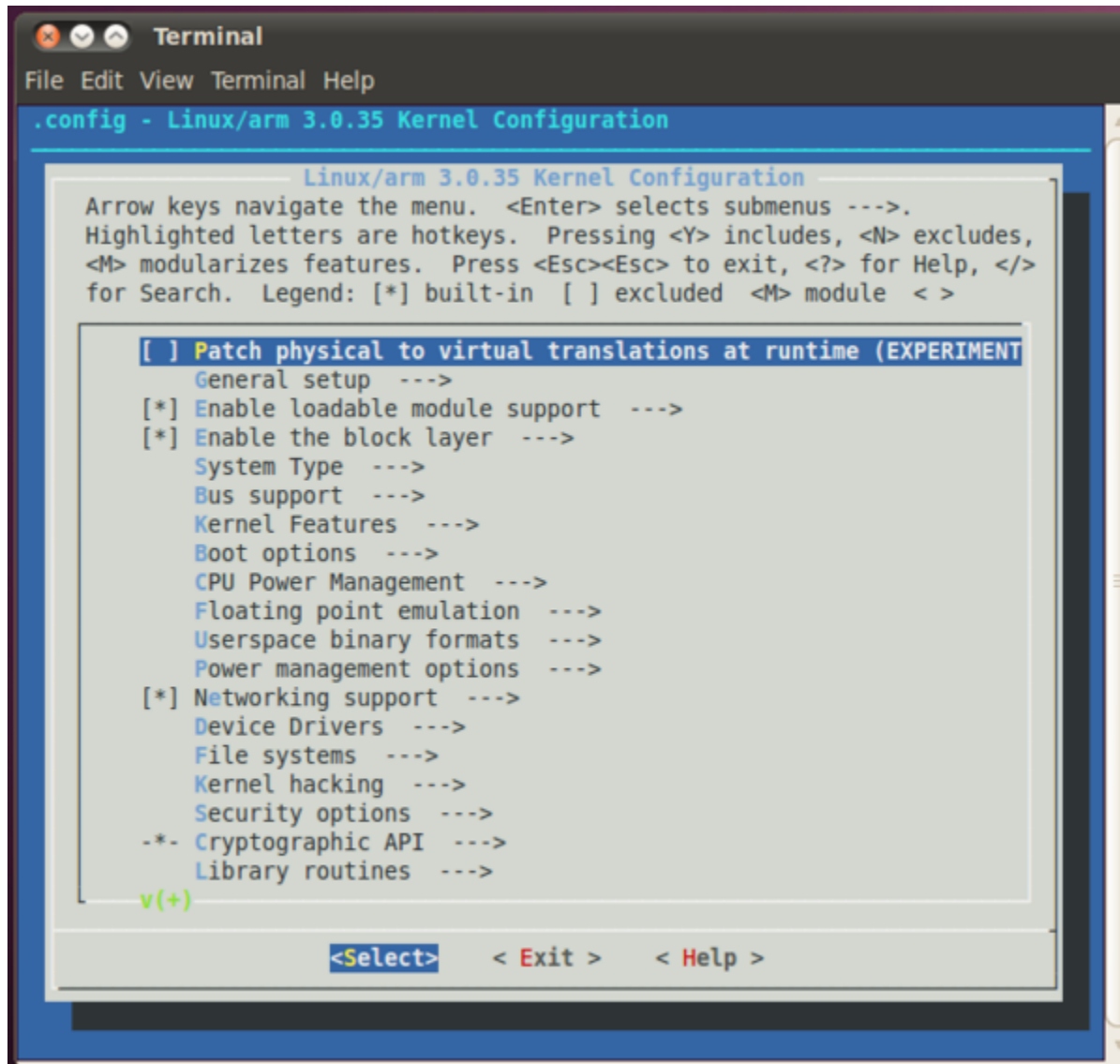
```
ptxdist platformconfig
```


4.2.2 Kernel

The kernel configuration menu allows the user to adjust the drivers and support included in a Linux kernel build. The settings are available by using the following command:
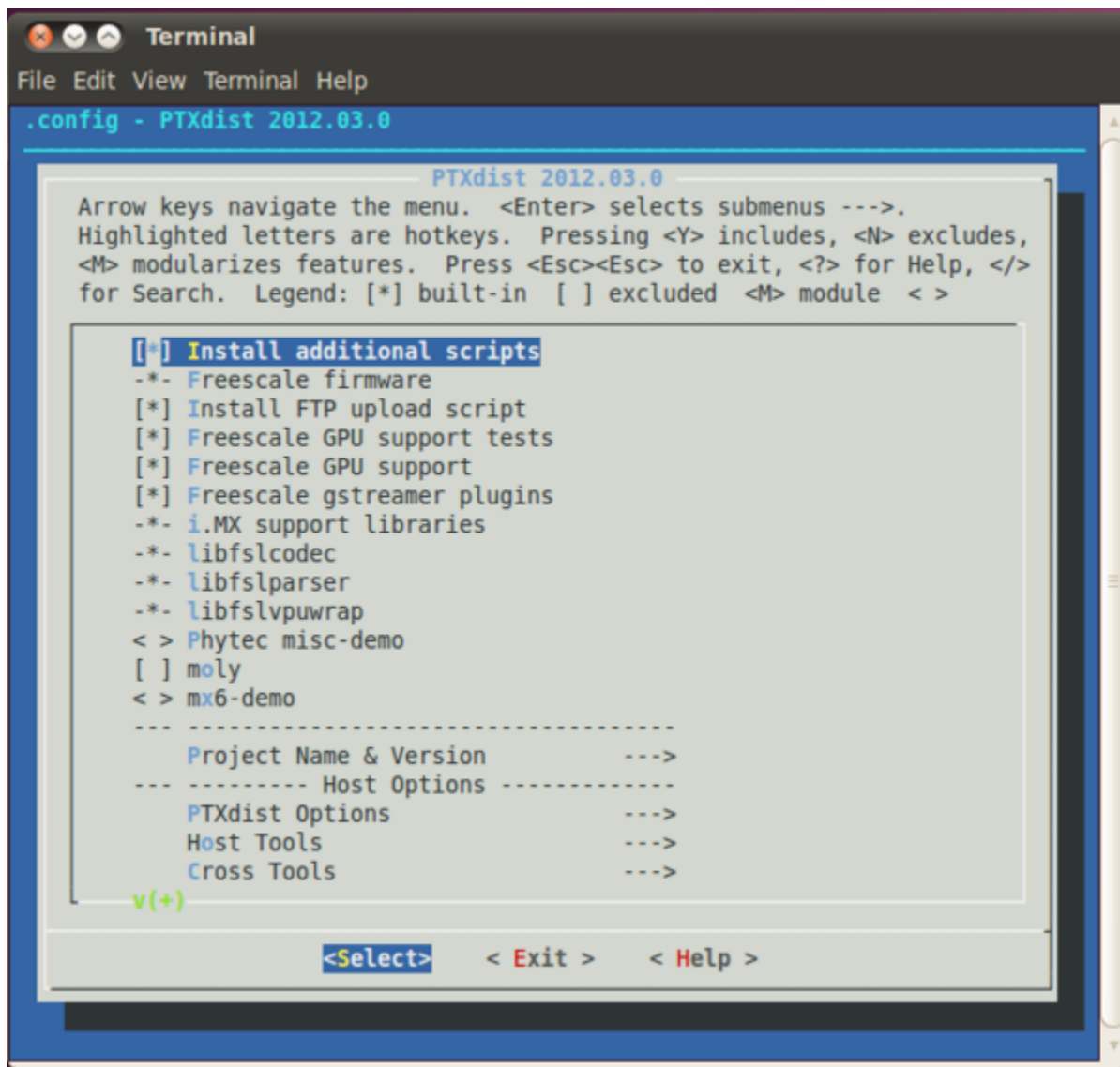
```
ptxdist kernelconfig
```


See Section 3.7.4 for information on configuring the Linux kernel to enable WiFi.

4.2.3 Root Filesystem

Configuration of the project's filesystem can be done in the overall menuconfig. By toggling the options in the base configuration, the applications and content built into the filesystem can be modified directly. This allows both minimal and more complete filesystem builds to be created easily. The following command will open the configuration menu:

```
ptxdist menuconfig
```

4.3 Building Images with PTXdist

Building and creating images with PTXdist is very simple, the phyFLEX PD12.0.3 BSP uses PTXdist version 2012.03.0 therefore affix all **ptxdist** command s with *2012.03.0* resulting in **ptxdist-2012.03.0** or see Section 2.2.4 for information on creating a symbolic link between **ptxdist** and **ptxdist-2012.03.0**.

All the required steps to compile sources and build packages in the correct order are done using the **ptxdist go** command. Following a successful build, the command, **ptxdist images**, will create images to deploy on the target.

Before starting a build, the userland and platform configuration must be specified:

```
ptxdist select configs/ptxconfig
ptxdist platform configs/platform-phytec-phycore-imx6/platformconfig
```

The toolchain should be automatically detected and read:

```
found and using toolchain:
'/opt/OSELAS.Toolchain-2011.11.1/arm-cortexa9-linux-gnueabi/gcc-4.6.2-glibc-2.14.1-binutils-2.21.1a-kernel-2.6.39-
sanitized/bin'
```

To compile and build the BSP, execute the **go** command from the BSP project directory:

```
ptxdist go
```

The build process relies on obtaining source information from the web, therefore in some cases the content to be downloaded may no longer exist and **ptxd ist go** will fail. If this occurs, use a search engine to find an alternative download link for the file as it appears in the build output, download the file and place it in BSP-Phytec-phyFLEX-i.MX6-PD12.0.3/src directory in its compressed form.

A root filesystem image can be created from the built content, by executing the **images** command in the BSP project directory:

```
ptxdist images
```

All images are stored in BSP-Phytec-phyFLEX-i.MX6-PD12.0.3/platform-mx6/images, the following can be expected:

barebox-default-environmentbarebox-imagehd.imglinuximage (rename to uImage-pfla02)root.ext2root.tgzroot.ubifsroot.ubi

5 Flashing Images

5.1 Update Command (Remote Flashing)

To update the bootloader, kernel, or filesystem in flash, Barebox provides the **update** command. This command allows the user to specify the file to be flashed, the mode over which it will be flashed, and the path to the image. The generic usage of this command is described by the following:

```
update -t <kernel|rootfs|barebox|bareboxenv|xload|oftree> -d <nor|nand> [-m tftp|xmodem] [-f imagename] -c
```

By typing **_update_help** or **update** in the barebox prompt, a summary of the syntax, options, and parameters are provided.

5.1.1 TFTP

TFTP is one of the simplest ways to apply modifications during development; it is commonly used for single file updates such as the bootloader, kernel, and files from the root filesystem.

The **update** command in Barebox can be used to flash images to NOR flash from the TFTP Server in the correct format and located, for example, at *<TFTP_DIRECTORY>/phyFLEX-iMX6/images* as specified in . The phyFLEX-i.MX6 PD12.0.3 release supports updating Barebox, Barebox default environment, and the Linux Kernel to NOR:

File	Flash to	Barebox Command
Barebox	NOR	`update -t barebox -d nor -m tftp -f phyFLEX-iMX6/images/barebox-image`
Barebox Environment	NOR	`update -t bareboxenv -d nor -m tftp -f phyFLEX-iMX6/images/default-barebox-environment`
Kernel	NOR	`update -t kernel -d nor -m tftp -f phyFLEX-iMX6/images/uImage-pfla02`