Chipsee

Industrial PC

Linux QT OS on AM335X User Manual

For AM335X Products

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

Contents

Linux QT OS 4

1. Preparation 6

1.1. Hardware Requirements 6

1.2. Software Requirements 6

2. Getting Started and Tests 7

2.1. DIP Switch Configuration 7

2.2. Downloading Images 7

2.3. Prebuilt Files Package 7

2.4. How to make a bootable SD card 9

2.5. How to flash Linux to eMMC 11

2.6. Start Linux QT OS 13

2.7. Tests 14

2.7.1. Touch screen and buzzer test 14

2.7.2. Audio and video test 16

2.7.3. 3D Test 17

2.7.4. Serial test 18

2.7.5. CAN test 20

2.7.6. GPIO test 21

2.7.7. Network 23

2.7.8. Date and Time 24

2.7.9. Backlight 26

2.7.10. WiFi 27

2.8. Modify OS Start up Logo 29

3. Linux QT OS debug 32

3.1. View Linux QT system via the serial port 32

3.2. Debug via NFS 33

4. Linux App Development 35

4.1. Preparation 35

4.2. Steps 35

4.3. New development kit 37

5. Disclaimer 43
6. Technical Support 43

Linux QT OS

Linux QT OS User Manual

This manual provides users with a fast guide of Chipsee Industrial Computer (Abbreviate as IPC) about Linux QT OS development. Through this manual, users can quickly understand the hardware resources; users can build a complete compilation of Linux development environment; users can debug Linux QT OS via serial and Internet.

Revision	Date	Author	Description
V1.0	2021-12-09	Randy	Initial Version

SUPPORTED BOARDS:

> *CS80480T050 CS80600T080 CS10600T101 CS10600T070 CS80480T070 CS10768T097*

PREBUILT FILES PACKAGE:

Prebuilt files for the various industrial PCs can be found in the OS Downloads.
Below are the links to the prebuilt files for each industrial PC model.

- CS80480T050
- CS80600T080
- CS80600T101
- CS10600T070
- CS80480T070
- CS10768T097

System Features

Feature	Comment
System	LinuxQt 4.8 & LinuxQt 5.5

Preparation

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

- **Power Supply Unit (PSU) with the appropriate voltages, as follows:**
    - Products with 5” display panel require 6V to 36V PSU
    - Products with 7” to 10.1” display panel and larger require 6V to 42V PSU
- USB to serial cable for debugging Chipsee Industrial Embedded Computers (Chipsee IPC)
- TF Card to create a bootable storage for re-flashing the system. Use the prebuilt files link above to re-flash the system.

Hardware Requirements

- Chipsee Industrial PC
- PSU according to the instructions above
- USB-to-serial or other serial cable for debugging
- TF Card (at least 4GB) and card reader
- USB A-A cable (used only if the hardware configured as OTG)
- Windows 7 PC

Software Requirements

- Linux QT OS Prebuilt Files Package (from the link above)
- Useful tools for Qt development

> **Note**
>
> In this documentation, all the commands are executed with `root` user privileges.

Getting Started and Tests

DIP Switch Configuration

Set the boot DIP switch, as shown on the figure below, to boot the system from the external SD Card.

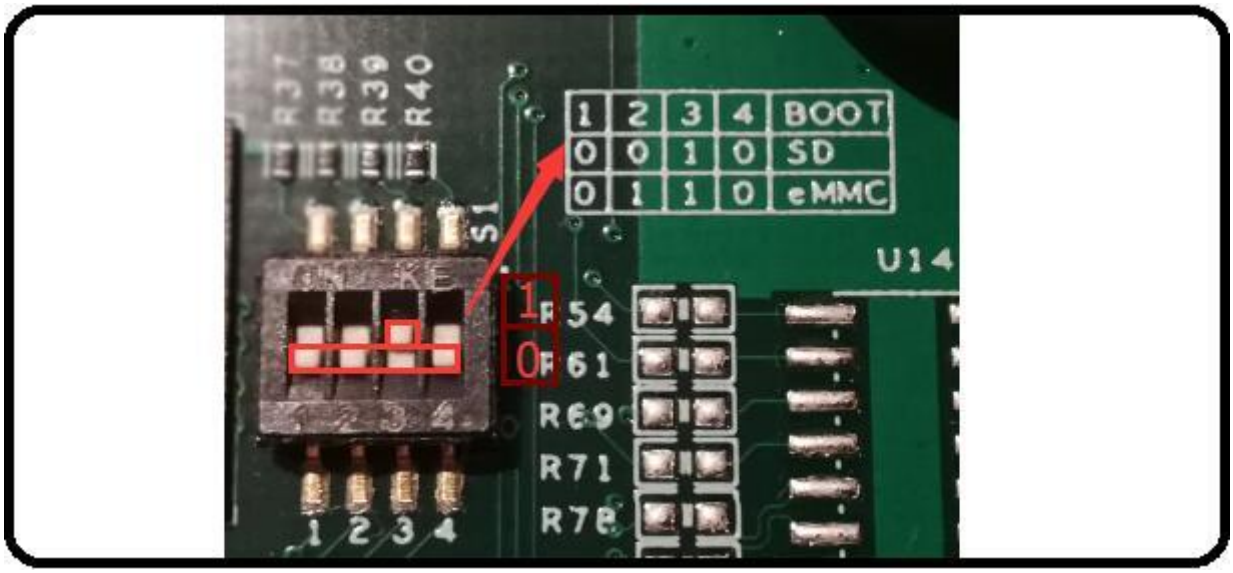

Figure 794: Boot Mode Setup

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.

> **Note**
>
> The operator should use the prebuilt file we provided in the CD to test the hardware before re-flashing the system.

Prebuilt Files Package

You can get the Prebuilt Files Package for each model from links mentioned at the beginning of this documentation. You can also get the Prebuilt Files Package from the DVD in /Linux QT/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:

Contents	Comment
boot/imx6ulipc.dtb	TF Card boot dtb file

Contents	Comment
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
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 271 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 framebufer, just rename these two files to `zImage` and `imx6q-eisd.dtb`.

How to make a bootable SD card

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

1. Copy the Prebuilt Files Package to a Linux environment (such as Ubuntu 14.04).
2. Insert the SD card into your computer. If you are using virtual machines, please ensure the SD card is mounted to the Linux operating system.
3. **Confirm the SD card mount point, `/dev/sdX` (e.g., `/dev/sdc` or `/dev/sdb`, be sure to use the right one). In a Linux system, you can use the command below to find out what *X* is.**

   ```
   $ sudo fdisk –l
   ```

4. Copy the `prebuilt-som-v3-csxxxxxtxx-v3-ezsdk-sd-yyyymmdd.tar.gz` to somewhere(such as $HOME).
5. **Extract the `prebuilt-som-v3-csxxxxxtxx-v3-ezsdk-sd-yyyymmdd.tar.gz`**

   ```
   $ tar -xzvf prebuilt-som-v3-csxxxxxtxx-v3-ezsdk-sd-yyyymmdd.tar.gz
   ```

6. **Go to the folder**

   ```
   $ cd ~/prebuilt-som-v3-csxxxxxtxx-v3-ezsdk-sd-yyyymmdd
   ```

7. **Use the following command to flash the Linux QT OS to the SD card**

   ```
   $ sudo ./mksdcard.sh --device /dev/sd<?>
   ```

   > **Note**
   >
   > • sd<?> means the SD card mount point, (e.g., `/dev/sdc` or `/dev/sdb`) in Ubuntu system.
   >
   > • The recommended SD card should be Sandisk Class4 level SD card or above.

8. The bootable SD Card is now ready. Power OFF the industrial PC and insert the SD Card.
9. Set the DIP switch to uSD BOOT mode. (refer to DIP Switch Configuration above)
10. Connect the industrial PC to PC via COM1. Power ON the IPC.

11. After 20 minutes, if the LED on industrial PC stays lit, flashing is completed. Using COM1, you can also find this message **>>>>>>> eMMC Flashing Completed <<<<<<<** which indicates that the system image was downloaded correctly to the eMMC.
12. Power OFF the IPC and set the DIP switch to eMMC BOOT mode. (refer to DIP Switch Configuration above)

How to flash Linux to eMMC

The Prebuilt Files Package has a shell tool that can help create a bootable SD card using a Linux platform (such as desktop PC or Virtual Machine running Ubuntu 14.04 distribution). Follow the steps below to create a bootable SD card:

1. Copy the Prebuilt Files Package to a Linux environment (such as Ubuntu 14.04).
2. Insert the SD card into your computer. If you are using virtual machines, please ensure the SD card is mounted to the Linux operating system.
3. **Confirm the SD card mount point, `/dev/sdX` (e.g., `/dev/sdc` or `/dev/sdb`, be sure to use the right one). In a Linux system, you can use the command below to find out what *X* is.**

   ```
   $ sudo fdisk –l
   ```

4. Copy the prebuilt file `prebuilt-som-v3-csxxxxxtxx-v3-ezsdk-emmc-yyyymmdd.tar.gz` to somewhere(such as $HOME).
5. **Extract the prebuilt file** `prebuilt-som-v3-csxxxxxtxx-v3-ezsdk-emmc-yyyymmdd.tar.gz`

   ```
   $ tar -xzvf prebuilt-som-v3-csxxxxxtxx-v3-ezsdk-emmc-yyyymmdd.tar.gz
   ```

6. **Go to the folder** `prebuilt-som-v3-csxxxxxtxx-v3-ezsdk-emmc-yyyymmdd`

   ```
   $ cd ~/prebuilt-som-v3-csxxxxxtxx-v3-ezsdk-emmc-yyyymmdd
   ```

7. **Use the following command to flash the Linux QT OS to the SD card**

   ```
   $ sudo ./mksdcard.sh --device  /dev/sd<?>
   ```

   > **Note**
   >
   > • sd<?> means the SD card mount point, (e.g., `/dev/sdc` or `/dev/sdb`) in Ubuntu system.
   >
   > • The recommended SD card should be Sandisk Class4 level SD card or above.

8. The bootable SD Card is now ready. Power OFF the industrial PC and insert the SD Card.
9. Set the DIP switch to SD BOOT mode. (refer to DIP Switch Configuration above)
10. Connect the industrial PC to PC via COM1. Power ON the IPC.

11. After 20 minutes, if the LED on industrial PC stays lit, flashing is completed. Using COM1, you can also find this message **>>>>>>> eMMC Flashing Completed <<<<<<<** which indicates that the system image was downloaded correctly to the eMMC.
12. Remove the SD card and Power OFF the IPC.
13. Set the DIP switch to eMMC BOOT mode (refer to DIP Switch Configuration above) and Power ON the IPC.

Start Linux QT OS

The first time you start Linux QT OS on the industrial PC will take a little time. But after the first time, Linux QT OS will start quickly. When the Linux QT OS starts up, you will see the Chipsee Logo on the LCD screen. It is a successful start if you see the Linux QT OS desktop such as the one shown in the figure below:

Figure 795: Chipsee Linux QT OS start-up screen

Tests

Touch screen and buzzer test

Click on the screen, the mouse arrow stays in a position that triggers the buzzer sounds, indicating that touch and buzzer work properly.
After working for some time, the resistive touch screen may not be accurate. The user must run a touch screen calibration test.
Click on the **Chipsee** icon on the desktop. Select **Calibrate Screen** to calibrate it, just as described in the figure below.

Figure 796: Resistive touch screen calibration app

The buzzer will sound when the screen is touched, if you want to disable it, you can do this:

- **On capacitive touchscreen:**

```
# echo 0 > /sys/devices/ocp.3/44e0b000.i2c/i2c-0/0-0038/buzopen
```

- **On resistive touchscreen:**

```
# echo 1 > /sys/devices/ocp.3/44e0d000.tscadc/tsc/buzopen
```

where:

  - 0 = disable

- 1 = enable

Audio and video test

Insert the microphone and earphones into the Audio IO interface (Audio IN coloured pink, Audio OUT coloured light blue).
as shown on the figure below, click the **Multimedia** icon on desktop then choose the **MPEG-4+AAC Dec** codec to test.

Figure 797: Audio and Video

3D Test

Click the **3D** icon on desktop, then choose **Film TV** to test perform 3D testing as shown on the figure below.

Figure 798: 3D test Film TV

Serial test

There are four serial ports on the Chipsee IPC: 2 X RS232 and 2 X RS485. The COM1(RS232) is used as the debug serial port. Users can communicate with the OS via COM1. Refer to the table below for the available serial device nodes.

Ports	Device Node
COM1(RS232, Debug)	/dev/ttyO0
COM2(RS232)	/dev/ttyO1
COM3(RS485)	/dev/ttyO2
COM4(RS485)	/dev/ttyO4

Table 272 Serial Ports Nodes on the System

If you want to use COM1 as a normal serial port, you can re-configure the port by following these steps:

- Open and edit the file `/etc/inttab` with any text editor.
- At the end of the file, edit this line `S:2345:repawn:/sbin/getty  115200  ttyO0` to

```
# S:2345:repawn:/sbin/getty 115200 ttyO0
```

- The code-block above, comments off the last line making it possible to use all the four serial ports as normal serial ports.
- You can verify the changes by running a serial test.
- **Run a serial test:**
    - Install **SecureCRT** or **Putty** software on a Windows 7 PC and use it to perform the serial port testing.
    - Click on the **Chipsee** icon on desktop, select **ChipseeTest** to run the **SerialTest** app to communicate with Windows 7 PC.
    - **From the ChipseeTest app, search for the serial area then configure the following settings, as shown on the figure below.**
        - set Com to COM2
        - set Baud to 115200
        - click on the **Open** button

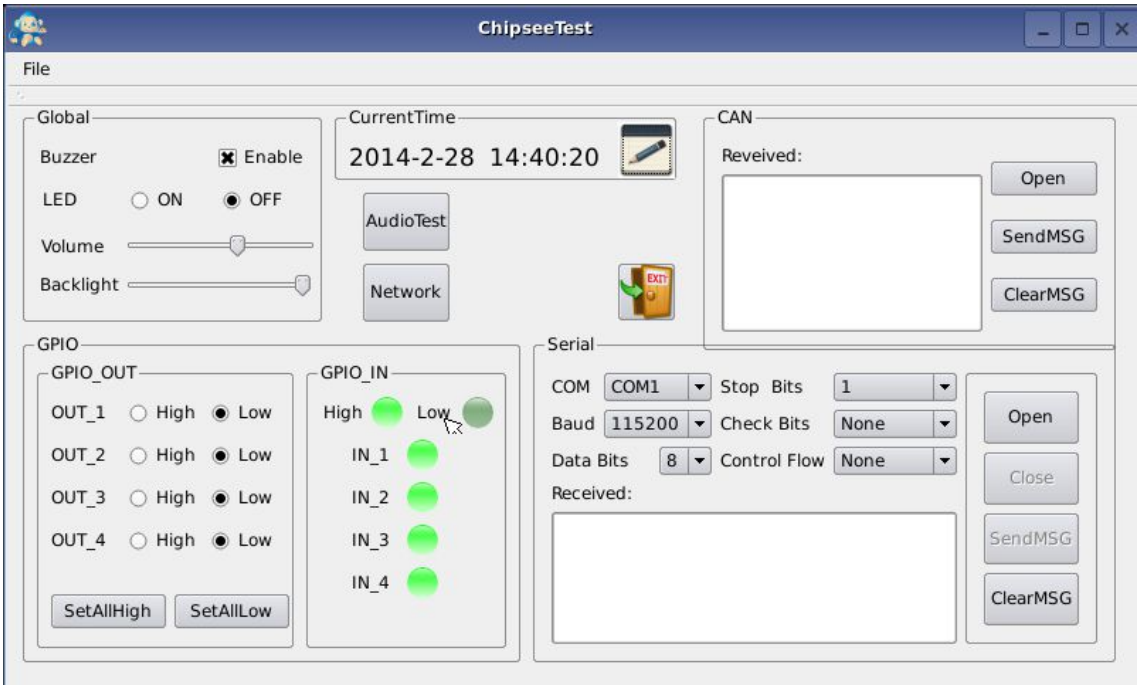

Figure 799: ChipseeTest

- It will send the string *Succeed in sending message!!!* every two seconds through the serial port to the Windows 7 PC.
- Click on the **SendMSG** button to send the string *Succeed in sending message-manual!!!*.
- Every two seconds, it will read the received buffer and show the result to the received area.

CAN test

To perform the test, the user will need the following: **2 x Embedded Industrial Computers, 2 x CAN bus connectors, 1 x 120Ω resistor, and oscillometer.**

- The user will connect the two CAN bus connectors directly to each other for testing.

  > Between CANH and CANL you should use a 120Ω resistor

- At the CAN area, click on the **Open** button, then click on the **SendMSG** button to send message: *11 22 33 44 55 66 77 88*
- On both embedded industrial PCs, you should see the message shown at the received area.
- If you have one embedded industrial PC, you can use an **oscillometer** to see the result.

GPIO test

There are (4) four input and (4) four output pins. LOW is 0V, HIGH is 5V.
The GPIO input terminals connect to the GPIO output terminals, respectively. IN1-4 corresponds to OUT1-4.
As a result, if you set the GPIO_OUT area, you will see the GPIO_IN region change as well.
You can control the LED light on the industrial PC by setting the LED **ON** or **OFF**.

GPIO	GPIO In System
OUT1	gpio49
OUT2	gpio50
OUT3	gpio51
OUT4	gpio52
IN1	gpio53
IN2	gpio54
IN3	gpio55
IN4	gpio56
USER_LED	gpio19

Table 273 GPIO Nodes on the System

You can read and write the GPIO by following the steps below. For this example, we are going to use **gpio49** (OUT1).

- **Use this command to export gpio.**

```
# echo 49 > /sys/class/gpio/export
```

- **Use this command to check if the directory `/sys/class/gpio/gpio49/` exist before writing to it**

```
# find /sys/class/gpio/gpio49/
```

- **Use this command to write gpio**

```
# echo 1 > /sys/class/gpio/gpio49/value
```

- **Use this command to read gpio**

```
# cat /sys/class/gpio/gpio49/value
```

Network

To view the network information on the industrial PC, follow these steps:

- Click on the **Network** tab, then click the **Ifconfig** button to view the network information on the industrial PC.
- Click on the **Refresh** button to restart the network service which will take five or six seconds to finish.

The figure below is an illustration of the network information on the industrial PC.

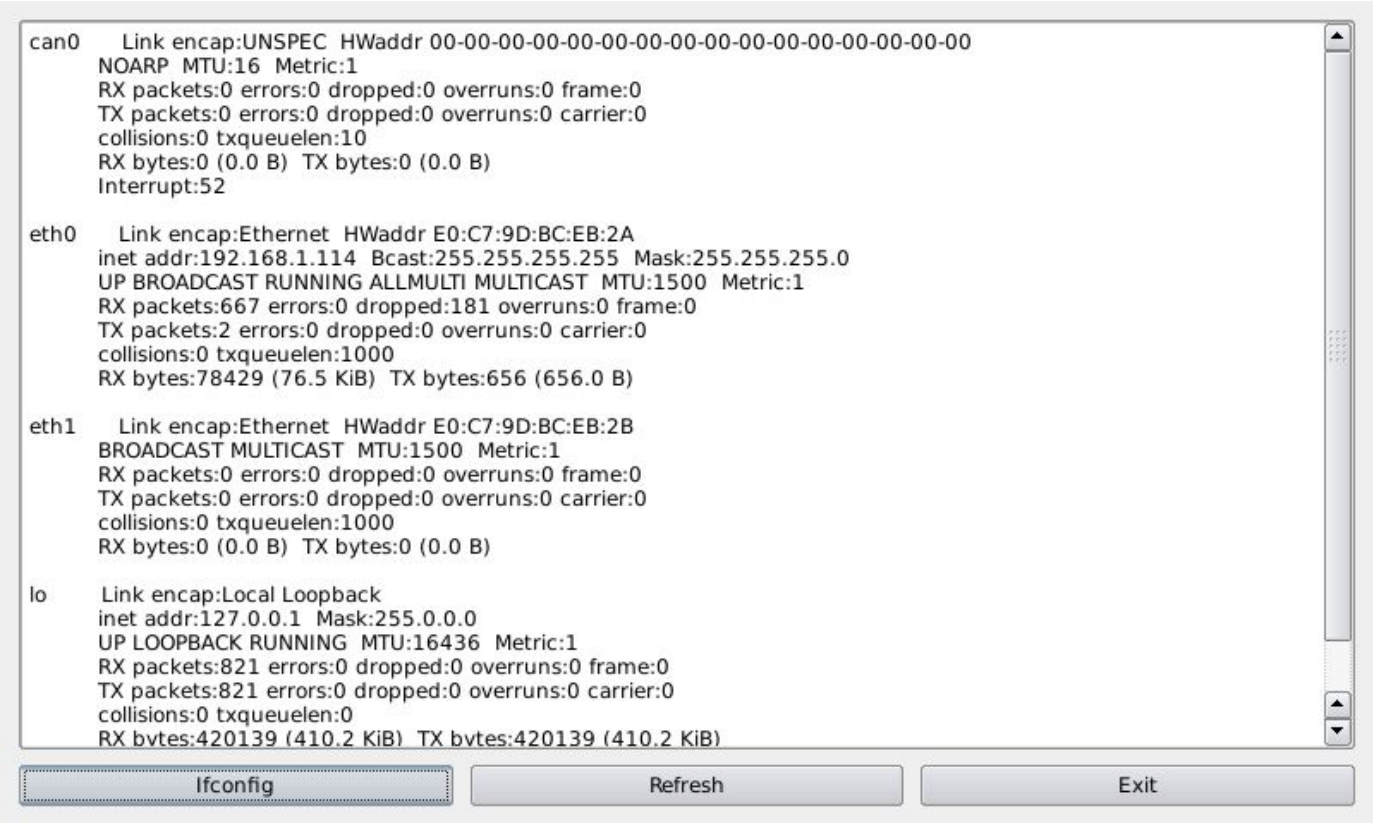

Figure 800: View Network Information

Date and Time

Click the **Edit** icon at the time display area to set the time and date, as shown on the figure below.

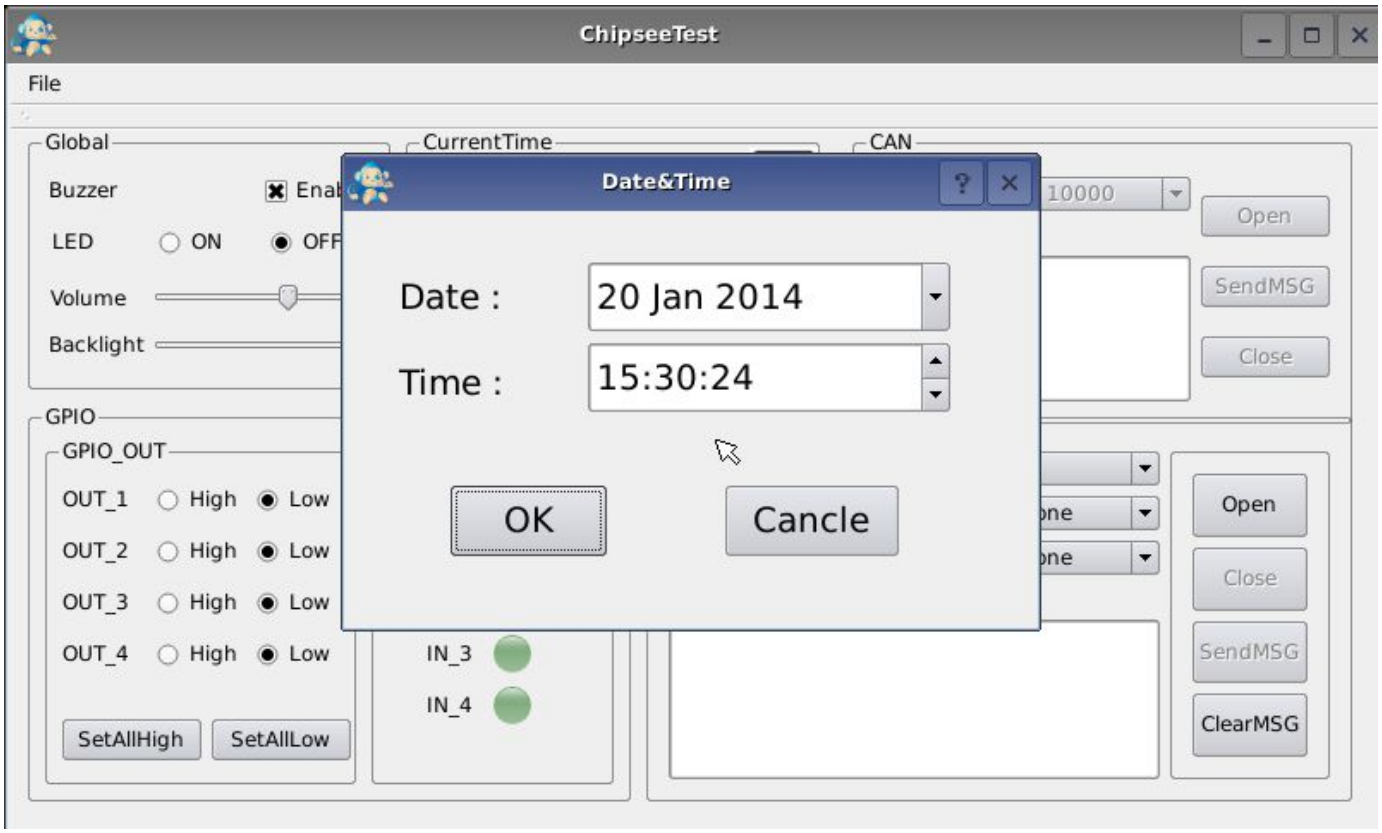

Figure 801: Set Date and Time

- **Check the system time**

```
# date
```

- **Set the system time**

```
# date –s "2014-03-15 10:30:30"
```

- **Check RTC**

```
# hwclock
```

- **Write RTC**

```
# hwclock -w
```

- **Modify the time zone to a different timezone, such as China**

```
# ln -sf /usr/share/zoneinfo/Asia/Hong_Kong /etc/localtime
```

Backlight

Modify this file `/sys/class/backlight/backlight` to adjust the screen brightness.
Brightness ranges from 0 to 100 where 0 means no backlight, and 100 is the MAX brightness value.
For example, you can adjust the screen brightness using this command:

```
# echo 50 > /sys/class/backlight/backlight.10/Brightness
```

WiFi

The Linux QT OS has a WiFi module. If you want to get Wifi module to work, you need to edit the configuration file: `/etc/wpa_supplicant.conf`.

1. **Set SSID and password in the config file, as shown in the code-block below:**

```
# vi /etc/wpa_supplicant.conf
network={
        ssid="Chipsee"    //set your wifi ssid
        psk="1chipsee234567890"   //set your wifi password
}
```

2. **Launch the Wifi**

Figure 802: Launching WiFi

3. After a few minutes, you can use the WiFi

Modify OS Start up Logo

Chipsee® provides a software to change the OS boot up logo. The software `ChipSee_LOGO_MOD_EN.exe` is provided on the CD for a product.
To change the logo, follow these steps:

1. **Open the software:** `Chipsee_LOGO_MOD_EN.exe` **in Windows**

Figure 803: Chipsee OS Boot-up Logo Modify Software

2. **Click the first Browse button. Select the picture file you want to use as the logo.**

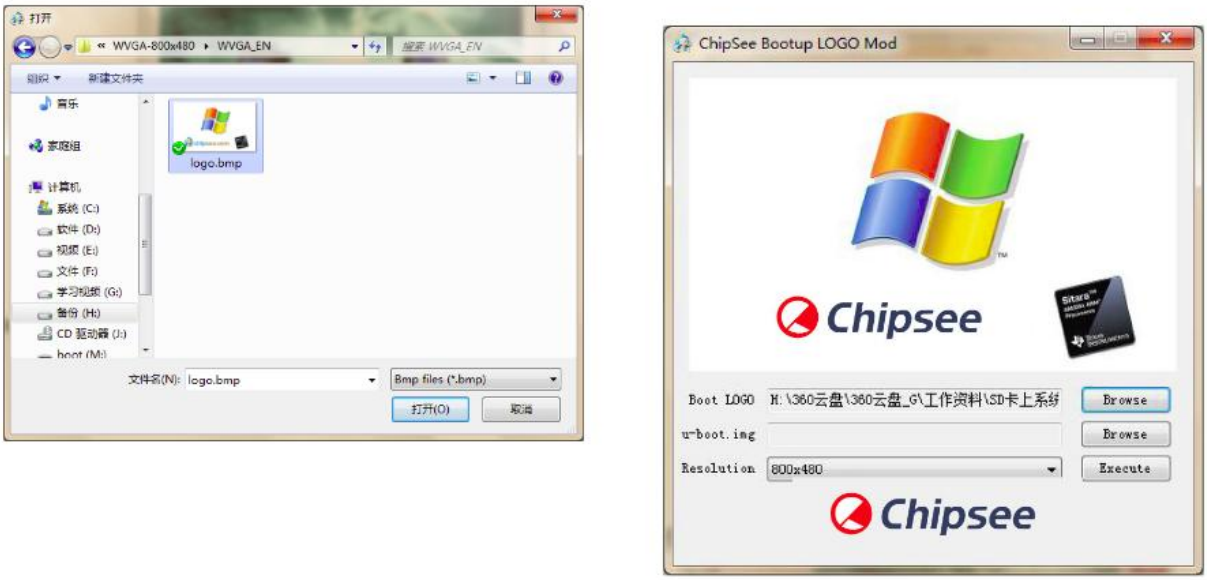

Figure 804: Choose the Logo you want

3. **Click the second Browse button. Select the `u-boot.img` file you want to use.**

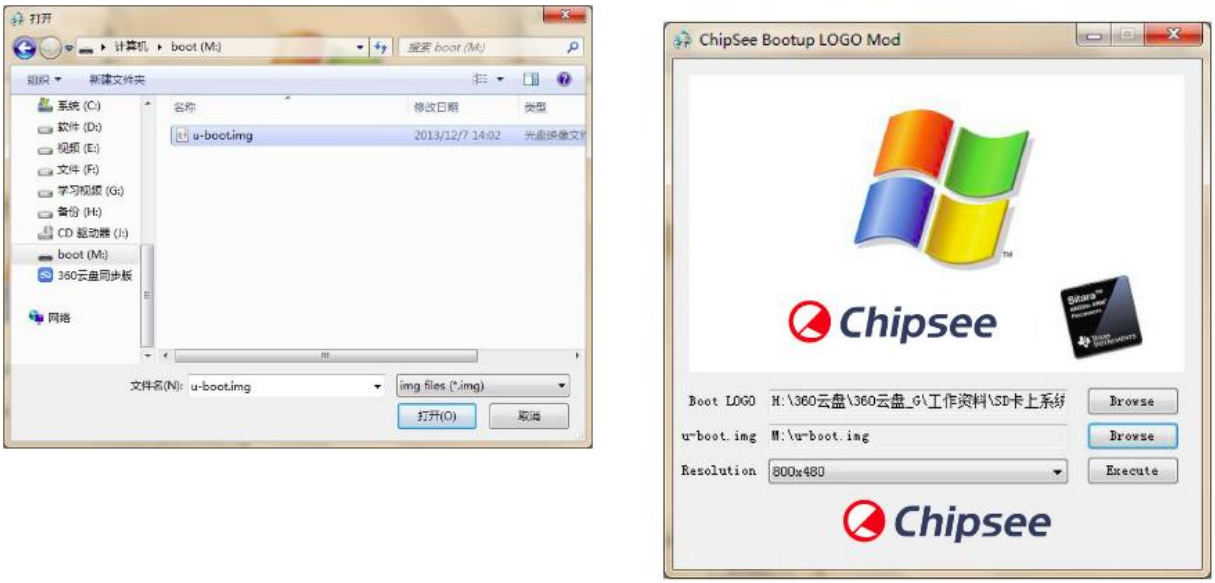

Figure 805: Choose the u-boot.img file

4. **Choose the correct resolution for your product, then click Execute.**

Figure 806: Change the Logo successful

5. Insert the SD card into the IPC. Power ON the IPC and the Logo will be replaced.

Note

If you want to run the system from NAND Flash and change the Logo, you should change the logo first then flash the NAND.
If the system is running in the NAND Flash, you can rewrite the `u-boot.img` file.
Boot from uSD card, hit the **space** key after power on, to switch boot into **u-boot** mode `U-Boot #`.
Rewrite the `u-boot.img` file.

```
U-Boot # nand erase.chip
```

Reboot

Linux QT OS debug

In this section, we will discover how to view the Linux QT system via the serial port on a Windows 7 PC.
Also, we will discover how to debug using NFS on a Ubuntu Linux PC.

View Linux QT system via the serial port

Install the **SecureCRT** or **Putty** software on a Windows 7 PC to view the Linux QT system via the serial ports.
Follow these steps to view Linux QT system via the serial port:

- Connect COM1 on the industrial PC board to Windows 7 PC.
- Open the **SecureCRT** or **Putty** software on the Windows 7 PC.
- Power ON the industrial PC. You will see the serial output information as shown on the figure below.
- When the system is fully booted, you can communicate with it by logging in with these details: user= root and password= empty.

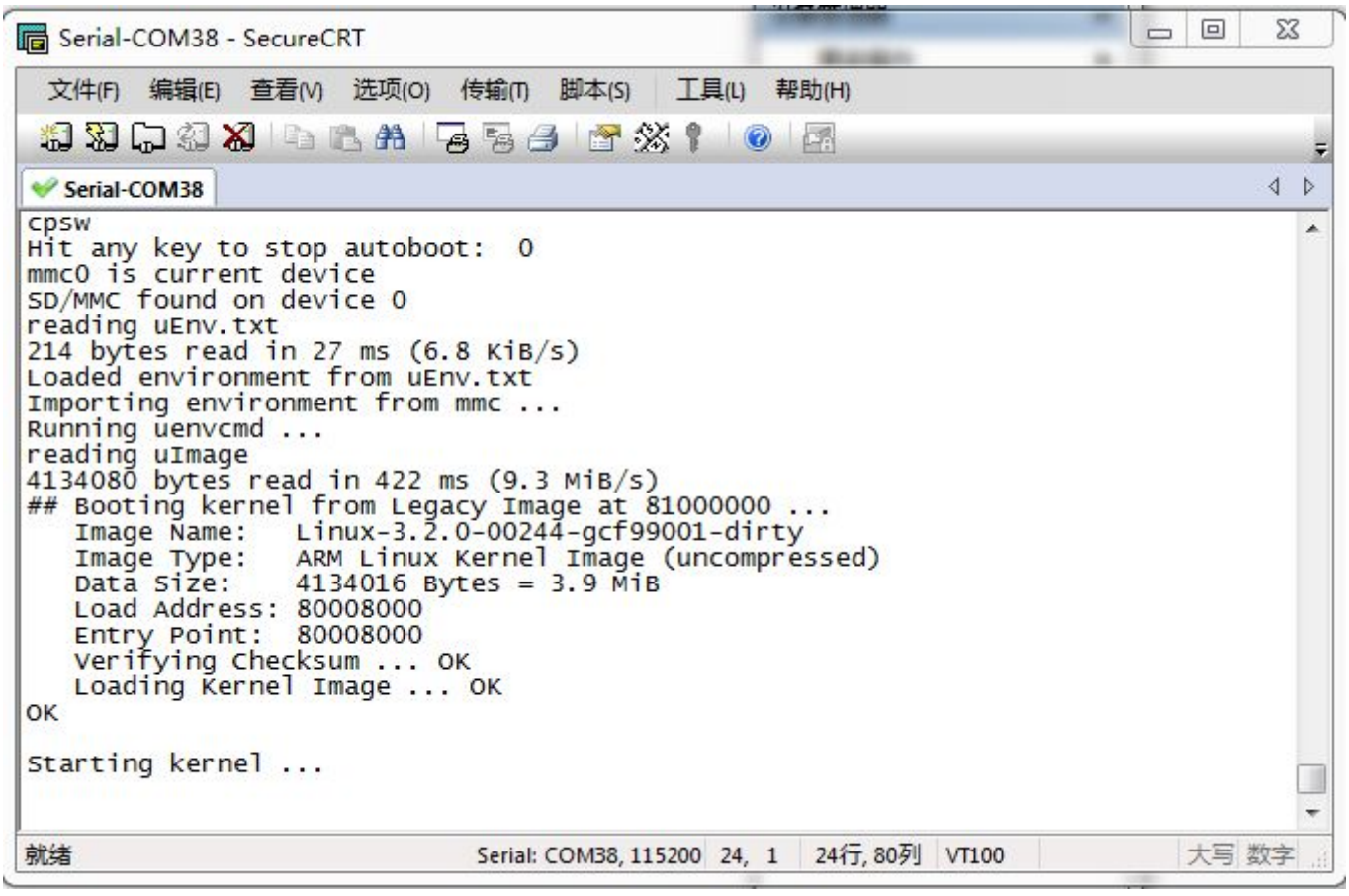

Figure 807: Serial output information

Debug via NFS

1. **Install NFS on Ubuntu Linux PC.**

```
$ sudo apt-get install nfs-kernel-server
```

2. **Configure the file `/etc/exports`, by adding this line at the end of the file.**

```
/qtprojects *(rw, sync, insecure, no_subtree_check)
```

> **Note**
>
> - `/qtprojects`: the shared folder in Ubuntu system
> - `*`: allows all other PC to get access to this system
> - `rw`: means this folder can be read and write by NFS client
> - `sync`: synchronous write memory and hard disk
> - `insecure`: sent message through the port above 1024
> - `no_subtree_check`: no check the parent directory permissions

3. **Restart NFS service.**

```
$ sudo /etc/init.d/portmap restart
$ sudo /etc/init.d/nfs-kernel-server restart
```

4. **Test**

```
$ showmount -e
```

or mount the shared folder to `/mnt`:

```
$ sudo mount -t nfs –o nolock localhost:/qtprojects /mnt
```

Use the command `df` to check out the result, then umount.

```
$ df -h
$ sudo umount /mnt
```

5. **Mount NFS on the industrial PC running Linux QT OS.**

Create the `nfsdir` directory

```
# mkdir /nfsdir
```

Mount the folder `/qtprojects` on the Ubuntu Linux PC to `/nfsdir` on the industrial PC.

```
# mount –t nfs :/qtprojects /nfsdir
```

If you have an executable program like **SerialTest** under folder `/qtprojects`, you can run it directly on the industrial PC.

```
# /nfsdir/SerialTest
```

Linux App Development

In this section, we will introduce how to develop applications for the industrial computer in Linux.

Preparation

Software:

1. Ubuntu system, we suggest Ubuntu 14.04.5 LTS x64
2. Install Qtcreator package: qt-linux-opensource-5.1.0-x86-offline.run.
3. Install package ti-sdk-am335x-evm-07.00.00.00-Linux-x86-Install.bin provided by TI.

Steps

1. We assume you have created a GUI program using Qtcreator, such as `HelloWorld`.
2. **Configure the environment variables for the TI package by using the command below to source a script file included in the TI package.**

   ```
   $ source /opt/ti-sdk-am335x-evm/linux-devkit/environment-setup
   ```

3. **Change directory into the folder of your GUI program (`HelloWorld`) and run these commands:**

   ```
   $ qmake –project
   $ qmake
   $ make
   ```

   Now there will be a file which you can run in the industrial PC. You can use the command `file  HelloWorld`, to check if the file can be executed in the ARM platform.

4. **Put the file `HelloWorld` on the industrial PC. Then run the command below in the Ubuntu Linux system (communicate via COM1).**

   ```
   # ./Hello
   ```

   The program will start running

5. **Add application to the desktop of Matrix**

- Put your program file `HelloWorld` into a folder which can be found by the system such as, `/usr/bin`.
- Put the program's icon into the folder: `/usr/share/matrix-gui-2.0/apps/images/`. For example, `/usr/share/matrix-gui-2.0/apps/images/YOURAPPIMG.png`
- Go to the folder `/usr/share/matrix-gui-2.0/apps/`, then create a new folder named `HelloWorld` using this command `$ mkdir HelloWorld`.
- Change directory to `HelloWorld` folder, then create a new file named `HelloWorld.desktop` using the command `$ touch HelloWorld.desktop`.
- Edit the `HelloWorld.desktop` file using the command:

```
1 #!/usr/bin/env xdg-open[Desktop Entry] Name= YOURAPPNAME // Can be changed
2 GenericName=Demo App
3 Icon=/usr/share/matrix-gui-2.0/apps/images/YOURAPPIMG.png
4 Exec=YOURAPPNAME
5 Type=Application
6 ProgramType=gui
```

- Refresh the system by clicking on Settings–>Refresh Matrix, then click on run.
- You will see your application's icon on the desktop after refreshing. If the icon does not change in time, you need to reboot your system.

New development kit

1. **Open a Terminal in Ubuntu using the CTRL + ALT + T key combination. Enter the command below.**

```
# source /usr/local/ti-sdk-am335x-evm/linux-devkit/environment-setup
```

2. **Then open QtCreator**

```
# /opt/Qt5.1.0/Tools/QtCreator/bin/qtcreator &
```

Choose Tool–>Options:

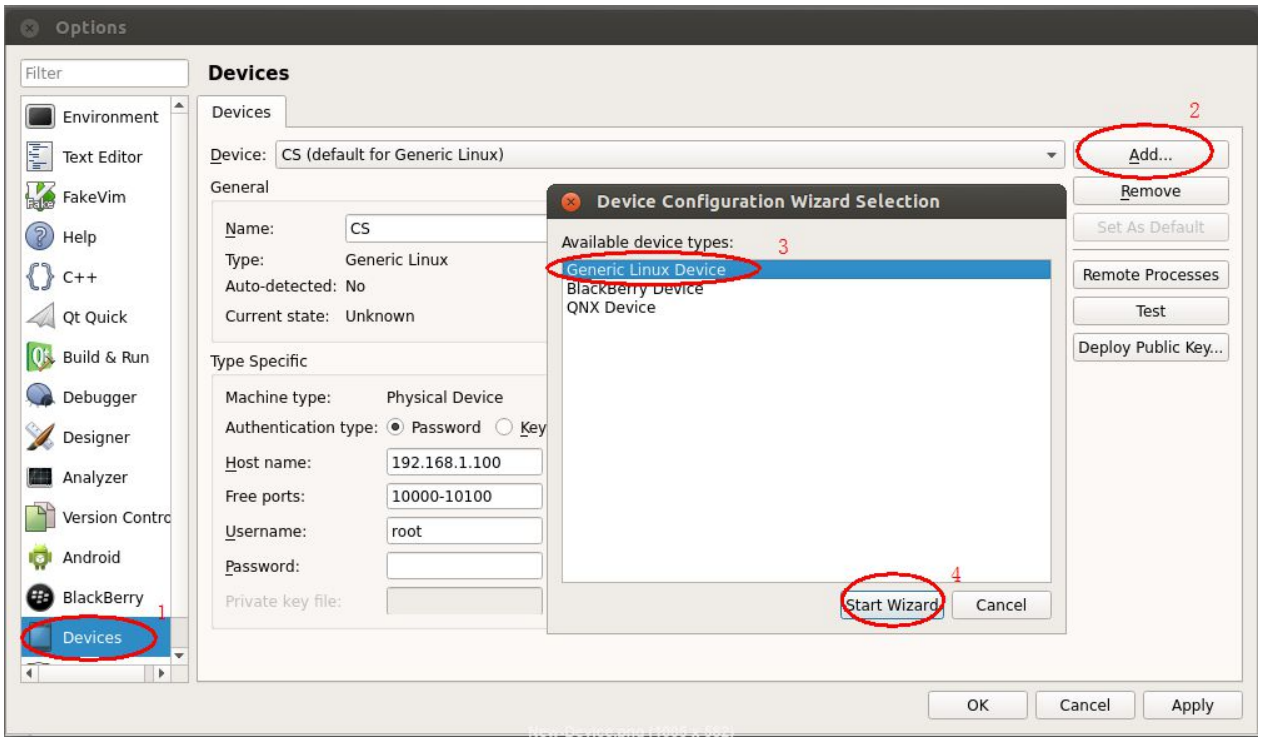

Figure 808: New Device

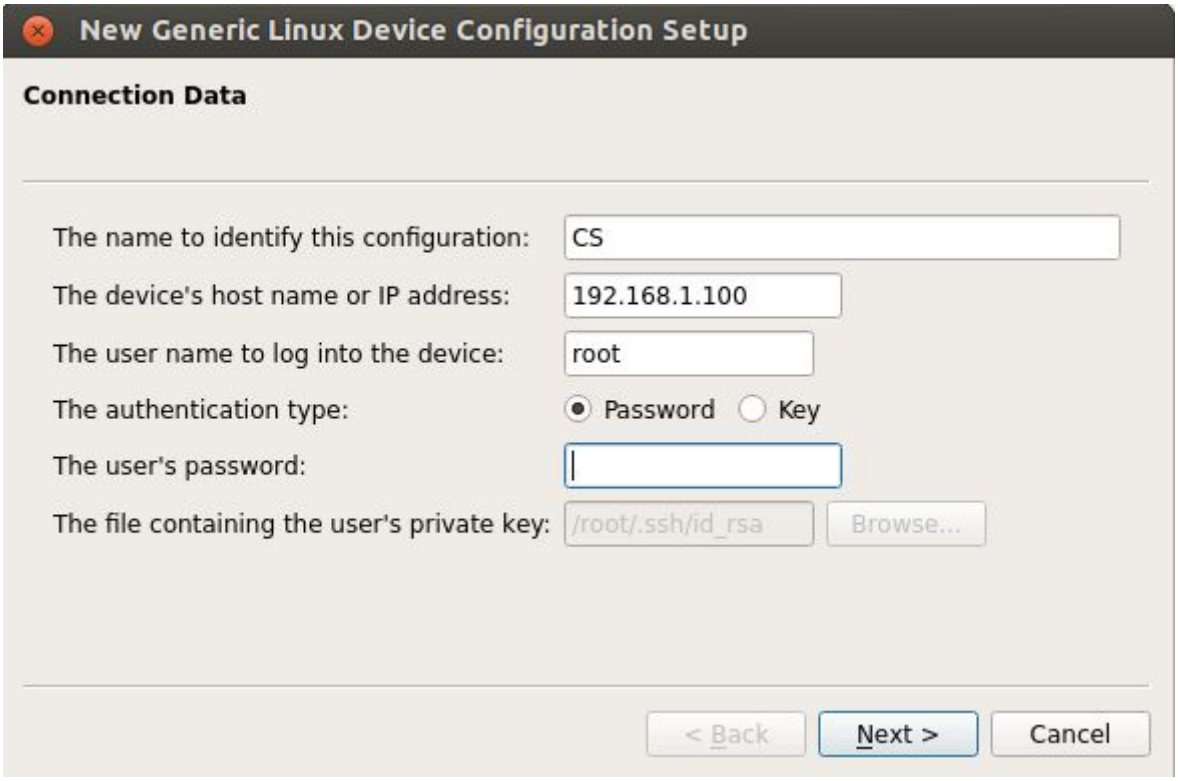

Figure 809: Properties of the Device

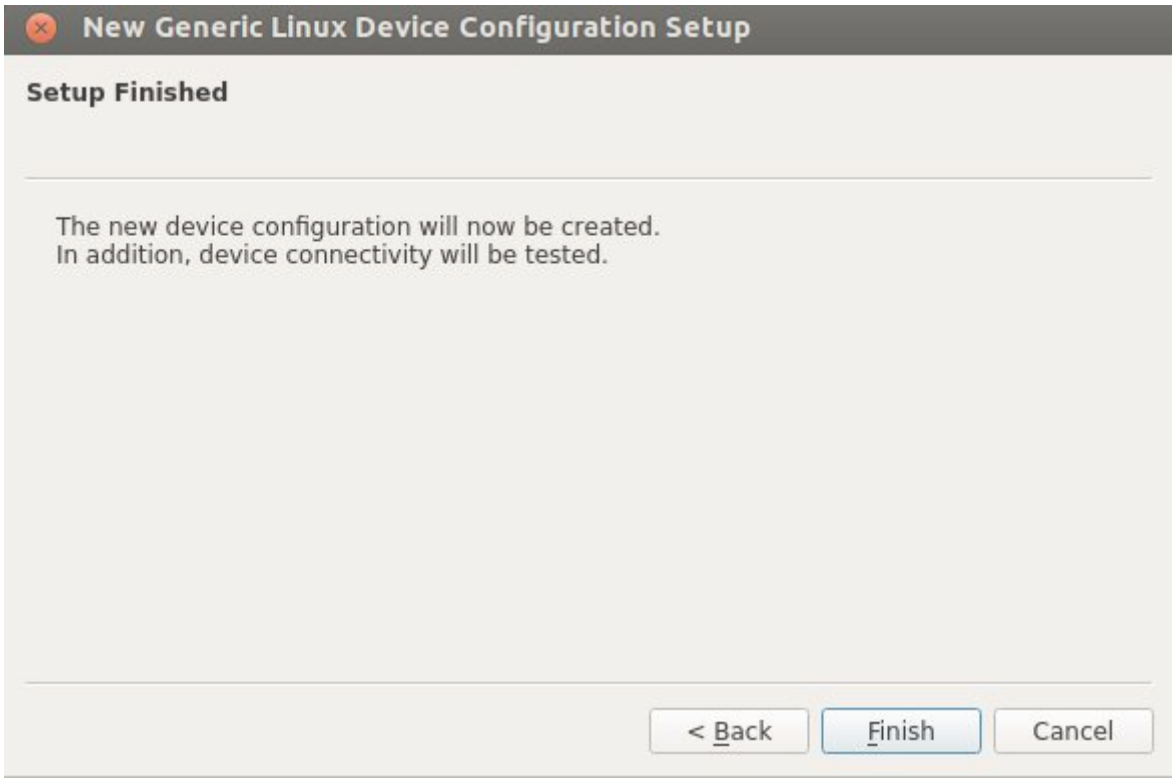

Figure 810: Succeed

Click the **Finish** button to test the connection between the Ubuntu Linux system and the industrial PC via internet.

Then click the **Build & Run** tab to select and apply a compiler, as shown on the figure below.

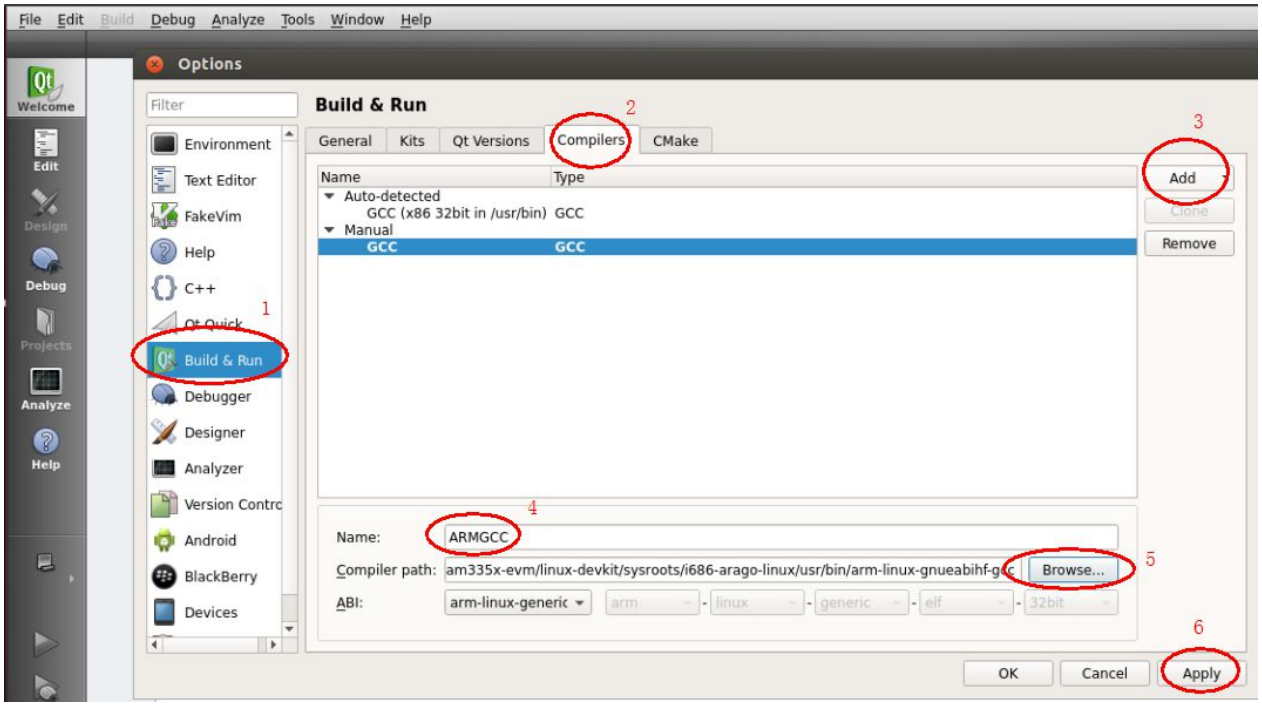

Figure 811: Compilers

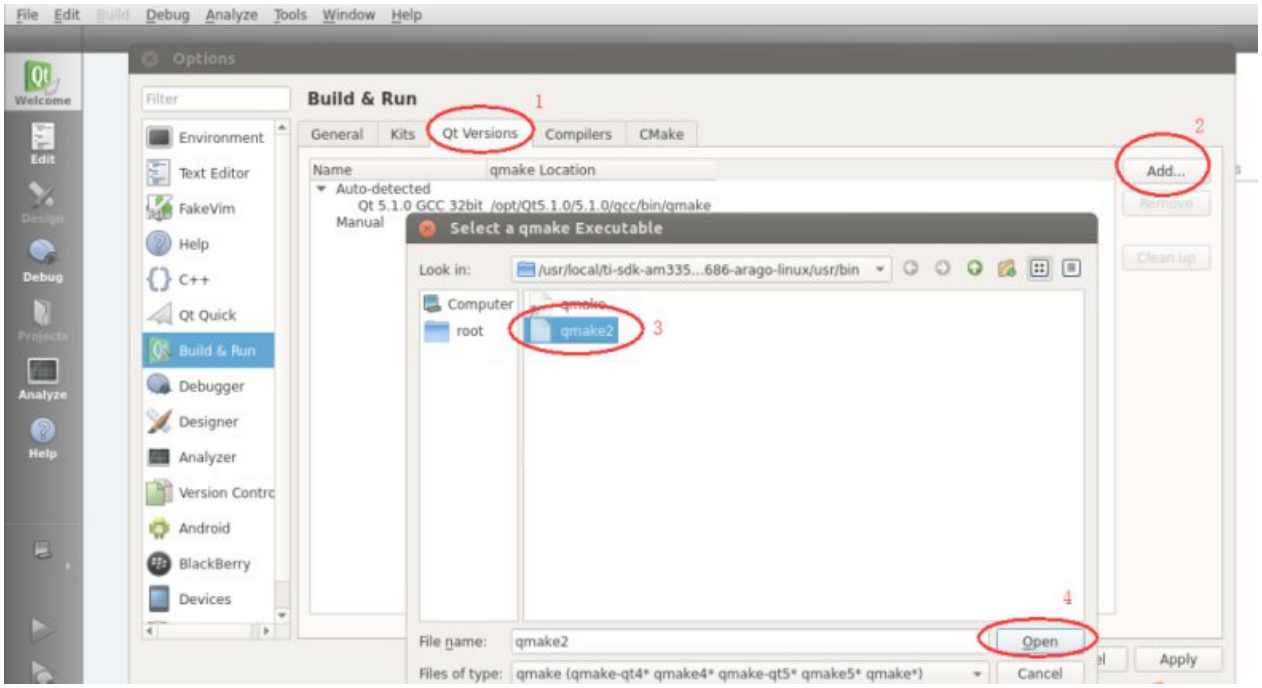

Figure 812: Qt Version

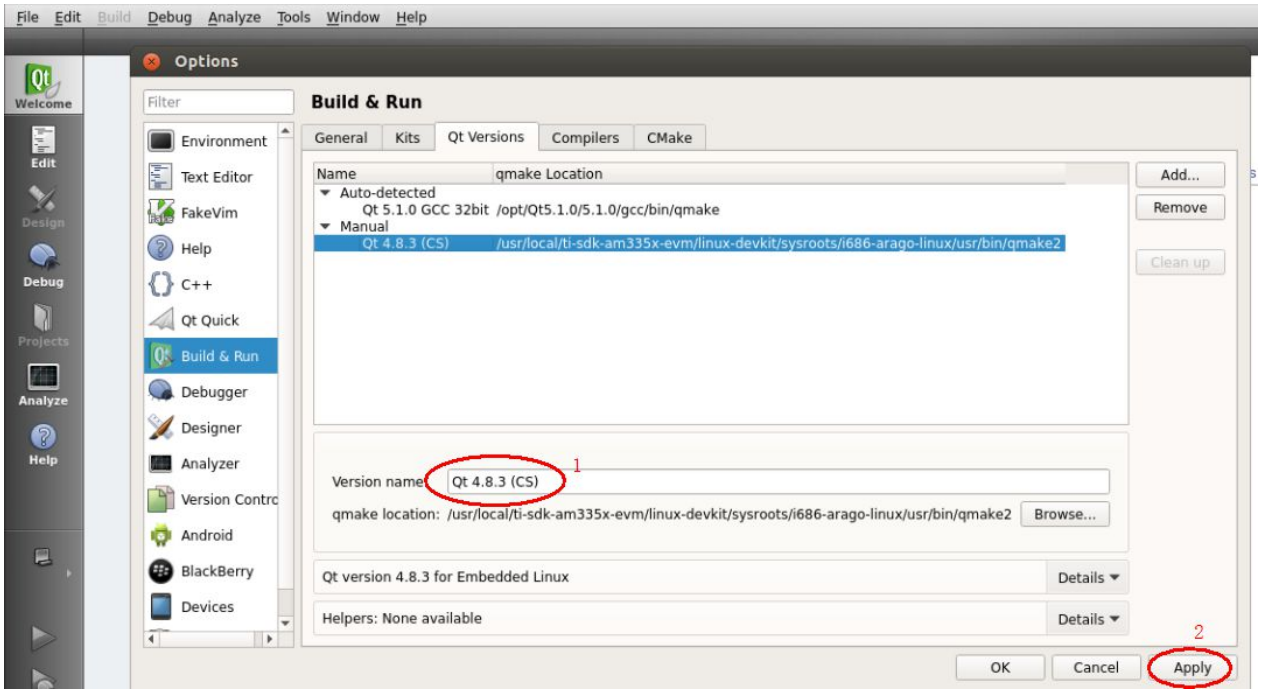

Figure 813: Change the name

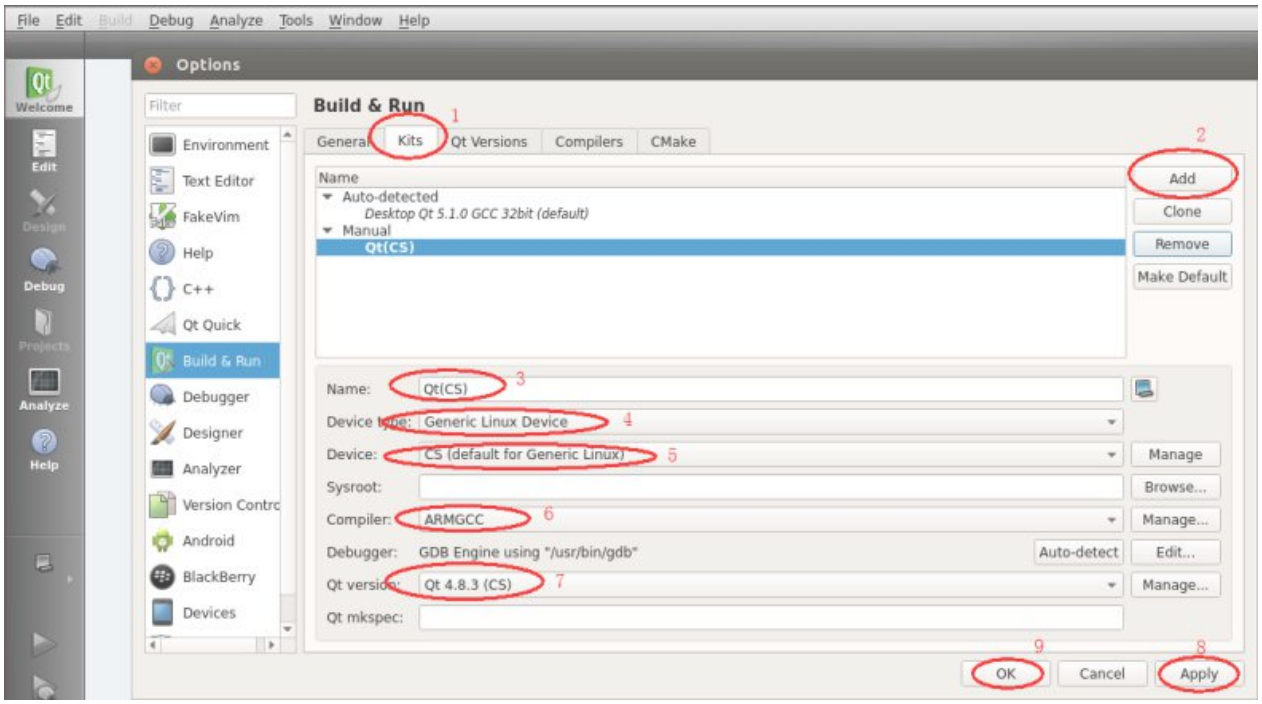

Figure 814: New kit settings

Follow the Steps above to build a new project named `Test` and choose the new development kit.

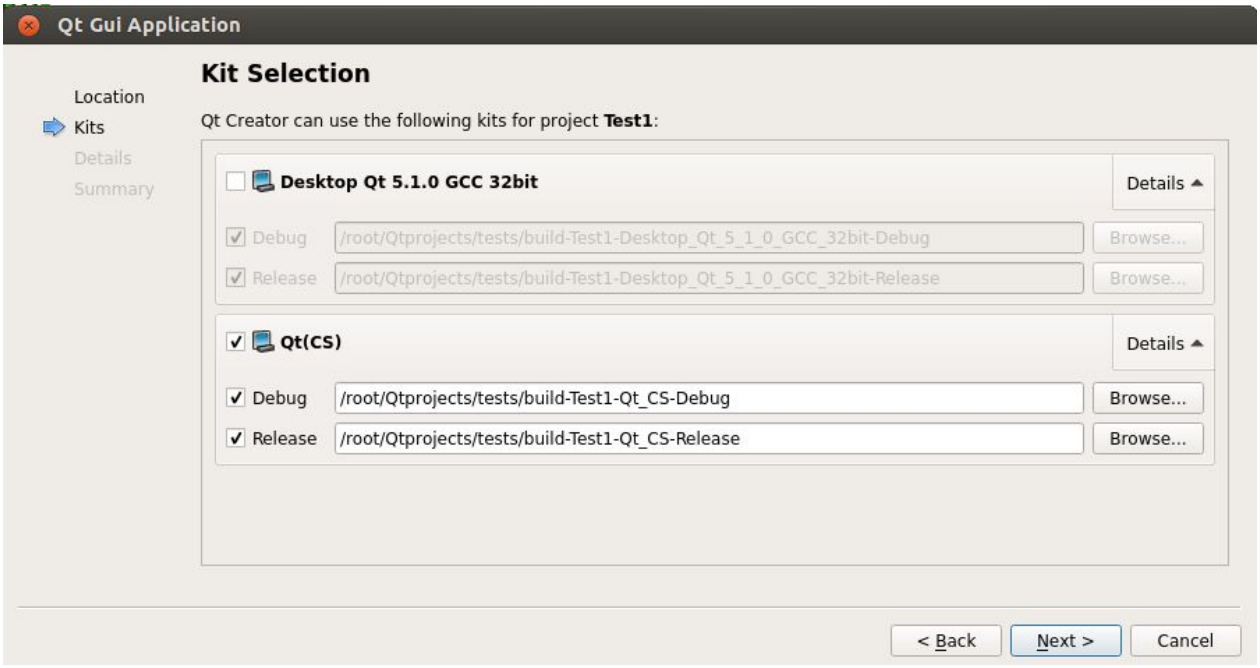

Figure 815: Choose new kit

Add the code below to the `Test.pro` file before you build & run.

```
target.files = Test
target.path = /home/root
INSTALLS += target
```

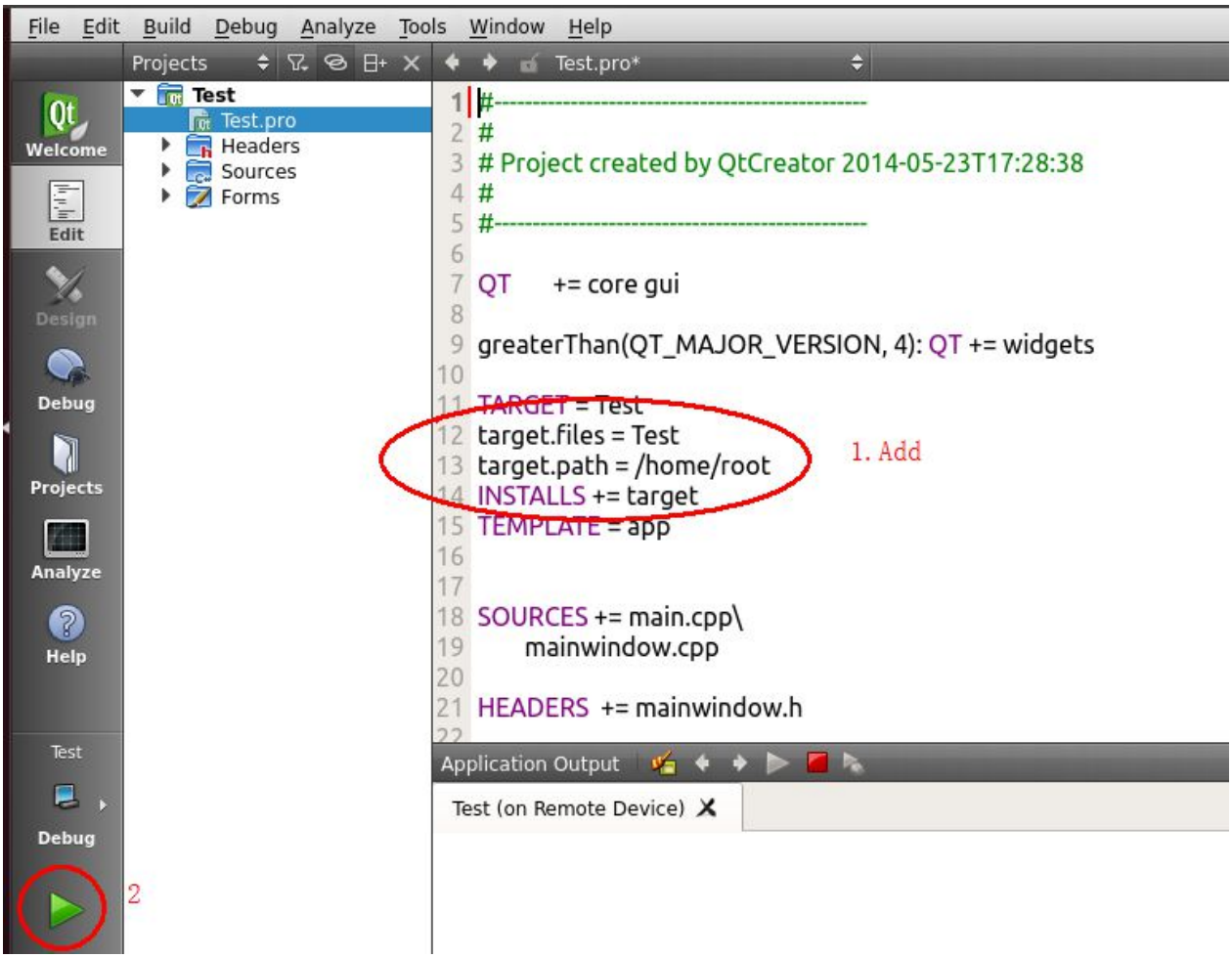

Figure 816: Test.pro file

Now you should see a window on the industrial PC.

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.