0.1 BeagleBone Black User guide

The most common way to interact with the BeagleBone Black (BBB) is by connecting to a terminal on it via a host computer. The software needed to do it differ depending on the operating system of the host.

0.1.1 Linux host

The Linux terminal can connect to a terminal in a BBB via Ethernet via the ssh command. SSH is included in most Linux distributions by default.

```
$ ssh ubuntu@192.168.1.1
```

On this line, ubuntu is the user name and 192.168.1.1 is the IP of the BBB. A prompt will ask for a password, the password is ubuntu.

0.1.2 Windows host

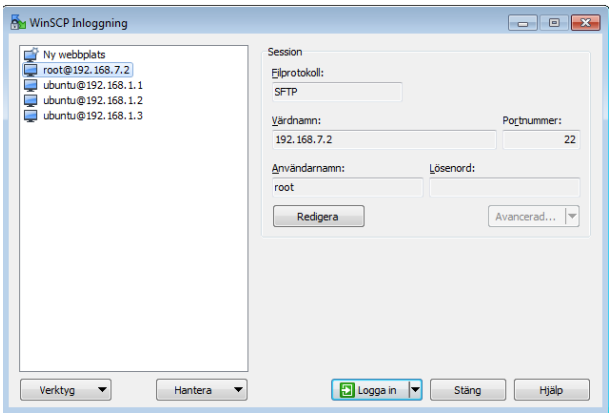

Figure 1: WinSCP

WinSCP[**?**], Secure Copy (SCP), is an application for managing files between Windows and BBB easier. One can move files freely between the both systems without terminal commands. If connected through Ethernet over the switch use hostname (”vrdnamn” in the picture) 192.168.1.n where n is dependent on which of the BBB one wants to access. If connected through USB use hostname 192.168.7.2. With drivers, this can also share connection to internet for installation of applications etc in BBB. Username (”Anvndarnamn” in the picture) is either root or ubuntu. If ubuntu then password is ubuntu.

PuTTY [**?**] is an application to Secure Shell (SSH) in to BBB from Windows. This gives the possibility to remote control the BBB. By doing this one can compile and run applications.

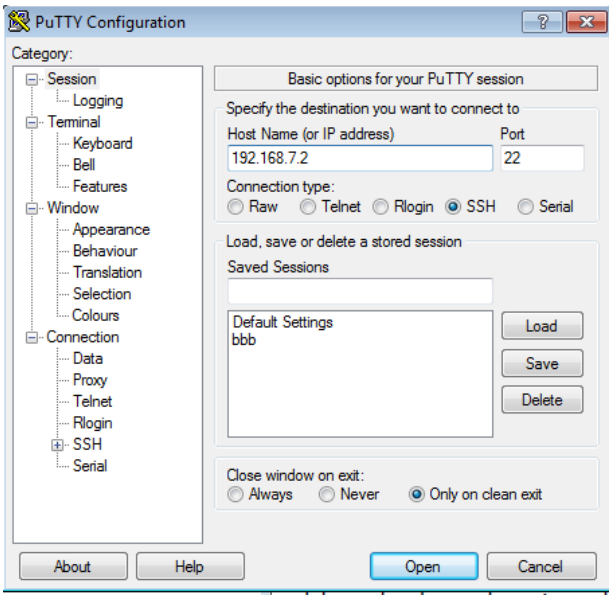

Figure 2: PuTTY

0.1.3 Navigating in a Linux environment

When connected, the user has to navigate in the Linux environment on BBB. This is not effected by the operating system of the host.
Navigation in Linux is mostly done with two commands:

```
$ ls
```

This command ls show all the folders in the current directory.

```
$ cd Software/
```

The cd command is used to enter a folder.

```
$ cd ..
```

The folder name .. will move up one level. Note that Linux is case sensitive regarding names.

All of the software is located in the folder /home/ubuntu/BBB/software/. All the binary files should be located in a folder called 'bin' within the folder Software. To transfer files or folders between a computer and a BBB, use the following command:

```
$ scp -r pid/ ubuntu@192.168.1.1:BBB/software/
```

The flag -r tells the program that the source 'pid/' is a folder while ubuntu@192.168.1.1 is the user on the target computer, in this case a BBB. The colon is the separator where the path target path begin. The default folder will be /home/ubuntu.

By adding BBB/software/, the folder will end up at /home/ubuntu/BBB/software/pid with all its content.
If gcc is installed, a project can be compiled with gnatmake.

```
$ gnatmake -P mainproject.gpr
```

Or to only compile a single file:

```
$ gnatmake main.adb
```

To make the process easier, a makefile can be used. Here is an example with three commands. Note that the formatting is important.

```
this:
	gnatmake -P mainproject.gpr
all:
	gnat clean -r -P mainproject.gpr
	gnatmake -P mainproject.gpr
clean:
	gnat clean -r -P mainproject.gpr
```

The command make will run the first command in the makefile, 'this'. With other words, the project mainproject.gpr will be compiled. To run another command, add the name after make.

```
$ make all
```

This will clean the project and rebuild it from scratch. Multiple makefiles can be combined with a shell script so all of them can be compiled by just running the shell script.

```
$ make all -C software/myprojectfolder/
```

This command will run a makefile located in the folder software/myprojectfolder/, multiple lines can be added to compile mulitple projects. To run an executable file, use ./ before the name.

```
$ ./main
```

To exit a program, press ctrl + c. To check which processes are running use ps aux.

```
$ ps aux
```

There are currently a script file with some predefined functions. The file system.sh is a shell script located at /home/ubuntu/BBB/. It can make some tasks easier. It got 5 commands, kill, run, build, restart and move. The command kill together with a name will end that process.

```
$ ./system.sh kill pid
```

This will end the PID controller. To kill all the running nodes, leave leave the name empty. All of the commands except build can be used this way. With other words, the run command will look like this.

```
$ ./system.sh run pid
```

The run command will start the node as a background process but it will still output all the messages to the terminal. To hide the output, redirect the standard output with a pipe to null.

```
$ ./system.sh run pid > /dev/null
```

The build command will compile all projects. If the parameter all is added, all projects will be cleaned first. The build command does also move all the executable files to a specific binary folder. This folder is used by run and restart. The restart command simply kills a specified process and tries to run it again. Finally, the command move will only copy the executable file(s) to the binary folder.