sensor networks

Stargate Developer's Guide

Rev. B, January 2006

Document 7430-0317-13

Crossbow

Table of Contents

1   Introduction

Stargate is a powerful single board computer with enhanced communications and sensor signal processing capabilities. The Stargate uses Intel's latest generation 400 MHz XScale® processor (PXA255). This product was designed within Intel's Ubiquitous Computing Research Program, and licensed to Crossbow for production. In addition to traditional single board computer applications, the Stargate directly supports applications around Intel's Open-Source Robotics initiative as well as TinyOS based Wireless Sensor Networks and Smart Dust Technology.

1.1   Stargate Features
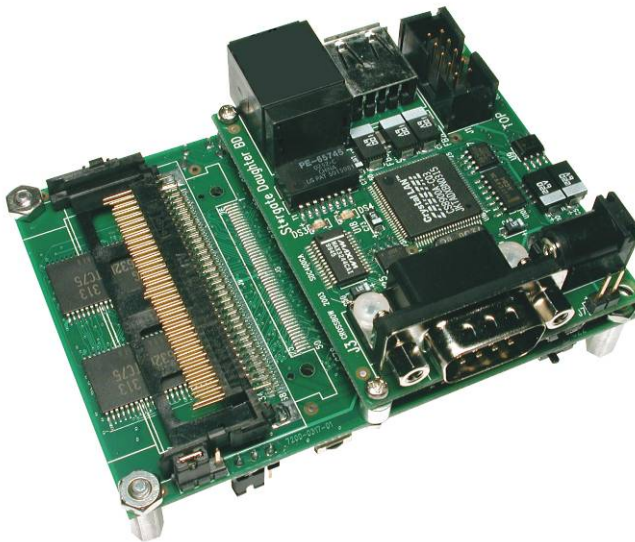
Figure 1. *Stargate Development Platform (Processor Board and Daughter Card)*

- Small form factor (3.5" × 2.5")
- 32-bit, 400 MHz Intel PXA255 XScale RISC processor.
- SA1111 StrongARM companion chip for multiple I/O access.
- 32 MB of Intel StrataFlash.
- 64 MB of SDRAM.
- 1 Type II CompactFlash slot.
- 1 PCMCIA lot
- Reset button
- Real time clock
- Lithium ion battery option
- MICA2 and MICAz Mote capability, GPIO/SSP and other signals via 51-pin expansion connector
- I2C connector via an installable header
- 51-pin daughter card interface for:
    o   Wired Ethernet via a 10 Base-T Ethernet port
    o   Host USB
    o   JTAG port
    o   External A/C power supply adapter
    o   RS-232 serial port via DB-9 connector

1.2   Software Environment

The Stargate Development Platform is shipped with pre-installed software on the board and additional software on the CDROM to enable application program development.

CDROM contents:

- An embedded Linux operating system (OS) kernel.

- A Linux board support package and file system.

- Additional drivers in source code with instructions on how to build and install them.

- A Bootloader for initial loading of the kernel and file system images.

- A flash programmer utility for programming the flash ROM.

- GNU cross platform development tools.

- System configuration support file archives.

1.3   Potential Applications

The Stargate processor board has a wide variety of applications such as:

- A single-board computer running embedded Linux OS

- Sensor network gateway

- Customizable 802.11a/b wireless gateway

- Cellular wireless gateway

- Robotics controller card

- Distributed computing platform

- Embedded sensor signal processing

2 Getting Started

2.1 Development Kit Overview

2.1.1 The Basic Stargate Development Kit

This basic kit contains the following items:

- The Stargate processor board
- The Stargate daughter card
- A null modem cable for serial RS-232 connection
- A CDROM
- A wall power DC adaptor supply
- The Stargate Developer's Manual
- Factory installed Linux image in flash with a number of driver modules for common devices

2.1.2 The Advanced Stargate Development Kit

This advanced kit contains the following items in addition to items in the basic Kit:

- A USB webcam
- A 802.11b wireless compact flash card

2.2   Hardware Layouts

Hardware Reset Button (SW3)

Power Switch (S1)

Compact Flash Card Slot

51-pin Mote, I/O Connector

Software Reset Button (S2)

Figure 2. *SPB400CB Processor Board (Top View)*

PCMCIA Slot

Daughter Card Connector

Li-Ion Battery Connector

Figure 3. *SPB400CB Processor Board (Bottom View)*
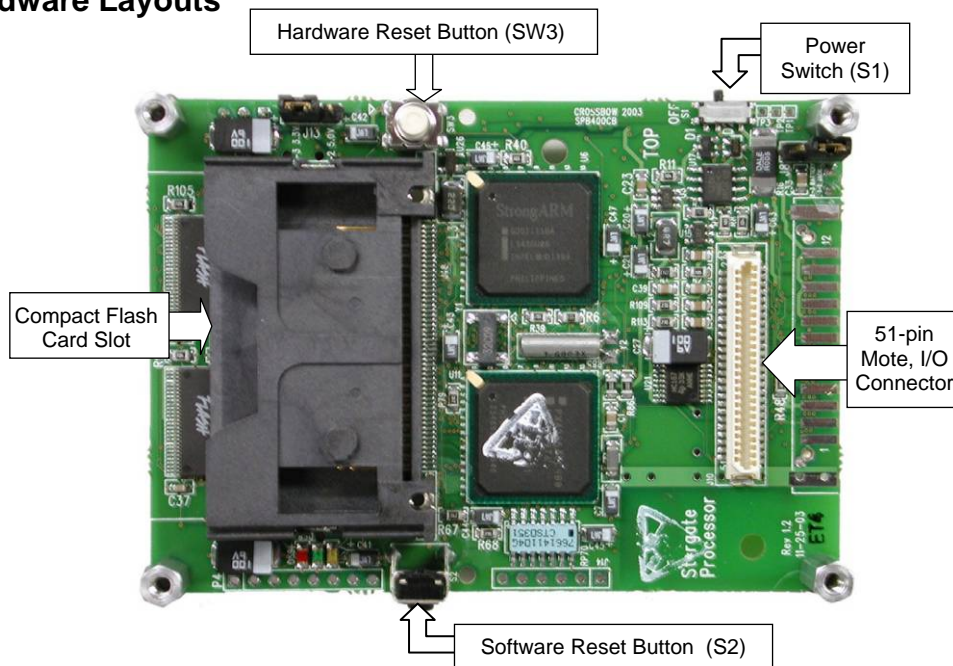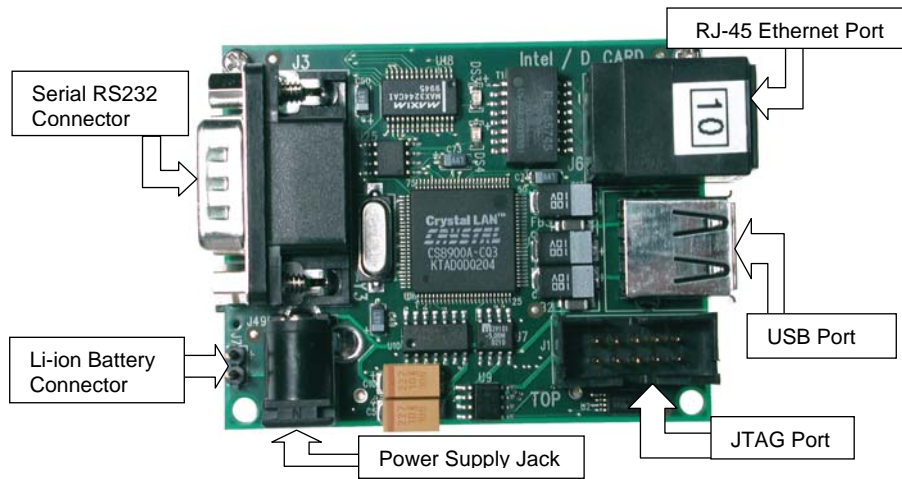
Figure 4. *SDC400CA Daughter Card (Top View)*
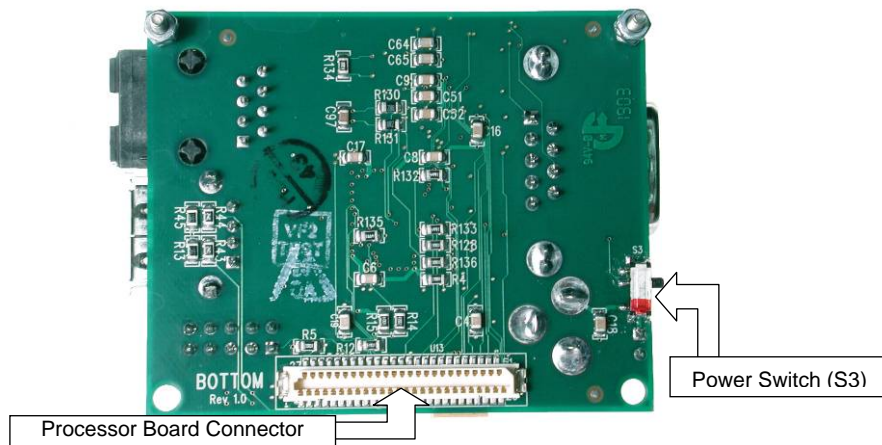
Figure 5. *SDC400CA Daughter Card (Bottom View)*

ESD WARNING: The Stargate Processor Board and Daughter Card are electronic devices and therefore can be damaged by Electrostatic Discharge (ESD). While handling these devices, follow ESD prevention procedures and use an ESD wrist strap.

2.3 Assembling the Stargate Development Platform

Connect the Stargate Processor Board (SPB400CB) with the Stargate Daughter Card (SDC400CA) by connecting the bottom of Stargate Processor Board (connector **U1**) to the bottom of Stargate Daughter Card (connector **U13**).

Fasten the 2 boards using 2 nylon spacers, screws and nuts provided in a plastic bag.

2.4 Verify successful setup on terminal console

To configure and boot the Stargate platform, complete the following steps:

1. Connect the Stargate to the host machine (a Linux or Windows PC): Section 2.4.1.

2. Configure and start a terminal emulation program on the host PC: For Linux hosts see Section 2.4.2; for Windows hosts see Section 2.4.3.

3. Plug in the power to the Stargate and turn the two power switches (S1 and S3) to the "ON" position: Section 2.4.4.

4. Recognize the boot prompt and boot the Linux Operating System.

2.4.1 Connecting Stargate to Host

Before you can begin, you need to connect your target board to your host machine (a Linux or Windows PC). To connect your target board to the host, attach a null modem serial cable (provided with your Development Kit) between **J3** of the target and an available serial port on the development host.

2.4.2 Configuring *Minicom* (for Linux Hosts)

Minicom is a terminal emulation and modem interface program included in most Linux distributions. It is used to communicate with the Stargate board.

To configure Minicom, complete the following steps:

1. Log in as **root** on your Linux host machine.

2. Type **minicom -s**

   The *Minicom* configuration menu appears.

3. Select **Serial port setup**. Configure *Minicom* to use the tty port that is connected to the target, for example /dev/ttyS0 (com1), **115200** baud, **8** data bits, **No** parity and **1** stop bit and flow control set to **None**.

```
A - Serial Device          : /dev/ttyS0
B - Lockfile Location       : /var/lock
C - Callin Program          :
D - Callout Program         :
E - Bps/Par/Bits            : 115200 8N1
F - Hardware Flow Control : No

G - Software Flow Control : No


              Change which setting?
```

4. Press **Enter** to return to the main configuration menu.

5. Select **Save setup** as **dfl** to save these as default settings.

6. Select **Exit from Minicom**.

You have now successfully configured *Minicom* to access the console port for Stargate.

1. When you need to enter *Minicom* again, you will be able to communicate with your target by using the command **minicom**

2. To exit Minicom, use the command: **Ctrl-AX**

3. You will be prompted to confirm that you want to exit Minicom. Press **Enter** to exit.

4. For help using Minicom, use the command: **Ctrl-AZ**

2.4.3 Configuring *HyperTerminal* (for Windows hosts)

- To start *HyperTerminal*, choose
  Start>Programs>Accessories>Communications>HyperTerminal.

- You will now see the *HyperTerminal* window come up, along with a dialog box for configuring
  a **New Connection**.

- Type **Stargate** in the Name text box and click **OK**.

- The **Connect To** dialog box will then be displayed, select the COM port you wish to use from
  the **Connect using** drop down list, then click **OK**.

- The **COM Properties** dialog box then opens up, choose the following parameters then click
  OK:

Table 1. *COM Port Settings.*

Bits per second	**115200**
Data Bits	**8**
Parity	**0**
Stop bits	**1**
Flow Control	**None**

- You have now successfully configured *HyperTerminal* to access the console port for Stargate.

2.4.4 Powering-Up the Stargate

If you are using an external power supply, you could use the power supply/adapter provided with the
Development kit. Connect the DC input of the AC wall power adapter to the **J49** connecter on the
daughter card.

If you are using a Li-Ion battery pack, there are 2 usage models:

- If both Stargate Processor Board and the Daughter Card are used, then **J8** must be jumpered on
  1-2 position (JACK) and the battery plugged in at the **J7** connector on the Daughter Card.

- If only Stargate Processor Board is used, the **J8** must be jumpered on 2-3 position (BATTERY)
  and the battery plugged in at **J23**.

2.4.5 Boot prompt

Once the target is powered up and properly connected to the host computer, the following message
is seen on the console. This verifies that the Stargate platform has booted up.

```
MAC address: 00:50:C2:2F:33:83FLASH_BASE=00000000
PXA: Evacuating 1MB of Flash to DRAM at: A3E00000
done
Map Flash virtual section to DRAM at: A3E00000
setting flashword=50000000
 Flash style = 2x16
verify flash size...
btflash_init: mfrid=00890089
  devid=00180018
```

```
mfrid=00890089
deviceid=00180018
walking flash descriptors (2x16)
btflash_init: found flash 28F128J3A
  flashDescriptor=0002DEC8
  flashSectors=0002C6A4
  nsectors=00000080
  flash_size=02000000
  flash_address_mask=01FFFFFF
mach_type  ->0000012F
reset_partitions: partition_table = 0x00000000
reset_partitions: flashDescriptor = 0x0002DEC8
define_partition: base 0x00000000
define_partition: partition_table 0x00000000
allocated partition_table
defining partition: bootldr
define_partition: base 0x00040000
define_partition: partition_table 0xA3A00014
defining partition: root
initialize_by_mach_type done
probing ram
Probing bank0 memory size...
 bank0 memory size=04000000
MDCNFG=00001ACB
Probing memory size bank=00000000
maybe mem_size=04000000
found=04000000
Probing memory size bank=00000001
seems to be no dram in bank=00000001
n_banks=00000001
dram_sizes[0]=04000000
dram_sizes[1]=00000000
SDRAM size: 0x04000000
 in megs: 64M
>> Compaq OHH BootLoader ported to Stargate 7.1, Rev 2-21-13 [BIG_KERNEL] [MONO]

>> 04-06-04_15:58
>> Last link date: Fri Jun  4 15:58:22 PDT 2004
>> Contact: bootldr@handhelds.org

Cpu company: INTEL
Cpu Architecture: XScale
Cpu Part: PXA255
>> ARM Processor Rev=00050000
>> (c) 2000-2001 Compaq Computer Corporation, provided with NO WARRANTY under
the terms of the GNU General Public License.
>> See http://www.handhelds.org/bootldr/ for full license and sourcesPress
Return to start the OS now, any other key for monitor menu
found partition table in params sector
pt->npartitions=00000004
partition_table_size=000000B8
define_partition: base 0x00000000
define_partition: partition_table 0xA3A00014
defining partition: bootldr
define_partition: base 0x00040000
define_partition: partition_table 0xA3A00014
defining partition: params
define_partition: base 0x00080000
define_partition: partition_table 0xA3A00014
defining partition: kernel
```

```
define_partition: base 0x00140000
define_partition: partition_table 0xA3A00014
defining partition: root
+ set macaddr "00:50:C2:2F:33:83"
  setting param <macaddr> to value <00:50:C2:2F:33:83>
+ set ipaddr "10.1.1.171"
  setting param <ipaddr> to value <10.1.1.171>
+ set gateway "10.1.1.221"
  setting param <gateway> to value <10.1.1.221>
+ set netmask "255.255.255.0"
  setting param <netmask> to value <255.255.255.0>
+ set serverip "10.1.1.46"
  setting param <serverip> to value <10.1.1.46>
+ set loadmethod "tftp"
  setting param <loadmethod> to value <tftp>
+ set netcfg "manual"
  setting param <netcfg> to value <manual>
+ set boot_flags 0x00000004
  setting param <boot_flags> to value <0x00000004>
+ set enable_mmu 1
  setting param <enable_mmu> to value <1>
```

Now you may press **Enter**, so as to start booting up the Linux OS. If no keys are pressed, after a timeout of about 10 seconds, the Linux OS boots up automatically. Note that the configuration parameters may be different for your board, due to unique assignment of the MAC address. The following message is seen during Linux Bootup:

```
booting flash...
booting kernel from partition >kernel<
kernel partition base 50080000
kernel_magic=E1A00000
kernel_region_words[9]=016F2818
Linux ELF flash_imgstart=50080000 size=000C0000 dest=A0000000 offset=00008000
MMU Control=00000079
MMU PIDVAM=00000000
copying Linux kernel ... done
root_filesystem_name=initrd
Grabbed linuxargs, argc = 00000002
…
…
Starting PCMCIA services: cardmgr.
cardmgr[106]: starting, version is 3.1.22
cardmgr[106]: watching 2 sockets
cardmgr[106]: exiting
Starting BlueZ HCI server: hcid.
Starting OpenBSD Secure Shell server: sshd.
usb.c: registered new driver usbdevfs
usb.c: registered new driver hub
Pre-allocating 32 buffers for pool 0xc3eef5a0
Pre-allocating 32 buffers for pool 0xc3eef4e0
usb-ohci.c: USB OHCI at membase 0xf4000400, IRQ 149
usb.c: new USB bus registered, assigned bus number 1
hub.c: USB hub found
hub.c: 1 port detected
Enabling remote wakeup...
SCSI subsystem driver Revision: 1.00
```

```
Initializing USB Mass Storage driver...
usb.c: registered new driver usb-storage
USB Mass Storage support registered.

stargate login:
```

At this point you are ready to log into the target system.

- For "login:" type **root**.

- For "Password:" type **rootme**.

```
login: root
Password:******

Linux stargate 2.4.19-rmk7-pxa2-star #1 Fri Aug 13 12:58:00 PDT 2004 armv5tel
unknown

login[150]: root login  on `tts/0'


stargate:/root#
```

Now you are ready to use the Stargate platform!

3   System Configuration

3.1   Networking Setup

To use the Stargate effectively it needs to be connected to a network. There are many options available to physically connect the Stargate to a network. The quickest and easiest way is to use the Ethernet port built into the daughter card.

3.1.1   Ethernet LAN - DHCP

The Stargate is pre-configured to connect to a LAN via the Ethernet port. If there is a DHCP service running on the LAN the Stargate will automatically be assigned an IP address, Gateway address and DNS server entries. If the LAN is configured for workstation Internet access (via Gateway and DNS entries) the Stargate will have Internet access.

The disadvantage to DHCP is you need to determine the IP address of the Stargate after it boots up - it could be different each time. Using a terminal program open a Stargate session and issue the following command:

```
$ ifconfig eth0
eth0      Link encap:Ethernet   HWaddr 00:50:C2:2F:3B:31
          inet addr:192.168.2.3  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:39931 errors:0 dropped:14 overruns:0 frame:0
          TX packets:1646 errors:0 dropped:0 overruns:0 carrier:0
          collisions:10 txqueuelen:100
          RX bytes:12777054 (12.1 MiB)  TX bytes:0 (0.0 b)
          Interrupt:26
```

The *ifconfig* command shows the current configuration of the ethernet port: eth0. The response shows the IP address to be 192.168.2.3. You can then connect to the Stargate remotely using this address.

3.1.2   Ethernet LAN – Fixed IP Address

A more convenient way to configure the Stargate is to assign a fixed IP address. There is file named *networking* located in the /etc/init.d folder. To assign a fixed IP address on boot-up edit this file and make sure the start section looks like the following:

```
    start)
        echo -n "Configuring network interfaces: "
        ifup –a

#        pump -i eth0

        ifconfig eth0 192.168.2.3 up
        route add default gw 192.168.2.1

        ifconfig lo 127.0.0.1 up
        echo "done."
        ;;
```

The key changes are to comment out the *pump* line and add the two additional lines below it – the *ifconfig* and *route* commands. The *pump* command was used previously to request the dynamic IP

address via the DHCP service. The *ifconfig* command assigns a fixed IP address and the *route* command adds the default route back through the LAN gateway machine.

The next time the Stargate is booted it will be assigned a fixed IP address of 192.168.2.3 and the default gateway will be 192.168.2.1. The addresses you choose are particular to the LAN you are connecting to.

There is one other important file named *resolv.conf* located in the /etc folder. This file contains the DNS entries for your LAN. The default DHCP configuration creates a valid resolv.conf file as a result of the *pump* command. If you switch to a fixed IP address configuration the entries in this file should still be valid. If you have problems resolving network names see your network administrator for a list of valid DNS entries.

3.2   Application Server Configuration

One of the more interesting uses for the Stargate is as an application server. An application server is a remotely deployed Stargate configured with software for local management of a Mote network. The basic software installed for this purpose is the PostgreSQL database and Xlisten. Xlisten is a Crossbow supplied program that attaches to the serial port attached through the Mote connector. Its purpose is to listen for incoming sensor data messages and log them to the PostgreSQL database. The application server can then be accessed remotely using MOTE-VIEW to view the sensor data.

Other server software that can be installed includes the Apache web server for web based applications and a Java runtime.

3.2.1   Installing Compact Flash Card / PostgreSQL / Xlisten

In order to have sufficient storage space for the PostgreSQL database you need to add an external compact flash (CF) memory card. These cards are available from various manufacturers such as SanDisk, Kingston and sizes ranging from 64MB – 1GB. The recommended minimum size is 256MB. This size should provide a couple years worth of space to store sensor data.

Figure 6. Typical Compact Flash Memory Card

Once the Stargate is configured to use the external CF card there is an archive file that can be downloaded and extracted onto the card to automatically install PostgreSQL and Xlisten.

Compact Flash Card Installation

Open Stargate console session using Windows HyperTerminal or Mincom on Linux (see instructions above)

Insert Compact Flash Card, observe the device id in the error message, ex. /dev/hda1. The error message indicates there is not yet a valid file system installed for the CF device.

(You need to create a valid file system on the CF card in order to mount and use it)

```
$ mkfs.ext2 /dev/hda1
```

(Mount CF card manually the first time – automatically mounted on subsequent reboots)

```
$ mount /dev/hda1 /mnt/cf1
```

PostgreSQL and Xlisten Installation

Locate the software update archive named *StargateCFCard.tar.gz* located in the <CDROM>\8160-0025-14_A\config folder.

This archive contains an image of the PostgreSQL database and Xlisten tools in a plug-and-play format. You simply expand the archive in the root of the CF card and reboot the Stargate.

Open Cygwin shell on PC –

```
$ scp StargateCFCard.tar.gz root@<Stargate IP Address>:/mnt/cf1
```

The <Stargate IP Address> can be determined using the `ifconfig` command at the Stargate console prompt.

Open Stargate shell –

```
$ cd /mnt/cf1
$ tar zxvf StargateCFCard.tar.gz
$ rm StargateCFCard.tar.gz
```

The next time the Stargate is rebooted PostgreSQL and Xlisten will start up automatically. There is a bootup script file named *cfcardapps* in the /etc/init.d folder that checks for the presence of the CF card and automatically starts the applications if they are present on the card.

3.2.2   *AmbiCom Wave2Net Wireless 802.11 Compact Flash Card*

To provide a short-to-medium range, high speed remote access link to the Stargate the advanced Stargate kit ships with the AmbiCom Wave2Net wireless 802.11 card. This card plugs into the available PCMCIA slot using the PCMCIA adaptor module. Although this is a compact flash card the CF slot is already allocated by the CF memory card thus the adaptor.

Figure 7. AmbiCom Wireless 802.11 Card (left) with PCMCIA adaptor (right)

The Stargate is preconfigured to recognize this card and to automatically load the required device drivers. However, some additional configuration is required to allow this card to join a network using the ad-hoc or access point based methods. There is a file named *wireless.opts* located in the /etc/pcmcia folder that allows you to specify the settings for your wireless 802.11 network.

Ad-Hoc Mode (Peer to Peer)

To enable ad-hoc mode, edit the /etc/pcmcia/wireless.opts file and include the following section:

```
# AmbiCom 802.11 Card
*,*,*,*)
    INFO="AmbiCom"
    ESSID="Id"
    MODE="Ad-hoc"
    CHANNEL="6"
    RATE="11M"
    KEY="1234-1234-12 open"
    ;;
```

Set the parameters according to your network settings. If encryption is not being used, comment out the *Key* parameter using a # character at the start of the line.

The "*,*,*,*)" line matches a WiFi card plugged into PCMCIA slot 0, or CF slot 1 - this activates the configuration.

Managed Mode (Access Point)

To enable managed mode make sure the wireless.opts file is as follows:

```
# AmbiCom 802.11 Card
*,*,*,*)
    INFO="AmbiCom"
    ESSID="Id"
    MODE="Managed"
    CHANNEL="6"
    RATE="11M"
    KEY="1234-1234-12 open"
    ;;
```

3.2.3   Sierra Wireless AirCard 555D PCMCIA Cellular Modem Card

To provide a low speed, long range remote access link to the Stargate you can purchase the Sierra Wireless AirCard 555D PCMCIA Cellular Modem Card. This card is essentially a modem that provides a PPP link across the Internet via the cellular network. In addition to purchasing the card you need to purchase a data service plan – Verizon Wireless is the recommended cellular carrier. When you purchase the plan from Verizon make sure you specify a fixed IP address when connecting, this way you can easily locate and connect to the Stargate remotely.
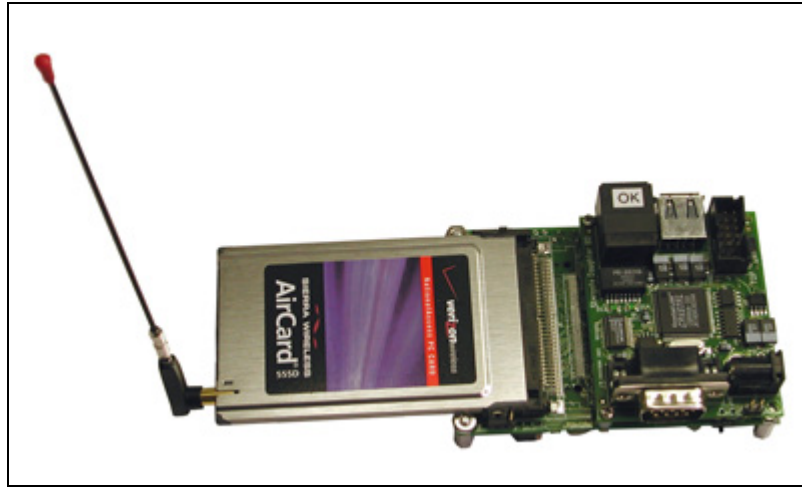
Figure 8. AirCard 555D plugged into the Stargate

<u>Sierra Wireless AirCard 555D Installation</u>

Locate the Sierra Wireless AirCard 555D software update archive named *CellModem.tar.gz* in the <CDROM>\8160-0025-14_A\config folder.

1.   The software update archive adds several configuration files to the Stargate /etc folder tree. You can download this file to your PC and then use scp (secure copy) to copy it to the Stargate / folder as follows:

Open Cygwin shell on PC –

```
$ scp CellModem.tar.gz root@<Stargate IP address>:/
```

2.   Open a console session on the Stargate and unarchive the support files as follows:

```
$ cd /
$ tar zxvf CellModem.tar.gz
etc/ppp/chap-secrets
etc/ppp/options
etc/ppp/pap-secrets
etc/ppp/peers/
etc/ppp/peers/ac555chat1x
etc/ppp/peers/ac555-1x
etc/ppp/peers/ac555qnc
etc/ppp/peers/ac555chatqnc
etc/ppp/resolv.conf
etc/pcmcia/aircard555.conf
etc/pcmcia/cis/aircard555.cis
etc/pcmcia/cis/aircard555.dat
etc/init.d/ppp
etc/init.d/xfer-driver.sh
```

3.   Edit the /etc/ppp/chap-secrets, /etc/ppp/pap-secrets and /etc/ppp/peers/ac555-1x files and make sure your dial up number is correct. Basically replace the 5555555555 placeholder with your assigned number.

4.   Insert the AirCard into the PCMCIA slot. Test the PPP dial-up connection using the following command:

```
$ pppd call ac555-1x
Serial connection established.
```

```
Using interface ppp0
Connect: ppp0 <--> /dev/modem
local  IP address 166.139.23.78
remote IP address 66.174.217.7
primary  DNS address 66.174.6.7
secondary DNS address 66.174.3.7
```

You should see a response from the *pppd* command similar to the above – this means the AirCard has been configured properly and is connecting to the service. The Stargate IP address is the *local IP address*, in this example 166.139.23.78. Make sure this is the same as the fixed IP address you are assigned by your service provider.

5. There is a startup script named *ppp* located in the /etc/init.d folder. To automatically connect to the PPP service on bootup, create a symbolic link to this file as follows:

```
$ ln –s /etc/init.d/ppp /etc/rc2.d/S99PPP
```

The /etc/ppp script will attempt to identify the AirCard and if found establish a connection to the PPP service. If successful you will be able to see the connection status of the PPP service using the following command:

```
$ ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:166.139.23.78 P-t-P:66.174.217.7
          Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1514  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:64 (64.0 b)  TX bytes:94 (94.0 b)
```

3.2.4   MOTE-VIEW

MOTE-VIEW is a PC application Crossbow provides as a free download that allows you to view sensor network data stored in the PostgreSQL database. The latest version of MOTE-VIEW can be downloaded from the software download section of the Crossbow web site.

To connect MOTE-VIEW to a PostgreSQL database running on the Stargate use the following steps:

1. Start MOTE-VIEW and click on the [icon] icon to bring up the "Database Server Configuration" dialog box.

2. Enter the IP address of the Stargate into the "Server" field and click "Connect", this will populate the "Table Name" drop down list with the available results tables.

3. Select the results table you wish to view from the "Table Name" drop down list and then click on "Apply" to view the sensor data.
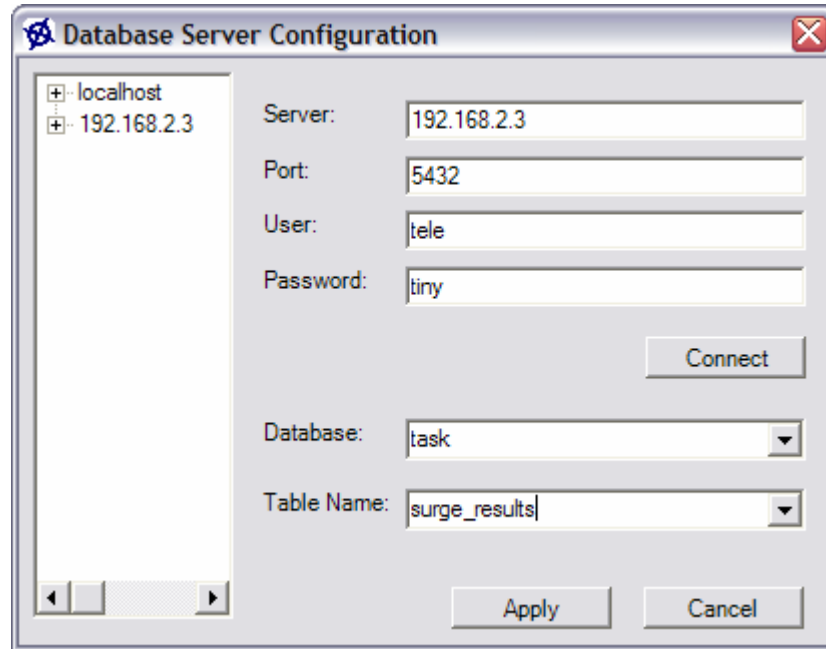
Figure 9. MOTE-VIEW Database Server Configuration

3.2.5   PostgreSQL Automated Database Replication

Depending on the type of remote access connection you have setup for the Stargate, i.e. high speed 802.11 vs. low speed Cellular Modem, it may be desirable to configure automatic PostgreSQL replication. This is especially true when connecting remotely to the sensor database using MOTE-VIEW. MOTE-VIEW has a "Live" mode that requires a constant stream of data be read from the PostgreSQL database.

Database replication allows for a remote *replica* database to be automatically synchronized with the Stargate database. Tools such as MOTE-VIEW can then connect to the *replica* database using high speed connection.

PostgreSQL does not have a built-in replication feature so Crossbow has devised an ftp (file transfer protocol) based replication scheme. The basic idea is to install and configure a script on the Stargate that takes a snapshot of the database periodically and transfers the new records as an ASCII file export to a remote Linux server. The remote Linux server has a similar script file running that will detect new snapshot files and import them into the replica PostgreSQL database.

You must have an ftp service running on the remote Linux server and an account setup for replication.

Stargate Configuration

To install the database replication feature perform the following steps:

1. Copy the <CDROM>\8160-0025-14_A\config\RmtAcc.tar.gz archive into the / folder on the Stargate.

   Open a command console on the Stargate –

```
$ cd /
$ tar zxvf RmtAcc.tar.gz
etc/xfer/
```

```
etc/xfer/etc/
etc/xfer/etc/PASSWORD
etc/xfer/etc/USERNAME
etc/xfer/do-file-xfer.sh
etc/init.d/xfer-driver.sh
```

2. Edit the /etc/xfer/PASSWORD and /etc/xfer/USERNAME files and enter the appropriate settings for the remote Linux server ftp access.

3. Edit the /etc/xfer/do-file-xfer.sh file and make sure the REMOTE symbol is set to the name or IP address of the remote ftp server. Make sure the TABLE symbol is set to the table name of the table you wish to replicate – default is *surge_results*.

4. Execute the /etc/init.d/xfer-driver.sh script to start the replication process. A successful output looks like the following:

```
$ /etc/init.d/xfer-driver.sh
Last Date 05/20/05 12:26:31
COPY
230 Login successful. Have fun.
LOGIN WAS OK
226 File receive OK.
File Sent OK
Connected to 192.168.2.50.
220 (vsFTPd 1.1.3)
331 Please specify the password.
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
250 Directory successfully changed.
local: /tmp/mytable-2005-05-20-14-07.txt remote: tmp4008
227 Entering Passive Mode (192,168,2,50,56,228)
150 Ok to send data.
226 File receive OK.
38752 bytes sent in 0.02 secs (1858.3 kB/s)
350 Ready for RNTO.
250 Rename successful.
local: /tmp/test-script.sh remote: new-script.sh
227 Entering Passive Mode (192,168,2,50,230,215)
550 Failed to open file.
221 Goodbye.
Renamed Remote file
ALL OK!
XFER OK!! etc/xfer/
```

At the ftp server end of the connection (Linux host) you should see time-stamped .txt files being created in the /tmp folder similar to the following:

```
$ ls /tmp/*.txt
/tmp/mytable-2005-05-20-12-26.txt   /tmp/mytable-2005-05-20-14-00.txt
/tmp/mytable-2005-05-20-12-38.txt   /tmp/mytable-2005-05-20-14-07.txt
```

Each time a file is created on the remote server the xfer-driver.sh script running on the Stargate will sleep for 2 hours. This value can be changed by editing the xfer-driver.sh script and re-starting it.

Linux Host Configuration

The Linux host end of the replication scheme completes the database replication by taking the .txt files uploaded to the /tmp folder and importing them into the locally running PostgreSQL database – i.e. the *replica*.

The first step is to make sure you have PostgreSQL installed and running on the Linux host.

The next step is to take a 'snapshot' of the Stargate database and copy it to the Linux host as follows:

Open a command console on the Stargate –

```
$ cd /mnt/cf1
$ pg_dump -U tele -C -F t -f taskdb.sql.tar task
$ scp taskdb.sql.tar root@<Linux IP address>:/tmp
```

The Stargate database snapshot is contained in a file named taskdb.sql.tar located in the /tmp folder. The next step is to import this snapshot into the Linux host PostgreSQL database as follows:

Open a command console on the Linux host –

```
$ createuser tele -d -A -U postgres
$ pg_dump -U tele -C -F t -f taskdb.sql.tar task
$ pg_restore -C -U tele -O -d template1 taskdb.sql.tar
```

The result is the Stargate and the Linux host have the same task database.

The last step is to run a script on the Linux host that will periodically import the new database records from the /tmp/*.txt files uploaded via ftp by the Stargate. Copy the <CDROM>\8160-0025-14_A\config\insert-pgsql.sh script file into the /etc/init.d folder on the Linux host. Edit this file and make sure the PSQL and TABLE symbols are set appropriately. The TABLE symbol needs to match the TABLE symbol used in the do-file-xfer.sh script running on the Stargate.

To start the Linux host end of the database replication, open a command console on the Linux host and execute the following:

```
$ cd /etc/init.d
$ ./insert-pgsql.sh
COPY
INSERT 0 2
```

This script will check for .txt files uploaded into the /tmp folder every 2 hours (default) and insert the records into the local PostgreSQL database. When new records have been inserted you will see output similar to that at the console prompt above.

You can now connect MOTE-VIEW to the Linux host database replica and you should see record updated every 2 hours.

3.2.6 Apache Web Server

Another option for hosting applications on the Stargate is using the Apache web server. The Apache web server allows for remote web based client connections using a web browser such as Internet Explorer or Netscape.

A version of the Apache web server is located on the Stargate support CDROM.

To install the Apache web server on the Stargate perform the following steps.

1. Copy the <CDROM>\8160-0025-14_A\apps\share\apache\apache.tgz archive into the / folder on the Stargate. Follow the instructions in the <CDROM>\8160-0025-14_A\apps\share\apache\README.txt file for the basic installation but **stop** after the /etc/apache/setup command step.

2. Make the following Stargate specific fixes:

   Open a command console on the Stargate –

```
$ cd /lib
$ ln -s libdb.so.3 libdb.so2
```

Edit the /etc/hosts file and make sure there is an entry for the Stargate IP address similar to the following but using your specific IP address:

```
192.168.2.3 Stargate Stargate.domain
```

Modify the Apache default configuration to fix the cgi-bin path.

```
$ cd /etc/apache
```

Edit the Apache configuration file named *httpd.conf*. Make sure you update the cgi-bin path to be /home/apache/cgi-bin in the sections below:

```
ScriptAlias /cgi-bin/ /home/apache/cgi-bin/
#
# "/usr/lib/cgi-bin" could be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory /home/apache/cgi-bin/>
    AllowOverride None
    Options ExecCGI FollowSymLinks Indexes Includes
    Order allow,deny
    Allow from all
</Directory>
```

Create the Apache cgi-bin folder and set the permissions -

```
$ cd /home/apache
$ mkdir cgi-bin
$ chown apache:www-data cgi-bin
```

Start the Apache web server to test.

```
$ /etc/init.d/apache start
```

Start a web browser session and enter http://<Stargate IP address>/index.html - you should see the default Apache web page appear.

The web root is located in /home/apache/public_html. CGI applications can be placed into the /home/apache/cgi-bin folder to provide dynamic application content.

3.2.7   Java Runtime

This section describes how to add the ability to run Java based applications on the Stargate. There are two Java runtimes available: Open Wonka and Sun CDC (Connected Device Configuration). Open Wonka has been reported to be slow and not very stable. Crossbow recommends using the CDC runtime version augmented with the Open Wonka class library – wre.jar.

Due to the size of the CDC java runtime installation it is recommended that it be installed on the CF card as follows:

1. Copy the <CDROM>\8160-0025-14_A\config\stargate-j2me-cdc-wre.tgz archive into the /mnt/cf1 folder on the Stargate using scp from a PC Cygwin session.

2. Unarchive the file and create a symbolic link to the Java CDC runtime as follows:

```
$ cd /mnt/cf1
$ tar zxvf stargate-j2me-cdc-wre.tgz
$ ln –s /mnt/cf1/cdc/bin/jre /usr/sbin/jre
```

The /mnt/cf1/cdc/bin/jre script file is linked to /usr/sbin/jre to be included in the default path. The jre script is used to start a compiled Java program.

To start a compiled Java program issue the following command:

```
$ jre <Java class>
```

The jre script accepts a single command line argument by default and includes the basic class libraries. This script can be modified to add additional command line arguments and class library references.

3.3 Starting Programs during System Boot-up

To start programs automatically when the Stargate boots up you need to do the following:

1. Create a script in the /etc/init.d folder, call it say myprog.sh. This script when called starts the /usr/local/myprog program in the background. You must run the program in the background otherwise the boot-up sequence will pause until the program completes.

    Example script file myprog.sh:

```
#!/bin/sh
# Start program as a background task
/usr/local/myprog >/dev/null 2>&1 &
```

The & tells the command processor to start the program as a background task then continue. The >/dev/null redirects the standard output to the null device. The 2>&1 sends the standard error output to the null device as well (via standard out). Re-directing the program output to the null device prevents messages from appearing on the console screen.

2. Create a link to the script you just created to /etc/rc2.d. The name of the link tells the command processor what order to start it.

    Example:

```
$ ln –s /etc/init.d/myprog /etc/rc2.d/S99myprog
```

The Stargate will then run the /etc/init.d script with a relative startup priority of 99. The links with 01 run first, and so on.

3.4 Programming Motes

The Stargate can be used to download programs onto motes via the on-board connector in a similar manner to the MIB510 & MIB600 cards. The application program must first be compiled within the Cygwin TinyOS environment on the PC. The resulting program image is then copied to the Stargate and then the programming instruction can be issued.

Open Cygwin shell on PC –

(cd to application folder and build)

```
$ make mica2
```

(set mote id, this example sets it to 1)

```
$ set-mote-id build/mica2/main.srec build/mica2/main.srec.out 1
```

(Copy image with mote id applied to Stargate for programming step)

```
$ scp build/mica2/main.srec.out root@<Stargate IP Address>/usr
```

Open Stargate shell -

(Stop xlisten-arm if running, i.e. need access to serial port for programming)

```
$ ps –A | grep xlisten-arm
$ kill <process id for xlisten-arm>
$ cd /usr
```

(Program Mote plugged into Stargate connector)

```
$ uisp -dprog=sggpio -dpart=ATmega128 --wr_fuse_h=d9 --wr_fuse_e=ff --erase --
upload if=main.srec.out
```

3.5 Setting the System Clock

The Stargate does not have a battery backed clock so the date and time are reset back to 1970 when the unit is powered down. If you have the Stargate configured with Internet access the /mnt/cf1/cfcard.rc script that runs on boot-up will set the correct date and time using the *ntpdate* service.

To set the Stargate clock manually issue a command similar to the following using the correct date and time:

```
$ date --set="Tues May 17 10:00:00 PDT 2005"
```

4   Developing and Running Applications on Stargate

4.1   Installing and Using the GNU tool chain

The CDROM shipped with the Stargate Processor Board, contains the GNU cross platform development tools, version 3.3.2, for the **Linux Host** platform only.

The zipped tool chain archive file, `cross-3.3.2.tar.gz`, shipped with the CDROM, is available at the following directory:

`<CDROM>/8160-0025-14_A/tools/`

To install these tools on your Linux Host Development machine, log in as **`root`**. Change to the root directory:

`cd /`

Extract the zipped archive files and install them on your Host machine, by entering the following command:

`tar -xvfz <CDROM>/8160-0025-14_A/tools/arm-linux-gcc-3.3.2.tar.gz`

This command will install the tools on your Host machine in the */usr/local/arm/3.3.2/* directory. Add the path to the directory containing the binaries for the tools, in your environment variable PATH. The following command can be used on the bash shell:

`export PATH=$PATH:/usr/local/arm/3.3.2/bin`

Alternatively, this line can be added in your shell configuration file such as, **`$HOME/.bashrc`** for bash shell.

You are now ready to build applications for the Stargate platform.

4.2   Running Applications

Once the cross platform development tools are installed and configured on the host machine, we are ready to develop applications for the Stargate Platform.

Given below is a sample `Makefile` for compiling an application with just two source files `foo.c` and `bar.c`. This can easily be extended to include more source files as application development proceeds.

```
CC=arm-linux-gcc
DEFS=
INCS=
LIBS=
OBJS= foo.o bar.o
foo-bar: ${OBJS}
   ${CC} -o foo-bar ${LIBS} ${OBJS}
foo.o: foo.c
   ${CC} ${DEFS} ${INCS} -c foo.c
bar.o: bar.c
   ${CC} ${DEFS} ${INCS} -c bar.c
```

A template for starting application development is provided in the CDROM shipped with the Stargate Processor Board, in the *apps/sample/template/* directory.

A number of sample test programs are provided to serve as examples for application development and verify the correct installation/configuration of the tools on the Host Computer and working of the various devices on the Stargate Platform. These include:

- an "Hello World" application
- a networking application
- a file system application
- a webcam video-capture application
- a Mote application

They are present in the *apps/sample/** directory on the CDROM. For more details on these applications, refer to the README file in each of these application directories.

In addition to the sample applications, some shareware, open source applications are provided to aid in the application development of your custom application for Stargate. These include:

- A Java Virtual Machine: *apps/share/open_wonka*
- A Web server: apps/share/apache
- A memory test utility: apps/share/memtester-2.93.1
- An image capture utility for a video4linux device: *apps/share/w3cam-0.7.2*

5 Creating a Linux boot image

5.1 How to build the Bootloader from Source

The Bootloader for the Stargate Development Platform is based on the Bootloader for the IPAQ Bootloader, version 2.21.13, with appropriate changes. The Bootloader has been enhanced to provide TFTP download capability for installing Kernel and Root File system modules using the Ethernet from a TFTP server. The bootloader image file *bootldr* is provided with the release distribution, in the CDROM shipped with the Stargate Processor Board, in the *<CDROM>/8160-0025-14_A/images/bootldr/* directory.

Sources for the Bootloader have also been provided in the distribution, and can be rebuilt or updated, if needed. The Bootloader sources are available in *<CDROM>/8160-0025-14_A /sources/bootloader/*.

To rebuild or update the Bootloader, you should copy this entire directory to your host machine where you plan to rebuild the Bootloader.

You can build the Bootloader image by entering the following command in the Bootloader sources directory:

```
make bootldr
```

This would result in the building of a new Bootloader image, *bootldr*. You can program this image on to your Stargate board by following the directions provided in section 5.5.1.

5.2 How to rebuild the Linux Kernel Image

The kernel image file `zImage` has been provided with the release distribution, in the CDROM shipped with the Stargate Processor Board, in the following location: *<CDROM>/8160-0025-14_A /images/zImage*.

Source code for the kernel, with appropriate patches applied, has been provided in the distribution, and can be used to compile a new version of the kernel, if needed. The kernel sources are available in the following directory: *<CDROM>/8160-0025-14_A /sources/os/linux-2.4.19/*.

To rebuild or update the kernel or OS modules, you should copy this entire directory to your Host machine where you plan to rebuild the kernel.

If you need to obtain the kernel sources and apply patches on your own, you can follow the directions provided in the section 5.2.1. Otherwise you can proceed to section 5.2.2 to configure and compile the kernel.

5.2.1 Extracting Kernel Sources

- Get the main Linux kernel sources from: ftp://ftp.arm.linux.org.uk:/pub/linux/linux-2.4/linux-2.4.19.tar.gz

- Get the ARM processor port patch from: ftp://ftp.arm.linux.org.uk/pub/linux/arm/kernel/v2.4/patch-2.4.19-rmk7.gz

- Get the XScale patch from: ftp://ftp.arm.linux.org.uk/pub/linux/arm/people/nico/diff-2.4.19-rmk7-pxa2.gz

- Extract the kernel source using the command:

```
tar zxvf linux-2.4.19.tar.gz
```

- Extract the ARM processor port using the command:

```
gunzip patch-2.4.19-rmk7.gz
```

- Change to the extracted Linux source directory and apply the patch using the following command:

```
patch –p1 < patch-2.4.19-rmk7
```

- Extract and apply the XScale patch similarly:

```
gunzip diff-2.4.19-rmk7-pxa2.gz
```

```
patch –p1 < diff-2.4.19-rmk7-pxa2
```

- Apply the Stargate patch provided in the CDROM, using the command:

```
patch –p1 < <CDROM>/8160-0025-14_A/sources/os/patch/patch-
2.4.19-rmk7-pxa2-star7.2
```

5.2.2 Configure and Compile the Kernel

The *README* file in the Linux sources directory provides more information about configuring and compiling the kernel image. We are only providing here the concise directions to rebuild the image as configured and shipped with the Stargate Development Kit. After you have copied the sources, you need to make sure you have no stale object files and dependencies from previous builds.

- To change to the Linux sources directory and enter the following command.

```
make mrproper
```

You now have the sources correctly installed to configure and compile a new kernel image.

- The default configuration for Stargate is installed by entering the following command.

```
make stargate_config
```

- To carry your existing configuration to a new version with minimal work, use the following command.

```
make oldconfig
```

- You can use the following command to interactively configure the kernel using text based color menus, radio lists and dialogs.

```
make menuconfig
```

- Next, to set up all the dependencies correctly, run the following command.

```
make dep
```

- Finally compile the kernel to obtain the downloadable image by entering the following command.

```
make zImage
```

- After completion of the compilation and linking, which can take up to a few minutes, you would have a kernel image in the following location:

```
arch/arm/boot/zImage
```

You can load this image on to your Stargate board by following the directions provided in section 5.5.2

5.3   How to add new device driver modules

5.3.1   New device drivers

In this section we will go through the steps needed to obtain, configure, build and load the device driver modules for Linux OS in general and the Stargate board's kernel in particular. We will do this using an example of a driver for a Webcam, the Logitech Pro 4000, which contains the Phillips chipset.

More information, for building modules, is available in the documentation provided with the Linux Kernel sources particularly the *README* file in the top directory and the *modules.txt* file in the Documentation directory.

1. The first step is to find out whether the device driver sources are available either in the kernel sources or from the web. Some device manufacturers make the drivers available for download on their web site. Section 7.2 lists some useful sources for such information. The sources for the example we are using have been obtained from http://www.smcc.demon.nl/webcam/

2. The next step is to copy the driver files in the appropriate directory. The drivers may be provided either in source form, pre-compiled for certain architectures, or a mixture of both. If the drivers are provided either in part or wholly in source form, they need to be copied in the appropriate directory in the Linux sources distribution copied installed in the Host machine. In this example the source files need to be placed in the directory *<Linux Kernel>/drivers/usb/*.

3. If there are source files provided in the driver distribution, after copying the source files, execute the following command to configure the Kernel build process to include the required drivers:

    make menuconfig

4. This menu driver configuration tool allows you to configure what is included in the Kernel image and in the loadable driver modules. In this example, we need to include:

    a.   The video drivers

    b.   The Phillips chipset (used in the Webcam) drivers

    c.   The compression and decompression routines used by these drivers. This is provided in object modules, without source code, so build is necessary.

5. To compile the video drivers, we scroll down to the item named "Multimedia devices," and select it by pressing **Enter**. Now, in the new menu, configure "Video for Linux" item by typing **M**. This entry will now look as follows:

    ```
    <M> Video for Linux
    ```

    You will see 2 new entries, the first of which is "Video for Linux --->". Select this by pressing **Enter** on this line, which leads to the next menu which should have the following line:

    ```
    [*] V4L information in proc filesystem
    ```

6. To configure the Phillips chipset drivers, in menuconfig, at the top- level menu, scroll down to the item named "USB support --->", and select it by pressing **Enter**. Configure it to be added as a module, in the next menu "Support for USB," by typing **M**.

Now, scroll down to locate the item USB Phillips Cameras and configure it to be built as a module by typing **M**.

7. Build the video driver and the Phillips chipset drivers, thus configured, by entering the following commands in the top level Linux Sources directory:

   ```
   make dep
   ```

   ```
   make modules
   ```

   This results in building the video driver object *videodev.o*, in the *drivers/media/video/* directory and the Phillips chipset driver *pwc.o* in the *drivers/usb/* directory.

8. Copy the `videodev.o` module to the Stargate board's file system at the directory */lib/modules/linux/kernel/drivers/media/video/* and the `pwc.o` and `pwcx-2.4.20.o` (precompiled compression routine) to the */lib/modules/linux/drivers/usb/* directory using either the serial port or Ethernet as described in section 5.5.3

   The drivers will be enabled by using the following commands:

   ```
   modprobe videodev
   ```

   ```
   modprobe pwc
   ```

   ```
   insmod -f /lib/modules/linux/kernel/drivers/usb/pwcx-2.4.20.o
   ```

   Create the video device file using the following command:

   ```
   mknod /dev/video0 c 81 0
   ```

5.4 How to make a File System

Generally you would not need to recreate the root file system, and should be able to use the `root.jffs2` file provided. However, should you need to remake the file system, follow the following steps.

1. Get the sources for root file system of the factory image of Stargate. The kernel sources are available in the following directory: *<CDROM>/8160-0025-14_A/sources/root_fs/*

2. Extract the root file system while logged in as **root** on the machine. This is important, because some of the files extracted are device files, and Linux will refuse to make them unless the extraction is done with root-level permissions. The following command extracts the file system:
   ```
   tar xvfz root.tgz
   ```

3. Make any changes that you want to the root file system.

4. Create a new file system into a file named `root.jffs2` using the following command:
   ```
   mkfs.jffs2 –r root -o root.jffs2 –e 0x40000 –p
   ```

You can load this image on to your Stargate board by following the directions provided in section 5.5.2

5.5 How to download images

5.5.1 JTAG Flash programming

A Flash programming utility is provided by Intel, as indicated in section 7.2. For your convenience it is also provided in the CDROM shipped with the Stargate Processor Board. The document *Flash Memory Programmer for Intel Development Platforms – Release Notes* contains more information about how to program Flash Memory.

The bootloader is typically loaded into the Flash memory of the Stargate board using JTAG. Copy the bootloader executable *bootldr* that you have built (or one that is shipped in the CDROM) to a Windows machine.

A JTAG pod is available from Insight (www.insight-electronics.com). The part number is DS-KIT-CABLE-JTAG. An alternate vendor is Mouser Electronics (www.mouser.com), part number 370-CABLEJTAG.

The following table shows how each JTAG port pin should be connected to the cable.

```
 • 2      • 4      • 6     • 8     • 10
 • 1      • 3      • 5     • 7     • 9
```

Figure 10. *JTAG port on Daughter Card*

Pin Number	Connection on JTAG Cable
1	TCK/CCLK
2	GND
3	TDO/D/P
4	VCC
5	TMS/PROG
9	TDI/DIN

Figure 11. *J1 Connector pin numbering*

Make sure that the platform file, dbpxa255.dat, and the bootloader binary, bootldr, are in the same directory as the flash programming utility *JflashMM.exe*.

To program the flash use the following command:

```
C:\Flash>JFLASHMM DBPXA255 BOOTLDR
```

5.5.2   Kernel and File System Using Serial port

After the Stargate board is reset and the Bootloader boots up the board, you will see the following message on the console (from either *HyperTerminal* or *Minicom*)

```
Press Return to start the OS now, any other key for monitor menu
type "?" or "help" for help.
boot>
```

At this `boot>` prompt, type the following command to set the load method to "serial" and to download the kernel:

```
set loadmethod serial
```

```
load kernel
```

In *HyperTerminal*, go to menu **Transfer > Send File**. Browse to select the **filename** zImage. Set the "Protocol" to **Xmodem**. Click on the **Send** button. This will start the transfer.

In *Minicom*, enter a **Ctrl-AS**. Scroll through the Upload menu to select "Xmodem" and press **Enter**. Now, in the directory browser, select the kernel file zImage by scrolling to it and selecting

it by pressing the **Spacebar**. The Xmodem upload menu comes up indicating the progress of the transfer.

◄ **NOTE:** The board waits for about 10 seconds[1] for the transfer to start and after that it times out. So, if you see that no transfer is occurring, that is probably the reason. Cancel, and retry the transfer.

The Bootloader is initially configured to create 3 partitions: 1) for itself, 2) for the boot parameters, and 3) for the kernel. We now need to create a partition for the File System by entering the following command at the Bootloader `boot>` prompt.

```
partition define root 0x00140000 0x01EC0000 16
```

You will see the following message confirming the action:

```
argv[1]=define
defining partition: root
```

Next, save this information in the Bootloader parameter section, by entering the following command:

```
partition save
```

You will see the following message confirming the action:

```
argv[1]=save
programming flash...erasing ...
Erasing sector 00040000
writing flash..
addr: 00040000 data: 646C7470
verifying ... done.
```

Now, we verify if the partitions are created properly, by entering the following command:

```
partition show
```

You should now see the following message from the Bootloader confirming that the partition has been successfully created.

```
npartitions=00000004
bootldr
base: 00000000
size: 00040000
flags: 00000002
params
base: 00040000
size: 00040000
flags: 00000000
kernel
base: 00080000
size: 000C0000
flags: 00000000
root
```

---

[1]To change the timeout value for File Transfer, use the following command on the Bootloader prompt: `set xmodem_initial_timeout X00000` where, X is the number of seconds, for 15 seconds you would type **1500000**

```
base: 00140000
size: 01EC0000
flags: 00000010
```

You can load the File System by typing the following command at the Bootloader `boot>` prompt:

`load root`

Follow the directions to download the kernel, as described previously in this section (using *HyperTerminal* or *Minicom* as appropriate for your situation). However, substitute the file `root.jffs2` for `zImage`.

5.5.3   Kernel and File System Using Ethernet from a TFTP Server

The Bootloader will download the modules (kernel and file system) from the *<TFTPROOT>/stargate/* directory of the machine designated as the TFTP server. So the image files `zImage` and `root.jffs2` should be placed in this directory. It is only necessary to place the files that you plan to download; it is not necessary to have all of the files present.

The Stargate comes with its MAC address factory configured. If you want to re-configure it, say after re-flashing the Bootloader, enter the following commands at the bootloader prompt. It is essential to assign a MAC address to the Stargate for proper functioning of the Networking features.

`set macaddr 00:50:C2:2F:3X:YZ`
`params save`
`loadmacaddr`

Substitute the "`F`," "`X`," and "`Z`" with the appropriate numbers from the label on your Stargate Processor board. This sets the address, saves it to the flash parameter area and loads the Ethernet chip with the indicated address.

> ◀ **NOTE:** The setting of the MAC address needs to be done only once. The IP parameters can configured either manually or with DHCP.

5.5.4   Configuring the IP Parameters

If you want to manually setup your IP configuration and save it, enter the following bootloader commands at the prompt. You would have to obtain the IP address (`ipaddr`) for your Stargate from your Network Administrator. The Network Address mask (`netmask`) and the Gateway IP address (`gateway`) are site specific and may be obtained from the Network Administrator.

- **`set ipaddr XXX.XXX.XXX.XXX`**

- **`set netmask NNN.NNN.NNN.NNN`**

- **`set gateway GGG.GGG.GGG.GGG`**

- **`set serverip SSS.SSS.SSS.SSS`**

- **`set netcfg manual`**

- **`params save`**

The "`serverip`" is the address of the TFTP server, on which you have placed the image files to be downloaded. Again, you should only need to do this once.

If you want to use DHCP for your network configuration, enter the following commands:

```
set netcfg dhcp
params save
```

If you want to check to ensure that DHCP works you can use the following command:

```
ipconfig
```

This will start the DHCP configuration process and set all the necessary parameters. To save the parameters so configured, enter the following command to make the DHCP-provided parameters permanent.

```
save params
```

Now you are set to download the required modules via TFTP. Once you have all the IP parameters setup, you declare your intent to use TFTP with the following command:

```
set loadmethod tftp
```

If the `loadmethod` parameter is not set, or is set to "xmodem", the load command will continue to use xmodem on the serial port for downloading, see section 5.5.2. Again you can use the `params save` command to make this permanent. The Kernel is loaded by entering the following command:

```
load kernel
```

At the end of the output for the TFTP transaction is a checksum that is used to ensure that the file is copied reliably. For example after loading the Kernel you would see the following message.

```
boot> load kernel
After receiving file, will automatically uncompress .gz images
loading flash region kernel
using tftp
IP: 10.1.1.171 netmask: 255.255.255.0 gateway: 10.1.1.221
Boot server: 10.1.1.46
TFTP loading 'stargate/zImage' to a0000400
eth addr: 00:b0:d0:4c:f2:74
TFTP from server 10.1.1.46
Filename 'stargate/zImage'.
Load address: 0xa0000400
Loading:  1400
1457 blocks done
Bytes transferred = 745748 (b6114 hex)
setenv called with filesize=b6114
Loaded 745748 bytes
BSD checksum: 28541
000B6114 bytes loaded to A0000400

programming flash...erasing ...
Erasing sector 00080000
Erasing sector 000C0000
Erasing sector 00100000
writing flash..
addr: 00080000 data: E1A00000
addr: 00090000 data: CFF71EC5
addr: 000A0000 data: 25CD7137
addr: 000B0000 data: 4E5B7E82
addr: 000C0000 data: 7F45E8BD
addr: 000D0000 data: 2783DF48
```

```
addr: 000E0000 data: 74CE0168
addr: 000F0000 data: E65D10EB
addr: 00100000 data: 6A3DEB91
addr: 00110000 data: 7EB88112
addr: 00120000 data: BBE675BD
addr: 00130000 data: 98997D09
verifying ...
done.
boot>
```

The file system is loaded by entering the following command:

`load root`

You can also download another Bootloader image by entering the following command:

`load bootldr`

5.5.5   Drivers and Applications via Ethernet or Serial

There are two ways to copy files to a Stargate board, after it has been properly initialized with the Kernel and the File System: The zmodem protocol can be used to transfer files to the board, via the serial port. This method copies the files to the current directory, but does not overwrite existing files. We need to make sure to change to the proper directory (with a **`cd`**), before copying the files.

1. In *HyperTerminal*, select the **Transfer>Send File** in the *HyperTerminal* menu. Browse to select the appropriate filename. Set "Protocol**"** to **`Zmodem`**. Click on the **Send** button. This will start the transfer. After the transfer completes, you should be able to see the file in the current directory, by doing an **`ls`** (for list files).

2. A faster method is to transfer the files, is using the secure copy program, **scp**. For this you need to be connected to the network. Check that the board has an IP address, by typing the following command at the Shell prompt:

`ifconfig`

If you see an output similar to the following, you have a valid network connection. If there is no output, you are not yet connected to the network.

```
eth0

Link encap:Ethernet  HWaddr 00:00:00:00:00:00

inet addr:10.1.1.82 Bcast:10.1.1.255 Mask:255.255.255.0

UP BROADCAST RUNNING  MTU:1500  Metric:1

RX packets:84 errors:0 dropped:8 overruns:0 frame:0

TX packets:3 errors:0 dropped:0 overruns:0 carrier:0

collisions:8 txqueuelen:100

RX bytes:6781 (6.6 Kb)  TX bytes:0 (0.0 b)

Interrupt:25
```

In the above example, the IP address is 10.1.1.82.

The following command on your host machine, where you have the driver modules file `modules.tgz`, will transfer this file to the */lib/modules/* directory on the Stargate's file system.

`scp modules.tgz root@10.1.1.82:/lib/modules/`

If you want to copy the files from Stargate itself, you would have to enable a secure shell, ssh, connection on your host machine. The following command can be used:

```
scp user@domain:<location>/modules.tgz /lib/modules/
```

The command scp means secure shell copy.

6   Linux Console Commands

6.1   cd – change directory (folder)

The cd command is used to change the current folder.

Example:

```
$ cd /usr/lib
```

6.2   ls – list files

The ls command is used to list the files located in the current folder.

Example:

```
$ ls
armsf*          postgres*    setup-task-db.sh*
```

6.3   pwd – print working directory (folder)

The pwd command will print the current working folder.

Example:

```
$ pwd
/usr/lib
```

6.4   scp – secure copy

The *scp* command is used to copy files between Unix/Linux based environments.

This example will copy all matching files in the current folder to the folder on the remote machine located at address 192.168.2.3 using the remote user *root*; you will be prompted for the remote user root password – the command will then complete:

```
$ scp *.tgz root@192.168.2.3:/usr/src
```

6.5   ps – list current processes

The *ps* command is used to list the currently running processes.

This example will list all the running processes:

```
$ ps –A
  PID TTY          TIME CMD
    1 ?        00:00:06 init
    2 ?        00:00:00 keventd
    3 ?        00:00:00 ksoftirqd_CPU0
    4 ?        00:00:00 kswapd
    5 ?        00:00:00 bdflush
    6 ?        00:00:00 kupdated
    7 ?        00:00:00 mtdblockd
    8 ?        00:00:00 jffs2_gcd_mtd3
  120 ?        00:00:00 cardmgr
  135 ?        00:00:00 sshd
  144 ?        00:00:00 khubd
  179 ?        00:00:00 postmaster
```

```
  180 ?        00:00:00 xlisten-arm
  185 ?        00:00:00 postmaster
  186 ?        00:00:00 postmaster
  188 tts/0    00:00:00 bash
  201 ?        00:00:00 pump
  329 ?        00:00:01 sshd
  330 pts/0    00:00:00 bash
  340 pts/0    00:00:00 ps
```

6.6   kill – stop a current process

The *kill* command is used to stop a current process from running.

This example will stop the xlisten-arm process (see ps output above):

```
$ kill 180
```

6.7   date – set system date & time

The *date* command is used to set the system date & time. The Stargate date is not battery backed like a PC so it must be set each time after the system is booted.

Example:

```
$ date --set "Mon May 16 13:35:00 PM PDT 2005"
```

6.8   shutdown – perform an orderly system shutdown

The *shutdown* command is used to perform an orderly shutdown of the Linux operating system. This is a required step before turning off or resetting the Stargate to ensure the integrity of the file system.

This example will shutdown the Stargate immediately to prepare for a power off or reset:

```
$ shutdown -h now
```

6.9   ifconfig – show the active network interfaces

The *ifconfig* command is used to show the parameters of one or all active network interfaces. The *ifconfig* command can also be used to start and stop an interface.

This example shows the status of the ethernet interface::

```
$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:C2:2F:3B:31
          inet addr:192.168.2.3  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:168978 errors:0 dropped:23 overruns:0 frame:0
          TX packets:137799 errors:0 dropped:0 overruns:0 carrier:0
          collisions:34 txqueuelen:100
          RX bytes:25771930 (24.5 MiB)  TX bytes:0 (0.0 b)
          Interrupt:26
```

This example shows how to start the ethernet device with a fixed IP address:

```
$ ifconfig eth0 192.168.2.3 up
```

6.10 iwconfig – show the active wireless network interfaces

The *iwconfig* command is used to show the details of one or all active wireless network interfaces.

Example:

```
$ ifconfig wlan0
```

```
wlan0      IEEE 802.11b  ESSID:"default"  Nickname:"stargate"
           Mode:Managed  Frequency:2.437GHz  Access Point: 00:01:24:F2:B8:EC
           Bit Rate=2Mb/s   Sensitivity=1/242700000
           Retry min limit:8   RTS thr:off   Fragment thr:off
           Encryption key:1234-1234-12   Security mode:open
           Power Management:off
           Link Quality:53/1  Signal level:-61 dBm  Noise level:-100 dBm
           Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
           Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

6.11 pump – show the active wireless network interfaces

The *pump* command is used to acquire an IP address dynamically for a particular network interface using the DHCP service running on the LAN.

Example:

```
$ pump –i eth0
```

7 Appendix A. References

7.1 Hardware

7.1.1 Power Switches

Switch	Function	Comments
S1	ON-OFF	When only Stargate Processor Board is used.
S1	ON	When both the Stargate Processor Board and the Daughter Card are used, it should be ON at all times.
S3	ON-OFF	When both the Stargate Processor Board and the Daughter Card are used.

7.1.2 Reset buttons

Name	Function
SW3	Hardware Reset Button.
S2	Software Reset Button.

7.1.3 Jumper Settings

Jumper	Settings	
J12	1-2	5v PCMCIA
J12	2-3	3.3v PCMCIA
J13	1-2	5v Compact Flash Card
J13	2-3	3.3v Compact Flash Card
J8	1-2	External Power Supply
J8	2-3	Li-Ion Battery Pack

7.1.4 LEDs

LED	Color	Where	Comments
DS3	Green	Daughter Card	Switched ON when a network cable is connected.
DS4	Green	Daughter Card	Indicates network activity, when ON.
DS7	Yellow	Processor Board	
DS8	Green	Processor Board	
DS9	Red	Processor Board	

7.1.5   Connector Information

Table 2. *Mote Connector Pinouts*

Pin	Description	Pin	Description
1	GND	27	GP25_M-UART_RXD0
2	Not Used	28	GP26_M-UART_TXD0
3	GP8_M-INT3	29	GP65_M-PW0
4	GP6_M-INT2	30	GP66_M-PW1
5	GP1_M-INT1	31	GP67_M-PW2
6	GP0_M-INT0	32	GP68_M-PW3
7	GP9_M-BAT-MON	33	GP69_M-PW4
8	GP29_M-LED3	34	GP70_M-PW5
9	GP28_M-LED2	35	GP75_M-PW6
10	GP27_M-LED1	36	GP74_M-ADC7_JTAG_TDI
11	Not Used	37	GP73_M- ADC7_JTAG_TDO
12	Not Used	38	GP72_M- ADC7_JTAG_TMS
13	Not Used	39	GP71_M- ADC7_JTAG_TCK
14	GP76_M-PW7	40	Not Used
15	GP24_M-USART1_CLK	41	Not Used
16	GP25_M-UART_RXD0	42	Not Used
17	GP26_M-UART_TXD0	43	Not Used
18	GP23_M-SPI_SCK	44	Not Used
19	IRTXD_M-USART1_RXD	45	Not Used
20	IRRXD_M-USRAT1_TXD	46	Not Used
21	I2C_SCL	47	Not Used
22	I2C_SDA	48	GP77_M-RSTN
23	GP31_M-PWM0	49	GP32_M-PWM1B
24	GP30_M-PWM1A	50	VCC3_3
25	Not Used	51	GND
26	Not Used		

Table 3. *Main Board Connector on Daughter Card*

Pin	Description	Pin	Description
1	USB_MINUS_D_CARD	27	ENET_HW_SLEEP# _D_CARD
2	USB_PLUS_D_CARD	28	GND
3	USB_PWR_SENSE_D_CARD	29	JTAG_NTRST_D_CARD
4	USB_PWRCNTL_D_CARD	30	TDI_D_CARD
5	VCC3_3	31	SA_TDO_D_CARD
6	SA_D15_D_CARD	32	JTAG_TMS_D_CARD
7	SA_D14_D_CARD	33	JTAG_TCK_D_CARD
8	SA_D13_D_CARD	34	SA_NWE_D_CARD
9	SA_D12_D_CARD	35	SA_NOE_D_CARD
10	SA_D11_D_CARD	36	SA_NCS [3]_D_CARD
11	SA_D10_D_CARD	37	5V_PWR_JACK_D_CARD
12	SA_D9_D_CARD	38	SA_A4_D_CARD
13	SA_D8_D_CARD	39	SA_A3_D_CARD
14	GND	40	SA_RDY_D_CARD
15	SA_D7_D_CARD	41	SA_A2_D_CARD
16	SA_D6_D_CARD	42	GP61_D_CARD
17	5V_PWR_JACK_D_CARD	43	FF_RI_D_CARD
18	SA_D5_D_CARD	44	FF_TXD_D_CARD
19	SA_D4_D_CARD	45	FF_RTS_D_CARD
20	SA_D3_D_CARD	46	FF_DTR_D_CARD
21	SA_D2_D_CARD	47	FF_RXD_D_CARD
22	SA_D1_D_CARD	48	FF_DCD_D_CARD
23	SA_D0_D_CARD	49	FF_DSR_D_CARD
24	RESET_IN#_D_CARD	50	FF_CTS_D_CARD
25	ENET_INT_D_CARD	51	VCC3_3
26	GND		

Table 4. *Daughter Card Connector on Main Board*

Pin	Description	Pin	Description
1	GND	27	VCC3_3
2	ENET_INT	28	FF_CTS
3	RESET_IN#	29	FF_DSR
4	SA_D0	30	FF_DCD
5	SA_D1	31	FF_RXD
6	SA_D2	32	FF_DTR
7	SA_D3	33	FF_RTS
8	SA_D4	34	FF_TXD
9	SA_D5	35	FF_RI
10	5V_PWR_JACK	36	GP61
11	SA-D6	37	SA_A2
12	SA_D7	38	SA_RDY
13	GND	39	SA_A3
14	SA_D8	40	SA_A4
15	SA_D9	41	5V_PWR_JACK
16	SA_D10	42	SA_NCS [3]
17	SA_D11	43	SA_NOE
18	SA_D12	44	SA_NWE
19	SA_D13	45	JTAG_TCK
20	SA_D14	46	JTAG_TMS
21	SA_D15	47	SA_TDO
22	VCC3_3	48	TDI
23	USB_PWRCNTL	49	JTAG_NTRST
24	USB_PWR_SENSE	50	GND
25	USB_PLUS	51	ENET_HW_SLEEP#
26	USB_MINUS		

Table 5. *Edge Connector P4 on Main Board*

Pin	Description
1	VCC3_3
2	GND
3	I2C_SCL
4	I2C_SDA
5	MBGNT
6	MBREQ
7	GP1_M-INT1
8	GP1_M-INT0

Table 6. *Edge Connector J14 on Main Board*

Pin	Description
1	GND
2	IRTXD_M-USART_RXD_MUX
3	VCC3_3
4	IRRXD_M-USART_TXD_MUX
5	GP25_M-UART_RXD0_MUX
6	GP26_M-UART_TXD0_MUX

7.1.6 Device Information

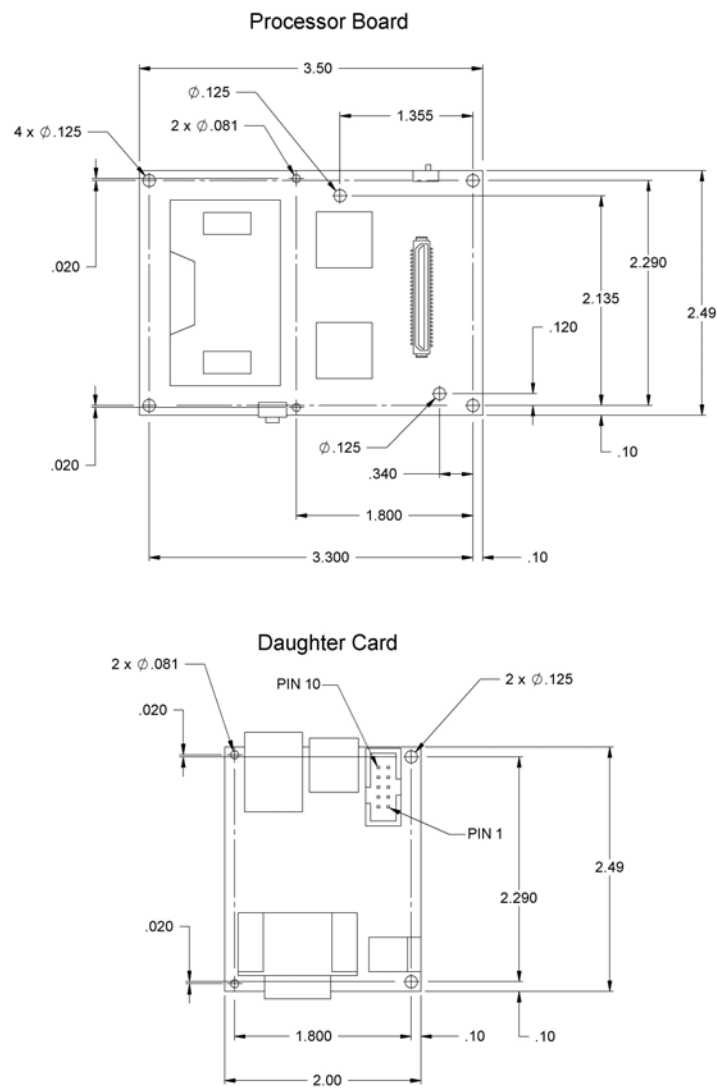| IC	URL for Manuals
PXA255	http://www.intel.com/design/pca/prodbref/252780.htm
SA1111	http://www.intel.com/design/strong/guides/278281.htm
CS8900A	http://www.cirrus.com/en/products/pro/detail/P46.html

7.2 Software

Software	URL for more information
Linux Kernel	ftp://ftp.arm.linux.org.uk/pub/linux/linux-2.4
GNU Tools	ftp://ftp.arm.linux.org.uk/pub/armlinux/toolchain/
Linux Drivers	http://www.linuxdevices.com/ http://linuxdrivers.foundries.sourceforge.net/ http://www.linuxvoodoo.com/
Misc. Linux	http://www.embedded-linux.org/
Bootloader	http://www.handhelds.org/
Misc. Stargate	http://platformx.sourceforge.net/
Flash Programmer	http://www.intel.com/design/pca/applicationsprocessors/swsup/jflashmm.htm

7.3 Community Resources

There is a large community of Stargate users and developers who can be reached via a mailing list. For more information refer to the following URL: http://www.cens.ucla.edu/mailman/listinfo/stargate-users

7.4   Mechanical Drawings

Processor Board

Daughter Card

(All dimensions are in inches)

8 Appendix B. Support and Warranty Information

8.1 Customer Service

As a Crossbow Technology customer you have access to product support services, which include:

- Single-point return service
- Web-based support service
- Same day troubleshooting assistance
- Worldwide Crossbow representation
- Onsite and factory training available
- Preventative maintenance and repair programs
- Installation assistance available

8.2 Contact Directory

United States: Phone: 1-408-965-3300 (8 AM to 5 PM PST)

                     Fax: 1-408-324-4840 (24 hours)

                     Email: techsupport@xbow.com

Non-U.S.:   refer to website   www.xbow.com

8.3 Return Procedure

8.3.1 Authorization

Before returning any equipment, please contact Crossbow to obtain a Returned Material Authorization number (RMA).

Be ready to provide the following information when requesting a RMA:

- Name
- Address
- Telephone, Fax, Email
- Equipment Model Number
- Equipment Serial Number
- Installation Date
- Failure Date
- Fault Description

8.3.2   Identification and Protection

If the equipment is to be shipped to Crossbow for service or repair, please attach a tag TO THE EQUIPMENT, as well as the shipping container(s), identifying the owner. Also indicate the service or repair required, the problems encountered, and other information considered valuable to the service facility such as the list of information provided to request the RMA number.

Place the equipment in the original shipping container(s), making sure there is adequate packing around all sides of the equipment. If the original shipping containers were discarded, use heavy boxes with adequate padding and protection.

8.3.3   Sealing the Container

Seal the shipping container(s) with heavy tape or metal bands strong enough to handle the weight of the equipment and the container.

8.3.4   Marking

Please write the words, "***FRAGILE, DELICATE INSTRUMENT***" in several places on the outside of the shipping container(s). In all correspondence, please refer to the equipment by the model number, the serial number, and the RMA number.

8.3.5   Return Shipping Address

Use the following address for all returned products:

Crossbow Technology, Inc.
  4145 N. First Street
  San Jose, CA 95134
  Attn: RMA Number (XXXXXX)

8.4   Warranty

The Crossbow product warranty is one year from date of shipment.

Crossbow

Crossbow Technology, Inc.
4145 N. First Street
San Jose, CA 95134
Phone: 408.965.3300
Fax: 408.324.4840
Email: info@xbow.com