Pronghorn Metro SBC Software Manual

Quad Radio Wireless Router Board

ADI Engineering, Inc.
1769A Worth Park
Charlottesville, VA 22911

http://www.adiengineering.com
Phone: (434) 978-2888
Fax: (434) 978-1803

LIMITED WARRANTY
ADI Engineering guarantees that every Pronghorn Metro SBC unit will be free from physical defects in material and workmanship for one year from the date of purchase, when used within the limits set forth in the Specifications section of this User Guide. If the product proves defective during this warranty period, call ADI Engineering Technical Support in order to obtain a Return Authorization number. When returning a product, mark the Return Authorization number clearly on the outside of the package and include a copy of your original proof of purchase. All customers shall be held responsible for return shipping and handling charges.

IN NO EVENT SHALL ADI ENGINEERING'S LIABILITY EXCEED THE PRICE PAID FOR THE PRODUCT FROM DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OF THE PRODUCT, ITS ACCOMPANYING SOFTWARE, OR ITS DOCUMENTATION. ADI ENGINEERING OFFERS NO REFUNDS FOR ITS PRODUCTS. ADI Engineering makes no warranty or representation, expressed, implied, or statutory, with respect to its products or the contents or use of this documentation and all accompanying software, and specifically disclaims its quality, performance, merchantability, or fitness for any particular purpose. ADI Engineering reserves the right to revise or update its products, software, or documentation without obligation to notify any individual or entity. Please direct all inquiries to:

ADI Engineering, 1769A Worth Park, Charlottesville, VA 22911.

SAFETY AND REGULATORY NOTICES

FCC STATEMENT
Pronghorn Metro SBC has been tested and found to comply with the specifications for a Class B digital device, pursuant to Part 15 of the FCC Rules. Operation is subject to the following two conditions:
(1) This device may not cause harmful interference, and
(2) This device must accept any interference received, including interference that may cause undesired operation.

These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used according to the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which is found by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment or devices
- Connect the equipment to an outlet other than the receiver's
- Consult a dealer or an experienced radio/TV technician for assistance

FCC Caution: Any change or modification to the product not expressly approved by ADI Engineering could void the user's authority to operate the device.

Revisions

Date	Revision	Page-paragraph	Remarks
6 Jul 07	1.00	All	Initial Document
9 Apr 08	1.10	20	Updated support link

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This document provides an overview of the different software components available with the Pronghorn Metro SBC as well as technical information regarding the use of any ADI-supplied software provided for the board.

2 What is Shipped?

Please refer to the Pronghorn Metro SBC Hardware Manual for a complete list of the physical components shipped with or available for the Pronghorn Metro SBC.

Boards are typically shipped from ADI with only the RedBoot boot monitor installed. RedBoot is the first software executed on the board and is responsible for configuration of the SDRAM, chip selects, GPIO pins, and peripheral components on the board. After RedBoot executes, a "RedBoot>" prompt is displayed on the serial console.

Boards shipped from a distributor or other 3rd party may have only RedBoot installed as described above or may have software from a 3rd party software vendor pre-installed. Always be sure to purchase the board with the software you intend to use or at least with a means to load it. This is because it is not always possible to easily replace 3rd party software with RedBoot or another software solution. It may require an expensive JTAG device to do it.

With RedBoot running on the board, the end user may install his own software or software from one of the many software vendors by executing a sequence of RedBoot commands. See a later section of this manual for more information about RedBoot commands.

3 How to Power up a Pronghorn Metro SBC

The following instructions are included as a Quick Start Guide with each board. They are reproduced here for easy reference.

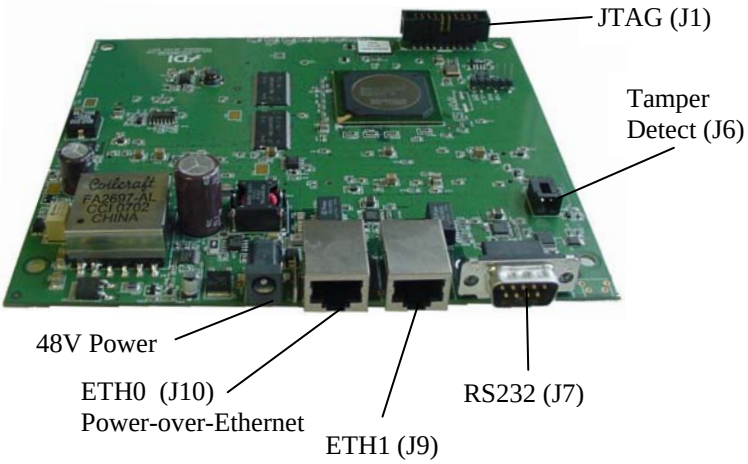

Figure 1 - Pronghorn Metro SBC Top Side

1. **Connect a serial null modem cable from the Host Computer to the DB9 connector at RS232 port (J7)**
2. **Configure the Host Computer terminal emulator for 115200-N-8-1-N**
3. **Connect power and Ethernet in <u>one</u> of the following configurations:**
   - **Line power, or**
     - Connect an Ethernet cable from a DHCP-enabled LAN to the RJ45 port at ETH0 (J10)
     - Connect a 48V/.620A DC adapter to the barrel connector at 48V Power (J12)
   - **Power-over-Ethernet injector**
     - Connect an Ethernet cable from a DHCP-enabled LAN to the Data port of a Power-over-Ethernet injector
     - Connect an Ethernet cable from the Data & Power port of a Power-over-Ethernet injector to the RJ45 port at ETH0 (J10)
4. **A "RedBoot>" prompt will appear on the terminal emulator**

LED	Description
STATUS	RedBoot turns this LED ON shortly after power up – indicates power present and software operational
ETH0 (WAN)	ON indicates Link on ETH0 (WAN), Blinking indicates activity
ETH1 (LAN)	ON indicates Link on ETH1 (LAN), Blinking indicates activity
PCI0	Depends on mPCI card in PCI0 slot – usually indicates card active
PCI1	Depends on mPCI card in PCI1 slot – usually indicates card active
PCI2	Depends on mPCI card in PCI2 slot – usually indicates card active
PCI3	Depends on mPCI card in PCI3 slot – usually indicates card active

Table 1- LED Descriptions

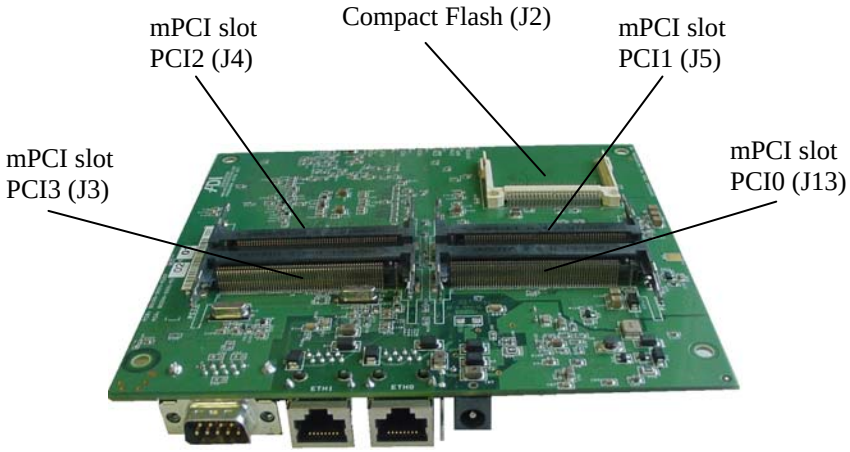

Figure 2 - Pronghorn Metro SBC Bottom Side

3.1 Power-Up Process

When the Pronghorn Metro SBC is first powered up, RedBoot starts, attempts to DHCP an IP address, and the following banner is displayed if successful:. Note the message "No devices on IDE controller 0" may appear if a compact flash device is not present in the J2 socket located on the back side of the board. A compact flash card is an option and is not required to run software on the board.

```
+No devices on IDE controller 0

Trying NPE-B...success. Using NPE-B with PHY 0.
... waiting for BOOTP information
Ethernet eth0: MAC address 00:08:a2:02:a6:c2
IP: 192.0.0.35/255.255.255.0, Gateway: 192.0.0.1
Default server: 192.0.0.1

RedBoot(tm) bootstrap and debug environment [ROM]
Red Hat certified release, version 1.30 - built 16:50:03, Apr 25
2007

Platform: ADI Engineering Pronghorn Metro (IXP42X 533MHz) BE
Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.

RAM: 0x00000000-0x04000000, [0x0002a0b8-0x01fc1000] available
FLASH: 0x50000000 - 0x51000000, 128 blocks of 0x00020000 bytes
each.
RedBoot>
```

Note the IP address that is displayed. If the board is connected to a DHCP-enabled LAN and an IP address is successfully acquired, it will be displayed here. If a DHCP-enabled LAN is not available it is possible to statically assign an IP address to the board. See the RedBoot **fconfig** or **ip_address** commands for more details.

3.2 RedBoot Commands

This section provides a brief summary of RedBoot commands and is intended only as a quick reference. For a thorough overview of RedBoot functionality please refer to the RedBoot User's Guide available on the ADI FTP site referenced in the Appendix.

RedBoot provides three basic classes of commands:

- Program loading and execution
- Flash image and configuration management
- Miscellaneous commands

The basic format for a RedBoot command is:

```
RedBoot> COMMAND [-S] [-s val] operand
```

Commands may require additional information beyond the basic command name. In most cases this additional information is optional, with suitable default values provided if they are not present. The type of information required affects how it is specified.

1. **[-S]** indicates an optional switch. If this switch is present, then some particular action will take place. For example in the command

   ```
   RedBoot> fis init –f
   ```

   the -f switch indicates to perform a full file system initialization.

2. **[-s val]** indicates an optional switch that requires an associated value. For example the command:

   ```
   RedBoot> load -b 0x20000 data_file
   ```

   specifies downloading a file (via TFTP) into memory, relocating it to location 0x20000.

3. **operand** This format is used in a case where a command has one operand which must always be present (no -s is required since it is always implied). For example the command

   ```
   RedBoot> go 0x20000
   ```

   specifies executing the code starting at location 0x20000.

The list of available commands, and their syntax, can be obtained by typing help at the command line:

```
RedBoot> help
Manage aliases kept in FLASH memory
   alias name [value]
Set/Query the system console baud rate
   baudrate [-b <rate>]
Manage machine caches
   cache [ON | OFF]
Display/switch console channel
   channel [-1|<channel number>]
Compute a 32bit checksum [POSIX algorithm] for a range of memory
   cksum -b <location> -l <length>
Display disks/partitions.
   disks
Display (hex dump) a range of memory
```

```
   dump -b <location> [-l <length>] [-s] [-1|2|4]
Execute an image - with MMU off
   exec [-w timeout] [-b <load addr> [-l <length>]]
        [-r <ramdisk addr> [-s <ramdisk length>]]
        [-c "kernel command line"] [<entry_point>]
Manage FLASH images
   fis {cmds}
Manage configuration kept in FLASH memory
   fconfig [-i] [-l] [-n] [-f] [-d] | [-d] nickname [value]
Execute code at a location
   go [-w <timeout>] [-c] [-n] [entry]
Help about help?
   help [<topic>]
Display command history
   history
Set/change IP addresses
   ip_address [-l <local_ip_address>[/<mask_len>]] [-h
<server_address>]
Load a file
   load [-r] [-v] [-d] [-h <host>] [-p <TCP port>][-m <varies>]
[-c <channel_num
ber>]
        [-b <base_address>] <file_name>
Dump information on PCI devices
   lspci
Compare two blocks of memory
   mcmp -s <location> -d <location> -l <length> [-1|-2|-4]
Copy memory from one address to another
   mcopy -s <location> -d <location> -l <length> [-1|-2|-4]
Compute a 128bit checksum [RSA Data Security, Inc. MD5 Message-
Digest Algorithm]
 for a range of memory.
   md5sum -b <location> -l <length>
Fill a block of memory with a pattern
   mfill -b <location> -l <length> -p <pattern> [-1|-2|-4]
Network connectivity test
   ping [-v] [-n <count>] [-l <length>] [-t <timeout>] [-r
<rate>]
        [-i <IP_addr>] -h <IP_addr>
Print information about the current settings within the main
processor
   processor
Reset the system
   reset
Swap bytes in 16-bit or 32-bit words in a block of memory
   swab -b <location> -l <length> [-2|-4]
Display RedBoot version information
   version
Display (hex dump) a range of memory
   x -b <location> [-l <length>] [-s] [-1|2|4]
RedBoot>
```

Commands can be abbreviated to their shortest unique string. Thus in the list above, **d,du,dum** and **dump** are all valid for the **dump** command. The **fconfig** command can be abbreviated **fc**, but **f** would be ambiguous with **fis**.

Also additional help can be found for commands by typing **help** before the command. A good example is the **fis** command which has many subcommands to manage the creation of partitions in the onboard flash device. See the section below on the RedBoot Flash Directory.

3.3 RedBoot Configuration

Use the command "**`fconfig –i"`** to initially set the configuration parameters. Use "**`fconfig –l"`** to list the configuration parameters, example below:

```
RedBoot> fconfig -l
Run script at boot: false
Use BOOTP for network configuration: true
Default server IP address: 0.0.0.0
Console baud rate: 115200
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
Default network device: npe_eth0
Network hardware address [MAC] for NPE eth0:
0x00:0x08:0xA2:0x02:0x0B:0x80
Network hardware address [MAC] for NPE eth1:
0x00:0x08:0xA2:0x02:0x0B:0x81
RedBoot>
```

The above represents the default RedBoot parameter settings for Pronghorn Metro. Note that it is setup to DHCP an IP address from the network using the NPE0 Ethernet port. This is the port at location J10 and is labeled ETH0. It is also the port that supports a Power-over-Ethernet connection.

Should you desire to change any of the settings, entering the "**`fconfig`**" command without any switches will enable you to change parameters.

```
RedBoot> fconfig
```

RedBoot will display the first parameter and its value. If you are satisfied with the current value, hit **`[enter]`**. Otherwise, type the desired value and then hit **`[enter]`**. When you have gone through the entire parameter list, a final confirmation is required before the modified configuration is written to flash. Upon completion, issue the "**`fconfig –l"`** command again to verify that your changes have been made. Then reboot the Pronghorn Metro to have the new boot parameters take effect.

3.4 RedBoot Flash Directory

The RedBoot Flash Directory system allows a user to partition the flash device and erase or program flash partitions. Usually this is used to install an operating system or application into the onboard flash device. Below is the content of the onboard flash as shipped from ADI.

Flash address	Content
0x50000000 … 0x5005FFFF	RedBoot boot monitor (384KB)
0x50060000 … 0x50FDFFFF	Free space (15.5MB)
0x50FE0000	RedBoot flash directory and configuration (128KB)

… 0x50FFFFFF	

Table 2 - Onboard Flash Memory Map

NOTE: For future reference, as long as RedBoot is operational, it is possible to restore the flash to the above condition at any time with the "**fis init –f**" command.

Loading and running an operating system or application depends on the provider of the software. Some software providers require RedBoot as the boot monitor while others replace the entire content of the onboard flash device. Some software providers require their software reside in the onboard flash while others require it reside in the optional Compact Flash card.

If the onboard flash must be programmed, it is important to know that the flash device on the Pronghorn Metro board always powers up with all sectors in the flash in the locked state. Therefore to program any sectors it is necessary to first unlock the flash. This is done with the following command. Note that only the Free Space shown in the above table is unlocked, thereby continuing to protect the RedBoot, flash directory, and configuration sectors.

RedBoot>**fis unlock –f 0x50060000 –l 0xF80000**

See the section below on using ADI's Linux operating system for an example of how to use the Flash Directory commands to store Linux images into the flash and automatically load and execute.

The RedBoot fis command provides all of the functions for the Flash Directory system. See the list below.

```
RedBoot> help fis
Manage FLASH images
  fis {cmds}
Create an image
  fis create -b <mem_base> -l <image_length> [-s <data_length>]
  [-f <flash_addr>] [-e <entry_point>] [-r <ram_addr>]
  [-n] <name>
Delete an image from FLASH Image System [FIS]
  fis delete name
Erase FLASH contents
  fis erase -f <flash_addr> -l <length>
Display free [available] locations within FLASH Image System
  fis free
Initialize FLASH Image System [FIS]
  fis init [-f]
Display contents of FLASH Image System [FIS]
  fis list [-c] [-d]
Load image from FLASH Image System [FIS] into RAM
  fis load [-d] [-b <memory_load_address>] [-c] name
LOCK FLASH contents
  fis lock [-f <flash_addr> -l <length>] [name]
LOCK DOWN FLASH contents
  fis lockdown [-f <flash_addr> -l <length>] [name]
UNLOCK FLASH contents
  fis unlock [-f <flash_addr> -l <length>] [name]
Write raw data directly to FLASH
  fis write -f <flash_addr> -b <mem_base> -l <image_length>
RedBoot>
```

3.5 Booting From a Compact Flash Card

RedBoot supports ADI's own Application-in-a-Flash (AiaF) feature – where customers can easily load and run applications stored on Compact Flash.

To use the AiaF feature, do the following:

1. Format a Compact Flash card with a Linux ext2 file system. Do this as follows:
    a. Insert a Compact Flash card into a Linux desktop PC. Determine what device the card has been mapped to. For this example, we use **/dev/sda**. **CAUTION**: It is absolutely necessary to determine the correct device name. If the wrong name is chosen, the following commands could result in reformatting the hard drive of the PC!
    b. Format the card with the ext2 file system
        i. Partition the disk using the command **fdisk /dev/sda**
        ii. In fdisk, use "**p**" to list any existing partitions
        iii. In fdisk, use "**d**" to delete all existing partitions
        iv. In fdisk, use "**n**" to create a new primary partition over the entire Compact flash card
        v. In fdisk, use "**w**" to perform all the above operations and exit the fdisk utility
        vi. Format the new partition using the command **mkfs.ext2 –m 0 /dev/sda1**

2. Mount the Compact Flash card and copy all the files needed to bring up the operating system or application to the Compact Flash card.
3. Create the RedBoot boot script and give it the filename, AIAFEXEC.RBS. Copy it to the Compact Flash card.
4. Unmount the Compact Flash card and insert it into slot J2 on the backside of the Pronghorn Metro board.
5. Power up the Pronghorn Metro board. RedBoot will detect the presence of the Compact Flash card, locate the AIAFEXEC.RBS file, load it and execute the commands it contains.

A sample AIAFEXEC.RBS file, an ASCII file, is shown below. The example loads a Linux ramdisk and kernel into memory and passes control to the kernel.

```
fis unlock -f 0x50060000 -l 0xF80000
load -r -v -b 0x800000 -m disk hda1:ramdisk.gz
load -r -v -b 0x1600000 -m disk hda1:zimage
exec
```

Table 3 - Sample AIAFEXEC.RBS file

3.6 MAC Addresses

Each Pronghorn Metro SBC is assigned a globally unique MAC address at the factory. This MAC address appears on a bar code label attached to the board.

The bar code label is scanned during manufacture and the MAC address is permanently stored in the one-time-programmable (OTP) section of the onboard flash device. The MAC addresses are assigned such that each Pronghorn Metro board can derive two MAC addresses from the one programmed, i.e., MAC and MAC+1. RedBoot and the Linux Intel NPE Ethernet driver have been modified to read the OTP, derive the second MAC address, and apply them to the appropriate Ethernet ports, first ETH0, then ETH1. The

RedBoot "**fconfig –l"** command can be used to view the MAC addresses. From Linux, the "**ifconfig –a**" command can be used.

The MAC addresses are allotted from the pool of addresses assigned to ADI Engineering by the IEEE and always begin with 0x0008A2, our organizationally-unique-identifier (OUI).

4 Operating Systems, Wireless Router Applications, etc

The RedBoot boot monitor by itself does not provide any operating system, wireless router application, or anything of the like. It merely initializes the hardware. But, in addition to doing that, it is used to install an operating system or application into the onboard flash. It can then be used to automatically load and run the operating system or initiate a boot from the Compact Flash card.

The following sections briefly describe some of the various software solutions available for the Pronghorn Metro. This is not a compete list. There may be others not known at the time of this printing.

Each software solution comes with its own requirements regarding onboard flash usage, compact flash usage, radio card requirements, etc. Some solutions are available free-of-charge, others must be purchased and/or licensed. Some must be loaded and run from the onboard flash and some must be loaded and run from Compact Flash. Some provide evaluation versions, some do not. For installation, operation, and support, please contact the particular software vendor.

4.1 ADI Linux

ADI provides a free Linux Board Support Package (BSP) for Pronghorn Metro SBC. This Linux BSP is based off an open-source Linux distribution from Snapgear, http://www.snapgear.org. The distribution includes a large variety of the typical utilities needed in an embedded environment, such as DHCP servers and clients, web servers, NFS, tftp, routing tables, wireless drivers, etc.

ADI's BSP adds a unique build target for the Pronghorn Metro and makes the necessary changes to the stock distribution to support all the interfaces on the Pronghorn Metro board. Also, additional utilities and/or drivers may be added to the distribution as needs surface.

It should be noted that ADI's BSP does not provide a turnkey wireless router implementation or any other application for that matter. Instead it provides a command-line only, base kernel distribution that may be used as a starting point to a proprietary wireless router implementation or other application. It could also be used as a source for code fragments specific to the Pronghorn Metro board to be ported to other distributions.

4.1.1 How to Get

The ADI Linux BSP is freely available for download from the ADIwebpage referenced in the Appendix. Available on the site are the following items:

- Full ADI/Snapgear Linux Distribution
  - 2.6 kernel
  - Intel Access Library already integrated
  - NPE Ethernet ports
  - PCI support for up to 4 miniPCI cards
  - MTD driver for on board flash device
  - I2C driver for Temperature/Voltage sensor
  - IDE driver for Compact Flash device
- Pre-built Linux kernel and ramdisk binaries

- Linux HowTo document

4.1.2 Makefile and TFTP Server

The vendors/ADI/PronghornMetro/Makefile contains a line that automatically copies the generated zImage and ramdisk.gz files to the /tftpboot/pmetro directory. This facilitates the subsequent downloading to the Metro board. Feel free to change or delete as appropriate to fit your particular setup.

4.1.3 Toolchain Path Setup

```
Adjust CROSS_COMPILE to reflect the tool location.

CROSS_COMPILE is defined in:

    snapgear-3.4.0/vendors/config/arm/config.arch

If the tools are in your path then CROSS_COMPILE can be set
to the tool prefix. This is typically arm-linux-.

    CROSS_COMPILE = arm-linux-

If the tools are not in your path then you must set
CROSS_COMPILE to the complete path and prefix.

    CROSS_COMPILE = <path>/arm-linux-
```

4.1.4 Compact Flash Access

To access the Compact Flash card from Linux, enter a command like the one below to mount the device to a mount point. It can be accessed as a normal file system from that mount point.

```
mount /dev/hda1 /mnt
```

4.1.5 Temperature/Voltage Sensor and the I2C Driver

The Temperature/Voltage sensor can be accessed with an included user program. Enter the following command to execute the program which reads and displays the current temperature and voltage. The source files for this program can be found in user/i2ctest/i2ctest.c.

```
# /home/i2c

MAX6652 Temperature/Voltage Sensor Display
/dev/i2c-0 opened successfully
Device Address (0x14) set successfully
Configuration Register 0x40 = 0x1 is successful
1.3  Volt Register 0x20 = 1.300 V
12.0 Volt Register 0x21 = 0.000 V
3.3  Volt Register 0x22 = 3.337 V
5.0  Volt Register 0x23 = 4.940 V
Temperature Register 0x27 = 34 degrees C
```

4.1.6 Onboard Flash and the MTD Driver

The onboard flash device can be mounted and used with the standard Linux MTD driver. Enter the following commands to examine the content of the flash device and to mount and use a JFFS2 flash image.

```
# cat /proc/mtd

dev:    size   erasesize  name
mtd0: 00060000 00020000 "RedBoot"
mtd1: 00300000 00020000 "jffs_01.img"
mtd2: 0001f000 00020000 "FIS directory"
mtd3: 00001000 00020000 "RedBoot config"
```

An example of mounting a JFFS2 image. (JFFS2 image not provided)

```
mount -t jffs2 /dev/mtdblock1 /mnt
```

4.1.7 Mounting a Remote File System Using NFS

This section gives an example on mounting a remote file system using NFS. This is useful during development of an application to run on Metro. Simply build your application on your host device, and execute from Metro by referencing your host file system.

Example configuration parameters: (your particular configuration may vary)

```
Host IP                              - 192.168.0.10
Target IP                            - 192.168.0.102
Target hostname                      - ADI-Pronghorn-Metro
Directory on host to mount on target - /tftpboot/pmetro/tmp
Directory on target to mount onto    - /mnt
```

4.1.7.1 Setting up the Host device to export a file sysystem

```
Add target hostname to /etc/hosts file.

Example /etc/hosts entry:
    192.168.0.102    ADI-Pronghorn-Metro

Add directory to be NFS mounted to /etc/exports and start/restart
nfsd.

Example /etc/exports entry:
    /tftpboot/pmetro/tmp 192.168.0.102(rw,sync,no_root_squash)
```

4.1.7.2 Setting up the target device to mount the file system using NFS

```
mount -t nfs <host-ip>:<host directory to mount>

Example corresponding to /etc/exports entry above:

    mount -t nfs -o nolock 192.168.0.10:/tftpboot/pmetro/tmp /mnt
```

4.1.8 Transferring Files to Metro SBC Using TFTP

This section describes the procedure to transfer files to Metro SBC from a TFTP server, i.e. a remote device.

```
     Example to download two files, where 192.168.1.10 is the IP of the
TFTP server.

     # tftp 192.168.0.10
     get pmetro/ramdisk.gz
     get pmetro/zImage
```

4.2 Antcor IkarusOS

IkarusOS "wireless (Wi-Fi) networking software", is a networking operating system which provides high-speed wireless connection. IkarusOS empowers modern OEMs with substantial benefits such as, compliance with the latest IEEE802.11 standards, innovative wireless features, support for WLAN chipsets from multiple vendors, accelerated time to market, lower development costs and risks, extensible for OEM differentiation and branding. Focusing on WI-FI innovation IkarusOS offers a number of added value features targeting performance and flexibility.

IkarusOS is a network services operating system which turns an embedded based system into a dedicated (WI-FI) router. IkarusOS enabled deployments are easily configured provisioned and monitored through IkarusOS Manager which is a software application that provides real-time visibility and control of all IkarusOS based systems of the network. The complete administration and configuration of IkarusOS systems is done through IkarusOS Manager executed from any platform that supports Java.

http://web.antcor.com/ikarusos.asp

4.3 RoamAD

RoamAD supplies a wireless networking platform for converged wireless networks including citywide Wi-Fi networks, public safety networks, highway Wi-Fi corridors, surveillance / CCTV networks as well as general use networks.

RoamAD's radio-agnostic, software-based platform is unmatched in flexibility and functionality supporting up to four radios per node in any combination for 802.11a/b/g and public safety networking. The frequency bands supported are 900 MHz, 2.4 GHz, 4.9 GHz, and 5 GHz.

RoamAD networks are scalable and, with their low-latency and fast-handoff attributes, are optimized to support mobile VoIP/VoWiFi.

Telecommunications carriers, ISPs, municipalities, and first responders are deploying RoamAD networks to cost-effectively and efficiently provide fixed and mobile voice and data access over large metro areas, campuses, and highway corridors.

http://www.roamad.com/roamad/

4.4 Valemount Networks StarOS

Valemount is the home of StarV3™, a powerful new Wireless platform, designed for the Valemount Networks high performance WAR (Wireless Advanced Router) systems, as well as do-it-yourself x86 users. With performance and features in mind, this new product is ideal for backhaul, and public access alike.

http://www.staros.com/index.php

4.5 Wilibox

WILI is a portable embedded Linux based software platform suited to implement highly functional, secure and manageable wired and wireless IP networking devices: HotSpot access points, access controllers, enterprise access points, fixed 802.11abg wireless network elements, base stations and customer premise equipment (CPE) and high performance, secure point to point or point to multi point wireless bridges.

WILI was originally developed for the Intel XScale (IXP-42x) family. Hardware encryption acceleration is fully exploited on IXP-42x CPU family based devices. WILI's scalable and portable nature allowed it to be ported on other hardware platforms like Atheros SoC AR2312, AR5312 (MIPS) and Kendin KS8695 SoC (ARM9) and Intel IA-32, with an upcoming porting to Broadcom BCM47xx SoC and other popular networking platforms.

The embeddable software is able to function on very constrained hardware that has 16 MB of RAM and 4 MB of flash memory. Latter amount of RAM and flash memory is sufficient for a fully functional 802.11abg Access Controller with Virtual Access Points, extended networking and wireless security capabilities. This makes WILI a great platform for a wide range of wireless network equipment: ranging from a simple access point or client device to a high speed access controller for busy HotSpot sites as well as component for secure fixed wireless networks.

WILI incorporates state of the art security functionality such as WEP/WPA/WPA2, dynamic key, 802.1x authenticator and supplicant. Security policy settings can be applied per BSSID or interface basis.

Multiple BSSID, VLAN, tunneling and per interface network policies allow creation of very flexible network structures where the same hardware infrastructure can be used for public access with WEB login and WPA2 protected company network or multiple WISP networks can run on the same hardware.

WILI was designed to address wireless network management problems as well. At the heart of WILI management subsystem is a simple text based configuration file and the RCMS (Remote Configuration Management System) agent, which in pair with an RCMS server creates a structure for fast wireless network deployment and robust administration.

Among other features, RCMS server has automatic provisioning, automatic configuration file and firmware image upload capabilities. RCMS agent can bypass most firewall and NAT protected routers without additional configuration. WILI has common management interfaces such as WEB, command line, SNMP and SYSLOG for troubleshooting.

http://www.wilibox.com/products/wili-embedded-linux

4.6 OpenWrt

OpenWrt is described as a Linux distribution for embedded devices. Instead of trying to create a single, static firmware, OpenWrt provides a fully writable filesystem with package management. This frees you from the application selection and configuration provided by the vendor and allows you to customize the device through the use of packages to suit any application. For developer, OpenWrt is the framework to build an application without having to build a complete firmware around it; for users this means the ability for full customization, to use the device in ways never envisioned.

http://openwrt.org/

5 Appendix

Additional Information

Hardware and Software Manuals, Linux BSP and RedBoot source trees, toolchains, etc are freely available on the ADI Engineering webpage for Pronghorn Metro SBC
http://www.adiengineering.com/php-bin/ecomm4/productDisplay.php?category_id=30&product_id=85

Technical Support

Register for access to FAQ, Message Boards, etc.
http://www.adiengineering.com/vanilla/