

Stargate Tutorial

Last Updated Friday, August 3, 2007 by Tony Le.



• Contents

- [Overview](#)
- [Parts/Components](#)
- [Configure Hardware](#)
- [Configure Software](#)
- [Configure Wireless](#)
- [Check Network Settings](#)
- [Install Toolchain](#)
- [Writing & Compiling Programs](#)
- [Uploading Files](#)
- [Executing Files](#)
- [Download Files](#)
- [Final Thoughts & Useful Links](#)

• Overview

Keywords: stargate, wireless network, Crossbow, Zagi

Crossbow's Stargate SPB400 gateway is a single board computer designed for enhanced communication and sensor signal processing. Going beyond the design of a tradition single board computer, it also integrates applications related to TinyOS based wireless sensor networks and Smart Dust technologies.

This Stargate Tutorial will is the basic run through on how to setup the Stargate for uses with the Zagi and μ Nav sensor suite. The tutorial will explain how to configure wireless communication, install the avr-linux-gcc toolchain to compile and build C executables, and lastly run the *avionics* auto-pilot code.

• Parts/Components

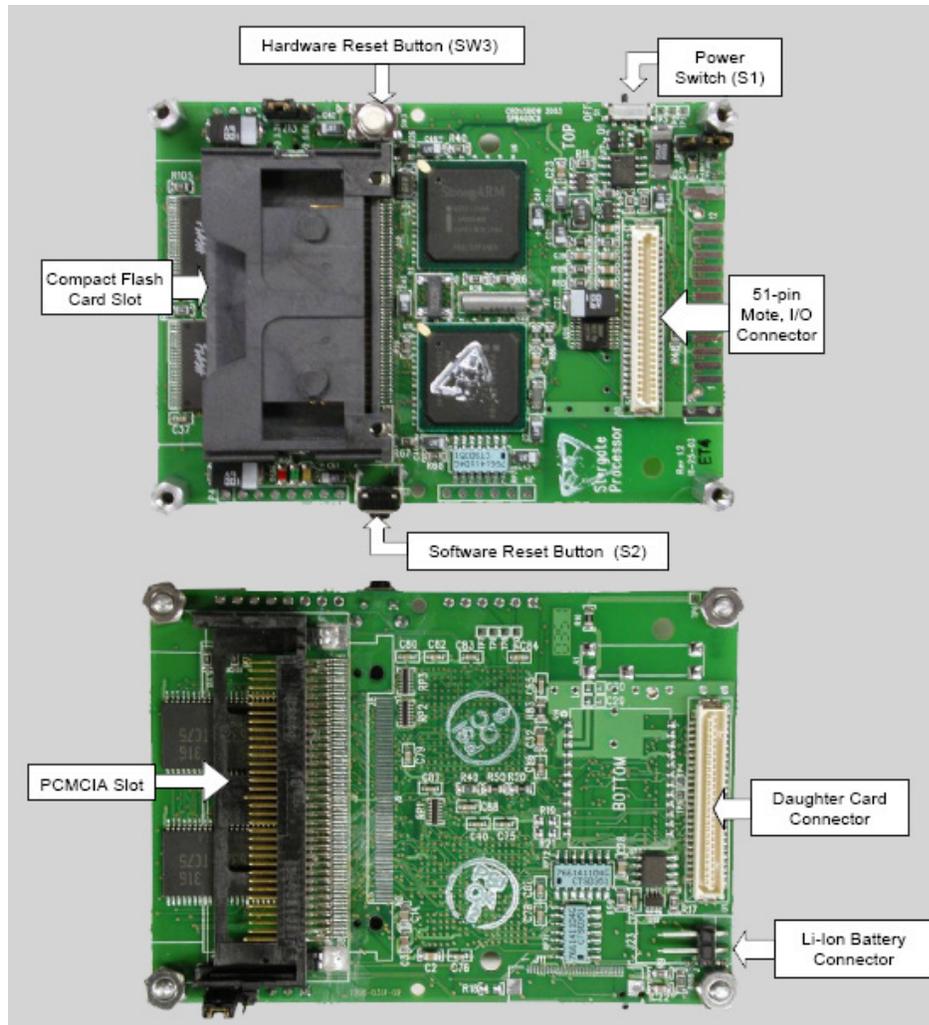
Table 1. Parts List

Part Description	Vendor	Part#	Price (2007)
Stargate Processor Board	Crossbow	SPB400CB	\$425.00
Stargate Daughter Board	Crossbow	SDC400CA	\$125.00
Stargate Power Supply	Crossbow		\$13.00
CF Wifi Card - 802.11b			
802.11			

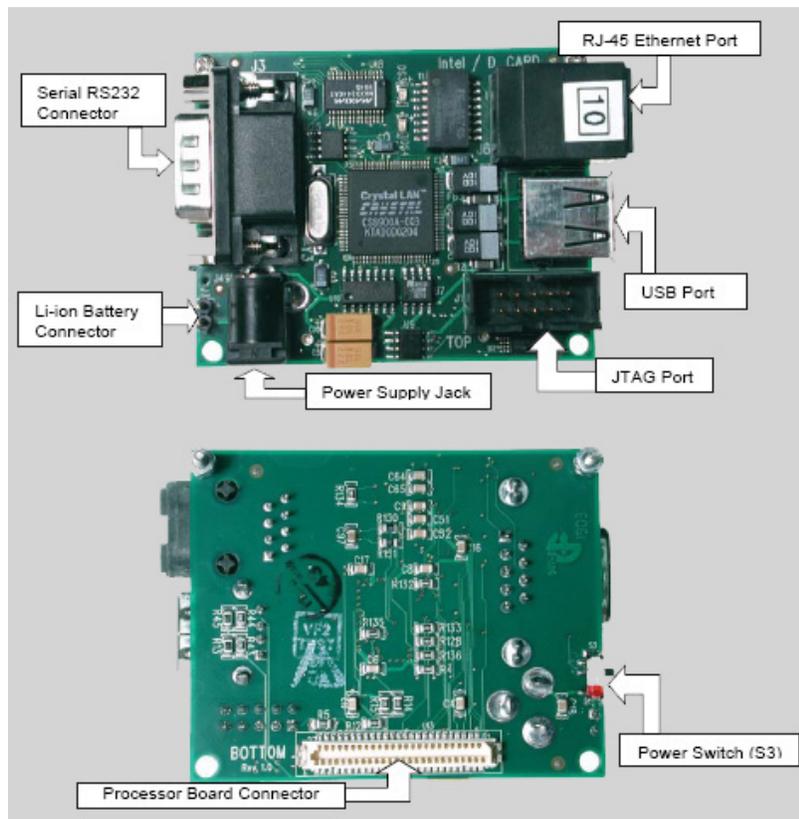
• Configure Hardware

- Required Hardware Components
 - Stargate Processor Board
 - 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.
 - Type II CompactFlash slot.
 - 16-bit PCMCIA slot.

- 51 pin expansion connector for Mica2, MicaZ, or μ Nav integration.



- Stargate Daughter Board
 - Wired Ethernet via a 10 Base-T Ethernet port
 - Host USB
 - JTAG port
 - External A/C power supply adapter
 - RS-232 serial port via DB-9 connector



- Null Modem RS-232 Cable & optional USB to Serial converter
- AmbiCom Wave2Net Wireless 802.11 Compact Flash Card
- Any working 802.11b supported wireless card
- DC Power Supply for Stargate and µNav

• **Configure Software**

- Establish Host Connection
 - Linux Host – Minicom

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:

- Log in as **root** on your Linux host machine.
- Type **minicom -s**
- The *Minicom* configuration menu appears.
- 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?
```

- Press **Enter** to return to the main configuration menu.
- Select **Save setup as dfl** to save these as default settings.

- Select **Exit from Minicom** to complete setup configuration.
 - When you need to enter *Minicom* again, you will be able to communicate with your target by using the command **minicom**
 - To get help on any Minicom commands, use the command **Ctrl+A-Z**
 - To exit Minicom, use the command: **Ctrl+A-X**
- Windows Host – HyperTerminal or SecureCRT
 - Establish a new Connection with the following COM Port settings:

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

- Power up the Stargate

- Power Settings

- If you are using an external power supply, you could use the power supply/adaptor provided with the Development kit. Connect the DC input of the AC wall power adapter to the **J49** connector on the daughter card. Make sure to check the jumper settings listed below.
- 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 is plugged in at **J23**.
- *Boot Prompt* - Once the power settings and host connection are properly established, the following message will appear on the console. This will verify a proper Stargate boot up. See [boot.txt](#).
- At this point you are ready to log into the target system.

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

```
stargate login: root
Password:
Linux stargate 2.4.19-rmk7-pxa2-star #1 Fri Aug 13 12:58:00
PDT 2004 armv5tel unknown
login[177]: root login  on `tts/0
```

- You are now logged in to the Stargate and ready to use it. Here are some basic linux commands that you can [use](#). Note: not all commands will be available since this is a compressed embedded linux version.

- **Configure Wireless**

- All file modifications will be made by the **vi** Editor. Here are some simple **vi** [commands](#).
 - Start vi using: "**vi filename**"
 - Save and write to file: "**:wq**"
 - Quit without saving any changes: "**:q!**"
- With the PCMCIA or Compact Flash wireless card inserted, use the **cardctl ident** command to identify the card.

```
stargate:/root# cardctl ident
Socket 0:
no product info available
Socket 1:
product info: "AmbiCom", "WL1100C 802.11b CF-Card", "2.2"
manfid: 0xd601, 0x0002
function: 6 (network)
```

- If the card is not identified the problem could be:

- That there is no driver associated with the card; therefore you must find a appropriate driver file that will work with a 2.4.19 Linux Kernel. Most likely you will have to install and rebuild the entire kernel. Refer to the Stargate manual for instructions for the 7.2 release.
 - If you are using an unsupported card, make sure if you are using a PCMCIA card that is a 16-bit card i.e. 802.11b. Also make sure to check the voltage ratings for card.
- Change your directory to `/etc/pcmcia` using the `cd` command.
 - We will first configure the wireless card. Using the vi Editor and the MAC Address of your wireless card, modify `wireless.opts` like the following:

```
case "$ADDRESS" in

# NOTE : Remove the following four lines to activate the samples below ...
# ----- START SECTION TO REMOVE -----
#*,*,*,*)
#    ;;
# ----- END SECTION TO REMOVE -----

# Here are a few examples with a few Wireless LANs supported...

# uNav Configuration
*,*,*,00:10:7A:*)
    INFO="uNav Configuration"
    ESSID="DASL"
    MODE="Ad-Hoc"
    #RATE="11"
    RATE="auto"
    CHANNEL="1"
    ;;
```

- Comment out (**#**) the two lines between the "SECTION TO REMOVE" lines
- Add a configuration for you wireless card like the uNav Configuration.
 - The last portion before the parenthesis is to associate the configuration to a card with a certain MAC address, used here is the first 3 sets of HEX values of the AmbiCom card.
 - The key things to make sure to have are the **ESSID** value for the network and also to set the card to an **Ad-Hoc MODE**. All the other values have defaults and are optional.
- Next is to configure the network. Again using the vi Editor, modify the file `/etc/pcmcia/network.opts` as seen:

```
# Network adapter configuration
#
# The address format is "scheme,socket,instance,hwaddr".
#
# Note: the "network address" here is NOT the same as the IP address.
# See the Networking HOWTO. In short, the network address is the IP
# address masked by the netmask.
#
case "$ADDRESS" in
*,*,*,*)
    INFO="Sample private network setup"
    # Transceiver selection, for some cards -- see 'man ifport'
    IF_PORT=""
    # Use BOOTP (via /sbin/bootpc, or /sbin/pump)? [y/n]
    BOOTP="n"
    # Use DHCP (via /sbin/dhccpd, /sbin/dhclient, or /sbin/pump)? [y/n]
    DHCP="n"
    PUMP='n'
    # If you need to explicitly specify a hostname for DHCP requests
    DHCP_HOSTNAME=""
    # Host's IP address, netmask, network address, broadcast address
    IPADDR="192.168.3.101"
    NETMASK="255.255.255.0"
    NETWORK="192.168.3.0"
    BROADCAST="192.168.3.255"
    # Gateway address for static routing
    GATEWAY="192.168.3.1"
    # Things to add to /etc/resolv.conf for this interface
    DOMAIN=""
    SEARCH=""
```

```

DNS_1=""
DNS_2=""
DNS_3=""
# NFS mounts, should be listed in /etc/fstab
MOUNTS=""
# If you need to override the interface's MTU...
MTU=""
# For IPX interfaces, the frame type and network number
IPX_FRAME=""
IPX_NETNUM=""
# Extra stuff to do after setting up the interface
start_fn () { return; }
# Extra stuff to do before shutting down the interface
stop_fn () { return; }
# Card eject policy options
NO_CHECK=n
NO_FUSER=n
;;
esac

```

- The key values to modify are:

- Set DHCP to no: **DHCP="n"**
- Host IP Address: **IPADDR="192.168.3.102"**
- Netmask: **NETMASK="255.255.255.0"**
- Network: **NETWORK="192.168.3.0"**
- Broadcast: **BROADCAST="192.168.3.25"**
- Gateway: **GATEWAY="192.168.3.1"**

- All other values are optional.
- (OPTIONAL) Now that you are using a wireless network communication, speed up boot-up by modifying */etc/init.d/networking*. The key thing is comment out the line *"pump -i eth0"*

```

#!/bin/sh
#
# start/stop networking daemons.

if ! [ -x /sbin/ifup ]; then
    echo "Unable to configure network. Missing /sbin/ifup."
    exit 0
fi

case "$1" in
    start)
        echo -n "Configuring network interfaces: "
        ifup -a
        #
        pump -i eth0
        ifconfig lo 127.0.0.1 up
        echo "done."
        ;;
    stop)
        echo -n "Deconfiguring network interfaces: "
        ifdown -a
        echo "done."
        ;;
    reload)

```

- Configuration is now complete, shutdown and restart the Stargate using the **shutdown -r now** command. This insures that all configuration files and hardware are properly shutdown and restarted.

• Check Network Settings

- After booting-up login into the Stargate again and check all the wireless network configurations you just did.
- Use the **iwconfig** command to verify the wireless settings.

```

stargate:/root# iwconfig
lo          no wireless extensions.

eth0       no wireless extensions.

```

Warning: Driver for device wlan0 has been compiled with version 14

of Wireless Extension, while this program is using version 16.
Some things may be broken...

```
wlan0      IEEE 802.11b  ESSID:"DASL"  Nickname:"stargate"  
Mode:Ad-Hoc  Frequency:2.412GHz  Cell: 02:10:81:7D:CE:2B  
Bit Rate:2Mb/s  Sensitivity=1/242700000  
Retry min limit:8  RTS thr:off  Fragment thr:off  
Encryption key:off  
Power Management:off  
Link Quality:0/1  Signal level:-100 dBm  Noise level:-100 dBm  
Rx invalid nwid:0  Rx invalid crypt:40  Rx invalid frag:0  
Tx excessive retries:8  Invalid misc:374  Missed beacon:0
```

- Use the **ifconfig** command to verify the network settings.

```
stargate:/root# ifconfig  
lo          Link encap:Local Loopback  
            inet addr:127.0.0.1  Mask:255.0.0.0  
            UP LOOPBACK RUNNING  MTU:16436  Metric:1  
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
            collisions:0 txqueuelen:0  
            RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)  
  
wlan0       Link encap:Ethernet  HWaddr 00:10:7A:58:CE:2B  
            inet addr:192.168.3.101  Bcast:192.168.3.255  Mask:255.255.255.0  
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
            collisions:0 txqueuelen:100  
            RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)  
            Interrupt:155
```

• Install Toolchain on PC

- Linux Installation

- This method is the ideal one since it will also accommodate the Inurse class. I haven't tried this on cygwin yet. I will update this once is done. For the time being, I am doing all my compiling in Linux. For those who prefer to not install Linux natively, consider using parallel OS programs like VMWare.
- The first step is to download all the required files:
 - Cross Compiler: [arm-linux-gcc-3.3.2](#) or [arm-linux-gcc-3.4.1](#)
 - Cross Tools: [crosstool-0.27-gcc-3.4.1](#)
 - NCurses: [ncurses-5.6](#)
- Save these files or move them to the top of your file system i.e. /
- Create the directory /usr/local/arm i.e. **mkdir /usr/local/arm**
- Extract the cross tool, [crosstool-0.27-gcc3.4.1.tar.gz](#), to the directory /
- Extract the cross compiler now from the / directory, it should default install to /usr/local/arm
- Add /usr/local/arm/3.3.2/bin to the PATH environment variable i.e. **export PATH=\$PATH:/usr/local/arm/3.3.2/bin**
- You can now compile using the `arm-linux-gcc` command
- Extract [ncurses-5.6.tar.gz](#) anywhere. NOTE: This is needed for MNAV versions 1.3+
- Navigate into the new extracted folder then run the following command(s):
./configure --host=arm-linux --target=arm-linux --build=i686-pc-linux (or i686-pc-cygwin) --without-ada --prefix=/usr/local/arm/3.3.2/
- **make**
- **make install**
- Copy the ncurses libraries in /usr/local/arm/3.3.2/lib to /usr/local/arm/3.3.2/arm-linux/lib
cp /usr/local/arm/3.3.2/lib/libncurse* /usr/local/arm/3.3.2/arm-linux/lib
- All these steps are very time and resource consuming. For more information, read the INSTALL file in the ncurses-5.6 folder. But this is needed for the auto-pilot "GUI" on the Stargate.

- Windows Installation

- This method I have yet to personally test but should be some what similar to that of the Linux installation. The difference is you will be using `cygwin` as your command line tool. The cross compiler will be different since you are using Windows.

- You will need the following files:
 - [Cygwin](#)
 - Cross Compiler: [arm-cygwin-gcc-3.3.2](#)
 - NCurses: [ncurses-5.6](#)
- The installation process is the same as above for the Linux version but instead of using the Linux Bash Terminal you will use *cygwin*. Therefore, *cygwin* will have to be installed first before you can install the other components.
- Follow the ncurses library files as instructed above but during the `configure` stage set the build to `i686-pc-cygwin`.
- The installation process will be very time consuming, be patient. If the make or make install process fails, just retry if you are getting a random Win32 error.
- Before compiling the files, you will need to make sure the ncurses libraries are in the right place. To do so run the following command **`cp /usr/local/arm/3.3.2/lib/libnc* /usr/local/arm/3.3.2/arm-linux/lib`**

• Writing & Compiling Files

- Compiling files for the Stargate cannot be done onboard. You must compile the files from your PC using the toolchain described above.
- You can find an example "Hello World" program and compile instructions are found [here](#).
- For the MNAV auto-pilot, you will only have to use the included `makefile`
- Use the **make** command to compile and build *avionics* and *decoder* executable files.
NOTE: This will only work if you have make package and toolchain installed.
- The command **make clean** removes all the objective files that were created during the last compilation and the *avionics* executable.

• Upload Files to the Stargate

- After compiling and building the executable *avionics* file, you can then upload it to the Stargate.
- The first step is to establish a wired communication with the Stargate using HyperTerminal or Minicom. The instructions to do that is in the "Configure Hardware" section listed above.
- The Zmodem protocol is used to transfer files to the Stargate. The method copies file to the current directory on the Stargate. NOTE: It does NOT overwrite existing files.
- After logging into the Stargate, make sure navigate to the directory where you want the files to be copied to then do the following:

- For HyperTerminal:

- Select the "**Transfer**" menu
- Select the "**Send File**" menu item.
- Set the protocol to Zmodem
- Browse for the file of interest
- Press the "**Send**" button

- For Minicom:

- **<ctrl> A**
- **s** -- send command
- **<return>** -- select Zmodem
- **<up/down arrow keys>** -- to move the cursor up and down
- **<space><space>** -- to move into the directory at the cursor
- **<space>** -- selects the file at the cursor
- **<return>** -- starts the transfer
- -- wait until it completes
- **<any key>** -- exit

- After uploading the *avionics* file, you will have to make sure it is executable by running this command in the same folder **`chmod +x avionics`**

• Executing the MNAV Auto-Pilot on the Stargate

- To use the MNAV auto-pilot, just run the **avionics** executable.
- There are a few options that you can use with the *avionics* code, you can list them all using **avionics -help**

```
./avionics -option1 -option2 ...
-wifi on/off : enable or disable WiFi communication with GS
-save on/off : enable or disable datalogging in /mnt/cfl/
-screen on/off : enable or disable dumping data to screen
```

```
-ip xxx.xxx.xxx.xxx : set GS i.p. address for WiFi comm.  
-help : display the help messages
```

- To use the new GUI that uses ncurses also needs you to do the following:
 - **cp** proper terminal to the stargate `/usr/local/arm/3.3.2/share/terminfo`
 - The stargate already has `/terminfo` in `/etc` directory.
 - The reason that `/terminfo` needs to be installed in `/usr/local/arm/3.3.2/share/terminfo` is simply because of `--prefix=/usr/local/arm/3.3.2`
 - If you used a different `--prefix` directory, then you need to install `/terminfo` to the corresponding directory.
 - `vt100`, `cp vt100` to `/usr/local/arm/3.3.2/share/terminfo/v/.`
 - `cygwin`, `cp cygwin` to `/usr/local/arm/3.3.2/share/terminfo/c/.`

- **Download Files from the Stargate**

- For detailed instructions to download Data Recorded from the MNAV auto-pilot, see my [SCP/Stargate Tutorial](#).

- **Final Thoughts & Useful Links**

The Stargate is a very versatile single-board computer. It's strengths lie in its size, it's 32-bit processor, and the fact that it has Linux already embedded to it. I've used it for interfacing with Motes and the μ Nav. This tutorial basically runs through its basic use and configuration to use it with the MNAV autopilot. The next tutorial will explain how to set-up the μ Nav and connect it to the Stargate.

- *Useful Links*
 - [MNAV Auto-Pilot Sourceforge Website](#)
 - [Stargate Datasheet](#)
 - [Stargate Crossbow Manual](#)
 - [PlatformX Sourceforge Website](#)
 - [Stargate User Help Mailing List](#)
 - [Contact me](#)