Using Skywire® Modems with the Digi ConnectCore 6™ Development Board

NimbeLink Corp.

Revised: July 2015

Table of Contents

1. Introduction

1.1. Overview

This document describes the process of using Skywire® 4G LTE modems with Digi's ConnectCore 6™ board. The easiest way to do this is to use PPPd, and either NimbeLink's Skywire mPCIe Adapter Board (NL-AB-MPCIE) or Skywire Development Kit (NL-SWDK).

This example was written using a Digi ConnectCore 6™ board and a NL-SW-LTE-TSVG (Verizon) Skywire modem. Both the NimbeLink mPCIe adapter and the Skywire Development Kit have been verified working with this example.

2. Installation and Setup

2.1 Downloading the SD Image

To begin, download the ".sdcard" images from the NimbeLink website. You have two choices: `dey-image-minimal-nl.sdcard`, and `dey-image-graphical-nl.sdcard`. The `minimal` image is a bare-bones installation with a small footprint that you need to access via SSH or serial cable, while `graphical` installs with the GNOME Sato GUI interface.

2.2 Writing the SD Image

Once you have the image downloaded, write the image to a micro-SD card at least 1 GB in size. For Windows, you can use Win32 Disk Imager (http://sourceforge.net/projects/win32diskimager/). For Mac OS X and Linux, unmount (but don't eject) the SD card and you can use the program `dd` via the Terminal:

Mac:
```
sudo dd bs=1m if=/path/to/sdcard/image of=/path/to/sd/card
```
Linux:
```
sudo dd bs=1M if=/path/to/sdcard/image of=/path/to/sd/card
```
OR
```
sudo dd bs=1m if=/path/to/sdcard/image of=/path/to/sd/card
```

2.3 Set Up SD Card Booting

Once the SD image has finished copying to the SD card, follow this guide based on your ConnectCore 6 board to set it up to boot from an SD card:

ftp://ftp1.digi.com/support/documentation/html/90001421/90001421_C/Files/step5.html

Insert the SD card, apply power, and depending on how you'd like to communicate with it, plug in the serial cable or HDMI cable and USB keyboard/mouse.

Note: PPP requires bringing down the Ethernet and WiFi interface on the ConnectCore board, so it is recommended not to connect to the board with SSH.

2.4 Set Up PPP

Once you are communicating with the board, please consult the NimbeLink guide for PPP with LTE on Linux:

https://www.nimbelink.com/wp-content/uploads/2015/04/Skywire-PPPd-for-LTE-Application-Note.pdf

Begin the process at Section 2.5 while following the instructions for “Ubuntu”.

3. Advanced: Building the Images

3.1. Overview

If needed, you can create your own Linux image using the Yocto Project’s tools. The Yocto Project is a Linux group that created tools to more easily write Linux operating systems for embedded devices.

3.2. Prerequisites

Compiling and building an image using Yocto requires a Linux workstation and time. Please consult this document for specifics:

http://www.yoctoproject.org/docs/current/yocto-project-qs/yocto-project-qs.html#the-linux-distro

Yocto recommends the following Linux distributions for development:

- Ubuntu
- Fedora
- openSUSE
- CentOS
- Debian

As for operating system versions, a general guideline Yocto recommends is using the “current release minus one” to be safe. It is important to note, however, that the images on the NimbeLink website were built on the latest version of Debian as of this writing (Debian 8.1 x64 “Jessie”).

For a rough estimate for the time it takes to do the initial build, a workstation with the following specifications:

- Debian 8.1 x64 “Jessie”
- Intel Core i5 quad-core processor running at 3.7 GHz
- 16 GB of RAM

- SATA 6 SSD

took about three hours to complete using all four cores.

Finally, Digi's "First Steps Guide" on Digi Embedded Yocto 1.6:

http://ftp1.digi.com/support/documentation/90001423_D.pdf

will serve as the basis of this example. As such, this guide will cover the additions we made to Digi's files to get PPP loaded and working.

3.3. Setup and Installation

Starting with the "Setting up your workstation" section of Digi's "First Steps Guide", complete each section up to but not including "Building images". Below are a few notes on the procedure we discovered while going through the process ourselves.

"Installing Digi Embedded Yocto"

In the "Installing Digi Embedded Yocto" section, the third command you enter is:

```
cd /usr/local/dey-1.6
```

Before that, type the following lines:

```
mkdir /usr/local/dey-1.6
chmod a+x /usr/local/dey-1.6
```

From this point, continue with the third command.

"Updating existing projects"

In the "Updating existing projects" section, Digi recommends deleting the **`tmp`** and **`sstate-cache`** folders whenever you rebuild the project. Doing so results in needing to build the image from scratch again; this can take a long time. However, leaving the folders will significantly reduce the time to build the image. Unless you are making major changes from build-to-build, it may be beneficial to leave the folders to build in minor changes from build-to-build.

3.4. Modifying the Kernel Recipe

Before you build the image, you need to add modules to the kernel recipe. Navigate to the following folder:

```
cd
/usr/local/dey-1.6/sources/meta-digi/meta-digi-arm/rec
ipes-kernel/linux/linux-dey-3.10/ccimx6sbc
```

which contains the file `defconfig`. This file contains the modules that will be added to the kernel during the build. Add the following lines to this file:

```
CONFIG_WLAN=y
CONFIG_INPUT_MOUSE=y
CONFIG_USB_G_SERIAL=y
CONFIG_USB_SERIAL=y
CONFIG_USB_SERIAL_OPTION=y
CONFIG_USB_NET_QMI_WWAN=y
CONFIG_PPP=y
CONFIG_PPPOATM=y
CONFIG_PPPOE=y
CONFIG_PPPOL2TP=y
CONFIG_PPP_ASYNC=y
CONFIG_PPP_BSDCOMP=y
CONFIG_PPP_DEFLATE=y
CONFIG_PPP_FILTER=y
CONFIG_PPP_MPPE=y
CONFIG_PPP_MULTILINK=y
CONFIG_PPP_SYNC_TTY=y
CONFIG_PPTP=y
CONFIG_USB_CATC=y
CONFIG_USB_CDC_PHONET=y
CONFIG_USB_HSO=y
CONFIG_USB_IPHETH=y
CONFIG_USB_KAWETH=y
CONFIG_USB_PEGASUS=y
CONFIG_USB_RTL8150=y
CONFIG_USB_USBNET=y
CONFIG_USB_ZD1201=y
CONFIG_PCI=y
```

and remove the following lines from the file:

```
CONFIG_USB_G_SERIAL=m
# CONFIG_WLAN is not set
# CONFIG_INPUT_MOUSE is not set
```

Once you make these changes, you can proceed with the next step to build the images.

3.5. Locating the Images

After the build completes, the folder:

```
/$HOME/workspace/ccimx6sbc/tmp/deploy/images/ccimx6sbc
```

will contain the build images for installation, including the SD card images.