

Terminus 400AP Linux OS User Guide



JANUS REMOTE
COMMUNICATIONS

Bulletin	JA07-LOS_UM
Revision	00
Date	08 Nov 2012

TABLE OF CONTENTS

Table of Contents and Disclaimer	2
1 Introduction	3
2 Janus Remote Communications Cellular Plug-in Terminals	3
3 Required Items	3
4 Getting Started	4-23
4.1 SIM Card Installation	4
4.2 External Connections	5-6
4.3 Serial Console Sessions	7
4.4 SSH Console Sessions	8
4.5 Sending AT Commands	8
4.6 Cellular Configurations and Connectivity	9-21
4.7 Python Demonstration Scripts	22-23
5 Linux Communication I/O Mapping for Terminus 400AP	24
6 External 20-Pin Header	24
7 Peripheral Initialization	25-27
7.1 Initialize and Power On Cellular Radio	25
7.2 Test Plug-In Terminal AT Command Interface (ttyS1)	26
7.3 Test Plug-In Terminal AT Command Interface (ttyUSBO)	26
7.4 Initialize RS-485 Interface	27
7.5 Initialize CANBUS Interface	27
7.6 Set RTC Time	27
7.7 Mount SD Card	27
8 400AP Buildroot Toolchain Installation	28
8.1 Overview	28
8.2 Downloading Buildroot	28
8.3 Installing Buildroot Toolchain	28
8.4 Configuring Toolchain	28
8.5 Linux Kernel and UBI Filesystem	28
9 Amtel SAM-BA Application Installation	29
10 Uploading Firmware Images to NAND Flash in the Terminus 400AP	30-41
10.1 Overview	30
10.2 Configuring 400AP RomBOOT Mode	31
10.3 Launching SAM-BA Application	32
10.4 Writing Bootstrap Images to NAND Flash Memory	33
10.5 Writing U-Boot Image to NAND Flash Memory	34-35
10.6 Writing Linux Kernel Image to NAND Flash Memory	36-37
10.7 Writing File System Image to NAND Flash Memory	38-39
10.8 NAND Flash Recovery	40-41
Revision History	42

DISCLAIMER

The information contained in this document is the proprietary information of Connor-Winfield Corporation and its affiliates (Janus Remote Communication). The contents are confidential and any disclosure to persons other than the officers, employees, agents or subcontractors of the owner or licensee of this document, without the prior written consent of Connor-Winfield, is strictly prohibited. Connor-Winfield makes every effort to ensure the quality of the information it makes available. Notwithstanding the foregoing, Connor-Winfield does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information. Connor-Winfield disclaims any and all responsibility for the application of the devices characterized in this document, and notes that the application of the device must comply with the safety standards of the applicable country, and where applicable, with the relevant wiring rules. Connor-Winfield reserves the right to make modifications, additions and deletions to this document due to typographical errors, inaccurate information, or improvements to programs and/or equipment at any time and without notice. Such changes will, nevertheless be incorporated into new editions of this application note.

All rights reserved 2011 Connor-Winfield Corporation

1 Introduction

The 400AP comes with an example Linux kernel and UBI Filesystem loaded during manufacturing. The example build allows a developer the opportunity to evaluate the 400AP with little to no knowledge of the build environment or architecture of the device. This allows a quick method to evaluate the end application without getting too deep into the hardware. Please understand that the 400AP is an embedded hardware platform and has a limited memory footprint. The example build has a limited amount of software packages included.

The Linux OS allows for many options while developing applications on the 400AP. Python was chosen as the language for application demonstrations. The example build includes a full python environment and some examples written by Janus Remote Communications. The demonstrations are intended to illustrate how your application can interact to the 400AP hardware.

This guide will help you get the 400AP example build up and running quickly. Hardware connections and OS interaction are defined with a focus on application development. The quick start section will guide you through the steps and required accessories to properly evaluate the functionality of the 400AP. The guide is not intended to teach you how to use the Linux OS, but to show how the Linux OS interacts with the 400AP peripherals.

The Buildroot toolchain was used to build the example kernel and filesystem. Buildroot is a set of Makefiles and patches that make it easy to generate a complete embedded Linux system. Buildroot setup and firmware upload instruction are defined in the user guide.

2 Janus Remote Communications Cellular Plug-in Terminals:

The 400AP series terminal is equipped with a cellular Plug-In terminal manufactured by Janus Remote Communications. The following table details 400AP models and the Plug-In terminal supplied.

400AP Model	Plug-in Terminal	Approved Firmware	Plug-in Terminal Description
GSM400AP V1.0	GSM865CF V1.0	10.00.003	GSM 2.5G radio, NavSync MS20 GPS module
GSM400AP V2.0	GSM865CF V2.0	10.00.003	GSM 2.5G radio, w/o GPS
CDMA400AP V2.0	CDMA864CF V2.0	9.01.004	Sprint, CDMA-1xRTT 3G radio, Integrated GPS
CDMA400AP V3.0	CDMA864CF V3.0	9.91.024	Verizon, CDMA-1xRTT 3G radio, Integrated GPS
CDMA400AP V4.0	CDMA864CF V4.0	9.01.054	Aeris, CDMA-1xRTT 3G radio, Integrated GPS
HSPA400AP V1.0	HSPA910CF V1.0	12.00.003	HSPA 3.5G radio, Integrated GPS

Terminus Plug-In documentation can be found at the following URL.

<http://www.janus-rc.com/terminuscf.html>

The website includes hardware user guides and links to the Telit AT Command and Software User Guides for control of the Plug-In Terminus.

3 Required Items:

The 400AP can be purchased individually or included in an accessory kit. We suggest ordering the accessory kit during initial product evaluation in order to insure you have all the required accessories to evaluate the 400AP. There are many accessories kits available that contain different models of the 400AP. The following accessory kits are currently available.

400AP Model	Description
GSM400AP Accessory Kit V1.1	GSM 2.5G radio, NavSync MS20 GPS module
GSM400AP Accessory Kit V2.0	GSM 2.5G radio, w/o GPS
CDMA400AP Accessory Kit V2.0	Sprint, CDMA-1xRTT 3G radio, Integrated GPS
CDMA400AP Accessory Kit V3.0	Verizon, CDMA-1xRTT 3G radio, Integrated GPS
CDMA400AP Accessory Kit V4.0	Aeris, CDMA-1xRTT 3G radio, Integrated GPS
HSPA400AP Accessory Kit V1.0	HSPA 3.5G radio, Integrated GPS

These kits include the following accessories:

- Antennas for cellular and GPS (if needed)
- Power Supply
- USB cable
- Serial port cable
- USB to serial port adaptor
- Network patch cable

4 Getting Started:

Before you start please check that your accessory kit includes the following items, Antennas for cellular and GPS (if needed), Power Supply, USB cable, serial cable, USB to serial port adaptor and network patch cable. These items will be required during product evaluation.

4.1 SIM Card installation:

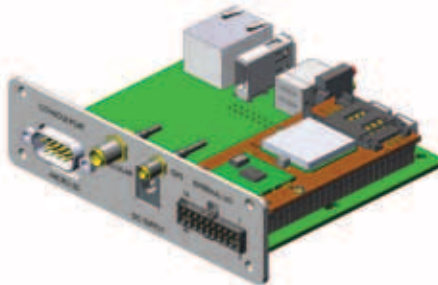
GSM and HSPA models of the 400AP require a SIM card in order to communicate with the network. The following section describes the SIM installation process.

4.1.1 Installing SIM Card into the GSM/ UMTS 400AP

Step 1. To access the 400AP board, remove the four TX-10 screws from the ruggedized aluminum enclosure.

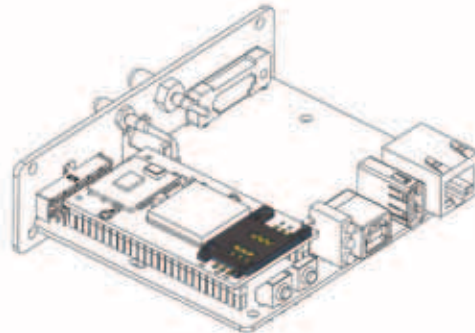


These screws are located on the back panel of the 400AP where the DB9 and lock header are located.



Step 2. Slide out the back panel which will include the 400AP board.

Step 3. Locate the SIM Card slot on top of the Terminus Plug-In board.



Step 4. To insert the SIM card in the SIM card slot, slide the top of the cover back to unlock. Insert the SIM card in the cover slot and close. Slide back to original position to lock in place.

Step 5. Slide the back panel and 400AP board into the aluminum enclosure. Replace the four TX-10 screws and tighten them.



4 Getting Started continued:

4.2 External Connections:

In order to power the 400AP and logon on to Linux you will need to make the following connections.

Connect the cellular antenna to the SMA connector port on the rear panel of the 400AP.

If you have the GSM400AP V2.0 that doesn't include a GPS receiver you will need to skip this step. Connect the GPS antenna to the MCX connector port on the rear panel of the 400AP.



Connection to the Linux console can be done via the serial port connector on the rear panel or via wired Ethernet connector on the front panel of the 400AP.



If you are using the serial console port you will need to connect the supplied serial cable between the 400AP and your development PC. For PCs that do not have a hardware serial port, we have supplied a USB to serial dongle.



4 Getting Started continued:

4.2 External Connections continued:

If you are using a wired Ethernet network to connect to the Linux console via SSH you will first need to review the default network settings of the 400AP.

The default wired Ethernet settings of the 400AP example build are:

IP Address: 192.168.1.50
Subnet: 255.255.255.0

If your existing network is on the 400AP default subnet and another device is not using the IP address, make the following connection. Connect the 400AP to a switch or hub in your network via the supplied network cable.



If your existing network is not on the 400AP default subnet or the IP address is already in use, make the following connection. Connect the 400AP to your development PC via the supplied network cable. In this configuration you will need to change your PC's NIC interface to match the 400AP default subnet.

Example PC network settings:

IP Address: 192.168.1.100
Subnet: 255.255.255.0



Connect power to the 400AP by plugging the barrel connector of the supplied AC transformer into the power connector on the rear panel of the 400AP. Plug the AC blades into an open AC receptacle.



4 Getting Started continued:

4.3 Serial Console Session:

A Terminal Emulator is needed in order to communicate with the 400AP console serial port(Front Panel DB9 Connector). The following instructions are for minicom, but you can use a terminal emulator you are comfortable with.

Step 1. Open a Linux Terminal window with user privileges that allow access to serial ports.

Step 2. Connect 400AP Console serial port to a PC serial port using a standard serial cable.

Step 3. Issue the following: minicom -s.

Follow the on screen menus to configure the following parameters:

Serial Device: /dev/ttyS0 (System Dependant)

Bps/Par/Bits: 115200 8N1

Hardware Flow Control: No

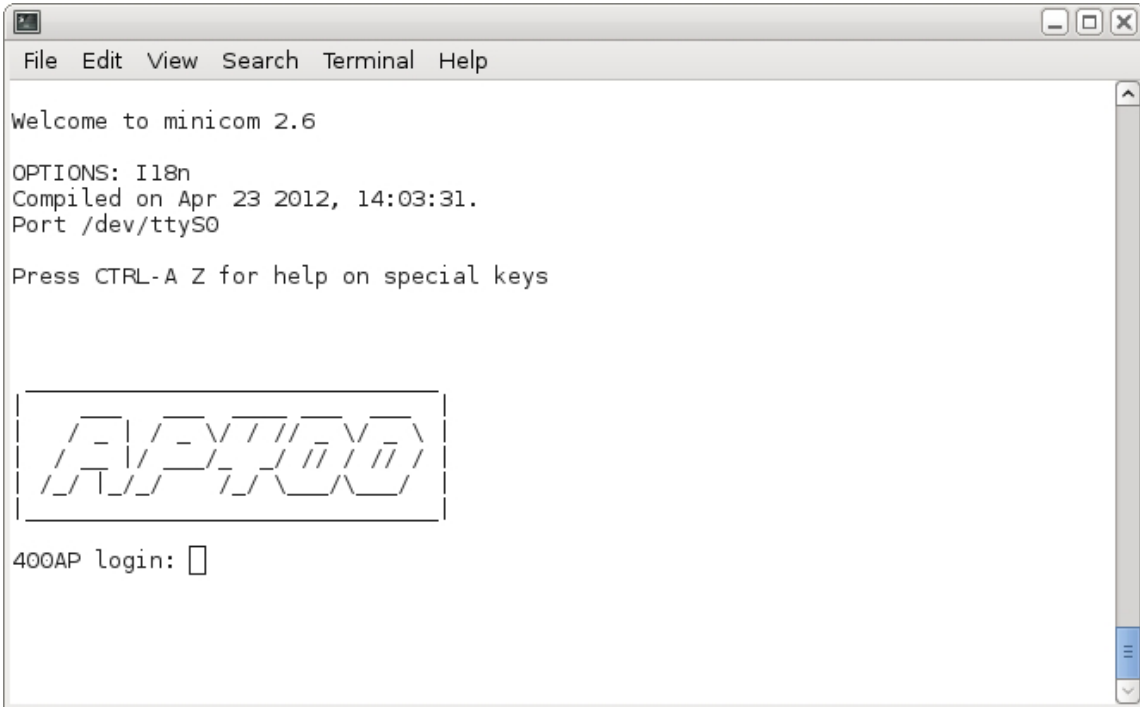
Software Flow Control: No

Save setup as DF1 before exiting.

System Dependant: This is the serial device of your PC it might not match this example. If you are using a USB to serial adaptor you will need to know the device name and enter it as the Serial Device.

Step 4. Issue the following to launch Minicom Terminal Emulator: minicom -o

If you have the terminal emulator setup up correctly you should see the 400AP boot if you cycle power or hit the reset button.



```
File Edit View Search Terminal Help

Welcome to minicom 2.6

OPTIONS: I18n
Compiled on Apr 23 2012, 14:03:31.
Port /dev/ttyS0

Press CTRL-A Z for help on special keys

[ ASCII art logo for JANUS ]

400AP login: [ ]
```

Login to console session with the following defaults:

User: root

Password: janus

Once logged into the console you will be able to control the 400AP as you would any Linux based PC. From this point out in the manual all references to opening a console session will be based on the pervious procedure.

4 Getting Started continued:

4.4 SSH Console Session:

In order to communicate with the 400AP via SSH console port you will need to setup a SSH session between your development PC and the 400AP. Issue the following command in a terminal session on your development Linux PC.

```
# ssh 192.168.1.50
```

Login to console session with the following defaults:

```
User:          root
Password:     janus
```

If connecting to the 400AP via SSH fails you will need to check network cable connection and verify that the network you are connected to is on the same subnet as the 400AP.

The default wired Ethernet settings of the 400AP example build are:

```
IP Address:    192.168.1.50
Subnet:        255.255.255.0
```

Once logged into the console you will be able to control the 400AP as you would any Linux based PC. From this point out in the manual all references to opening a SSH console session will be based on the pervious procedure.

4.5 Sending AT Commands:

The 400AP example build comes with a Python script that turns on the cellular radio and opens the microcom terminal emulation program. This will allow you to control the cellular radio via AT commands. Issue the following command to run this script.

```
#python /opt/demo400ap/scripts/demoMenu.py
```

Using the menu select option # 2.

```
400AP Demonstration (v1.0)
```

- 1) Edit demo configuration file (nano)
- 2) Open terminal to send AT commands (microcom)
- 3) Start GPS demo (demoGps.py)
- 4) Start a PPP connection (demoPPP.py)
- 5) Start router (demoRouter.py)
- x) Exit

```
Selection:
```

Wait for the script to display:

```
microcom terminal (ttyS1):
exit -> CTRL-X
```

At this point you can enter AT commands. The following sections will describe the process to verify provisioning

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity

The following sections will guide you through the AT commands to configure and test cellular network connectivity. Follow the section for the 400AP model you are evaluating.

4.6.1 GSM400AP

4.6.1.1 Telit Reference Documentation

The AT Command Reference Guide should be referenced for detailed command explanation while using the following sections to configure and test the cellular connectivity of the 400AP. On utilizing different commands for other applications than those described in this section, please refer to the following Telit reference documents. The Telit reference documents are firmware specific to our certified cellular terminals so please download from the Janus website. See Telit Documentation tab.

<http://www.janus-rc.com/gsm865cf.html>

Telit AT Commands Reference Guide

Telit Modules Software User Guide

Telit IP Easy User Guide

4.6.1.2 Verify Telit Firmware version

The Plug-in Terminus embedded in the 400AP has been certified with PTCRB and carrier approved with a specific Telit radio firmware. The following command returns the firmware version of the Telit radio. Please confirm firmware matches the 400AP model listed in section 2.

Issue: AT+CGMR<cr>

Response: 10.00.00x

4.6.1.3 Network Set-Up

Network settings for the Plug-in Terminus will vary depending on the cellular carrier you are using. Below are two cases for these settings within the US.

T-mobile® & MNVO (Raco®, Sensor Logic®, Nexaira® Jasper Wireless®)

Enter the following AT Commands:

- AT#SELINT=2
- AT#STIA=2,10 or AT#STIA=1
- AT#BND=3
- AT#AUTOBND=1
- AT#PLMNMODE=1
- AT#ENS=0
- AT&P0
- AT&W0

AT&T/Cingular® & MNVO (Kore®, Aeris®, nPhase®)

Enter the following AT Commands:

- AT#SELINT=2
- AT#ENS=1
- AT&P0
- AT&W0

If the Plug-in Terminus is being used in a different country or with a different carrier please refer to Telit AT Command Reference Guide regarding the use of the AT#BND command to set the proper frequency band.



The Plug-in Terminus will need to be power cycled after entering either of the AT command sets above. Exiting the current terminal window by selecting CTRL-X will allow the demoMicrocom.py script to power down the Plug-in Terminus.

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.1 GSM400AP continued

4.6.1.4 Check Signal Quality

The Plug-in Terminus reports its signal strength with the following command.

Issue: AT+CSQ<cr>
Response: +CSQ:<rssi>,<ber>

rssi	Signal Strength
99	Not known or not detectable
0-31	$\text{dBm} = (\text{rssi} * 2) - 113$

Note: A result of 31 indicates -51dBm or greater.

High rssi values are desired in order to maintain the fastest possible data rate. The rssi will vary depending on distance to base station tower and quality of attached antenna. As the signal strength reduces the network will start to adapt the coding scheme to insure data integrity. As the data coding scheme increases in complexity the data rate decreases.

4.6.1.5 Check Network Status

If the Plug-in Terminus is setup correctly and an active SIM card is installed the module will register to the carrier network and attach to GPRS service. Issue the following commands to verify network setup.

Ensure the module is registered on the home network or roaming.

Issue: AT+CREG?<cr>
Response: +CREG:0,1 or +CREG:0,5

Ensure the module is attached to GPRS service on the home network or roaming.

Issue: AT+CGREG?<cr>
Response: +CGREG:0,1 or +CGREG:0,5

If responses are different than listed above please refer to the Telit AT Command Reference Guide for more information.

Note: If the Plug-in Terminus indicates no network registration check ENS and BND settings. If these settings are correct for the carrier SIM you have installed your SIM card might not be active. Contact the network carrier and verify your account and SIM are active.

4.6.1.6 Sending an SMS

SMS (Select Message Service) mode allows you to send a text message to an SMS capable subscriber unit. Issue the following commands to send an SMS message.

Set SMS mode to text. This must be entered every power cycle.

Issue: AT+CMGF=1<cr>
Response: OK

Enter the receiving subscriber unit phone number.

Issue: AT+CMGS="8885551234" <cr>
Response: >

Send SMS message text.

Issue: "text message up to a max of 160 ASCII characters"
Issue: "Ctrl-z" or ASCII character 0x1A to send the message
Response: +CMGS: <message reference number>

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.1 GSM400AP continued

4.6.1.7 Establishing a Socket Connection

Issue the following commands to establish a TCP/IP socket via the Internet.

Set up the PDP context parameters.

Issue: AT+CGDCONT=1, "IP",APN<cr>
Response: OK

APN (Access Point Name) is a configurable network identifier used by a mobile device when connecting to a GSM carrier. The carrier examines this identifier to determine what type of network connection you have provisioned.

Please contact your carrier for the APN that belongs to your account type.

Check if PDP context is active.

Issue: AT#SGACT?<cr>
Response: #SGACT=1,0 or AT#SGACT=1,1

If response is #SGACT=1,0 then activate PDP context.

Issue: AT#SGACT=1,1,USERID,PASSWORD<cr>
Response: #SGACT:"IP address assigned to radio"

USERID: Context userid, optional parameter
PASSWORD: Context password, optional parameter

If carrier context doesn't require USERID and PASSWORD then omit and issue: AT#SGACT=1,1.

Note: If you activate PDP context when already active you will receive and ERROR response.

Open socket connection to server on the Internet.

Issue: AT#SD=1,0,PORT,ADDRESS<cr>
Response: CONNECT

PORT: Designation IP port (0 to 65535)
ADDRESS: Designation IP address in the format: "xxx.xxx.xxx.xxx"
or any host name to be solved with a DNS query.

Example socket connection: AT#SD=1,0,80,"www.google.com"

After the CONNECT response is received the AT command port transitions from command mode to data mode. Data sent to the AT command port is sent to the server. Data received on the AT command port was sent from the server.

Exit data mode.

Issue: +++
Response: OK

Close socket connection.

Issue: AT#SH=1<cr>
Response: OK

4.6.2 CDMA400AP

4.6.2.1 Telit Reference Documentation

The AT Command Reference Guide should be referenced for detailed command explanation while using the following sections to configure and test the cellular connectivity of the 400AP. On utilizing different commands for other applications than those described in this section, please refer to the following Telit reference documents. The Telit reference documents are firmware specific to our certified cellular terminals so please download from the Janus website. See Telit Documentation tab.

<http://www.janus-rc.com/cdma864cf.html>

Telit AT Commands Reference Guide
Telit Modules Software User Guide
Telit IP Easy User Guide

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.2 CDMA400AP continued

4.6.2.2 Verify Telit Firmware version

The Plug-in Terminus embedded in the 400AP has been carrier approved with a specific Telit radio firmware. The following command returns the firmware version of the Telit radio. Please confirm firmware matches the 400AP model listed in section 2.

```
Issue:          AT+CGMR<cr>
Response:       09.01.0xx
```

4.6.2.3 CDMA Radio Provisioning

CDMA radios need to be provisioned in order to communicate with the cellular network. The following sections will describe the various CDMA carries provisioning procedures.

4.6.2.3.1 Sprint Provisioning

Sprint uses the protocol OMA Device Management specified by the Open Mobile Alliance (OMA). The following sections will guide you through the provisioning process.

4.6.2.3.1.1 Sprint Account Set-up

Before your radio can be provisioned over the air you will need to contact Sprint or a Sprint MVNO and set-up a service contract. Sprint will request the following product information.

```
Product Model Number:  CDMA864CF V2.00
Product Manufacture:   Janus Remote Communications
Mobile Equipment Identifier (MEID) of each radio you plan to provision.
```

Query radio MEID.

```
Issue:          AT#MEID? <cr>
Response:       #MEID: A10000,00000001
```

Note: When supplying Sprint with MEID remove the comma from MEID response (A1000000000001).

If you need help during provisioning please contact us.

Contact information:
Dave Jahr, Janus Remote Communications
djahr@janus-rc.com
630-499-2121

4.6.2.3.1.2 Automatic OMA-DM Sprint Provisioning

After receiving verification that your service has been activated you will need to power up the Plug-in Terminus. Upon power up, the module will attempt to provision itself if not already done. It will continue to attempt this until it is successful. During the OMA-DM process you will see the following messages if the radio is provisioning.

```
#904 (initiate OMA)
#914 (provisioning successful)
```

To determine if your terminal has been provisioned to work with the Sprint network you need to review the current profile settings.

```
Issue:          AT$QCMIPGETP<cr>
Response:       Profile:0 Enabled
                NAI:A1000009D010CA@hcm.sprintpcs.com
                Home Addr:0.0.0.0
                Primary HA:68.28.15.12
                Secondary HA:68.28.31.12
                MN-AAA SPI:1234
                MN-HA SPI:1234
                Rev Tun:1
                MN-AAA SS:Set
                MN-HA SS:Set
```

A terminal not provisioned will have the following characteristics.

```
Profile:        0 Enabled
NAI: YOUR_TERMINALS_MEID@hcm.sprintpcs.com
```

A terminal provisioned will have the following characteristics.

```
Profile :       1 Enabled
NAI: YOUR_COMPANY_NAME@sprintpcs.com
```

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.2 CDMA400AP continued

4.6.2.3 CDMA Radio Provisioning continued

4.6.2.3.1.3 Manual OMA-DM Sprint Provisioning

In the event that Sprints automatic OMA-DM provisioning does not function properly follow the steps below to trigger the automated program.

Ensure the module is registered on the network.

Issue: AT+CREG?<cr>
Response: +CREG:0,1 or +CREG:0,5

Initiate the “Sprint Hands Free Activation” (HFA) session.

Issue: AT#HFA<cr>
Response: OK
#904
#914

Complete the “Sprint Hands Free Activation” (HFA) session.

Issue: AT#HFAOK<cr>
Response: OK

Upon reboot determine if your radio has been provisioned to work with the Sprint network.

Issue: AT\$QCMIPGETP<cr>
Response: Profile:0 Enabled
NAI:A1000009D010CA@hcm.sprintpcs.com
Home Addr:0.0.0.0
Primary HA:68.28.15.12
Secondary HA:68.28.31.12
MN-AAA SPI:1234
MN-HA SPI:1234
Rev Tun:1
MN-AAA SS:Set
MN-HA SS:Set

A radio not provisioned will have the following characteristics.

Profile: 0 Enabled
NAI: YOUR_RADIO_MEID@hcm.sprintpcs.com

A radio provisioned will have the following characteristics.

Profile : 1 Enabled
NAI: YOUR_COMPANY_NAME@sprintpcs.com

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.2 CDMA400AP continued

4.6.2.3 CDMA Radio Provisioning continued

4.6.2.3.2 Verizon Provisioning

Verizon uses the protocol OMA Device Management specified by the Open Mobile Alliance (OMA). The following sections will guide you through the provisioning process.

4.6.2.3.2.1 Verizon Account Set-up

Before your radio can be provisioned over the air you will need to contact Verizon or a Verizon MVNO and set-up a service contract. Verizon will request the following product information.

Product Model Number: CDMA864CF V3.00
Product Manufacture: Janus Remote Communications
Mobile Equipment Identifier (MEID) of each radio you plan to provision.

Query radio MEID.

Issue: AT#MEID? <cr>

Response: #MEID: A10000,00000001

Note: When supplying Verizon with MEID remove the comma from MEID response (A1000000000001).

If you need help during provisioning please contact us.

Contact information:

Dave Jahr, Janus Remote Communications

djahr@janus-rc.com

630-499-2121

4.6.2.3.2.2 Manual OMA-DM Verizon Provisioning

After receiving verification that your service has been activated you will need to manually start the Verizon provisioning process.

Ensure the module is registered on the network.

Issue: AT+CREG? <cr>

Response: +CREG: 0,1 or +CREG: 0,5

Select profile 0.

Issue: AT\$QCMIPP=0<cr>

Response: OK

Enable profile 0.

Issue: AT\$QCMIPPEP=1<cr>

Response: OK

Start OTASP provisioning session.

Issue: ATD*22899;<cr>

Response: #OTASP:0

#OTASP:1

#OTASP:2

If you receive a response #OTASP:5, or do not receive #OTASP:1 and #OTASP:2, the provisioning has failed somewhere. Please verify that your account has been set up, activated, and that you have been given an MDN and MSID for your module's MEID/ESN.



Do not skip the following steps or the provisioning process will not complete and you will not be able to communicate with the network.

Activate the PDP context.

Issue: AT#SGACT=1,1<cr>

Response: #SGACT:"IP address assigned to radio"

Note: AT#SGACT must return a valid IP address. If not the provisioning process has failed.

Wait a minimum of 10-15 seconds.

De-activate the PDP context.

Issue: AT#SGACT=1,0

Response: OK

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.2 CDMA400AP continued

4.6.2.3 CDMA Radio Provisioning continued

4.6.2.3.3 Sprint MVNO - CrossBridge Provisioning

Crossbridge provisioning is a manual process and requires careful entry of network parameters. The following sections will guide you through the provisioning process.

4.6.2.3.3.1 CrossBridge Account Set-up

Before your radio can be provisioned over the air you will need to contact Crossbridge and set-up a service contract. Crossbridge will request the following product information.

Product Model Number: CDMA864CF V2.00

Product Manufacture: Janus Remote Communications

Mobile Equipment Identifier (MEID) of each radio you plan to provision.

Query radio MEID.

Issue: AT#MEID? <cr>

Response: #MEID: A10000,00000001

Note: When supplying Crossbridge with MEID remove the comma from MEID response (A1000000000001).

If you need help during provisioning please contact us.

Contact information:

Dave Jahr, Janus Remote Communications

djahr@janus-rc.com

630-499-2121

4.6.2.3.3.2 CrossBridge Manual Provisioning

After receiving verification that your service has been activated you will need to manually start the Crossbridge provisioning process. The following information will be sent to you when the ESN has been activated.

The module's 10 digit MDN

The module's 10 digit MSID

The module's NAI , example (<MSID>@spp106.dl.sprintpcs.com)

The password associated to the NAI <pw>

The Primary Home Address <pha>

The Secondary Home Address <sha>

The HA-SS

Ensure the module is registered on the network.

Issue: AT+CREG? <cr>

Response: +CREG: 0,1 or +CREG: 0,5

Set the MDN value that you received from Crossbridge.

Issue: AT#ENG=9:MDN<cr>

Response: OK

Set the MSID value that you received from Crossbridge.

Issue: AT#ENG=10:<MSID><cr>

Response: OK

Ensure the modules MDN and MSID have been entered correctly.

Issue: AT#MODEM?<cr>

Response: #MODEM: <mdn>, <msid>

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.2 CDMA400AP continued

4.6.2.3 CDMA Radio Provisioning continued

4.6.2.3.3 Sprint MVNO - CrossBridge Provisioning continued

Select NAI profile 0.

Issue: AT\$QCMIPP=0

Response: OK

Disable NAI profile 0.

Issue: AT\$QCMIPPEP=0

Response: OK

Select NAI profile 1.

Issue: AT\$QCMIPP=1

Response: OK

Enable NAI profile 1.

Issue: AT\$QCMIPPEP=1

Response: OK

Enable mobile IP.

Issue: AT\$QCMIP=2

Response: OK

Enter NAI for profile 1.

Issue: AT\$QCMIPNAI=<nai>,1

Response: OK

Enter Home Address.

Issue: AT\$QCMIPHA=0.0.0.0,1

Response: OK

Enter the Primary Home Address.

Issue: AT\$QCMIPPHA=<pha>,1

Response: OK

Enter the Secondary Home Address.

Issue: AT\$QCMIPSHA=<sha>,1

Response: OK

Enter the AAA Server Security Parameter Index.

Issue: AT\$QCMIPMASPI=1234,1

Response: OK

Enter the Home Agent Security Parameter Index.

Issue: AT\$QCMIPMHSPI=1234,1

Response: OK

Enter the MN-AAA Shared Secret.

Issue: AT\$QCMIPMASS=<pw>,1

Response: OK

Enter the MN_HA Shared Secret.

Issue: AT\$QCMIPMHSS=oursecretmnhakey,1

Response: OK

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.2 CDMA400AP continued

4.6.2.3 CDMA Radio Provisioning continued

4.6.2.3.3 Sprint MVNO - CrossBridge Provisioning continued

Enable Reverse Tunneling.

Issue: AT\$QCMIPRT=1,1
Response: OK

Display the current NAI profile 1 settings and verify all values are correct.

Issue: AT\$QCMIPGETP
Response: Profile:1 Enabled
NAI: <MSID>@spp106.dl.sprintpcs.com
Home Addr:0.0.0.0
Primary HA:<pha>
Secondary HA:<sha>
MN-AAA SPI:1234
MN-HA SPI:1234
Rev Tun:1
MN-AAA SS:Set
MN-HA SS:Set

Note: Passwords will not be displayed, but if a value has been entered and saved it will be displayed as "Set." If you see them displayed as "Unset" go back and enter the value again.

Perform a reset.

Issue: AT\$SPRESET<cr>
Response: OK

The module will be unresponsive while it reboots. The module should be operational again in approximately 10 seconds.

4.6.2.4 Sending an SMS

SMS (Select Message Service) mode allows you to send a text message to an SMS capable subscriber unit. Issue the following commands to send an SMS message.

Set SMS mode to text. This must be entered every power cycle.

Issue: AT+CMGF=1<cr>
Response: OK

Set the text mode parameters. This is recommended as it will allow functionality of SMS on all networks

Issue: AT+CSMP="callback_address",4098,0,2<cr>
Response: OK
callback_address: The MDN of the provisioned module.

Save your CSMP settings.

Issue: AT+CSAS<cr>
Response: OK

To retrieve the CSMP saved settings.

Issue: AT+CRES<cr>
Response: OK

Enter the receiving subscriber unit phone number.

Issue: AT+CMGS="8885551234" <cr>
Response: >

Send SMS message text.

Issue: "text message up to a max of 160 ASCII characters"
Issue: "Ctrl-z" or ASCII character 0x1A to send the message
Response: +CMGS: <message reference number>

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.2 CDMA400AP continued

4.6.2.5 Establishing a Socket Connection

Issue the following commands to establish a TCP/IP socket via the Internet.

Select NAI Profile 1.

Issue: AT\$QCMIPP=1<cr>

Response: OK

Enable NAI Profile 1.

Issue: AT\$QCMPEP=1<cr>

Response: OK

Enable Mobile IP

Issue: AT\$QCMIP=2<cr>

Response: OK

Check if Context is active.

Issue: AT#SGACT? <cr>

Response: #SGACT=1,0 or AT#SGACT=1,1

If response of AT#SGACT=1,0 then activate context.

Issue: AT#SGACT=1,1<cr>

Response: #SGACT:“IP address assigned to radio”

Note: If you activate context when already active you will receive and ERROR response.

Open socket connection to server on the Internet.

Issue: AT#SD=1,0,PORT,ADDRESS<cr>

Response: CONNECT

PORT: Designation IP port (0 to 65535)

ADDRESS: Designation IP address in the format: “xxx.xxx.xxx.xxx”
or any host name to be solved with a DNS query.

Example socket connection: AT#SD=1,0,80,”www.google.com”

After the CONNECT response is received the AT command port transitions from command mode to data mode. Data sent to the AT command port is sent to the server. Data received on the AT command port was sent from the server.

Exit data mode.

Issue: +++

Response: OK

Close socket connection.

Issue: AT#SH=1<cr>

Response: OK

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.3 HSPA400AP

4.6.3.1 Telit Reference Documentation

The AT Command Reference Guide should be referenced for detailed command explanation while using the following sections to configure and test the cellular connectivity of the 400AP. On utilizing different commands for other applications than those described in this section, please refer to the following Telit reference documents. The Telit reference documents are firmware specific to our certified cellular terminals so please download from the Janus website. See Telit Documentation tab.

<http://www.janus-rc.com/HSPA910cf.html>

Telit AT Commands Reference Guide

Telit Modules Software User Guide

Telit IP Easy User Guide

4.6.3.2 Verify Telit Firmware version

The Plug-in Terminus embedded in the 400AP has been carrier approved with a specific Telit radio firmware. The following command returns the firmware version of the Telit radio. Please confirm firmware matches the 400AP model listed in section 2.

Issue: AT+CGMR<cr>
Response: 12.00.00x

4.6.3.3 Network Set-Up

Network settings for the Plug-in Terminus will vary depending on the cellular carrier you are using. Below are two cases for these settings within the US.

T-mobile® & MNVO (Raco®, Sensor Logic®, Nexaira® Jasper Wireless®)

Enter the following AT Commands:

- AT#SELINT=2
- AT#STIA=2,10 or AT#STIA=1
- AT#BND=3,3
- AT#AUTOBND=1
- AT#ENS=0
- AT&P0
- AT&W0

AT&T/Cingular® & MNVO (Kore®, Aeris®, nPhase®)

Enter the following AT Commands:

- AT#SELINT=2
- AT#ENS=1
- AT&P0
- AT&W0

If the Plug-in Terminus is being used in a different country or with a different carrier please refer to Telit AT Command Reference Guide regarding the use of the AT#BND command to set the proper frequency band.z



The Plug-in Terminus will need to be power cycled after entering either of the AT command sets above. Exiting the current terminal window by selecting CTRL-X will allow the demoMicrocom.py script to power down the Plug-in Terminus.

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.3 HSPA400AP continued

4.6.3.4 Check Signal Quality

The Plug-in Terminus reports its signal strength with the following command.

Issue: AT+CSQ<cr>
Response: +CSQ:<rsqi>,<ber>

rsqi	Signal Strength
99	Not known or not detectable
0-31	$\text{dBm} = (\text{rsqi} * 2) - 113$

Note: A result of 31 indicates -51dBm or greater.

High rssi values are desired in order to maintain the fastest possible data rate. The rssi will vary depending on distance to base station tower and quality of attached antenna. As the signal strength reduces the network will start to adapt the coding scheme to insure data integrity. As the data coding scheme increases in complexity the data rate decreases.

4.6.3.5 Check Network Status

If the Plug-in Terminus is setup correctly and an active SIM card is installed the module will register to the carrier network and attach to GPRS service. Issue the following commands to verify network setup.

Ensure the module is registered on the home network or roaming.

Issue: AT+CREG?<cr>
Response: +CREG:0,1 or +CREG:0,5

Ensure the module is attached to GPRS service on the home network or roaming.

Issue: AT+CGREG?<cr>
Response: +CGREG:0,1 or +CGREG:0,5

If responses are different than listed above please refer to the Telit AT Command Reference Guide for more information.

Note: If the Plug-in Terminus indicates no network registration check ENS and BND settings. If these settings are correct for the carrier SIM you have installed your SIM card might not be active. Contact the network carrier and verify your account and SIM are active.

4.6.3.6 Sending an SMS

SMS (Select Message Service) mode allows you to send a text message to an SMS capable subscriber unit. Issue the following commands to send an SMS message.

Set SMS mode to text. This must be entered every power cycle.

Issue: AT+CMGF=1<cr>
Response: OK

Enter the receiving subscriber unit phone number.

Issue: AT+CMGS="8885551234" <cr>
Response: >

Send SMS message text.

Issue: "text message up to a max of 160 ASCII characters"
Issue: "Ctrl-z" or ASCII character 0x1A to send the message
Response: +CMGS: <message reference number>

4 Getting Started continued:

4.6 Cellular Configurations and Connectivity continued

4.6.3 HSPA400AP continued

4.6.3.7 Establishing a Socket Connection

Issue the following commands to establish a TCP/IP socket via the Internet.

Set up the PDP context parameters.

Issue: AT+CGDCONT=1, "IP",APN<cr>
Response: OK

APN (Access Point Name) is a configurable network identifier used by a mobile device when connecting to a GSM carrier. The carrier examines this identifier to determine what type of network connection you have provisioned.

Please contact your carrier for the APN that belongs to your account type.

Check if PDP context is active.

Issue: AT#SGACT?<cr>
Response: #SGACT=1,0 or AT#SGACT=1,1

If response is #SGACT=1,0 then activate PDP context.

Issue: AT#SGACT=1,1,USERID,PASSWORD<cr>
Response: #SGACT:"IP address assigned to radio"

USERID: Context userid, optional parameter

PASSWORD: Context password, optional parameter

If carrier context doesn't require USERID and PASSWORD then omit and issue: AT#SGACT=1,1.

Note: If you activate PDP context when already active you will receive and ERROR response.

Open socket connection to server on the Internet.

Issue: AT#SD=1,0,PORT,ADDRESS<cr>
Response: CONNECT

PORT: Designation IP port (0 to 65535)

ADDRESS: Designation IP address in the format: "xxx.xxx.xxx.xxx"
or any host name to be solved with a DNS query.

Example socket connection: AT#SD=1,0,80,"www.google.com"

After the CONNECT response is received the AT command port transitions from command mode to data mode. Data sent to the AT command port is sent to the server. Data received on the AT command port was sent from the server.

Exit data mode.

Issue: +++
Response: OK

Close socket connection.

Issue: AT#SH=1<cr>
Response: OK

4 Getting Started continued:

4.7 Python Demonstration Scripts

4.7.1 Configuring Python Demos

The 400AP example build comes with a Python script that opens the demo400ap.conf file in the nano text editor. Issue the following command to run this script.

```
#python /opt/demo400ap/scripts/demoMenu.py
```

Using the menu select option # 1.

400AP Demonstration (v1.0)

- 1) Edit demo configuration file (nano)
- 2) Open terminal to send AT commands (microcom)
- 3) Start GPS demo (demoGps.py)
- 4) Start a PPP connection (demoPPP.py)
- 5) Start router (demoRouter.py)
- x) Exit

Selection:

After the configuration file opens you will need to change the following parameters to match your setup.

CGMM
CGMR
ENS
APN

All other parameters should be unchanged at this time.

CGMM represents the Telit radio that is embedded in the 400AP. The scripts use this parameter to verify you have the specified Telit radio installed in the 400AP. The following table describes the correct settings for the CGMM parameter.

400AP Model	CGMM
GSM400AP	GE865
CDMA400AP	CC864-DUAL
HSPA400AP	HE910

Example: CGMM=HE910

CGMR specifies the firmware of the Telit radio that is embedded in the 400AP. The scripts use this parameter to verify that the specified Telit firmware is installed in the 400AP. The scripts have been tested with this firmware version. The following table describes the correct settings for the CGMR parameter.

400AP Model	CGMR
GSM400AP	10.00.003
CDMA400AP V2.0	09.01.004
CDMA400AP V3.0	09.01.024
CDMA400AP V4.0	09.01.054
HSPA400AP	12.00.003

Example: CGMR=12.00.003

ENS selects if enhanced network selection should be enabled. The following table describes the correct settings for the ENS parameter.

ENS	Description
0	Standard GSM Network
1	AT&T Network

Example: ENS=1

4 Getting Started continued:

4.7 Python Demonstration Scripts continued

4.7.1 Configuring Python Demos continued

Access Point Name (APN) is a configurable network identifier used by a mobile device when connecting to a GSM carrier. The carrier examines this identifier to determine what type of network connection you have provisioned. Please contact your carrier for the APN that belongs to your account.

Example: APN=internet

4.7.2 Running Python Demonstration Scripts:

The 400AP example build comes with the following demonstration scripts.

```
/opt/demo400ap/scripts/demoGPS.py
/opt/demo400ap/scripts/demoPPP.py
/opt/demo400ap/scripts/demoRouter.py
```

The scripts can be called directly or via a helper script that launches the scripts via a menu.



If the script encounters an exception it might be useful to run the demo scripts directly without use of the demoMenu.py script in order to see exception information.

```
/opt/demo400ap/scripts/demoMenu.py
```

4.7.3 GPS Demo Script

The demoGPS.py script reports GPS data to a Janus web server that displays a single location on a Google™ map. Script settings are defined in the demo400ap.conf file.

Issue the following command to run this script.

```
#python /opt/demo400ap/scripts/demoGPS.py
```

Location data sent to the Janus server is displayed at the following URL.

```
http://gps.conwin.com/gpstrack
```

4.7.4 PPP Demo Script

The demoPPP.py script creates a network interface ppp0. When the script exits the ppp0 interface can be used by applications running on the 400AP to communicate via socket API. The intention of this script is to illustrate one method to create a network interface using ppp via the 400AP cellular radio. Issue the following command to run this script.

```
#python /opt/demo400ap/scripts/demoPPP.py
```

4.7.5 Router Demo Script

The demoRouter.py script creates a network interface ppp0. Re-configures the eth0 network interface to the following settings.

```
IP Address:    192.168.2.1
Sub-Net:       255.255.255.0
```

Starts a DHCP service to handle DHCP requests on eth0. Starts iptables and routes data from ppp0 to eth0. When the script exits the 400AP acts as a cellular router. The intention of this script is to illustrate one method to route network traffic from eth0 via ppp to the cellular network. Issue the following command to run this script.

```
#python /opt/demo400ap/scripts/demoRouter.py
```

5 Linux Communication I/O Mapping for Terminus 400AP

I/O	Linux Device	Dir	Driver	Run State	Note
Linux Console Port	ttyS0	BI	atmel_serial	n/a	1
Cellular AT Command Port	ttyS1	BI	atmel_serial	n/a	
External Serial Port (RS-232)	ttyS2	BI	atmel_serial	n/a	
External Serial Port (RS-485)	ttyS3	BI	atmel_serial	n/a	
MS20 NMEA Port	ttyS4	BI	atmel_serial	n/a	2
Telit Trace Port	ttyS5	BI	atmel_serial	n/a	3
ON_OFF	GPIO104	OUT	sysfs	0	
RESET	GPIO105	OUT	sysfs	0	
GPS_RESET	GPIO106	OUT	sysfs	0	2
SERVICE	GPIO108	OUT	sysfs	0	4
ENABLE	GPIO111	OUT	sysfs	0	
ENABLE_VBUS	GPIO99	OUT	sysfs	0	3
ENABLE AT PORT	GPIO94	OUT	sysfs	0	
PWRMON	GPIO95	IN	sysfs	n/a	
CAN_DRIVE_ENABLE	GPIO77	OUT	sysfs	0	
External GPIO_1	GPIO37	BI	sysfs	n/a	5
External GPIO_2	GPIO83	BI	sysfs	n/a	5
External GPIO_3	GPIO96	BI	sysfs	n/a	5
External GPIO_4	GPIO97	BI	sysfs	n/a	5

Notes:

1. Console serial port settings (115200,8,N,1)
2. GSM400AP V1.0 model only.
3. Not available on Beta units, pin/port will be connected on production units.
4. GSM400AP models only.
5. For 3.3Vdc CMOS level I/O, see Hardware User Guide

6 External 20-Pin Header

PIN	Description	Direction	LEVEL
1	GPIO_3	IN / BI-DIR	Analog / 3.3v CMOS
2	GPIO_1	BI-DIR	3.3v CMOS
3	GPIO_4	IN / BI-DIR	Analog / 3.3v CMOS
4	GPIO_2	BI-DIR	3.3v CMOS
5	GND	N/A	N/A
6	RS485_A	BI-DIR	RS-485
7	CAN0_HI	BI-DIR	CAN
8	RS485_B	BI-DIR	RS-485
9	CAN0_LO	BI-DIR	CAN
10	GND	N/A	N/A
11	GND	N/A	N/A
12	RS232 TX	OUTPUT	RS-232
13	CAN1_HI	BI-DIR	CAN
14	RS232 RX	INPUT	RS-232
15	CAN1_LO	BI-DIR	CAN
16	RS232 RTS	OUTPUT	RS-232
17	Enable	IN	12.0v (Active High)
18	RS232 CTS	INPUT	RS-232
19	VIN	IN	12.0v
20	GND	N/A	N/A

7 Peripheral Initialization

7.1 Initialize and Power On Cellular Radio

The following steps use the sysfs GPIO drivers to initialize the Plug-In terminal installed in the 400AP.

Step 1. Open a Terminal Emulator application. Login as root if needed, password is blank.

Note: Refer to Terminal Emulator for use with the 400AP in this user guide.

Step 2. Run modem_init script in Linux console.

This script needs to be run one time after the 400AP has been powered up.

usr/local/bin/modem_init Script:

```
echo 94 >/sys/class/gpio/export
echo 95 >/sys/class/gpio/export
echo 104 >/sys/class/gpio/export
echo 105 >/sys/class/gpio/export
echo 106 >/sys/class/gpio/export
echo 108 >/sys/class/gpio/export
echo 111 >/sys/class/gpio/export
echo 99 >/sys/class/gpio/export

echo "high" >/sys/class/gpio/gpio94/direction
echo "in" >/sys/class/gpio/gpio95/direction
echo "low" >/sys/class/gpio/gpio104/direction
echo "low" >/sys/class/gpio/gpio105/direction
echo "low" >/sys/class/gpio/gpio106/direction
echo "low" >/sys/class/gpio/gpio108/direction
echo "low" >/sys/class/gpio/gpio111/direction
echo "low" >/sys/class/gpio/gpio99/direction
```

Step 3. Run modem_on_off script in Linux console.

This script toggles the on off state of the Plug-In terminal every time it is run.

usr/local/bin/modem_on_off Script:

```
echo "0" >/sys/class/gpio/gpio99/value
echo "1" >/sys/class/gpio/gpio94/value
echo "1" >/sys/class/gpio/gpio104/value
sleep 3
echo "0" >/sys/class/gpio/gpio104/value
```

Step 4. Check PWRMON state to verify that the Plug-In terminal has been turned on.

Issue: `cat </sys/class/gpio/gpio95/value`

A return value of 0 indicates that the Plug-In terminal has been turned on.

Note: The PWRMON input can read high if GPIO94 is set high when the Plug-In terminal is off. Make sure to set GPIO94 and GPIO113 low before turning Plug-In terminal off. This will ensure an accurate read of the PWRMON input.

7 Peripheral Initialization continued

7.2 Test Plug-In Terminal AT Command Interface (ttyS1):

All Plug-In terminals have an AT Command port exposed via a USART interface. The following instructions show how to send AT commands with the Microcom tool.

Step 1. Run modem_use_uart script in Linux console.

This script is used to enable the CMOS drivers that connect between the 400AP and the Plug-In terminal.

usr/local/bin/modem_use_uart script:

```
echo "0" >/sys/class/gpio/gpio94/value
```

Step 2. Launch microcom terminal emulator on 400AP Console port.

Issue: `microcom -s 115200 /dev/ttyS1`

Step 3. In microcom issue an AT Command.

Issue: `AT+CGMI`

Response: Telit

Note: Refer to Telit AT command guide.

7.3 Test Plug-In Terminal AT Command Interface (ttyUSB0):

The CDMA and UMTS Plug-In terminals have an AT Command port exposed via USB interface. The following instructions show how to send AT commands with the Microcom tool.

Step 1. Run modem_use_usb script in Linux console.

This script is used to enable the VBUS supply to the Plug-In terminal.

usr/local/bin/modem_use_usb script:

```
echo "1" >/sys/class/gpio/gpio99/value
```

Step 2. After the Plug-In terminal is powered the USB driver will detect and initialize the following devices.

AT Command Port: `/dev/ttyUSB0`

GPS NMEA Port: `/dev/ttyUSB1`

Trace Port: `/dev/ttyUSB2`

Step 3. To create the USB devices issue the following commands.

CDMA and UMTS models only: `modprobe option`

HSPA model only: `modprobe cdc-acm`

Scan /sys and populate /dev: `mdev -s`

Note: These commands need to be executed after every boot.

Step 4. Launch microcom terminal emulator on 400AP Console port.

Issue: `microcom -s 115200 /dev/ttyUSB0`

Step 5. In microcom issue an AT Command.

Issue: `AT+CGMI`

Response: Telit

7 Peripheral Initialization continued

7.4 Initialize RS-485 Interface:

The RS-485 interface is connected to ttyS3, which is configured for RS-232 mode when the 400AP boots. In order to change mode it is required to change the UART Mode Register at address 0xFFFFB8004.

Step 1. Open a Terminal Emulator application. Login as root, password is janus.

Note: Refer to Terminal Emulator for use with the 400AP in this user guide.

Step 2. Issue: DEVMEM 0xFFFFB8004 w 0xC00008C1

Step 3. To create ttyS3-ttyS5 devices issue: "mdev -s".

7.5 Initialize CANBUS Interface:

The CANBUS interfaces are configured during boot. In order to transmit via CAN0 and CAN1 network interfaces, you need to enable the drivers.

Step 1. Open a Terminal Emulator application. Login as root, password is janus.

Note: Refer to Terminal Emulator for use with the 400AP in this user guide.

Step 2. Enable CAN0 and CAN1 Drivers.

Issue: `echo77>/sys/class/gpio/export`

Issue: `echo"0">/sys/class/gpio/gpio77/value`

Step 3. List Network Devices.

Issue: `ifconfig`

7.6 Set RTC Time:

The RTC interface is configured during boot. The default build of the 400AP has the timezone set as UTC. To change the timezone, please edit the "/etc/TZ" file. The following instruction will guide you through setting Linux system time and saving to the hardware RTC chipset.

Step 1. Scan /sys and populate /dev

Issue: `mdev -s`

Step 2. Set system time.

Issue: `date MMDDHHMMYYYY`

Example: `date 091716002012`
`Mon Sept 17 16:00:00 UTC 2012`

Step 3. Set hardware clock from system time.

Issue: `hwclock -w`

Step 4. Verify hardware clock has been updated.

Issue: `hwclock -r`

7.7 Mount SD Card:

The SD Card needs the driver loaded and the memory device mounted before use. The following procedures describe the process.

Step 1. Load Atmel MCI driver

Issue: `modprobe atmel-mci`

Step 2. Scan /sys and populate /dev

Issue: `mdev -s`

Step 3. Mount memory device

Issue: `mount /dev/mmcblk0p1 /mnt/sdcard`

Note: This example uses /mnt/sdcard as the mount point. If you don't have a directory /mnt/sdcard you will need to issue "mkdir /mnt/sdcard" or choose a different mount point.

8 400AP Buildroot Toolchain Installation.

8.1 Overview

The following section describes the process of installing the Buildroot toolchain for use with the 400AP. The Buildroot toolchain creates the Linux Kernel and filesystem images used to run the Linux OS on the 400AP. The Buildroot toolchain requires a PC running a Linux OS and an Internet connection.

8.2 Downloading Buildroot:

The Buildroot website should be referenced for all source code and documentation.

<http://buildroot.uclibc.org/>

Step 1. Browse to the Janus-RC website and download the 400AP buildroot setup archive. The archive includes a setup script, buildroot toolchain and patches required for use with the 400AP.

http://www.janus-rc.com/400ap_downloads.html

Download the Buildroot toolchain setup archive that references the first two characters of your products serial number. The 400AP has its serial number listed on the product label.

Step 2. Extract the archive into a suitable working directory with at least 3GB of storage space available.

Example: `tar -xjvf 400ap_demo_buildroot buildroot-xxxx.xx.tar.bz2`

8.3 Installing Buildroot toolchain:

The following instructions will configure and compile the Buildroot toolchain for use with the 400AP. After the make process finishes a Linux Kernel and UBI filesystem will be created for use on the 400AP. The images created are identical to the demonstration images loaded during production of the 400AP

Step 1. Execute the following command to install toolchain.

```
./400ap_demo_setup
```

Note

- 1. This process requires approximately 3GB of storage.**
- 2. This script should only be run one time during toolchain installation.**
- 3. Script will create a directory named buildroot-xxxx.xx as referenced in the tar file name. Please insure there is no existing directory with this name in your working directory.**
- 4. This step takes some time to complete. The time will depend on the speed of your PC and Internet connection. If your Linux PC is missing any required package dependencies the make process will fail and report the missing package. Install missing packages. Then issue 'make' to continue the setup process.**

8.4 Configuring toolchain:

Once the toolchain has been installed, you might need to add packages or driver support into the toolchain or Linux Kernel. The following instruction should be used to modify the Buildroot tool chain.

To configure the Buildroot toolchain: `make menuconfig`

To configure the Linux Kernel: `make linux26-menuconfig`

To configure busybox: `make busybox-menuconfig`

After making configuration changes, you will need to issue 'make' in order to generate new Kernel and filesystem images.

8.5 Linux Kernel and UBI filesystem:

After the make process has completed a Linux Kernel and UBI filesystem image will need to be uploaded to the 400AP.

Kernel and file system image location: `/path/to/buildroot/output/images`

Reference Uploading Firmware Images to NAND Flash in this user guide for instructions to upload the Linux Kernel and UBI filesystem.

9 Atmel SAM-BA Application Installation

Support References:

Atmel SAM-BA Application:

<http://www.atmel.com/tools/atmelsam-bain-systemprogrammer.aspx>

Linux4sam Tools Support:

<http://www.at91.com/linux4sam/bin/view/Linux4SAM/SoftwareTools>

SAM-BA:

SAM-BA is an application written and supplied by Atmel to allow the upload of code into the connected memory of the SAM family of Atmel processors. Janus uses SAM-BA to upload the following code into Flash.

AT91 Bootstrap

UBOOT

Linux Kernel

Linux Filesystem

SAM-BA Installation:

Use the URL listed above the register and download SAM-BA from Atmel. Once downloaded follow the installation directions supplied by Atmel.

SAM-BA Required User Modifications:

In order to use SAM-BA with the 400AP you will need to manually alter the following tcl file. System administrative rights might be required to alter the tcl file.

```
/path/to/sam-ba/tcl_libs/at91sam9g20-ek/at91sam9g20-ek.tcl
```

Alter the following parameter within the at91sam9g20-ek.tcl file:

```
variable extRamVdd 1
```

```
variable extRamDataBusWidth 16
```

After these these modifications are saved SAM-BA is ready to use with the 400AP.

10 Uploading Firmware Images to NAND Flash in the Terminus 400AP

10.1 Overview

This section describes the process of uploading the required firmware images into flash. The following instructions include illustrations from the Linux version of SAM-BA. If you are using the Windows version of SAM-BA the illustrations might be slightly different. You will need to open a Console connection to the front DB9 serial port. Refer to Getting Started - Serial Console Session section of the user guide.

The following images are required to run the Linux OS on the 400AP:

- Bootstrap
- U-Boot
- Linux Kernel
- UBI filesystem



You must use Bootstrap and U-Boot images supplied by Janus. This software has been customized to work with the 400AP.

The 400AP comes with these images pre-loaded in flash during production. If these images have been erased from flash memory, replacement images are located on the Janus website under 400AP downloads.

http://www.janus-rc.com/400ap_downloads.html

The Linux Kernel and UBI filesystem images are generated from the Buildroot toolchain. To install the Buildroot toolchain refer to Buildroot Toolchain Installation section of this user guide.



You don't need to install the Buildroot toolchain if you are using firmware images from the Janus download page. Install the Buildroot toolchain if you need to add packages or drivers not included in images from download page.

Atmel's SAM-BA application is used to write firmware images to flash memory of the 400AP. To install the SAM-BA application refer to Atmel SAM-BA Application Installation section of this user guide.

10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

10.2 Configuring 400AP RomBOOT Mode:

The 400AP must be configured in RomBOOT mode in order to communicate with SAM-BA.

The following instructions assume the custom version of AT91bootstrap for the 400AP is loaded into flash. This version of AT91bootstrap evaluates the state of the Mode button and vectors to RomBOOT if the Mode button is pressed for some time during boot.

If no bootable code is found after the 400AP is powered or reset, RomBOOT mode will be entered automatically. Under this condition the only indication of RomBOOT mode is that "RomBOOT" is transmitted from the 400AP console port.

Step 1. Power the 400AP.

Step 2. Press and hold the Mode button.

Step 3. Press and release the Reset button while continuing to press the Mode button.

Step 4. Release Mode button after green LED illuminates.

Step 5. The 400AP will be in RomBOOT mode when the green LED illuminates.

RomBOOT will be transmitted from the Console port if you are in RomBOOT mode.

Note 1: Under some circumstances the preceding instructions can fail depending on the state of the following files: Bootstrap, U-Boot.

If RomBOOT is not displayed follow these alternate instructions.

Alt. 1.) Press and hold the Mode button.

Alt. 2.) Power cycle the 400AP.

Alt. 3.) Release Mode button after green LED illuminates.

Note 2: If the 400AP during boot vectors to Bootstrap and the code is corrupt you might not be able to enter RomBOOT mode. If this occurs refer to the NAND Flash recovery section.

Step 6. Connect a USB cable to the USB device port of the 400AP. The other end of the USB cable connects to a USB host port of your PC running the SAM-BA application.

The following should display in the terminal emulator: ">"

The ">" prompt indicates that the 400AP is in RomBOOT mode and the driver has been initialized by the host PC.

If the ">" prompt doesn't display when the USB cable is connected to the host PC then please check the following URLs for SAM-BA support.

<http://www.at91.com/linux4sam/bin/view/Linux4SAM/SoftwareTools>

<http://www.atmel.com/tools/ATMELSAM-BAIN-SYSTEMPROGRAMMER.aspx?>

10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

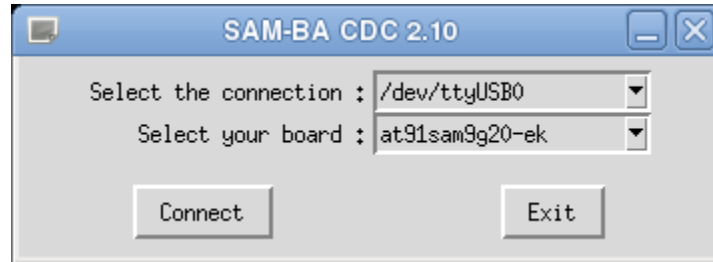
10.3 Launching SAM-BA Application:

Step 1. Configure 400AP for RomBOOT mode.

*Note: Refer to **Configuring 400AP RomBOOT Mode** section of this user guide.*

Step 2. Launch SAM-BA application.

Step 3. When prompted configure the following options and then select Connect.

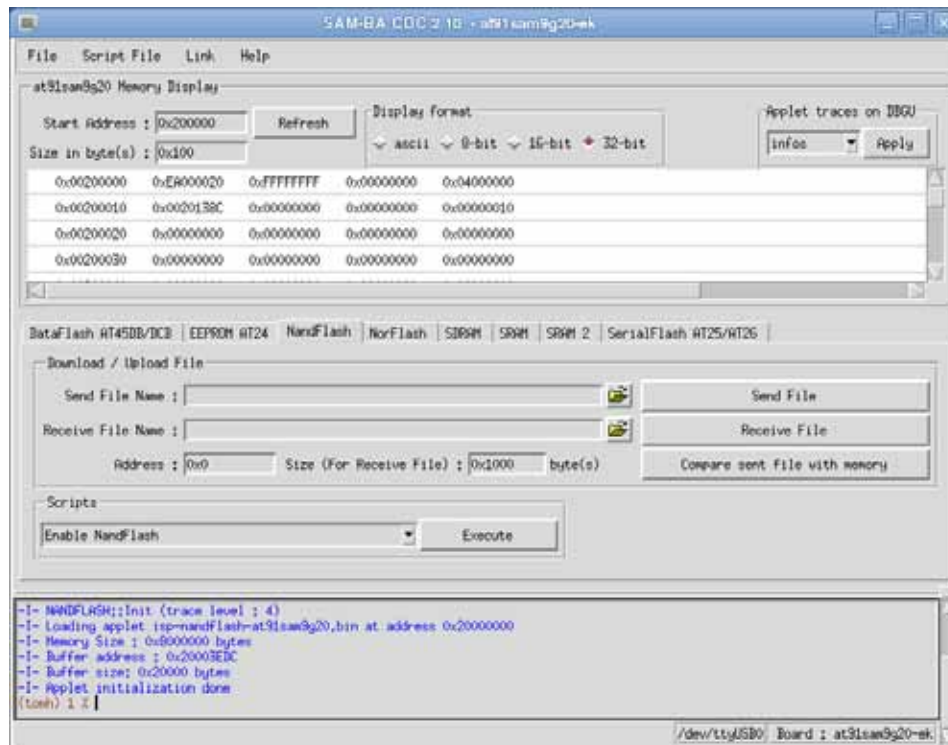


Step 4. To program the Flash memory you need to select Enable NandFlash.

To enable flash select the **Execute** button.



NAND Flash needs to be enabled every time you launch the SAM-BA application.



10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

10.4 Writing Bootstrap Image to NAND Flash Memory:

Step 1. Launch SAM-BA application.

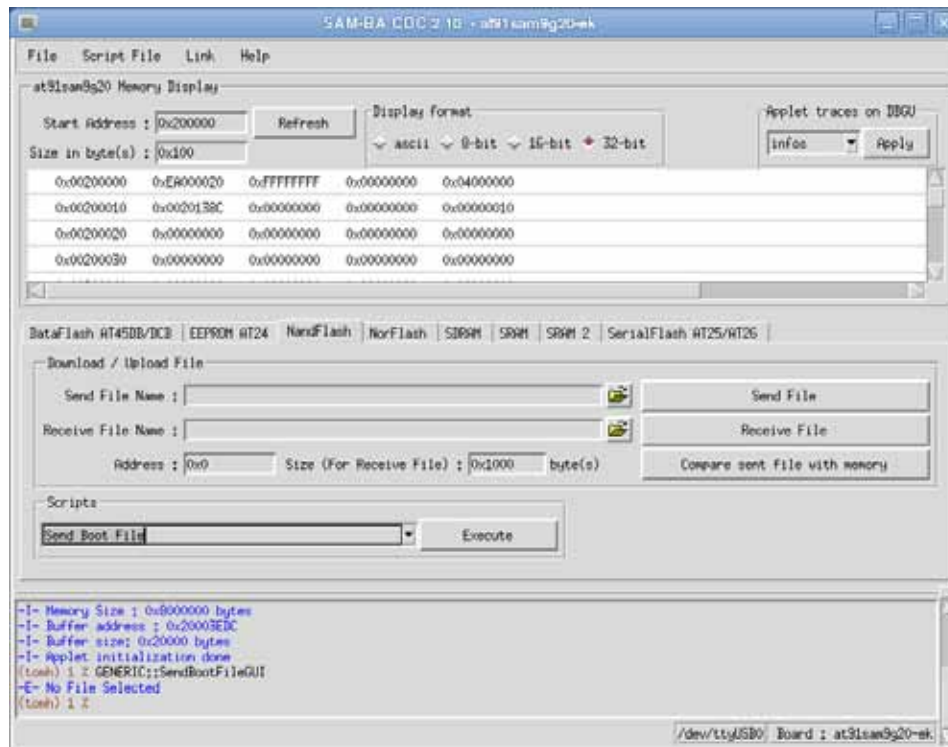
Note: Refer to Launching SAM-BA Application section in this user guide.



Remember to Enable NandFlash!

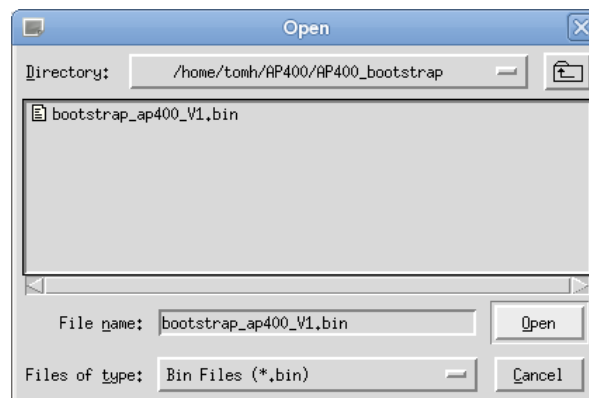
Step 2. To program to 400AP Bootstrap image you need to select Send Boot File.

Select the **Execute** button.



When prompted select the Bootstrap image.

Select **Open** button to write the Bootstrap image to flash memory.



10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

10.5 Writing U-Boot Image to NAND Flash Memory:


Step 1. Launch SAM-BA application.

Note: Refer to Launching SAM-BA Application section in this user guide.



Remember to Enable NandFlash!

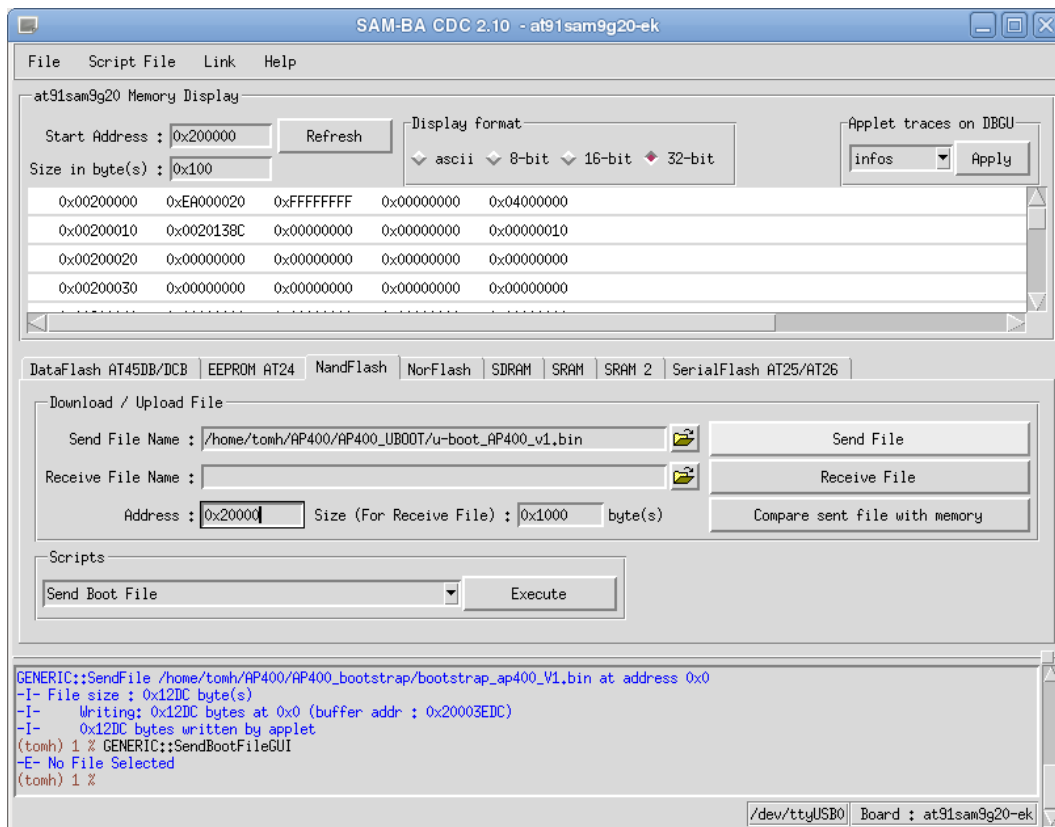
Step 2. Enter 0x20000 in the Address field.

Click the  icon next to the **Send File Name** field.

When prompted select the U-boot image and select **Open**.

To write the U-boot image to flash select **Send File**.

Verify image has been written correctly by selecting **Compare sent file with memory**.



10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

10.5 Writing U-Boot Image to NAND Flash Memory continued:

- Step 3.** After writing the U-Boot image to flash memory you will need to update the boot arguments. Open a Terminal Emulator application and reset 400AP.

Note: Refer to Terminal Emulator for use with the 400AP section in this user guide.

- Step 4.** Exit out of SAM-BA application.

- Step 5.** When prompted by U-Boot, exit out of autoboot.

Hit any key to stop autoboot:

The following prompt should be displayed:

U-Boot>

- Step 6.** The following commands are for the 400AP using a UBI File System image.

At the U-Boot prompt enter the following U-Boot commands:

```
setenv bootcmd nand read 0x22000000 0x00200000 0x00200000\; bootm 0x22000000
```

```
setenv bootargs mem=64M console=ttyS0,115200 ubi.mtd=1 root=ubi0:rootfs rw rootfstype=ubifs
```

```
setenv ethaddr xx:xx:xx:xx:xx:xx
```

```
saveenv
```

```
boot
```

Note: *xx:xx:xx:xx:xx:xx should be substituted with the MAC address listed on the 400AP product label.*

10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

10.6 Writing Linux Kernel Image to NAND Flash Memory:

Step 1. If the flash memory location where the Linux Kernel is to be written is blank proceed to Step 4.

You will need to erase the flash memory location where the Linux Kernel is going to be written. Open a Terminal Emulator application and reset 400AP.

*Note: Refer to **Terminal Emulator for use with the 400AP** section in this user guide.*

Step 2. When prompted by U-Boot, exit out of autoboot.

Hit any key to stop autoboot:

The following prompt should be displayed:

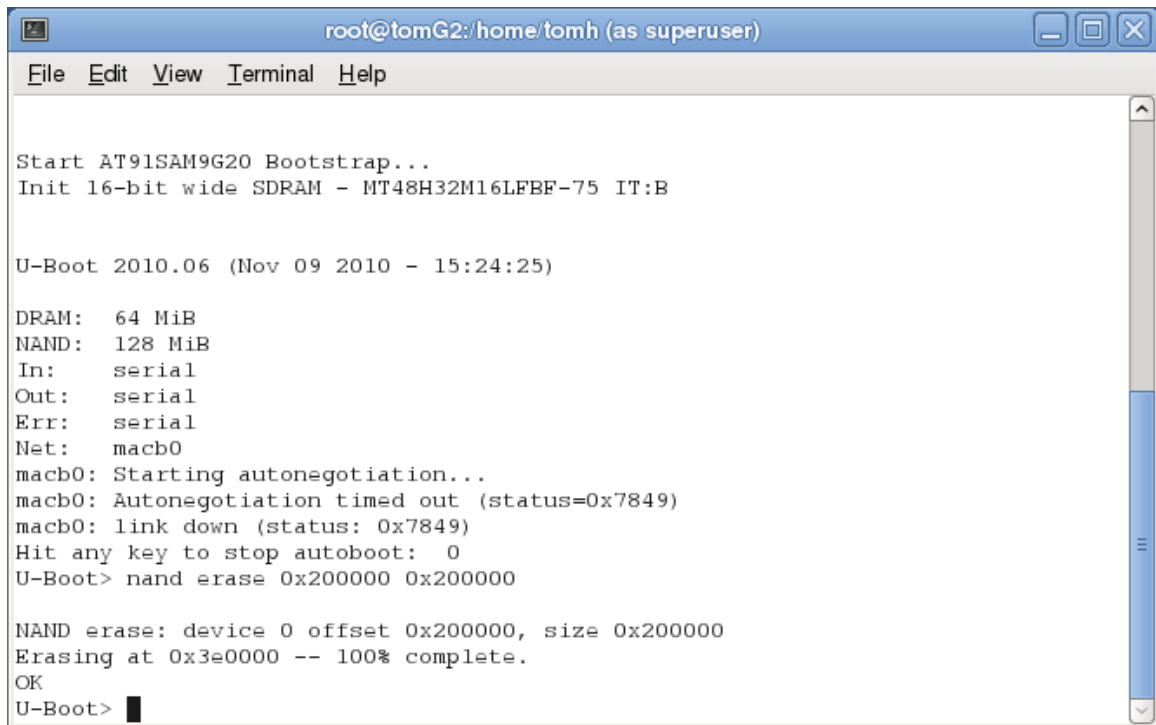
U-Boot>

Step 3. At the U-Boot prompt enter the following command.

nand erase 0x200000 0x200000

Note: If you are also going to write the File System image then issue the following command to erase memory for Kernel and File System at once.

nand erase 0x200000



```
root@tomG2:/home/tomh (as superuser)
File Edit View Terminal Help

Start AT91SAM9G20 Bootstrap...
Init 16-bit wide SDRAM - MT48H32M16LFBF-75 IT:B

U-Boot 2010.06 (Nov 09 2010 - 15:24:25)

DRAM: 64 MiB
NAND: 128 MiB
In: serial
Out: serial
Err: serial
Net: macb0
macb0: Starting autonegotiation...
macb0: Autonegotiation timed out (status=0x7849)
macb0: link down (status: 0x7849)
Hit any key to stop autoboot: 0
U-Boot> nand erase 0x200000 0x200000

NAND erase: device 0 offset 0x200000, size 0x200000
Erasing at 0x3e0000 -- 100% complete.
OK
U-Boot> █
```

10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

10.6 Writing Linux Kernel Image to NAND Flash Memory continued:

Step 4. Launch the SAM-BA application.

Note: Refer to Launching SAM-BA Application section in this user guide.



Remember to Enable NandFlash!

Step 5. Enter 0x200000 in the Address field.

Click the  icon next to the Send File Name field.

When prompted select Linux Kernel image and select Open.

Note: If using the Buildroot toolchain as configured for the 400AP the Kernel image will be named, ulmage. You will not be able select this image from within SAM-BA application without renaming the file with a file extension.

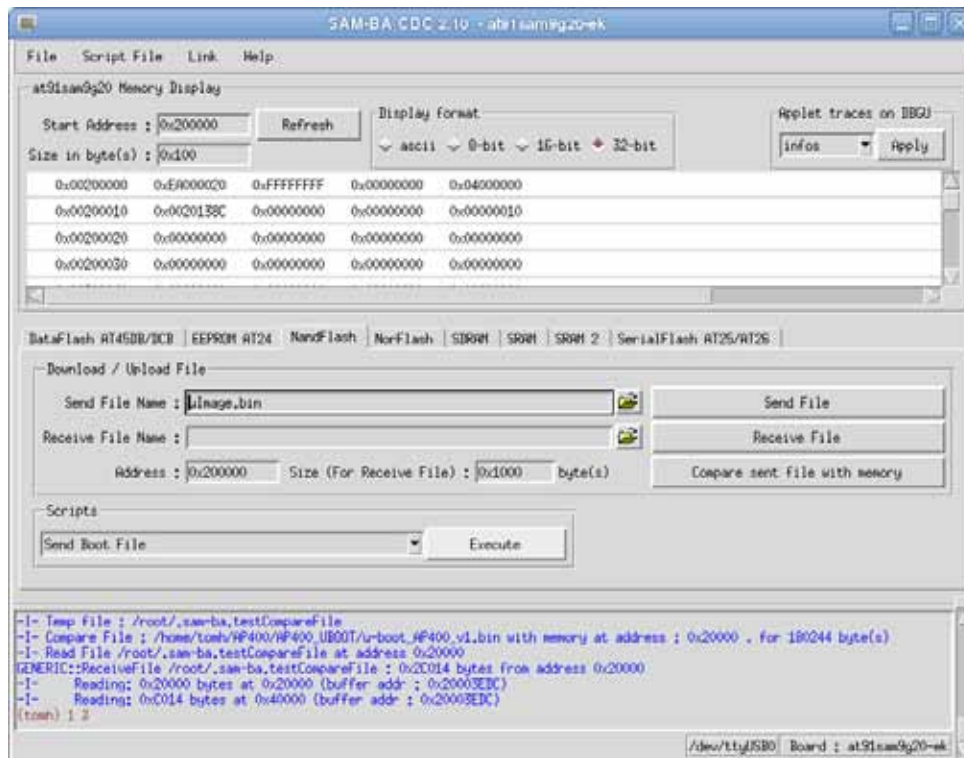
Example: Rename file to ulmage.bin

Another option is to create a symbolic link for ulmage.bin that references ulmage.

Example: `~/buildroot/output/images $ ln -s ulmage ulmage.bin`

To write Linux Kernel to flash select Send File.

Verify image has been written correctly by selecting Compare sent file with memory.



10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

10.7 Writing File System Image to NAND Flash Memory:

- Step 1.** If the flash memory location where the File System is to be written is blank proceed to Step 4. You will need to erase the flash memory location where the File System is going to be written. Open a Terminal Emulator application and reset 400AP.

Note: Refer to Terminal Emulator for use with the 400AP section in this user guide.

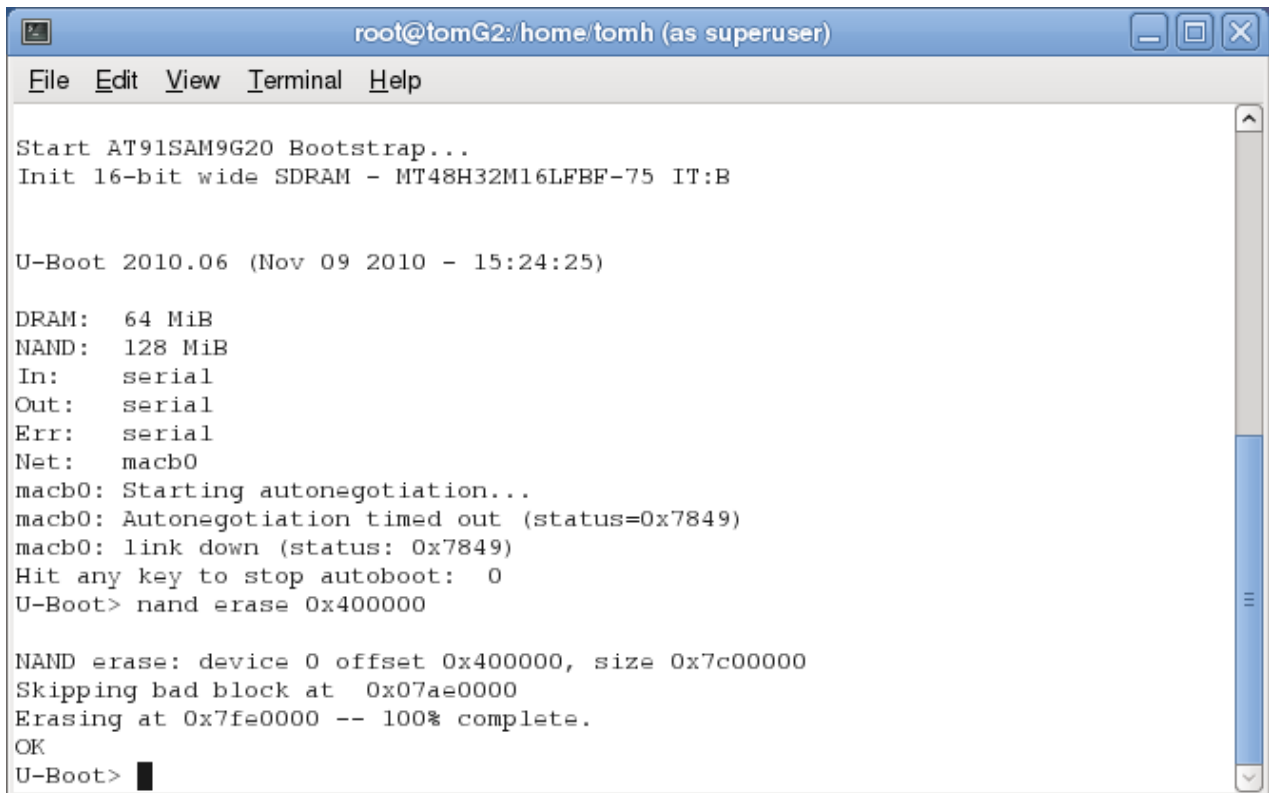
- Step 2.** When prompted by U-Boot, exit out of autoboot.

Hit any key to stop autoboot:

The following prompt should be displayed: U-Boot>

- Step 3.** At the U-Boot prompt enter the following command.

nand erase 0x400000



```
root@tomG2:/home/tomh (as superuser)
File Edit View Terminal Help
Start AT91SAM9G20 Bootstrap...
Init 16-bit wide SDRAM - MT48H32M16LFBF-75 IT:B

U-Boot 2010.06 (Nov 09 2010 - 15:24:25)

DRAM: 64 MiB
NAND: 128 MiB
In: serial
Out: serial
Err: serial
Net: macb0
macb0: Starting autonegotiation...
macb0: Autonegotiation timed out (status=0x7849)
macb0: link down (status: 0x7849)
Hit any key to stop autoboot: 0
U-Boot> nand erase 0x400000

NAND erase: device 0 offset 0x400000, size 0x7c00000
Skipping bad block at 0x07ae0000
Erasing at 0x7fe0000 -- 100% complete.
OK
U-Boot> █
```

10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

10.7 Writing File System Image to NAND Flash Memory continued:


Step 4. Launch the SAM-BA application.

Note: Refer to *Launching SAM-BA Application* section in this user guide.



Remember to Enable NandFlash!

Step 5. Enter 0x400000 in the Address field.

Click the  icon next to the **Send File Name** field.

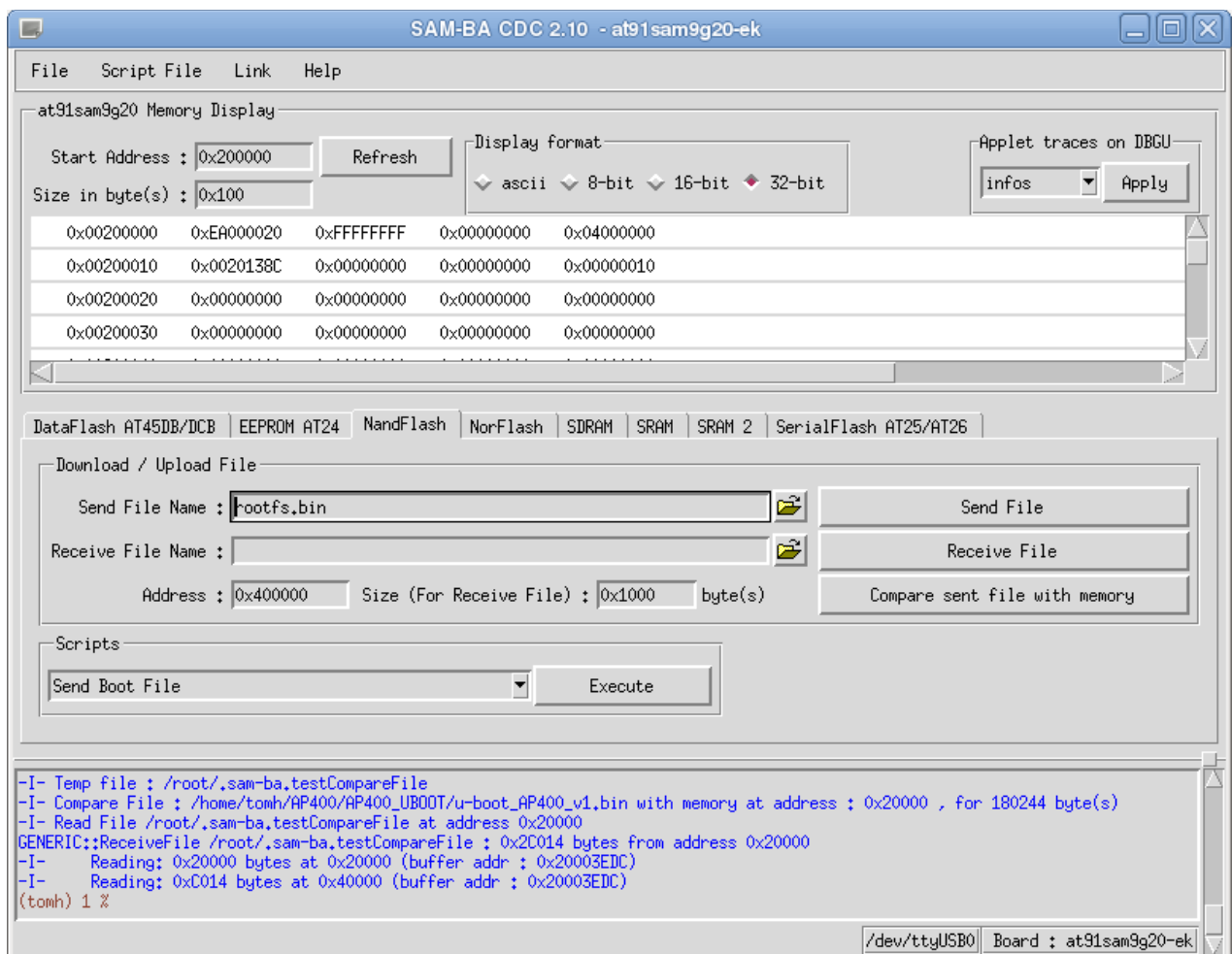
When prompted select File System image and select **Open**.

Note: If using the Buildroot toolchain as configured for the 400AP the filesystem will be named, **rootfs.ubi**.

To write the File System to flash select **Send File**.

Verify image has been written correctly by selecting

Compare sent file with memory.



SAM-BA CDC 2.10 - at91sam9g20-ek

File Script File Link Help

at91sam9g20 Memory Display

Start Address : 0x200000 Refresh Display format: ascii 8-bit 16-bit 32-bit

Size in byte(s) : 0x100 Applet traces on DBGU: infos Apply

0x00200000	0xEA000020	0xFFFFFFFF	0x00000000	0x04000000
0x00200010	0x0020138C	0x00000000	0x00000000	0x00000010
0x00200020	0x00000000	0x00000000	0x00000000	0x00000000
0x00200030	0x00000000	0x00000000	0x00000000	0x00000000

DataFlash AT45DB/DCB EEPROM AT24 **NandFlash** NorFlash SDRAM SRAM SRAM 2 SerialFlash AT25/AT26

Download / Upload File

Send File Name : rootfs.bin Send File

Receive File Name : Receive File

Address : 0x400000 Size (For Receive File) : 0x1000 byte(s) Compare sent file with memory

Scripts

Send Boot File Execute

```
-I- Temp file : /root/.sam-ba.testCompareFile
-I- Compare File : /home/tomh/AP400/AP400_UBOOT/u-boot_AP400_v1.bin with memory at address : 0x20000 , for 180244 byte(s)
-I- Read File /root/.sam-ba.testCompareFile at address 0x20000
GENERIC::ReceiveFile /root/.sam-ba.testCompareFile : 0x20014 bytes from address 0x20000
-I- Reading: 0x20000 bytes at 0x20000 (buffer addr : 0x20003EDC)
-I- Reading: 0xC014 bytes at 0x40000 (buffer addr : 0x20003EDC)
(tomh) 1 %
```

/dev/ttyUSB0 Board : at91sam9g20-ek

10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

10.8 NAND Flash Recovery

The Atmel ARM9 is configured on boot to execute its RomBOOT code. This RomBOOT code interrogates connected memory to detect an application to execute. If no application is detected the RomBOOT code enters RomBOOT mode and allows the Atmel SAM-BA application to access connected memory in order to upload firmware images. The 400AP has all application code stored in the NAND flash. The first location in Flash has a custom Bootstrap application that controls low-level hardware initialization. After Bootstrap has initialized the ARM9 hardware it detects the state of the external Mode button of the 400AP. If during this check the Mode button is pressed the green LED will illuminate and Bootstrap will vector to RomBOOT mode. If for any reason Bootstrap becomes corrupt the Mode button will not be able to interrupt the boot process and RomBOOT mode will not be entered. In this bricked condition firmware can only be reloaded with use of Atmel SAM-ICE emulator pod and the SAM-BA software.

The following directions are based on the Windows version of the Atmel SAM-BA application. The following emulator pod is required during the NAND flash recovery instructions.

Atmel Part: AT91SAM-ICE

User Guide: http://www.atmel.com/dyn/resources/prod_documents/doc6206.pdf

Follow the SAM-ICE user guide for how to install the Windows drivers.

Once the drivers are installed follow these directions to erase NAND flash contents.

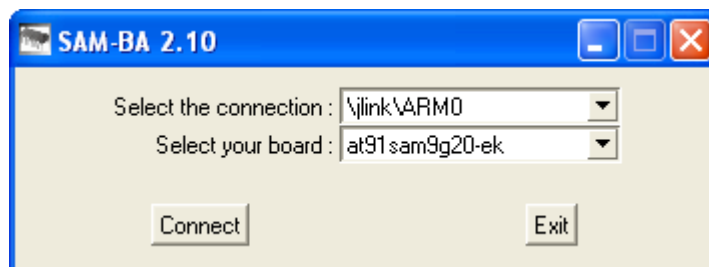
Step 1. Connect the SAM-ICE emulator pod to your PC and the 20-pin ribbon cable to the 400AP JTAG port (P5).

Step 2. Power the 400AP.

Step 3. Launch the SAM-BA application.

Note: If you are using an older version of SAM-BA do not use the SAM-BA Vx.xx_cdc version of the software. You might also consider upgrading to a newer version of SAM-BA.

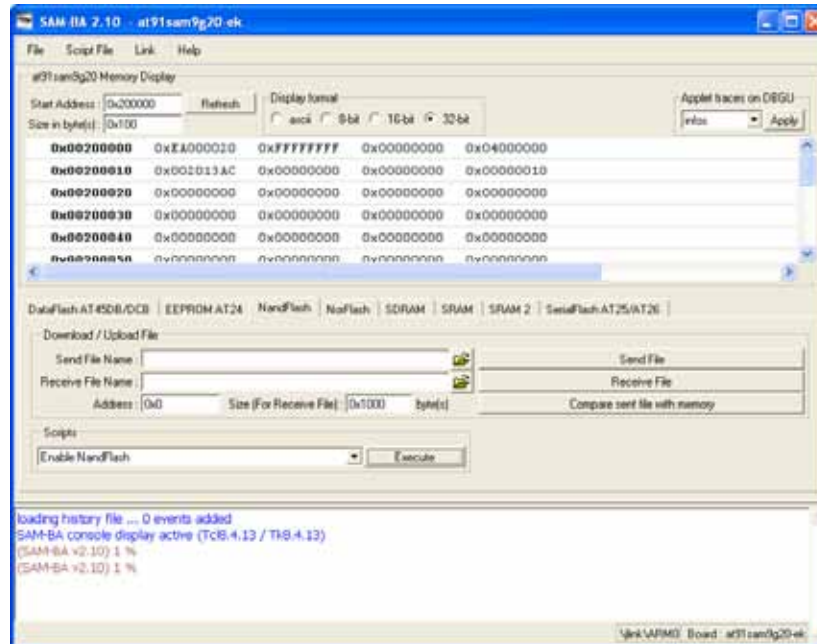
Step 4. When prompted make the following settings and select Connect.



10 Uploading Firmware Images to NAND Flash in the Terminus 400AP continued

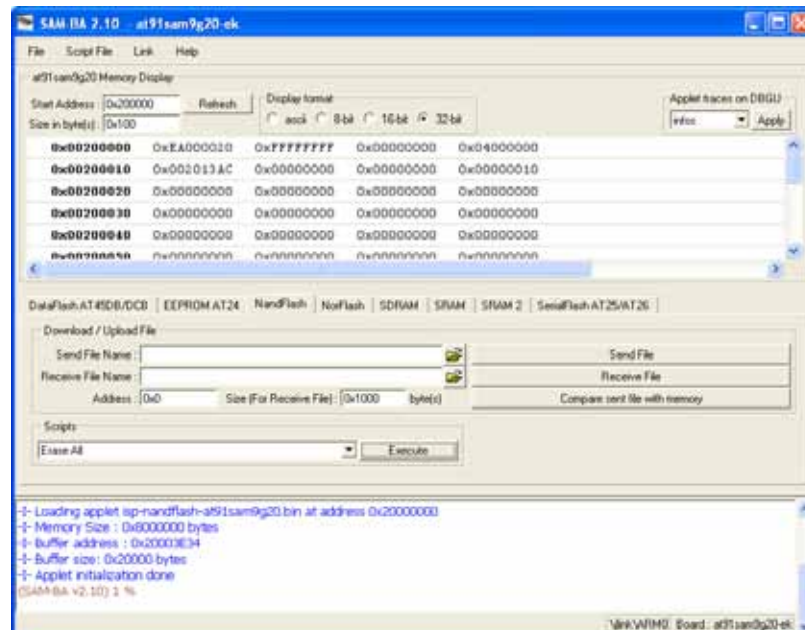
10.8 NAND Flash Recovery continued

Step 5. Select Enable NandFlash script and select Execute



Step 6. Select Erase All script and select Execute.

 **Do not select Scrub NandFlash script.**



Step 7. Remove power from 400AP then disconnect the SAM-ICE pod from 400AP.

At this point you will need to follow the directions in this application note to restore the following firmware.

Bootstrap

U-Boot

Linux Kernel

UBI filesystem

Terminus 400AP Linux OS User Guide



Revision History

Revision	Revision Date	Note
00	11/08/12	Released Linux OS User Guide



Division of The Connor-Winfield Corporation
2111 Comprehensive Drive • Aurora, Illinois 60505
630.499.2121 • Fax: 630.851.5040

www.janus-rc.com

Janus Remote Communications Europe
Bay 143
Shannon Industrial Estate
Shannon, Co. Clare, Ireland
Phone: +353 61 475 666