Chipsee

Industrial PC

Debian 8.10 OS on iMX6UL User Manual

For iMX6UL Products

Content can change at anytime, check our website for latest information of this product.
www.chipsee.com

Contents

Debian 8.10

Chipsee Debian 8.10 User Manual

Revision	Date	Author	Description
V1.0	2018-4-14	Madi	Initial Version

SUPPORTED BOARDS:

CS10600U070-V1.0

PREBUILT FILES PACKAGE:

`prebuilt-cs10600u070v1-debian-emmc-20210201.tar.gz`

System Features

Feature	Comment
Kernel	Kernel version: 3.14.52
Bootloader	Uboot 2015.04
System	Debian8.10
Python	Python version: 2.7.9
Qt	Needs to be installed
Desktop	lxde
user/password	[root/root] or [chipsee/chipsee]

Preparation

You will need to prepare the following items before you can start using the Prebuilt Files Package to reflash the system.

- Power Supply Unit (PSU) with the appropriate voltages, as follows:
  > - Products with 7” display panel require 6V to 36V PSU
  > - Products with 10” display panel and larger require 15V to 36V PSU
- USB to serial cable for debugging Chipsee Industrial Embedded Computers (Chipsee IPC)
- TF Card to create a bootable storage for reflashing the system

Use the Prebuilt Files Package from the link above to reflash the system. You can use the XShell terminal emulator to debug Chipsee IPC in Windows. You can also use VNC® Viewer to control Chipsee IPC remotely, over Ethernet. The Cross-toolchain software is used to compile the software for flashing.

Hardware Requirements

- Chipsee Industrial (Embedded) Panel PC
- PSU according to the instructions above
- USB-to-serial or other serial cable for debugging
- TF Card (at least 4GB)

Software Requirements

- Prebuilt Files Package (from the link above)
- XShell or similar terminal emulation software
- Cross-toolchain
- VNC-Viewer

Debug

This documentation covers the use of XShell terminal emulation software to debug Chipsee IPC. However, you can use other tools as well, such as SecureCRT or Minicom.

Serial Debug

The first serial port is used for debugging (serial port 1). It consists of *RS232_1_TXD*, *RS232_1_RXD* and, *GND* terminals. Please refer to 1.6.1. RS232/RS485/CAN chapter in the EPC/PPC-A7-070HB-C Hardware Documentation for additional information on serial ports.

> **Note**
>
> More information on how to connect different Chipsee IPCs to a personal computer (PC) via a serial connection can be found in the PDF document below:
>
> `How to Connect Board by Serial.pdf`

After the connection is successfully established, set up the XShell terminal as shown in figures below:

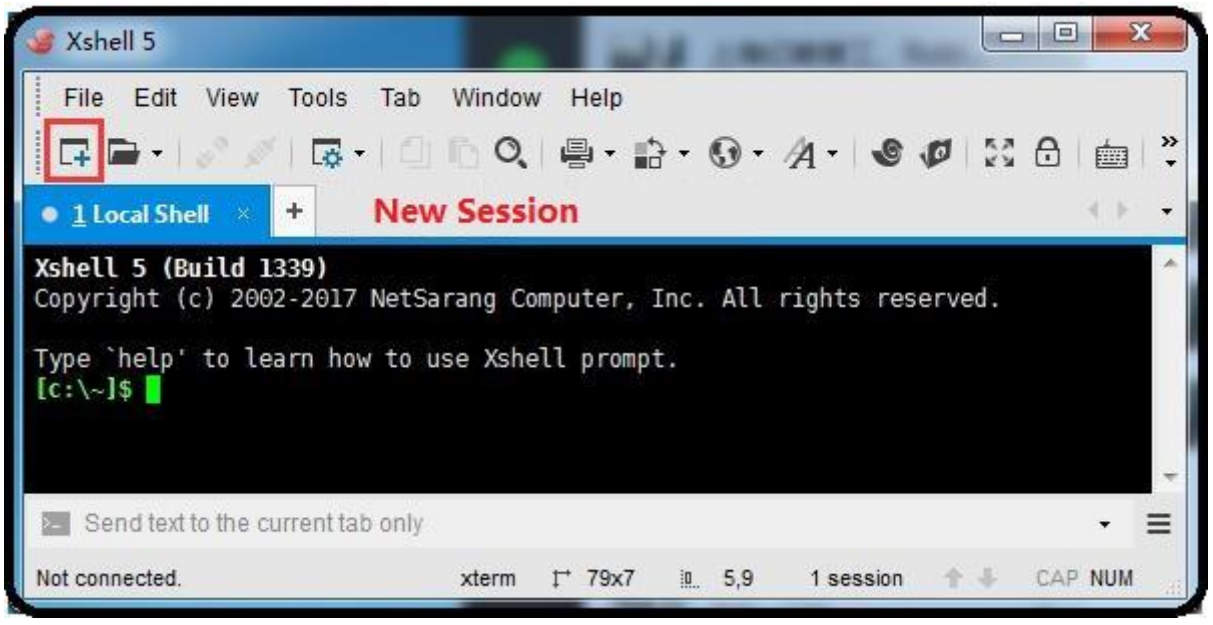

Figure 342: Figure 1: Add Session

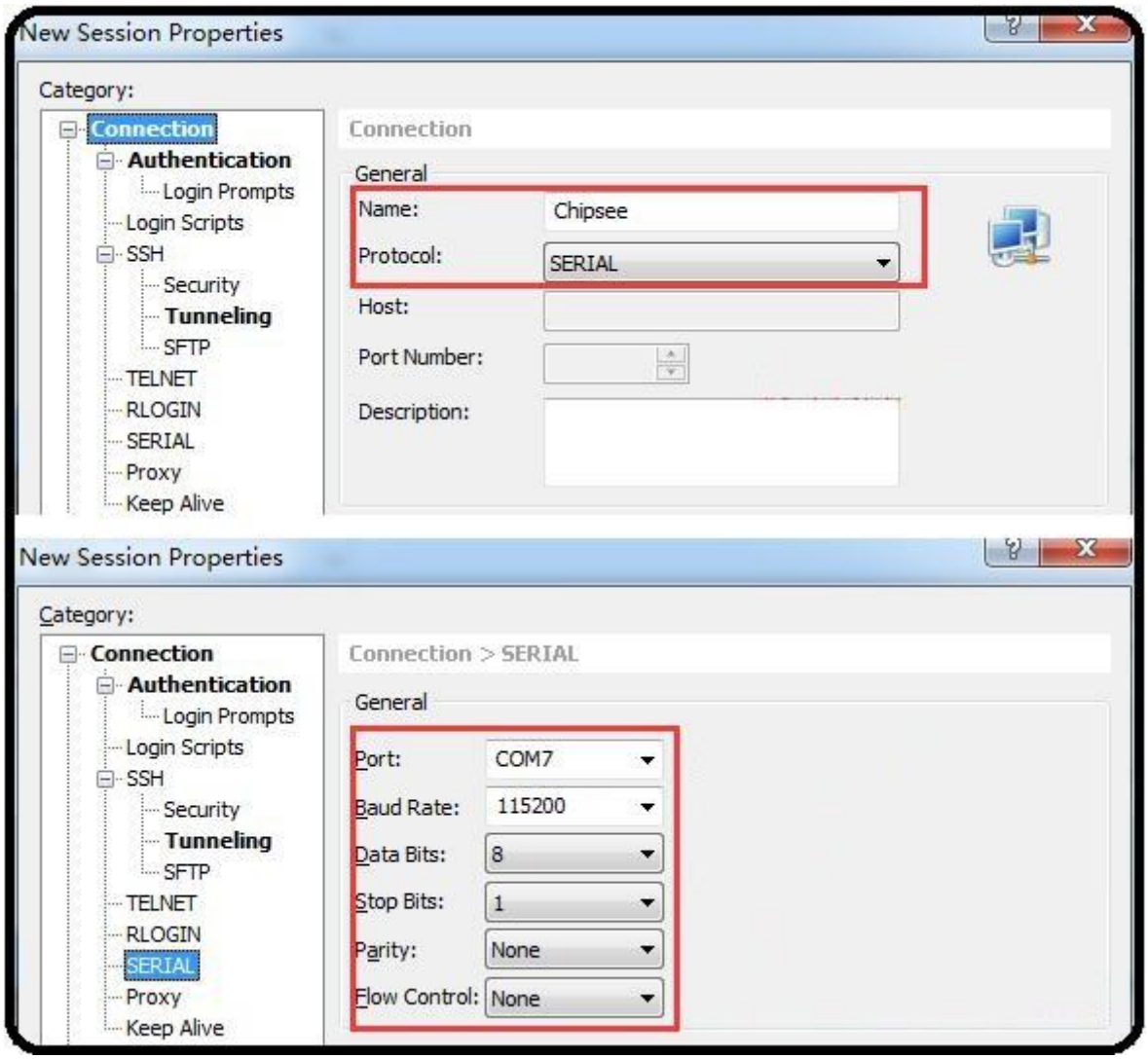

Figure 343: Figure 1a: Session Properties

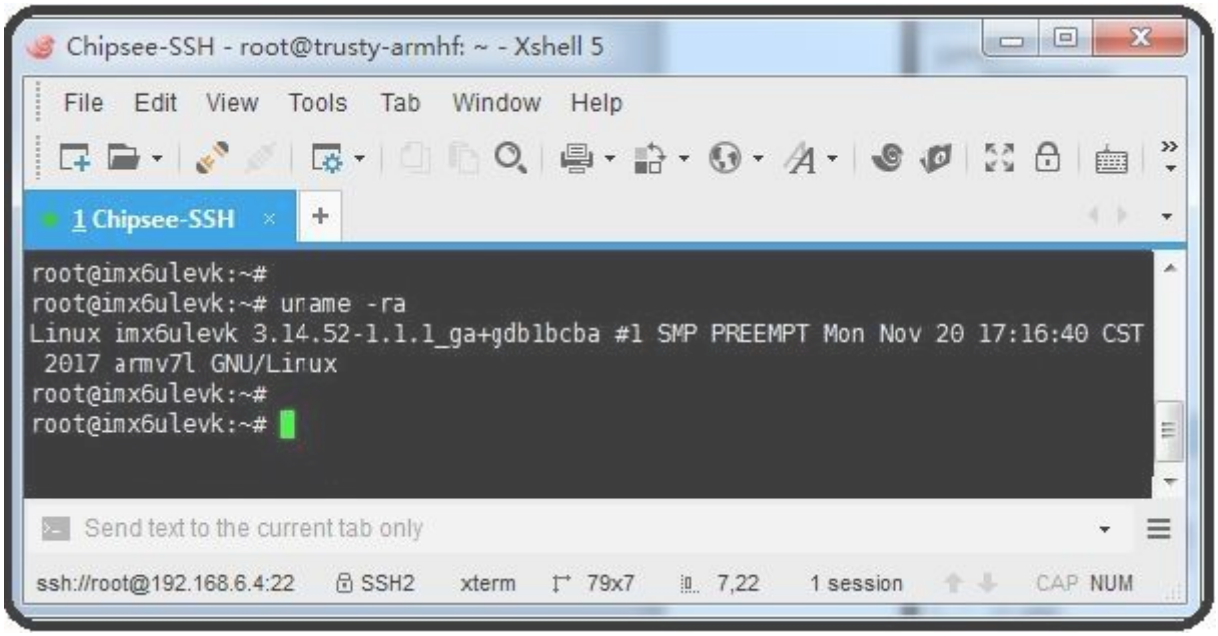

Figure 344: Figure 1b: Serial Debug

SSH Debug

Connect Chipsee IPC to the Internet and get the IP address. Then config XShell or use the SSH tool on the Linux PC host directly. In this documentation, we will cover XShell SSH debugging procedure.

You must first add a new session, as shown in *Figure 1*. The new session should be set as in *Figure 2* below.

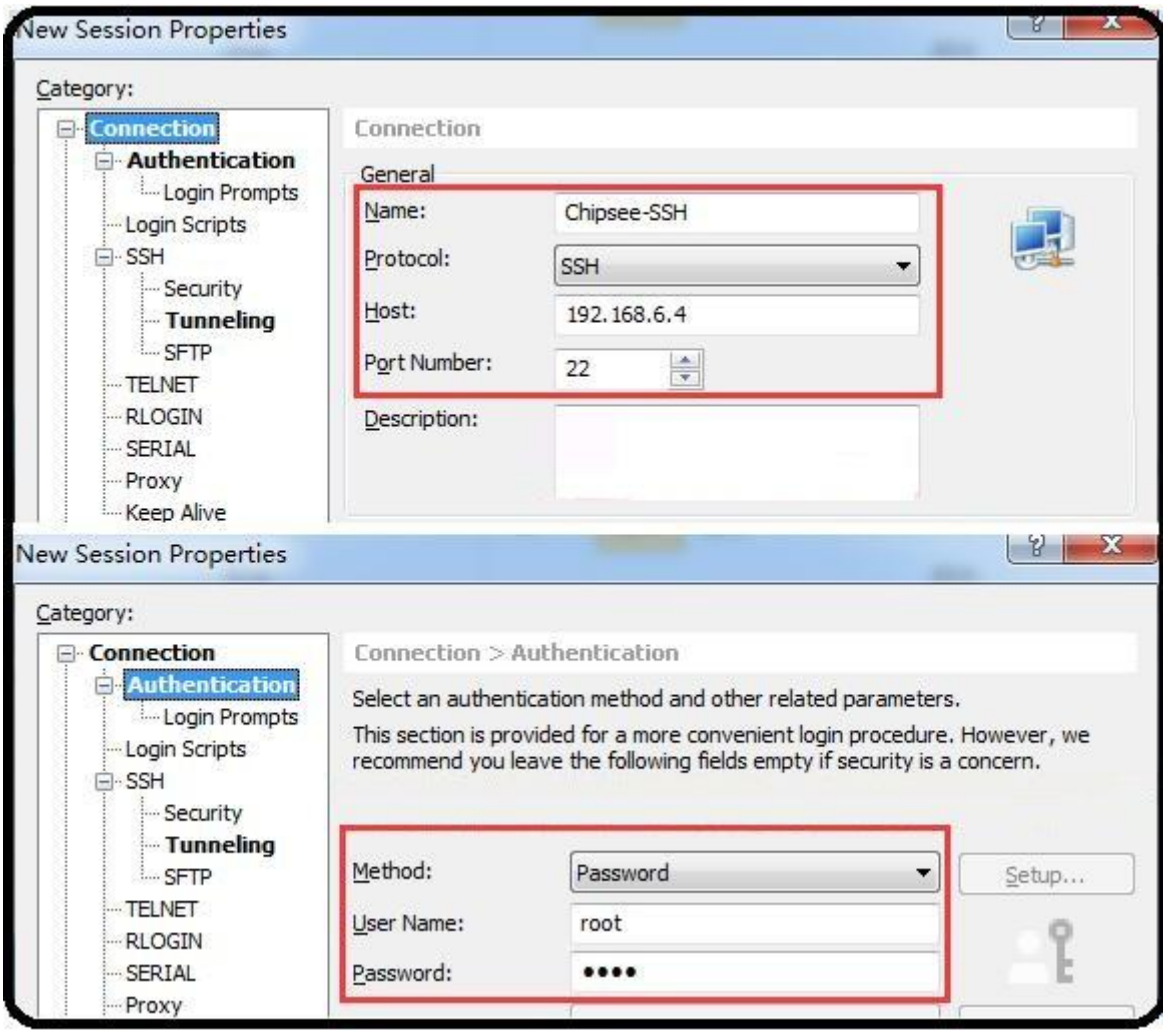

Figure 345: Figure 2: SSH Settings

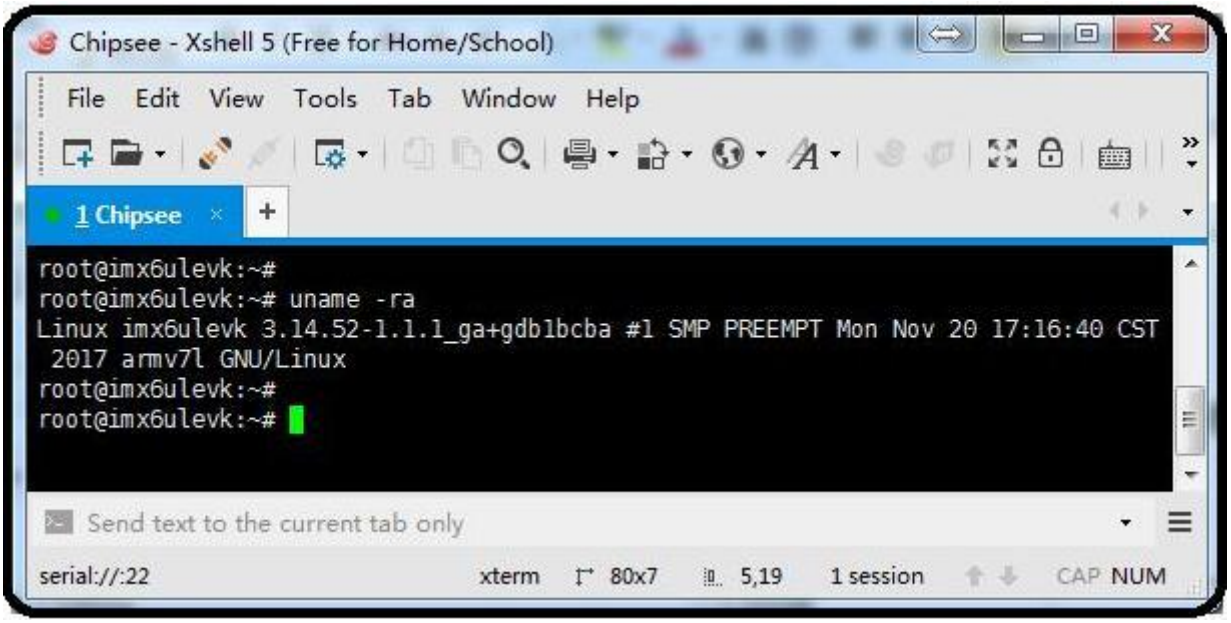

Figure 346: Figure 2a: SSH Debug

VCN Debug

You can use VNC Viewer in Windows to control Chipsee IPC over Ethernet, as mentioned above.

- Use XShell serial or SSH to connect to Chipsee IPC
- Login with the default credentials, using the commands below
- The default login credentials are: `chipsee/chipsee`

$ x11vnc -storepasswd
- -set password for VNC-Viewer access–
$x11vnc -display :0 -forever -bg -rfbauth /home/chipsee/.vnc/passwd -rfbport 5900 -o /home/chipsee/.vnc/x11vnc.log

- Use VNC Viewer in Windows to control Chipsee IPC over Ethernet, as shown in figures 2b, 2c, and 2d.

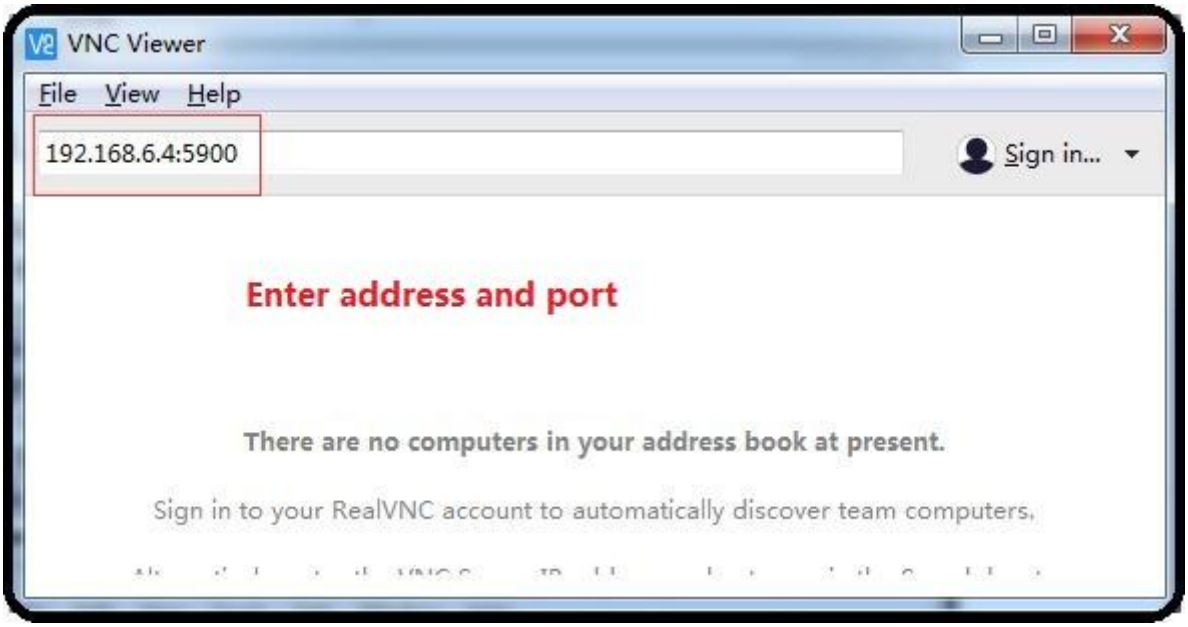

Figure 347: Figure 2b: VNC Viewer Connect

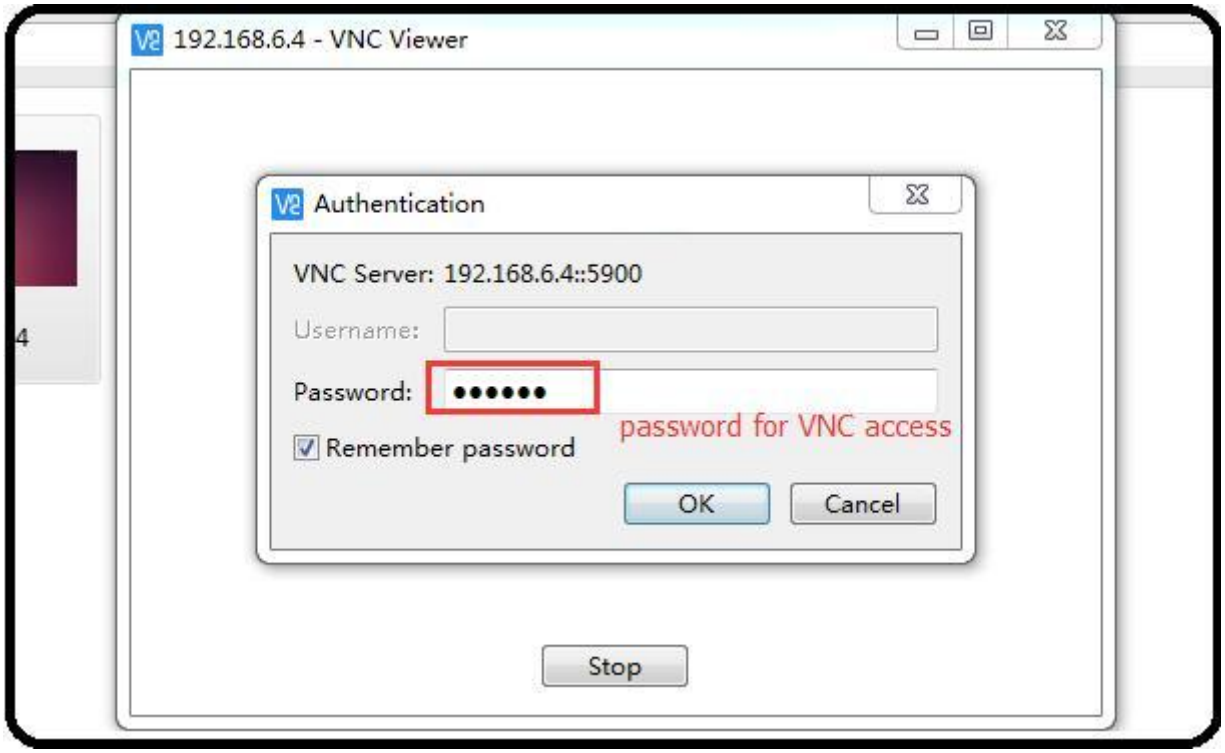

Figure 348: Figure 2c: Authentication

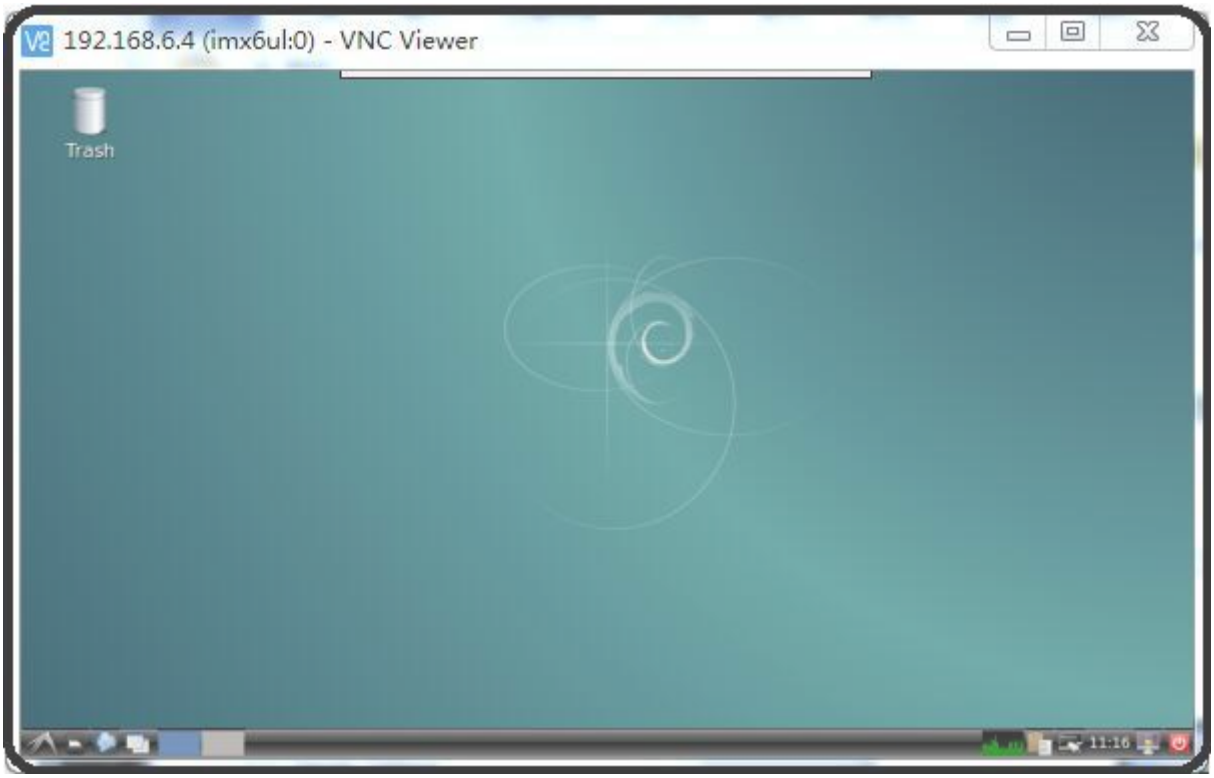

Figure 349: Figure 2d: VNC Desktop

Downloading Images

Chipsee IPC supports booting from an integrated eMMC or an external TF Card (also known as the micro SD card). Booting from the external TF Card allows flashing the system OS.

DIP Switch Configuration

Set the boot DIP switch as shown in *Figure 3* to boot the system from the external TF Card.

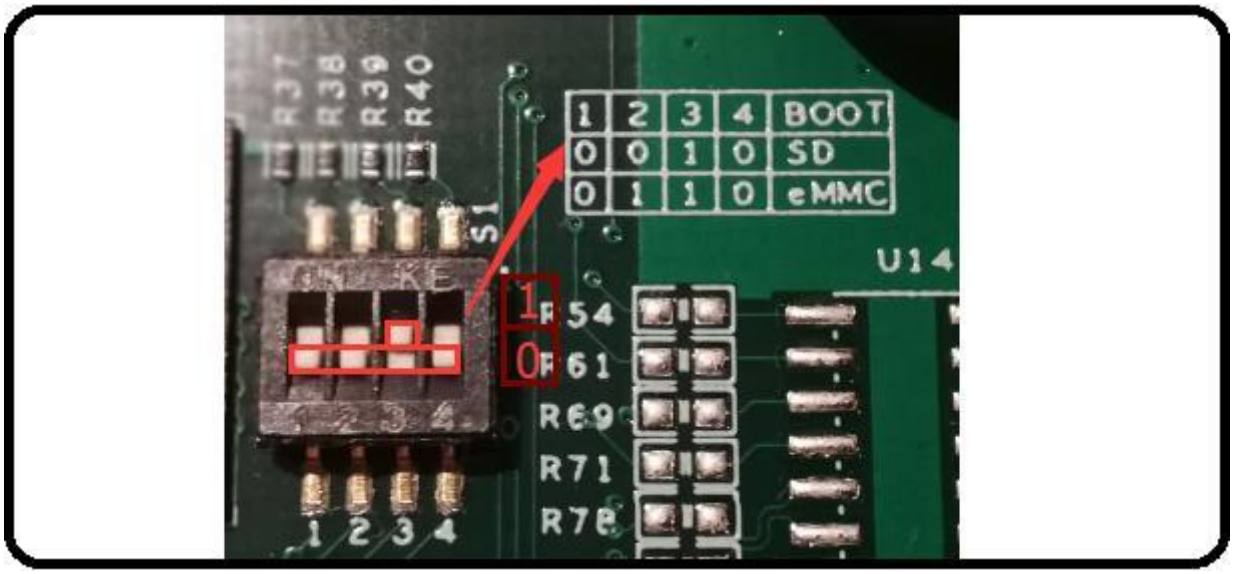

Figure 350: Figure 3: Boot Mode Setup

Prebuilt Files Package

You can get the Prebuilt Files Package from the Prebuilt Files Package link mentioned at the beginning of this documentation. You can also get the Prebuilt Files Package from the DVD in /Debian8.10/Prebuilds folder. However, it may be outdated so always compare the versions (the last number in the filename is the release date).

The prebuilt package has the following content (*Table 1*):

Contents	Comment
boot/imx6ulipc.dtb	TF Card boot dtb file
boot/u-boot.imx	TF Card boot bootloader
boot/zImage	TF Card boot kernel file
filesystem/rootfs-emmc-flasher.tar.bz2	TF Card boot rootFS
mksdcard.sh	Shell tools to make bootable TF Card
README	Simple guidelines
S1.jpg	Boot Switch Config Figure

Contents	Comment
emmc-flash/emmc/rootfs.tar.gz	RootFS in target eMMC
emmc-flash/emmc/u-boot.imx	Bootloader in target eMMC
emmc-flash/emmc/zImage	Kernel file in target eMMC
emmc-flash/emmc/imx6ul-eisd.dtb	dtb file in target eMMC
emmc-flash/mkemmc.sh	Shell tools to download images

Table 73 Table 1: Prebuilt Files Package

Note

The default `zImage` and `imx6q-sabresd.dtb` files support *'keep the logo from uboot to kernel'* but do not support framebuffer. Chipsee provides `zImage_framebuffer` and `imx6q-eisd.dtb_framebuffer` file versions that support the framebuffer function but do not support the *'keep the logo from uboot kernel'* feature. If you need the framebuffer, just rename these two files to `zImage` and `imx6q-eisd.dtb`.

Downloading images onto the TF Card

The Prebuilt Files Package has a shell tool that can help create a bootable TF card on the Linux platform (such as desktop PC or Virtual Machine running Ubuntu 14.04 distribution). Use the TF Card to download the bootable system image onto it:

- Copy the Prebuilt Files Package to a Linux environment (such as Ubuntu 14.04)
- Insert the TF Card and check the device node, (e.g., `/dev/sdc` or `/dev/sdb`, be sure to use the right one)
- Un-tar the prebuilt package and use the following command:

$ sudo ./mksdcard.sh –device /dev/sdc

- The bootable TF Card is now ready. Power OFF the IPC and insert the TF Card
- Set the DIP switch to SD BOOT mode (refer to *Figure 3* above)
- Power ON the IPC: the message below indicates that the system image was downloaded correctly to the eMMC

>>>>>>> eMMC Flashing Completed <<<<<<<

- Power OFF the IPC and set the DIP switch to eMMC BOOT mode (refer to *Figure 3* above).

System Resources

This chapter covers the resources available on Chipsee IPC.

TF Card/USB Storage

Both the TF Card and USB storage support the hot plug functionality. They will be automatically mounted on `/media/chipsee/`, as in *Figure 4*. Also, both storage types support NTFS and FAT32 file system.

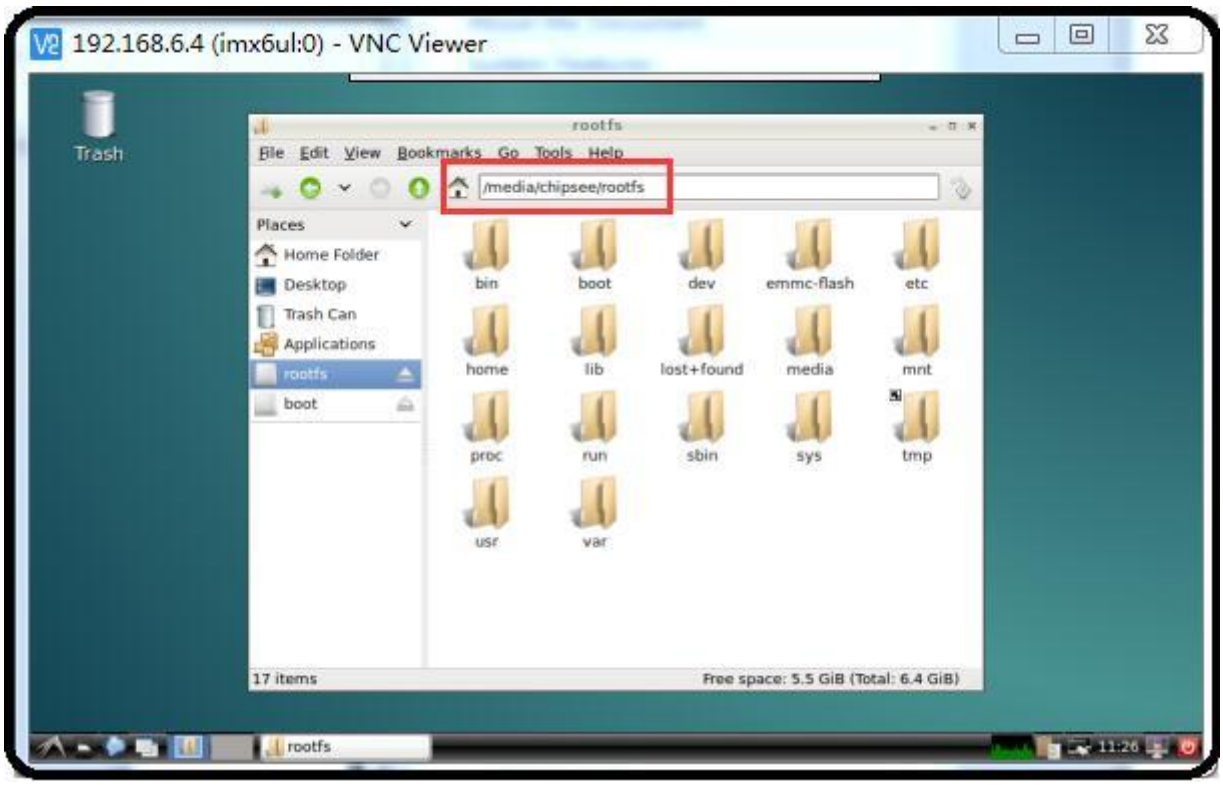

Figure 351: Figure 4: TF Card Contents

Network

The system uses WICD Network Manager to control Ethernet configuration. You can get the assigned IP address from DHCP, or you can set static IP. After you set the static IP, reboot the system to enable it (*Figure 5a* and *Figure 5b*):

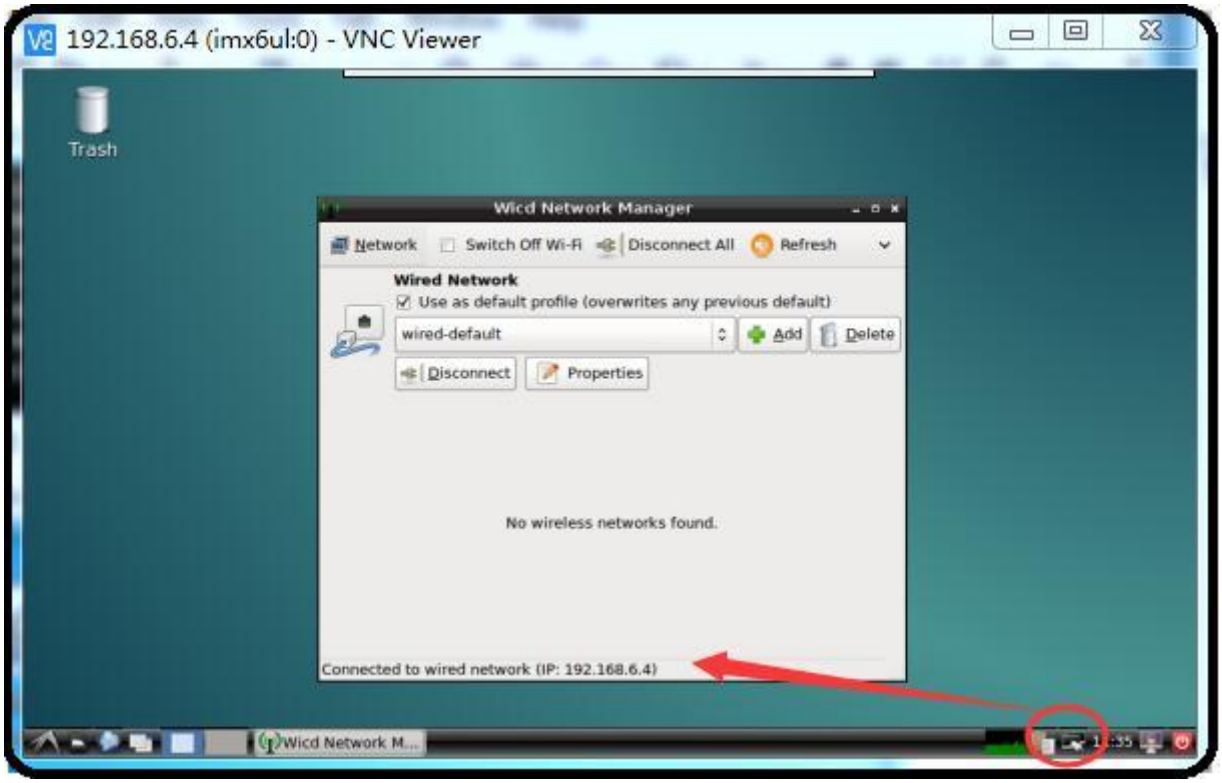

Figure 352: Figure 5: Ethernet Settings (Wired Network Manager)

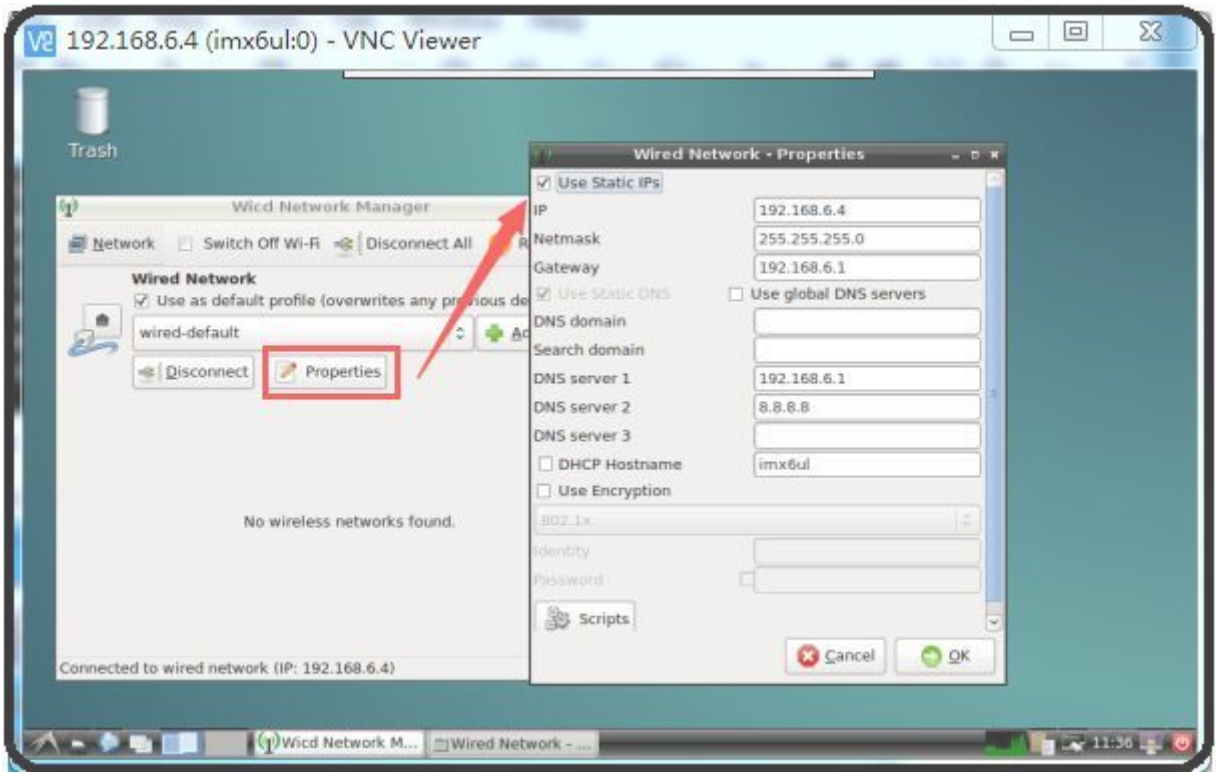

Figure 353: Figure 5a: Setting up Static IP

Sound

The following command example is used to record sound:

$ arecord -N -M -r 44100 -f S16_LE -c 2 -d 18 test.wav

The example above interrupts recording after 18 seconds (set by the `-d` parameter), records sound at a sampling rate of 44100 kHz (the `-r` parameter), and saves it as the `test.wav` file.

The following command can be used to playback the recorded sound from the example above:

$ aplay -N -M test.wav

Serial Port

There are five serial ports on the Chipsee IPC: 2 X RS232 and 3 X RS485. Refer to *Table 2* below for the available serial device nodes.

Ports	Device Node
RS232_1	/dev/ttymxc0
RS232_2	/dev/ttymxc1
RS485_3	/dev/ttymxc2
RS485_4	/dev/ttymxc3
RS485_5	/dev/ttymxc4

Table 74 Table 2: Serial Ports Device Nodes

- You can install the CuteCom serial terminal to test the serial ports by using the following command:

$ sudo apt-get install cutecom

- Only the root user can use the serial ports:

$ sudo cutecom

> **Note**
>
> 120Ω termination resistors are not mounted or included with the device.

CAN

Chipsee Industrial PC is equipped with two CAN busses (CAN1 and CAN2). You can test the CAN busses by using the HT application. Two devices can be interconnected as on the *Figure 6* below:

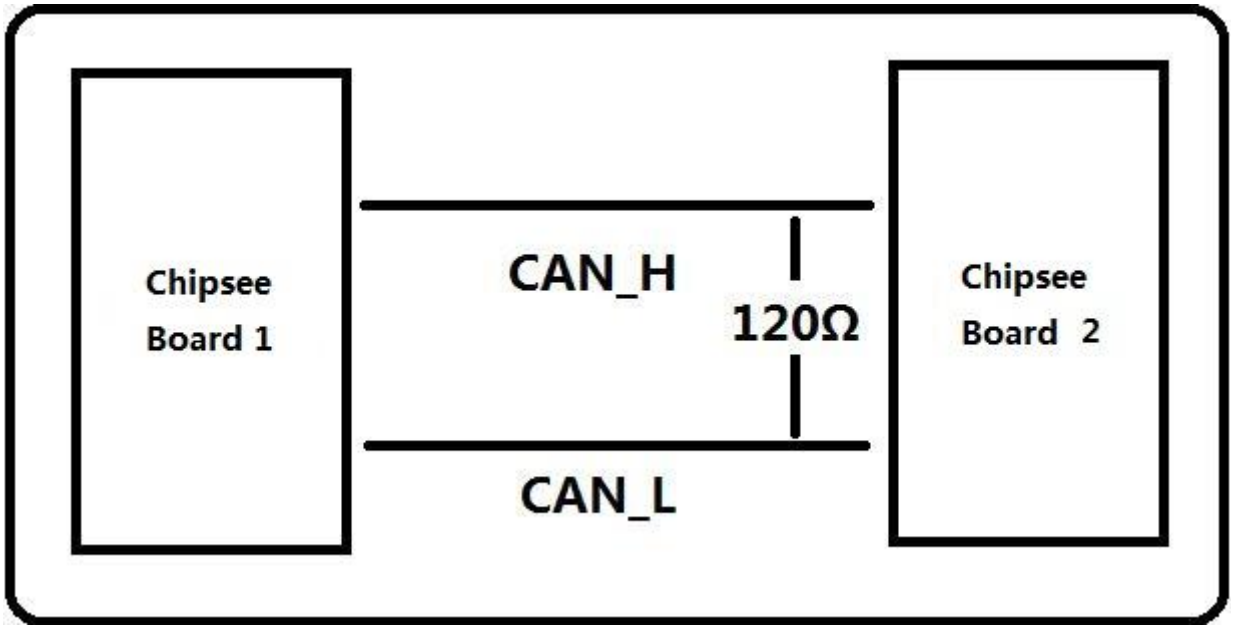

Figure 354: CAN connection

The following example can be used to perform testing:

- Set the bit-rate to 50kbps with triple sampling, using the following command as the root user:

ip link set can0 type can bitrate 50000 triple-sampling on

- Bring up the device using the command:

ip link set can0 up

- Transmit 8 bytes with standard packet ID number as 0x10

cansend can0 010#1122334455667788

- Transmit 8 bytes with extended packet id number as 0x800

cansend can0 800#1122334455667788

- Bring down the device

ip link set can0 down

- Receive packets

```
#candump can0
```

GPIO Ports

There are 8 GPIO ports on the Chipsee IPC, as explained in the GPIO chapter of the EPC/PPC-A7-070HB-C Hardware Documentation. The table below contains the related device nodes:

Pin Number	Definition
1	VDD_24V
2	GND_ISO
3	/dev/chipsee-gpio1(out)
4	/dev/chipsee-gpio2(out)
5	/dev/chipsee-gpio3(out)
6	/dev/chipsee-gpio4(out)
7	/dev/chipsee-gpio5(in)
8	/dev/chipsee-gpio6(in)
9	/dev/chipsee-gpio7(in)
10	/dev/chipsee-gpio8(in)

Table 75 Table 3: GPIO Ports

You can use the following commands to test the GPIOs easily:

- Set GPIO1 to HIGH logic level:

```
# echo 1 > /dev/chipsee-gpio1
```

- Set GPIO2 to LOW logic level:

```
# echo 0 > /dev/chipsee-gpio2
```

- Check the input level on GPIO5:

```
# cat /dev/chipsee-gpio5
```

Development

In this chapter, you can learn how to set up QT development environment and develop the first QT application on CS10600U070 IPC.

Set Environment

By default, there is no Qt and build environment set up in the system. Before you start the development, you need to install the environments by using the following set of commands:

$ sudo apt-get update $ sudo apt-get install build-essential git libudev-dev $ sudo apt-get install qt5-default // or qt4-default if you want to use qt4 $ sudo apt-get clean

Prepare Source Packages

There are some Qt source demo packages on the provided DVD in the `/Debian8.10/QT/` folder. You can use SSH or USB storage to transfer them to Chipsee IPC.

Build & Run

We will use the `hardwarewaretest_serial_ok_20170223.tar.gz` demo package to demonstrate how to build and run Qt applications and projects. This demo requires Qt serial port support before it can be used. You can install it as follows:

$ cd ~ $ git clone git://code.qt.io/qt/qtserialport.git $ cd qtserialport $ git checkout 5.3 // for qt4 is “git checkout qt4-dev” $ cd ../ $ mkdir qtserialport-build $ cd qtserialport-build $ qmake ../ qtserialport/ qtserialport.pro $ make $ sudo make install

After installing the Qt serial port support, copy the `hardwareretest_serial_ok_20170223.tar.gz` package to Chipsee IPC, as described above (using SSH or USB storage).

- Open Debian system console and use the following set of commands to build the `hardwaretest_serial` demo application:

$ tar zxvf hardwaretest_serial_ok_20170223.tar.gz $ cd hardwaretest_serial $ qmake $ make

- Modify the permission for the serial ports device node, using the following:

$ sudo chmod 666 /dev/ttymxc

- Finally, run the `hardwaretest_serial` application

$ cd hardwaretest_serial $ export DISPLAY=:0 $ ./hardwaretest_serial

Disclaimer

This document is provided strictly for informational purposes. Its contents are subject to change without notice. Chipsee assumes no responsibility for any errors that may occur in this document. Furthermore, Chipsee reserves the right to alter the hardware, software, and/or specifications set forth herein at any time without prior notice and undertakes no obligation to update the information contained in this document.

While every effort has been made to ensure the accuracy of the information contained herein, this document is not guaranteed to be error-free. Further, it does not offer any warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document.

Despite our best efforts to maintain the accuracy of the information in this document, we assume no responsibility for errors or omissions, nor for damages resulting from the use of the information herein. Please note that Chipsee products are not authorized for use as critical components in life support devices or systems.

Technical Support

If you encounter any difficulties or have questions related to this document, we encourage you to refer to our other documentation for potential solutions. If you cannot find the solution you're looking for, feel free to contact us. Please email Chipsee Technical Support at **support@chipsee.com**, providing all relevant information. We value your queries and suggestions and are committed to providing you with the assistance you require.