IdeaT41

User manual

Ingenic T41 Dual-core platform

Overview

The content of this document is only described for the development board **IdeaT41**, aiming to help users quickly understand, apply and develop the board.

System Support

Development Board	Android	Debian	Linux
IdeaT41_V1 CMT41Z V2	x	x	√

Revision History

Version	Date	Author	Revision History
V1.0	2024-05-29	Boardcon Team	Initial version
V2.0	2025-05-14	Boardcon Team	Modify CMT41Z V2 download

Instructions

No warranty of accuracy is given concerning the contents of the information contained in this document. Boardcon reserves the right to change details in this document without notice.

Boardcon embedded design limited
2508 Haofang Tianji Plaza, 11008 Beihuan Avenue, Nanshan District,
Shenzhen, Guangdong, China. 518051

Website: www.armdesigner.com | www.boardcon.com
Email: market@armdesigner.com
Technical Support Inquiries: support@armdesigner.com
Tel: +86-755-26481393 | +86-755-27571591

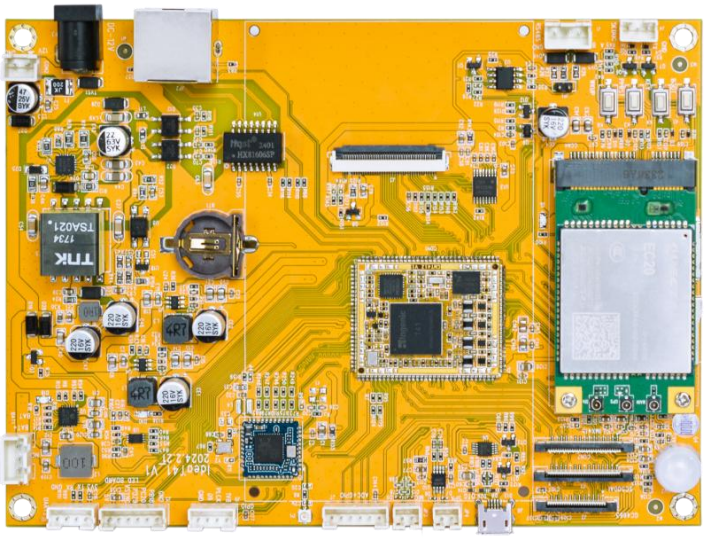
Content

Introduction 4
1. Overview 4
2. Product Parameters 5
3. Hardware Interface Introduction 7
4. T41 SDK Introduction 8
4.1 SDK Directory introduction 9
4.2 SDK Hierarchy 9
Compile SDK 10
1. Compiler Environment 10
1.1 Why do you need to build a development environment 10
1.2 Installing Linux server 10
1.3 Install Tools 11
2. Compile Source 11
2.1 Auto compile 11
2.2 Complie each module 12
2.2.1 unzip SDK 12
2.2.2 Toolchain Installation 12
2.2.3 Compile uboot 12
2.2.4 Compile kernel 13
2.2.5 Build file system 13
2.2.6 Compile driver 13
2.2.7 Compile sample app 14
Burn images 14
1. Preparation 14
2. Install serial debugging assistant 14
2.1.1 How to connect the serial port tool 14
2.1.2 Install driver 15
2.2 Install Serial Terminal Tool 16
3.Burn by USB 19
3.1 Install driver 19
3.2 Run the Burning tool 21
4.Burn by SD card 26
4.1 Make a startup SD card 26
4.2 Burn with SD card 28
5. Burn with TFTP 29
Test System 32
1. Default user 32
2. Ethernet 33
3. LCD/camera 33
4. PIR/KEY 35

5. 4G (EC20) 36
6. SD card 37
7. Light sensor 37
8. Audio 38
9. wifi 39
10. Bluetooth 40
11. Test UART2 40
12. RTC 41
13. Browsing video in PC 41
13.1 setup the board 41
13.2 Browsing video with Carrier 42
13.3 Browsing video with VLC 43

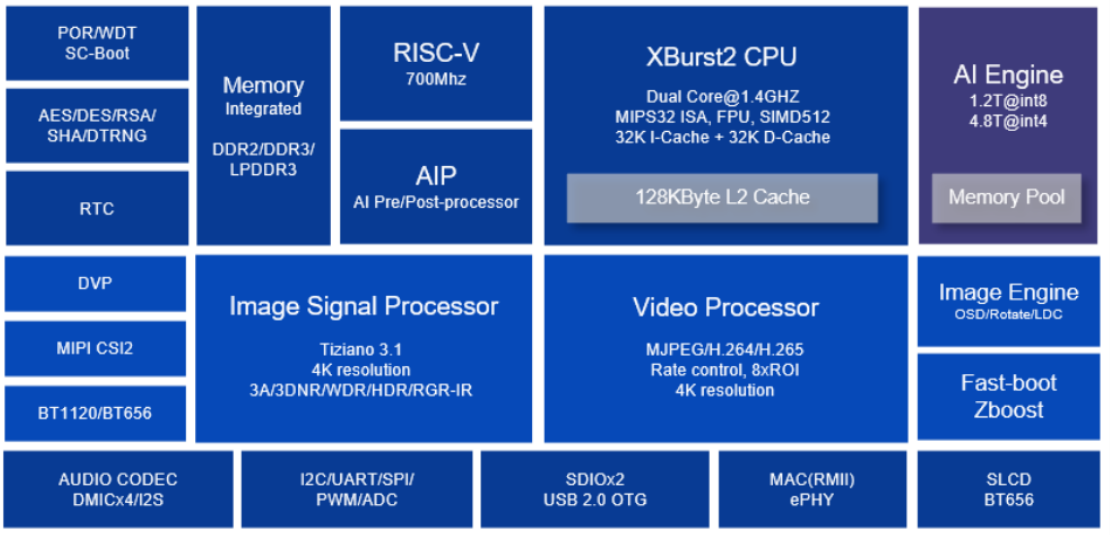
Introduction

1. Overview



Ingenic T41 key features:

1.2T@int8 NPU, professional ISP and video coding, cost-effective, extreme low power consumption, next-generation battery product application. T41 series provide rich version selection to meet different product requirements.

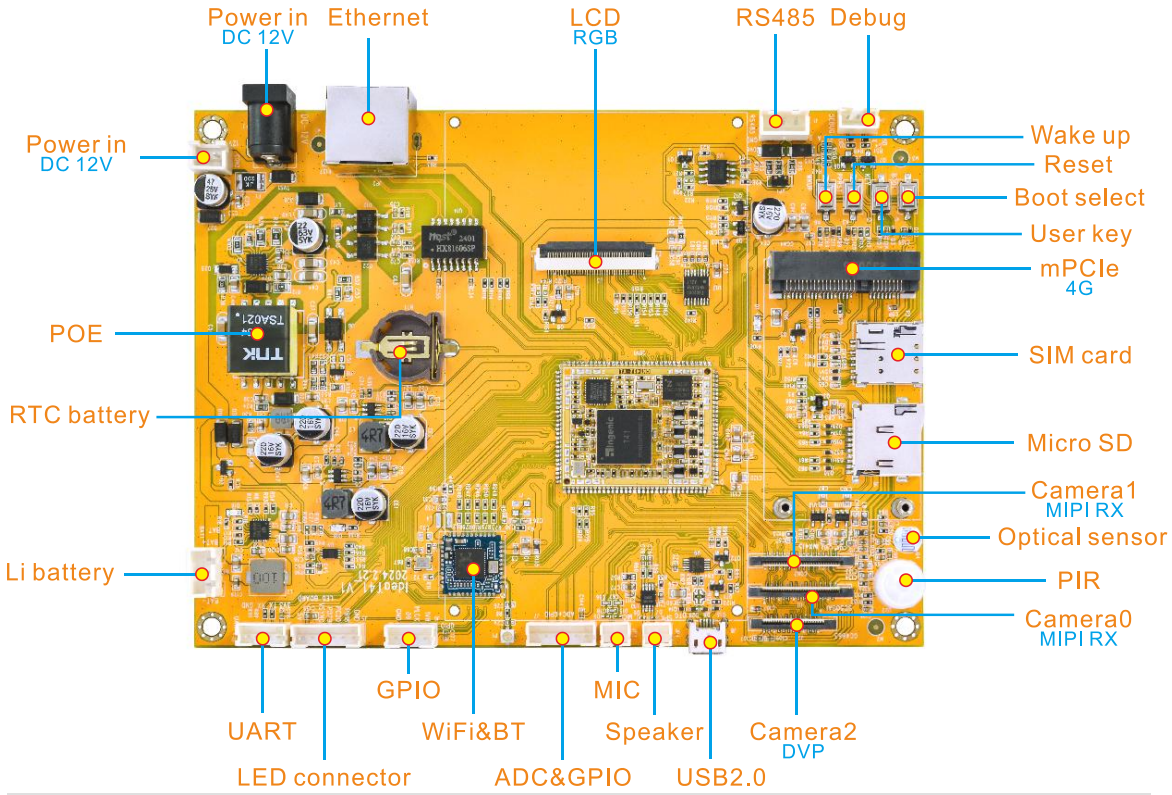


2. Product Parameters

Basic Parameters	
SOC	Ingenic T41
CPU	• XBurst2 Dual Core 1.0~1.4GHz • 512bit SIMD ISA • 128KB L2 cache
MCU	• 700MHz RISC-V coprocessor • RV32IM instruction set
AI Engine	• 1.2Tops@int8, 4.8Tops@int4 • Support int16/int8/int4/int2 bit width • Magik AI algorithm develop platform available
AI Pre/Post-Processing	• Color conversion, Resize • Hardware matrix operations
Video Encoder	• H.264/H.265/JPEG Encoder • Maximum performance:3840x2160@20fps • Support CBR/VBR/SVBR control, 8 ROIs • JPEG snapshot at 8 megapixels
Image Signal Processor	• 3A (AE, AWB, AF) function • BLC, LSC, CSC, DPC, Gamma, Defog • Adaptive Dynamic Range Compression • 2 frames WDR (DOL/frame) • Adaptive Edge-Based Demosaic • Adaptive Local Contrast Enhancement • Multi-Level NR(2DNR/3DNR) and sharpening • X+Y Lens Distortion Correction • Scaler, Clip, Flip and Mirror, 90/270 rotation
RAM	64MB DDR2
ROM	32MB Nor flash

Support system	Linux4.4
Hardware Parameters	
Extended Storage	• Support 1x MicroSD Card
Display	• Support 1x 480*800 TFT LCD output
Audio	• Support 1x Speaker output • Support 1x MIC input
Network	• Support 1x 100Mbps Ethernet • Support 1x WIFI/BT module • Support 1x 4G module
Camera	Support 1x Camera
Peripheral communication	• Support 1x UART
Other parameters	Support 1x Debug, 1x PIR, 1x RTC, Light sensor
Electrical Parameters	
Supply Voltage	12V/3A
RTC input voltage	3V/0.6uA
Operating temperature	0~70°
Storage temperature	-40~85°
Structural Parameters	
Core board dimensions	31.7mm x 25.5mm
Motherboard dimensions	150.0mm x 110.0mm

3. Hardware Interface Introduction

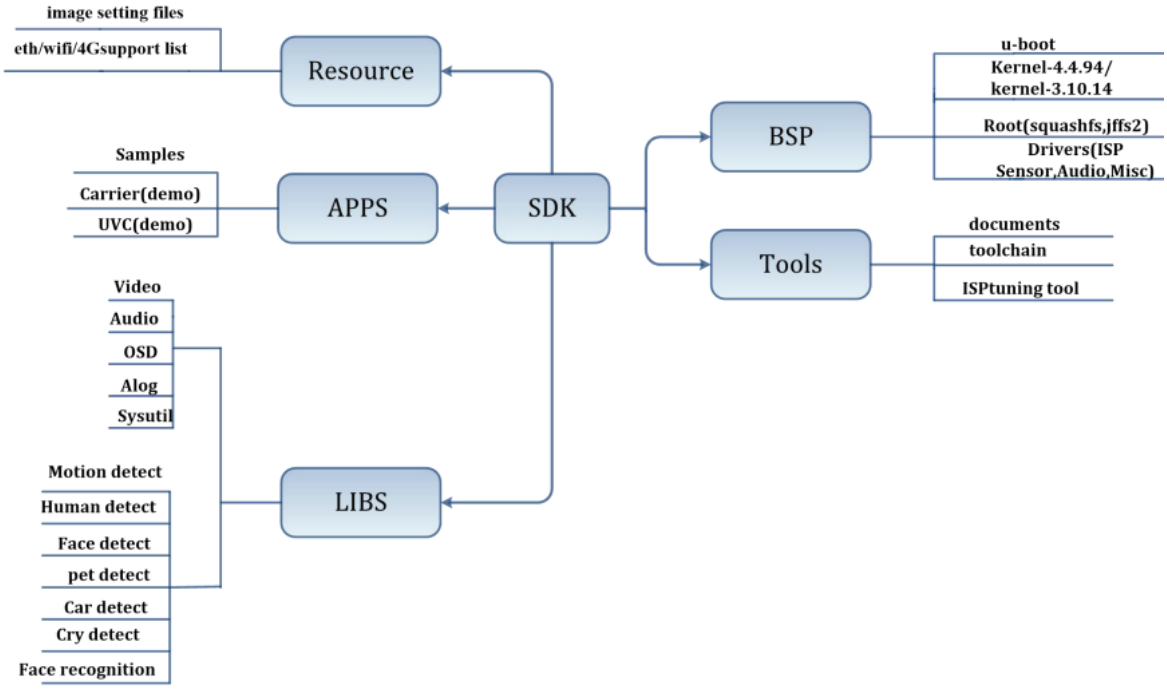


Interface parameters	
Power in	12V DC power input interface
Ethernet	100M Ethernet RJ45 interface
LCD	4.3 inch（480x800） TFT LCD
RS485	RS485 communication interface (Multiplexed with BT, BT is supported by default)
Debug	serial Debug port
Key	Wake up(power key), Reset(HW reset) User key(Back key), Boot select(Home key)
mPCIe 4G	4G model interface(EC20) (Multiplexed with Micro USB, Unplug the Micro USB when using 4G)
Micro SD	MicroSD card slot
Camera0	SC500AI module
Camera1	IMX415 module

Camera2	GC4663 module
USB2.0	As device (Flash images to ROM)
Speaker	2-pin speaker connector
MIC	2-pin MIC connector
ADC&GPIO	ADC/GPIO extension interface
WiFi&BT	Realtek RTL8723DS module
GPIO	GPIO extension interface
LED connector	LED extension interface
UART	UART2, TTL level interface
Li battery	Li battery interface
RTC battery	RTC battery

4. T41 SDK Introduction

T41 ISVP SDK, the software development kit, including API libraries, open-source codes, documentation, samples, etc. It can be used for quick product development. Below is an overview of the ISVP SDK:



4.1 SDK Directory introduction

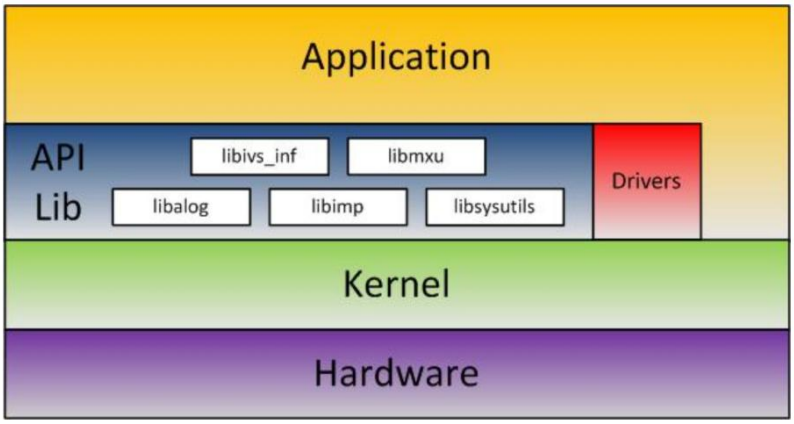
T41 directory structure is as follows:

```
├─── Image
│    ├─── T41L_gcc720_uclibc_32M_202404301536.img
│    └─── T41L_gcc720_uclibc_32M_202404301536.txt
├─── mulu.txt
├─── Schematic
│    ├─── cmt41z v1sch.pdf
│    └─── ideat41_v1.pdf
├─── Source
│    ├─── Ingenic-SDK-T41-1.2.3-20240308-en.7z
│    ├─── t41_env_setup_zh.sh
│    └─── t41_image_mk_zh.sh
├─── Tools
│    ├─── Carrier-V5.1.0.7z
│    └─── cloner-2.5.36-windows_alpha.rar
└─── User manual
     ├─── BSP
     │    ├─── T41 Audio AEC Debugging Guide.pdf
     │    ├─── T41 BSP Development Guide V2.0.pdf
     │    ├─── T41 GPIO Register Operation Instructions.pdf
     │    ├─── T41 SFC Development Guide.pdf
     │    └─── T41 WiFi Porting Guide.pdf
     ├─── IdeaT41 user manual.pdf
     ├─── IMP
     │    ├─── IMP T41 SDK Development GuideV1.0.pdf
     │    └─── T41 Bit Rate Control.pdf
     ├─── Ingenictool
     │    ├─── Ingenic Carrier Tool Usage Guide.pdf
     │    └─── USBCloner Burning tool guide.pdf
     ├─── T41 Development resource compilation.pdf
     ├─── T41 File System Building Guide.pdf
     ├─── T41 SDK Debugging Guide.pdf
     └─── T41 SDK Installation and Usage Guide.pdf
```

4.2 SDK Hierarchy

- Linux Kernel: Kernel layer. Perform basic system functions and define hardware resources.
- drivers: ko module drivers, used for performing hardware operations.
- API lib: Realize the abstraction of hardware functions and facilitate the development of the application layer. API main libraries are set into 5 sections:
- libimp: Multimedia functions. Such as H265/H264 encoding, JPEG encoding, IVS and Audio, etc.
- libsysutils: System functions. Such as reboot, system's time and battery functions, etc.
- libalog: ISVP-SDK log library.

- libivs_inf: Library for IVS algorithm. Such as line detection, perimeter detection, etc.
- libmxu: 512-bit mxu accelerated instruction operator library.
- Application: Features implementation. Kernel interfaces can be used in case of special function requirements, otherwise it is recommended to use the provided drivers and API from SDK for development.



Compile SDK

1. Compiler Environment

1.1 Why do you need to build a development environment

Due to the limited resources of the embedded board, the development and debugging tools cannot be run on the board. It usually needs to be developed and debugged in the way of cross compilation and debugging, that is, the form of “host + target”. Host computer and target computer usually use serial port connection to display interactive information, and network port connection to transmit files.

The host and target processors are generally different. The host needs to establish a cross-compilation environment suitable for the target machine. The program obtains the executable file through “compile-connect-locate” on the host computer. Burn the executable file to the target machine through certain methods, and then run it on the target machine.

1.2 Installing Linux server

It is recommended to use Ubuntu 16.04. If you encounter an error during compilation, you can check the error message and install the corresponding software packages accordingly. Other Linux versions may need to adjust the software package accordingly. In addition to the system requirements, there are other hardware and software requirements.

Hardware requirements	Software requirements

64-bit system, hard disk space should be greater than 40G. If you do multiple builds, you will need more hard drive space.	Ubuntu 16.04 system

1.3 Install Tools

The contents of this directory only provide the software package installation commands that are needed to build the compiled SDK environment. Please install other tools such as samba and ssh by yourself.

PC OS	Network	Permission
ubuntu 16.04	online	root

```
# sudo apt install p7zip-full lzop u-boot-tools mtd-utils git ssh make gcc expect g++ patchelf chrpath gawk texinfo chrpath diffstat binfmt-support qemuuser-static live-build bison flex fakeroo  unzip device-tree-compiler
```

2. Compile Source

2.1 Auto compile

Copy Ingenic-SDK-T41-1.2.3-20240308-en.7z,t41_env_setup_en.sh,t41_image_mk_en.sh to Linux server (ubuntu16.04 is recommended), use the command:

```
$ ls
```

Ingenic-SDK-T41-1.2.3-20240308-en.7z t41_env_setup_en.sh t41_image_mk_en.sh

```
$ ./ t41_env_setup_en.sh
```

After compilation is done, you can get image_t41/T41L_gcc720_uclibc_32M_202405151057.img in _release/.

```
IO/realtek/rtl8723DS/8723ds.mod.o
  LD [M]  /home/qinxueqin/t41/en/Ingenic-SDK-T41-1.2.3-20240308-en/opensource/drivers/wifi/drivers/SD
IO/realtek/rtl8723DS/8723ds.ko
make[1]: Leaving directory '/home/qinxueqin/t41/Ingenic-SDK-T41-1.2.1-20231222-zh/opensource/kernel-4
.4.94'
455+1 records in
233268+0 records out
233268 bytes (233 kB, 228 KiB) copied, 0.533101 s, 438 kB/s
4744+1 records in
9+1 records out
2429299 bytes (2.4 MB, 2.3 MiB) copied, 0.0107543 s, 226 MB/s
2928+0 records in
0+1 records out
1499136 bytes (1.5 MB, 1.4 MiB) copied, 0.00724857 s, 207 MB/s
55808+0 records in
5+1 records out
28573696 bytes (29 MB, 27 MiB) copied, 0.0996114 s, 287 MB/s
The /home/qinxueqin/t41/en/_release/image_t41/T41L_gcc720_uclibc_32M_202405151057.img is ok!
The img sensor driver has:
8723ds.ko hci_uart.ko sensor_sc500ai_t41.ko
 Please use source ./t41_env_setup_en.sh
 Please use source ./t41_env_setup_en.sh
 Please use source ./t41_env_setup_en.sh
qinxueqin@boardcon:~/t41/en$
```

2.2 Complie each module

2.2.1 unzip SDK

```
$ 7z x Ingenic-SDK-T41-1.2.3-20240308-en.7z
```

2.2.2 Toolchain Installation

Toochain, the cross-compiling toolchain, is a collection of tools used on Linux Host machines to compile and debug embedded device programs. ISVP's Toolchain version information is as below:

1) gcc version: 7.2.0

2) libc version: glibc: 2.29; uclibc: 0.9.33.2.

Installation process:

Step1, decompress gcc toolchain:

```
$ cd Ingenic-SDK-T41-1.2.3-20240308-en/resource/toolchain
$ tar xvfj mips-gcc720-glibc229-r5.1.9.tar.bz2
```

Step 2, modify the ~/.bashrc to add the bin path of the toolchain as an environment variable:

```
$ vi ~/.bashrc
```

Add as follow in .bashrc(Please modify your path in fact)

```
$ export
PATH=/home/qinxueqin/t41/Ingenic-SDK-T41-1.2.3-20240308-en/resource/toolchain/mips-gcc720-glibc229-r5.1.9/bin/:$PATH
```

Step 3, test toolchain:

```
$ source ~/.bashrc
$ mips-linux-gnu-gcc --version
```

After executing the test command, if the following message is displayed then your toolchain is well installed.

```
$ mips-linux-gnu-gcc --version
```

mips-linux-gnu-gcc (Ingenic Linux-Release5.1.9-Default_xburst2_glibc2.29 Fix: uclibc popen and pclose 2023.08-15 09:56:16) 7.2.0

Copyright (C) 2017 Free Software Foundation, Inc.

This is free software; see the source for copying conditions. There is NO

warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

2.2.3 Compile uboot

```
$ cd Ingenic-SDK-T41-1.2.3-20240308-en/opensource
$ 7z x uboot.7z
$ cd uboot
$ make distclean
$ make isvp_t41l_msc0          // For SD boot
$ make isvp_t41l_sfc_nor       // For Nor Flash boot
```

Compile the corresponding type of uboot according to your need. After compiling, you can get u-boot-with-spl.bin in current directory.

2.2.4 Compile kernel

```
$ cd Ingenic-SDK-T41-1.2.3-20240308-en/opensource
$ 7z x kernel-4.4.94.7z
$ cd kernel-4.4.94
$ make isvp_marmot_defconfig
$ make uImage
```

After compiling, you can get **uImage** in *arch/mips/boot/.*

2.2.5 Build file system

Nor-Flash Based system has limited storage space, so it uses a compressed file system. The following two file systems are recommended:

A. squashfs: Read-only file system, high compression rate.

B. jffs2: Read/Write file system, optional compression mode.

The recommended system-building scenario is to use squashfs file system for root file system(rootfs) and system partitions that do not need to be modified frequently; jffs2 file system for systems partitions that require frequent reading and writing.

The default partition size in ISVP Demo are as follows:

A. u-boot: 256K

B. uImage: 2560K

C. rootfs: 2048K

D. system: 27904K

So, it requires build two file system:

A, build squashfs file system

```
$ cd Ingenic-SDK-T41-1.2.3-20240308-en/resource/rootfs_720_r519
$ tar xvfj root-uclibc-toolchain720-r519.tar.bz2
$ mksquashfs root-uclibc-toolchain720-r519 root-uclibc-toolchain720-r519.squashfs -comp xz
```

Now, you can get root-uclibc-toolchain720-r519.squashfs in current directory.

B, build jffs2 file system

```
$ cdIngenic-SDK-T41-1.2.3-20240308-en/resource/rootfs_720_r519
$ tar xvfj system_uclibc.tar.bz2
$ mkfs.jffs2 -o system_uclibc.jffs2 -r system_uclibc -e 0x8000 -s 0x1000 -n -l -X zlib --pad=2857369
```

Now, you can get **system_uclibc.jffs2** in current directory.

For details about how to create and use a file system, see "*T41 File System Building Guide*"

2.2.6 Compile driver

T41 Driver compilation considerations:

(1) The kernel should be compiled before compiling the driver.

(2) After compiling the kernel, give priority to compiling the ISP driver. Before compiling, enter the Makefile to specify the kernel path.

(3) After compiling the ISP driver, compile the sensor driver. To compile the sensor driver, you need to enter the Makefile and specify the kernel path and ISP driver path.

(4) Other drivers only need to specify the kernel before compiling.

For example, compile ISP driver:

Modify Ingenic-SDK-T41-1.2.3-20240308-en/opensource/drivers/isp-t41/tx-isp-t41/Makefile

```
-       # ISVP_ENV_KERNEL_DIR = $(PWD)/../../../kernel-4.4.94/
+       ISVP_ENV_KERNEL_DIR = $(PWD)/../../../kernel-4.4.94/
$ cd Ingenic-SDK-T41-1.2.3-20240308-en/opensource/drivers/isp-t41/tx-isp-t41
$ make
```

After compiling, you can get **tx-isp-t41.ko** in current directory.

2.2.7 Compile sample app

Enter isvp-sdk path:
Ingenic-SDK-T41-1.2.3-20240308-en//sdk/samples/. You can see ibimp-samples and libsysutils-samples folders. Enter the folder you need to compile, you will see a Makefile file. Execute the make command to compile the SDK sample file. The generated sample is located in the current path and can be copied to the development board for direct use.
Before compiling, you need to specify the current sensor, open the sample-common.h file, modify the corresponding sensor information, and then compile the sample.
For details about how to compile SDK and run app, please see "*T41 Development resource compilation.pdf*".

Burn images

1. Preparation

First, read the "*T41 BSP Development Guide V2.0.pdf*" to understand the functions, structures, interfaces and other information of the T41 SDK.
1) If the board does not have uboot can burn images by Micro USB or SD card.
2) If the board already has uboot can use the network port to burn uboot, kernel and rootfs into nor flash. We recommend burning by Micro USB.

2. Install serial debugging assistant

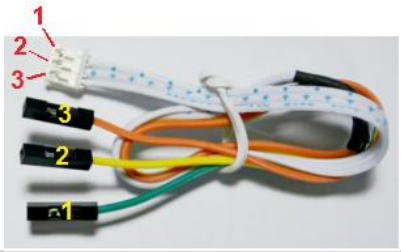
It needs to install serial debugging assistant to interact with the board, the following drivers and software need to be installed (for Windows OS PC):

Number	Driver name	Driver	Use
1	CH9102X	SETUP.EXE	Serial port debugging driver
2	Serial Terminal Tool	SecureCRT.exe	Debugging tool

Install CH9102X Driver

2.1.1 How to connect the serial port tool

Pin	Connection Description
RXD	Receive, connect to RX pin of the board.
TXD	Transmit, connect to TX pin of the board.
GND	Ground, connect to GND pin of the board.
3V3	No need to connect.



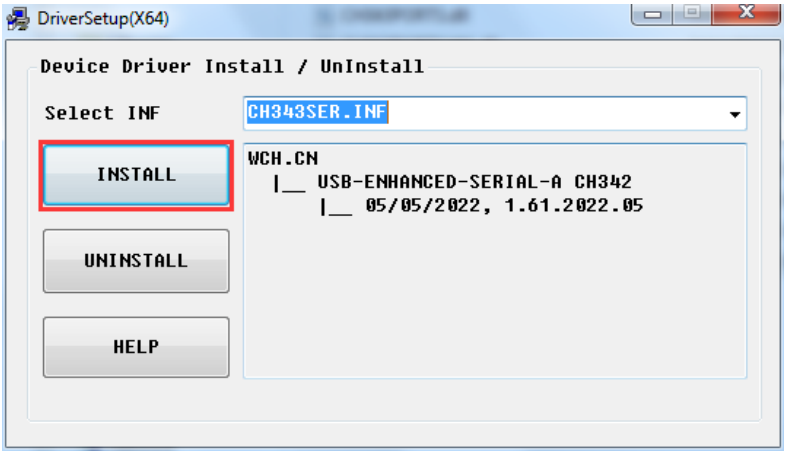
Pin	Connection Description
1	RX, connect to TXD pin of the CH9102X Module.
2	TX, connect to RXD pin of the CH9102X Module.
3	Ground, connect to GND pin of the CH9102X Module.

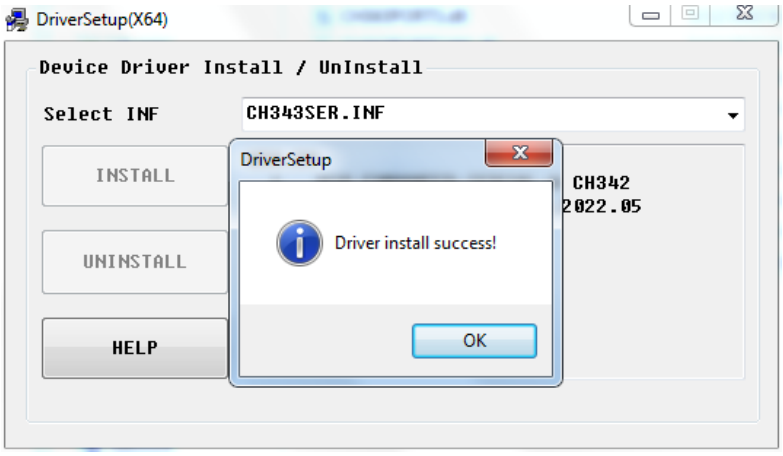
2.1.2 Install driver

Step 1, plug the CH9102X Module to the PC.

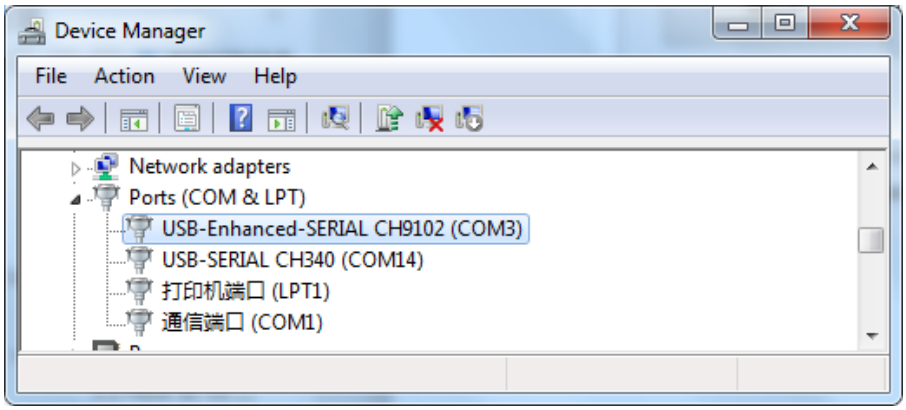
Step 2, unzip CH343SER.ZIP on Windows.

Step 3, select and install the corresponding SETUP.EXE according to the computer properties.





Step 4, after the installation is completed, the device will be listed under **Device Manager -> ports** with unique serial port assigned.



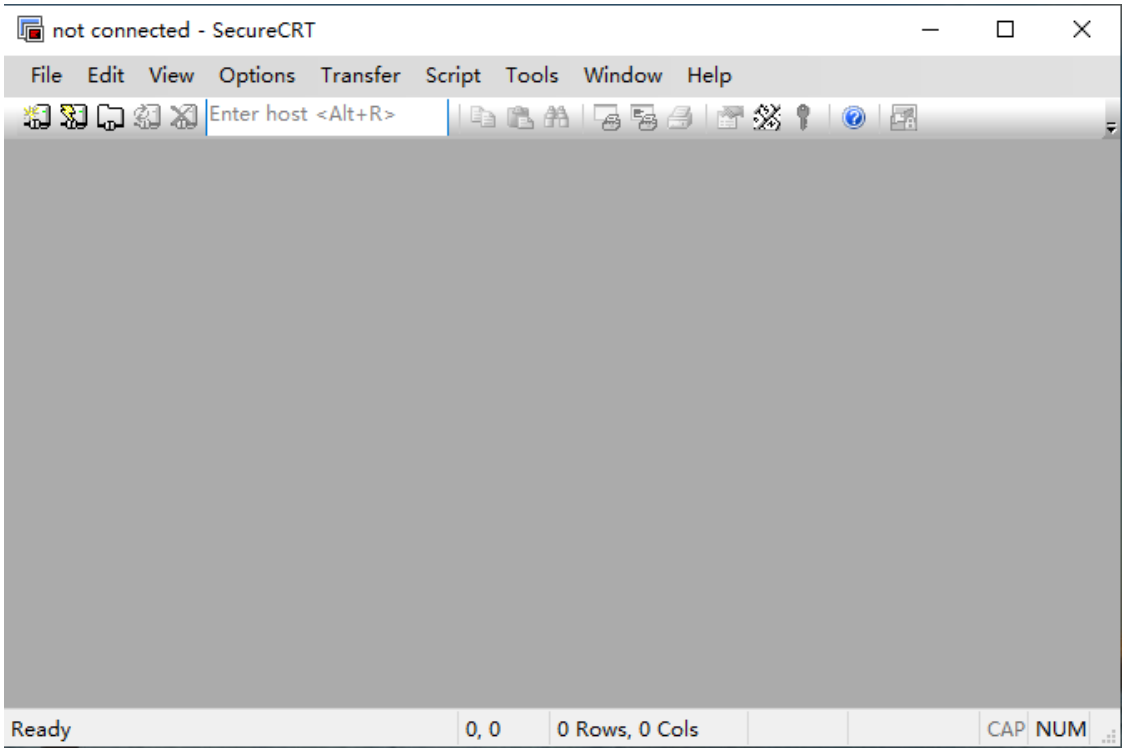
2.2 Install Serial Terminal Tool

The serial port of the development board is responsible for the transmission of interactive information. The kernel print will be sent to the display through serial port, and development board receives commands from keyboard through serial port. Common serial port tools include SecureCRT, putty, Xshell, MobaXterm and other software. The connection between the PC and the serial port of the development board requires configuration parameters, such as serial port number, baud rate, parity bits, data bits, stop bits, etc.

For example: set the serial port number to com3, the baud rate to 115200, 8 data bits, one stop bit and no parity bits. SecureCRT can be used directly after decompression, set detail as following:

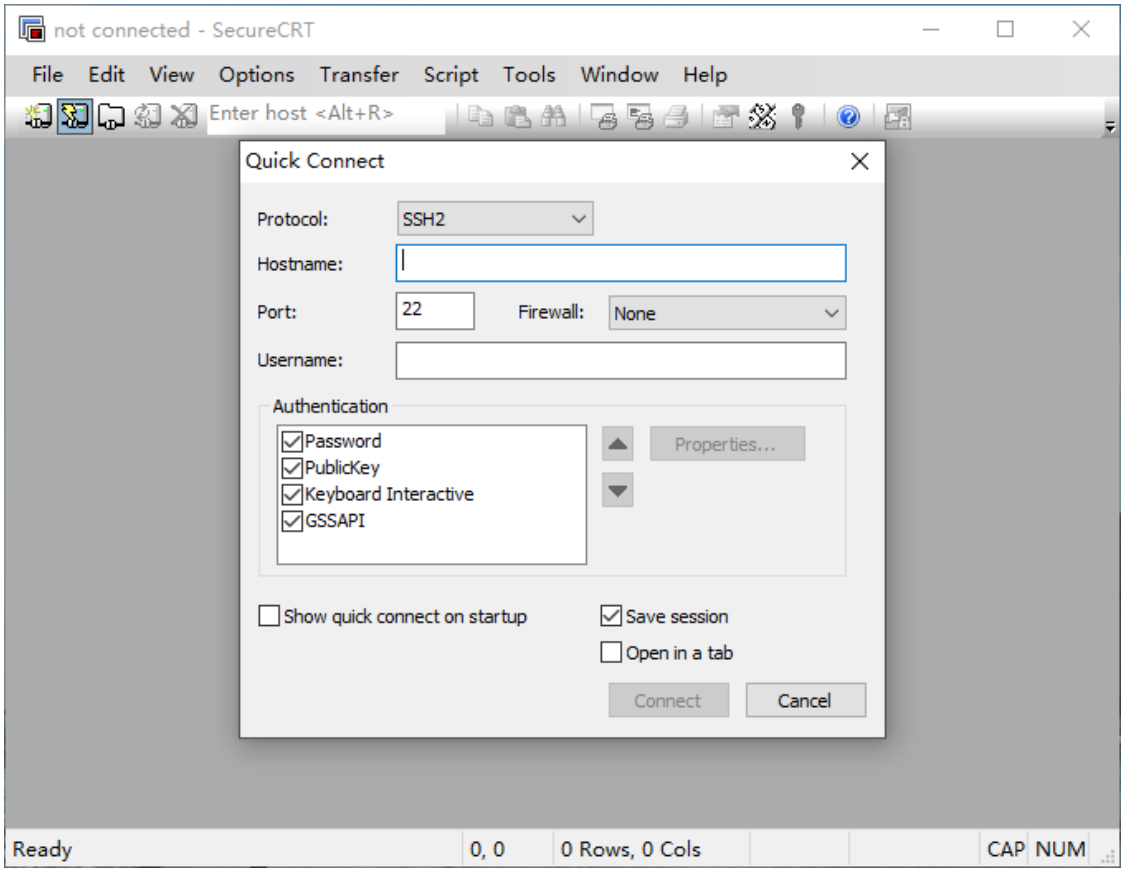
Step 1, unzip Platform/SecureCRT.rar on PC.

Step 2, click SecureCRT/SecureCRT.exe open the SecureCRT.

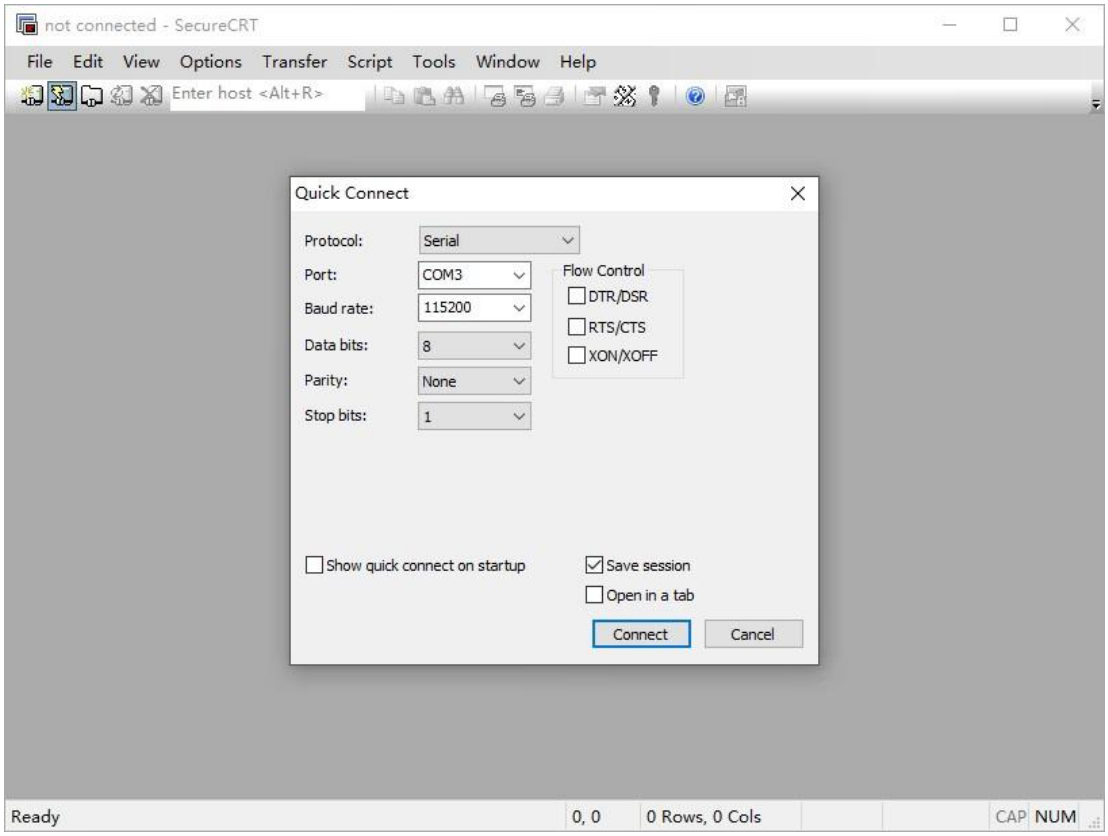


Step 3, confirm that the CH9102X driver has been installed and the CH9102 module is connecting to the PC.

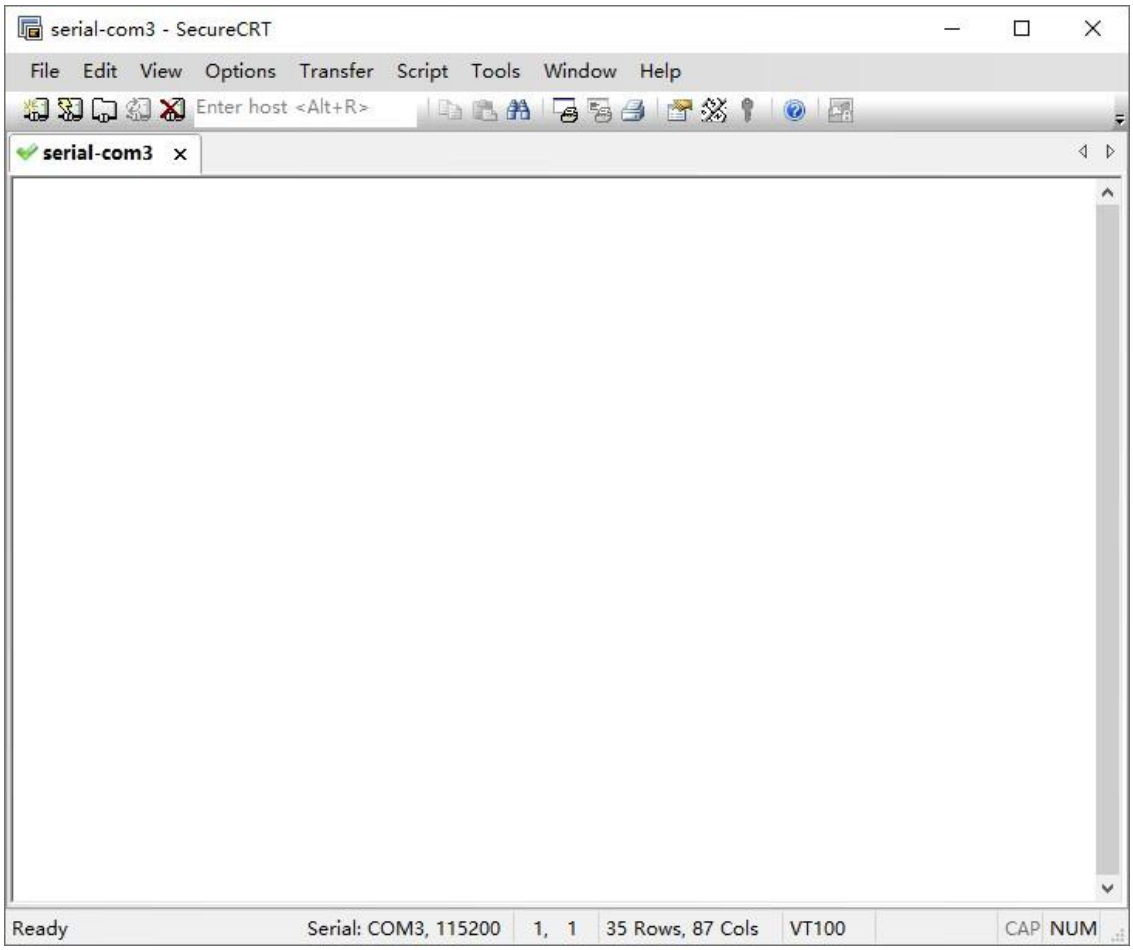
Step 4, click the **Quick Connect** button to go to the Quick Connect configuration screen.



Step 5, configure as shown in the following figure:



Step 6, after clicking connect button, the terminal serial interface will be successfully accessed.



3.Burn by USB

USBCloner is a burning tool developed by Ingenic that based on a new code architecture. USBCloner supports the following systems Windows XP, Win7 and above version（32bit, 64bit).

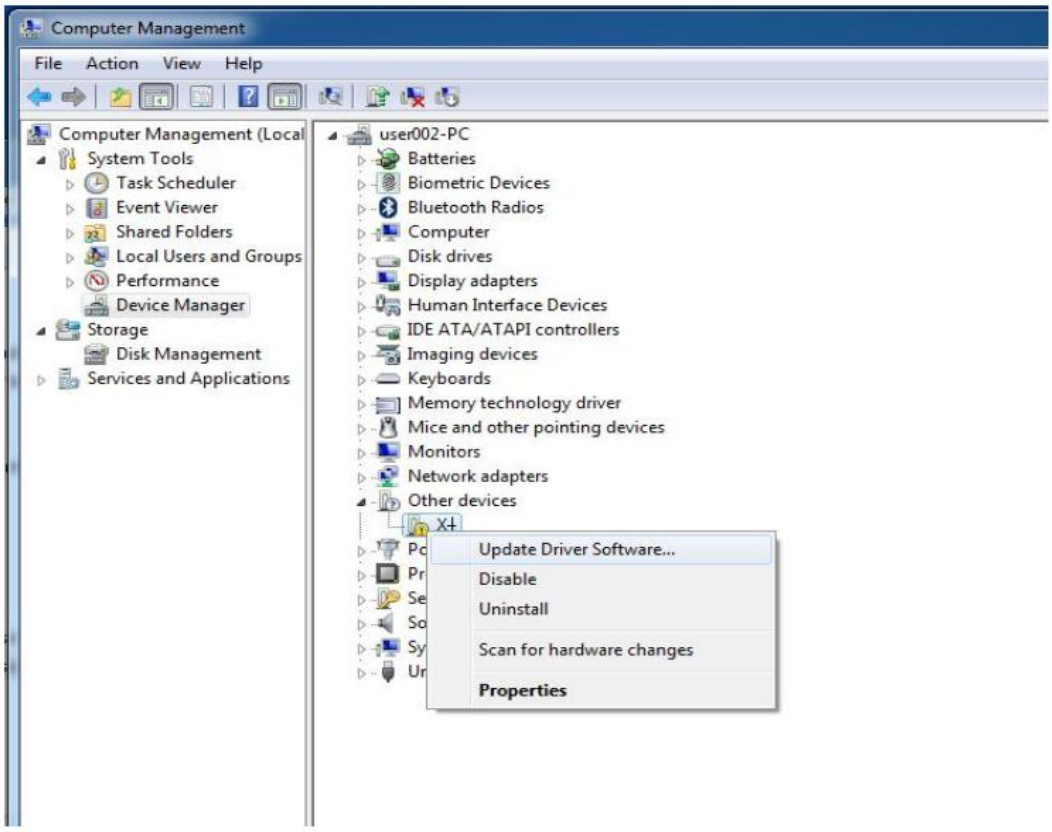
3.1 Install driver

This section uses Windows7 as an example to describe how to install the driver on a Windows host. USBCloner drivers do not require Microsoft signature authentication.

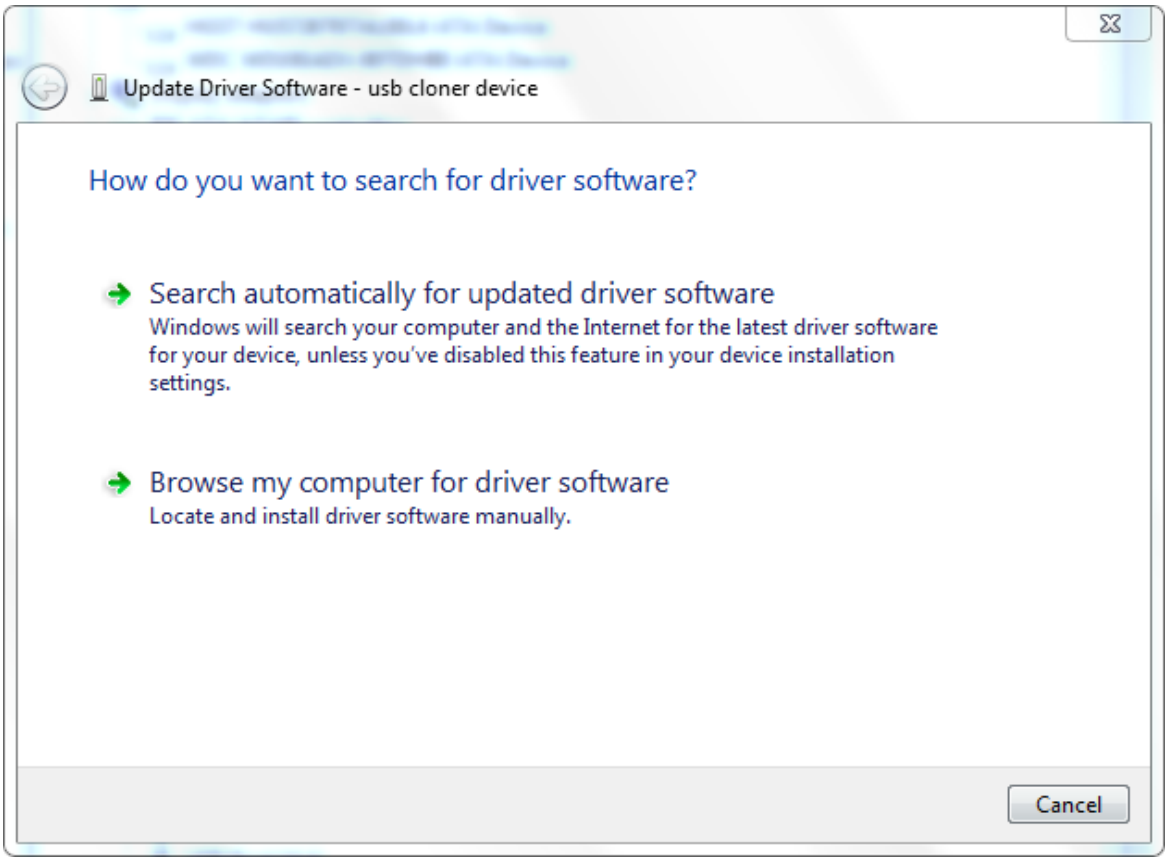
Note: When installing a driver in Windows, be sure to follow the instructions in the documentation. Before installing drivers in Windows 8, you need to configure Disable Driver forced signature. You can find the details online.

Step 1: Please connect the device to the PC, and make sure it's under burning mode (by pressing the **BOOT** key and the **PPRST** key). Power the board then the desktop will pop up a dialog with a wizard of installation and prompt the driver installation failure.

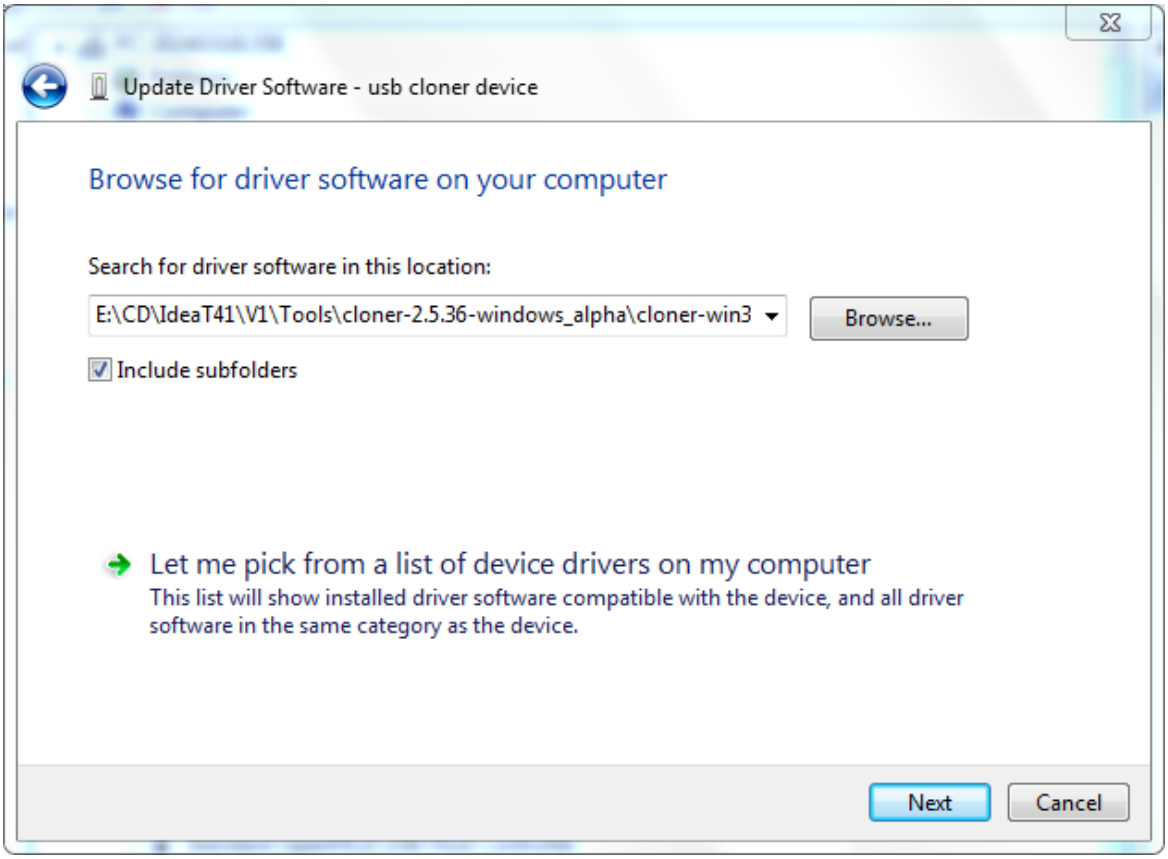
Step 2: Right-click "**Computer**" and enter "**Manage**". In "**Device Manager**" you can find that the current installation is not successful under other devices.



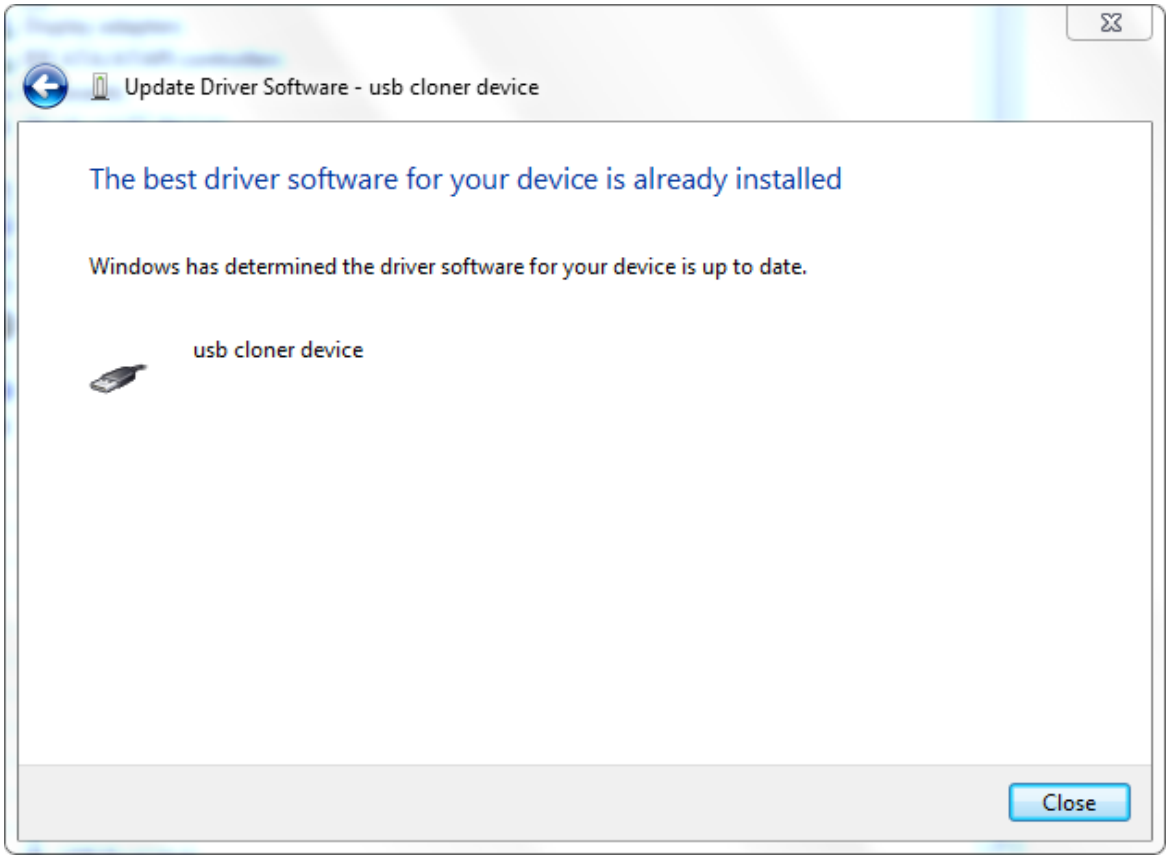
Step 3: Right-click on the current USB device and choose "**Update Driver Software**".

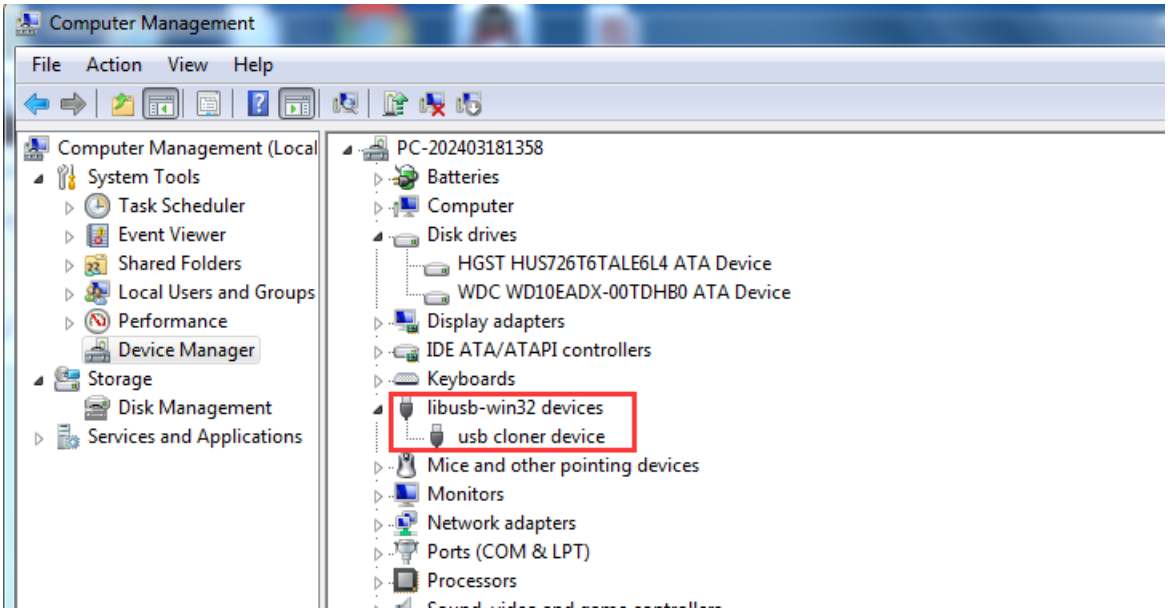


Step 4: Select "**Browse my computer to find driver software**" click "**Browse**" and select the driver directory **Tools\cloner-2.5.36-windows_alpha\cloner-win32-driver**.



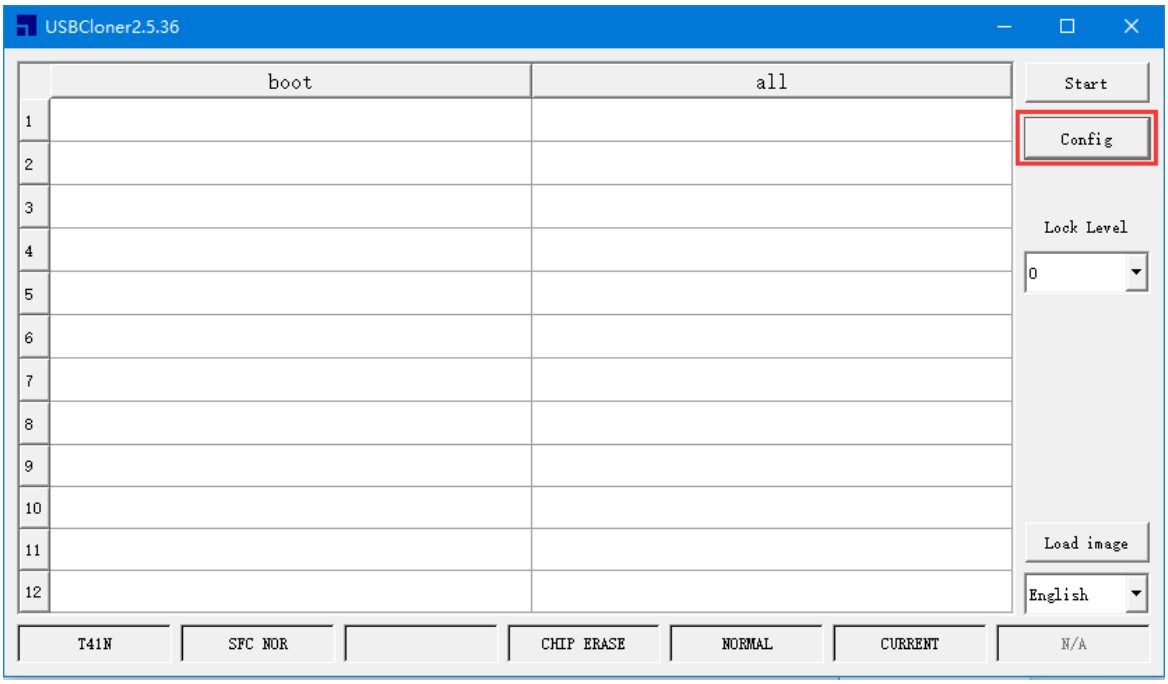
Step 5: Click "OK" and then "Next" to install the driver.



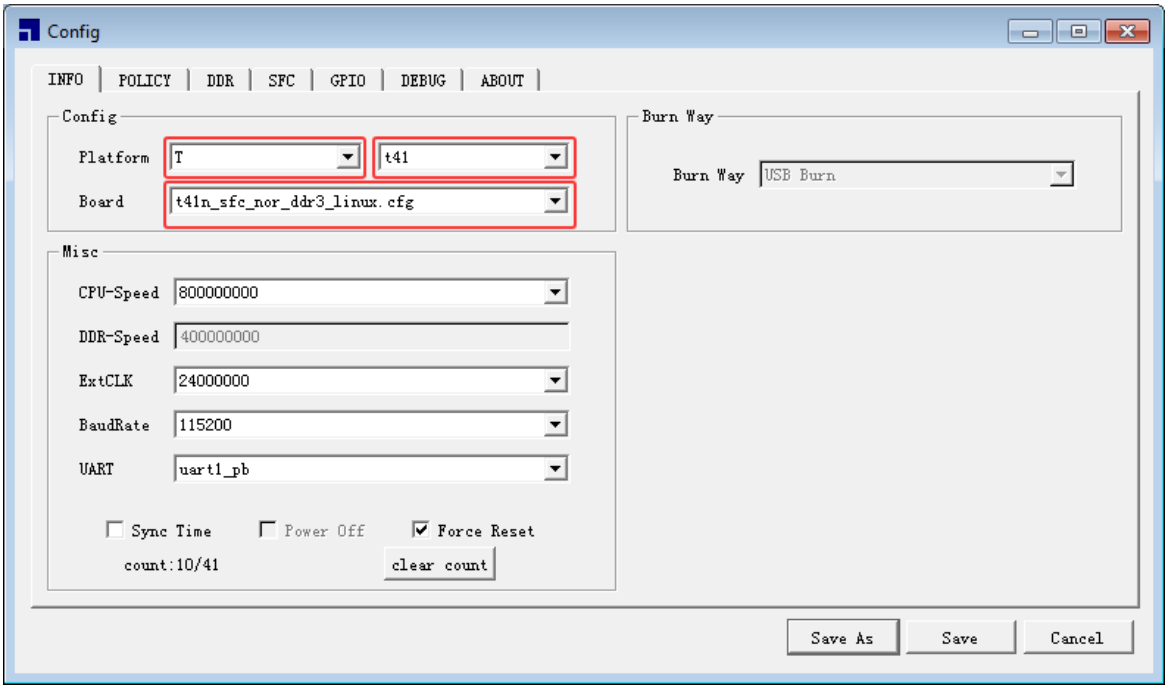


3.2 Run the Burning tool

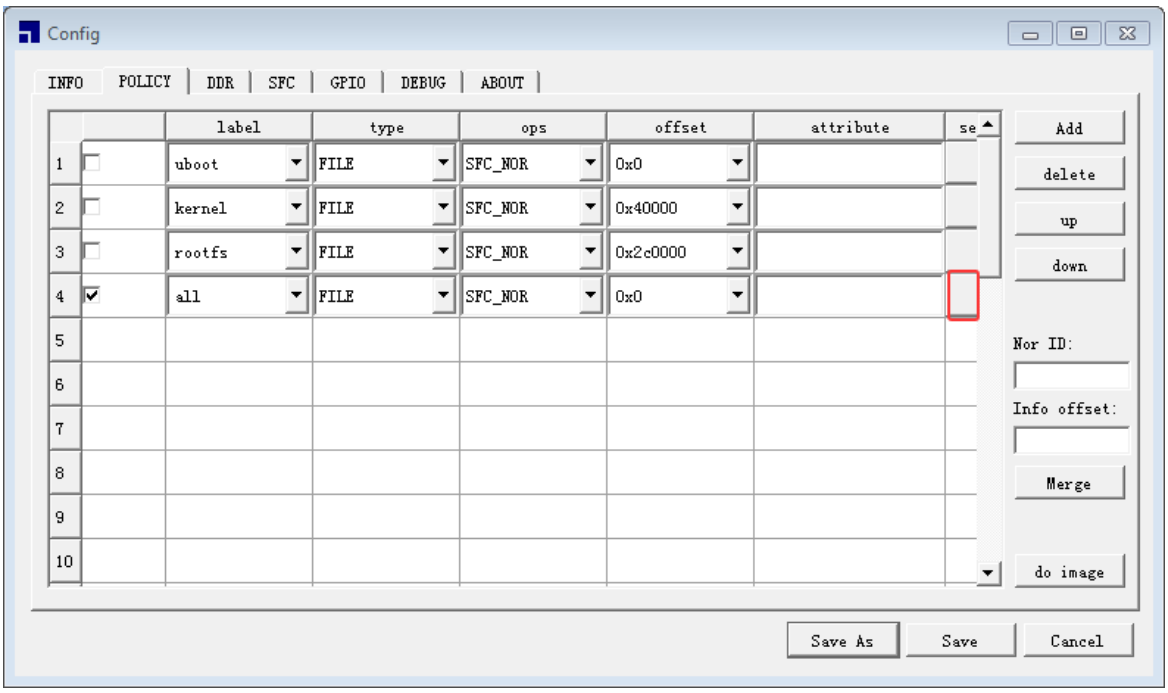
Step 1, decompress the cloner-2.5.36-windows_alpha.rar and double-click cloner.exe in the cloner-2.5.36-windows_alpha to run the burning tool, the operating interface will pop up as follow:



Step 2, Click "**Config**" button to enter the configuration interface. In the "**INFO**" table select the **platform** and **board** level support as follow picture:



Step 3, In the "**POLICY**" table select the image you want to burn, as follow picture:



uboot: u-boot-with-spl.bin

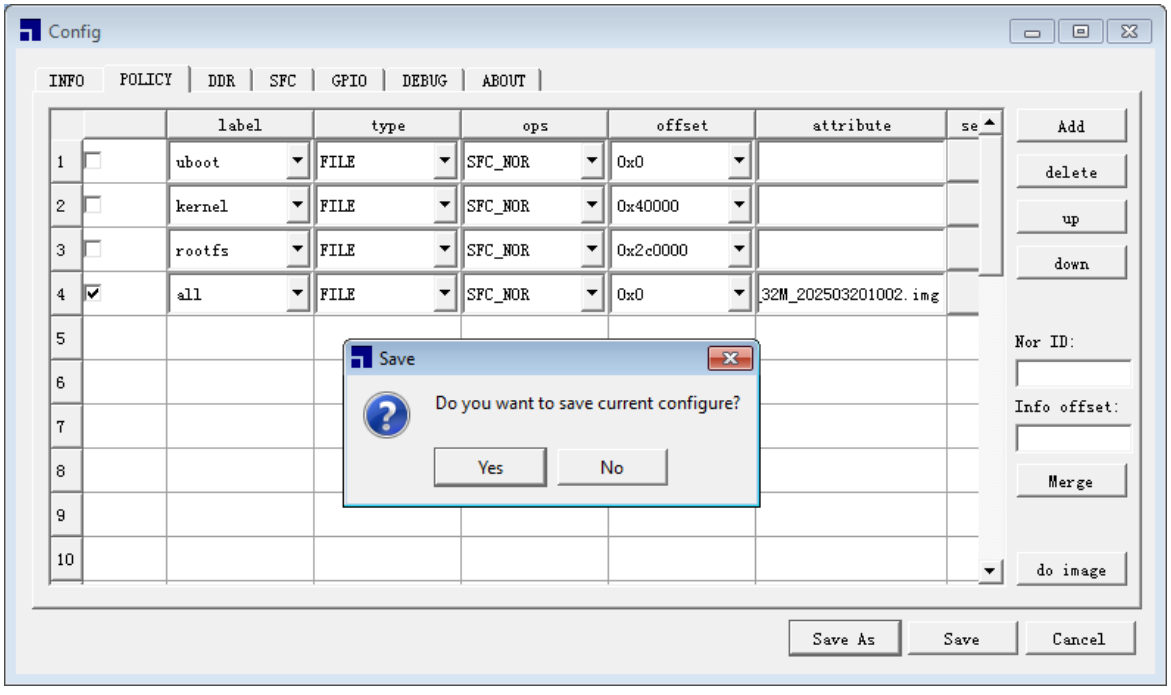
kernel: uImage

rootfs: root-uclibc-toolchain720-r519.squashfs

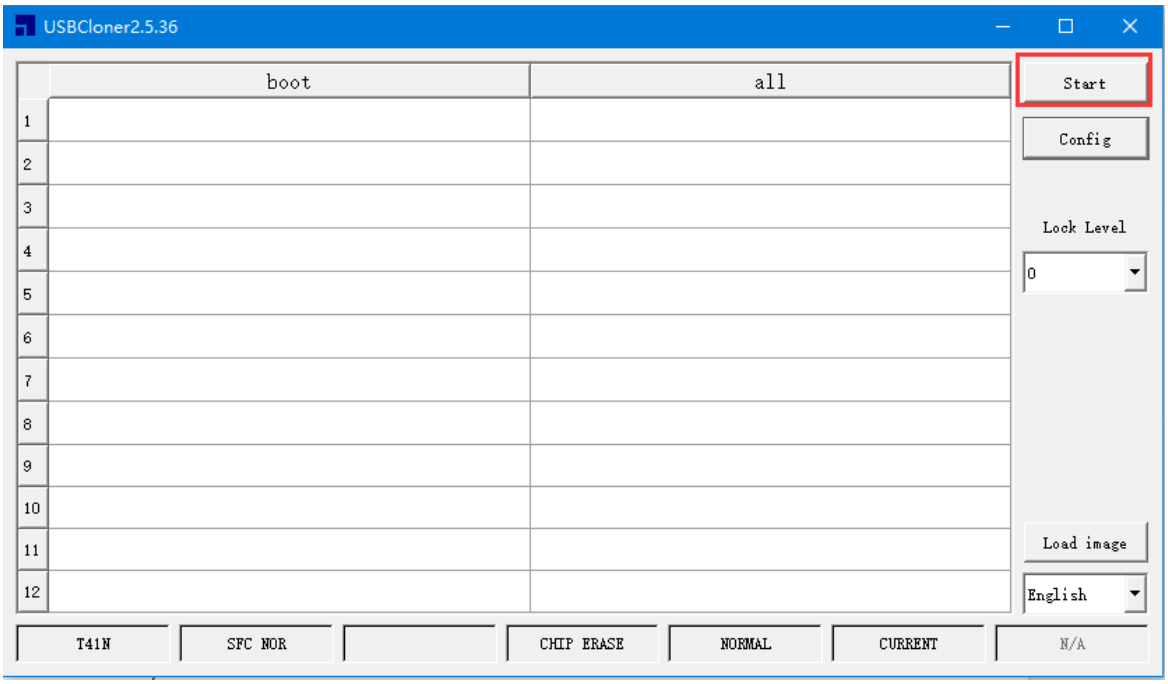
system: system.jffs2

all: T41N_gcc720_uclibc_nor_32M_202503201002.img (include uboot, kernel, rootfs, system)

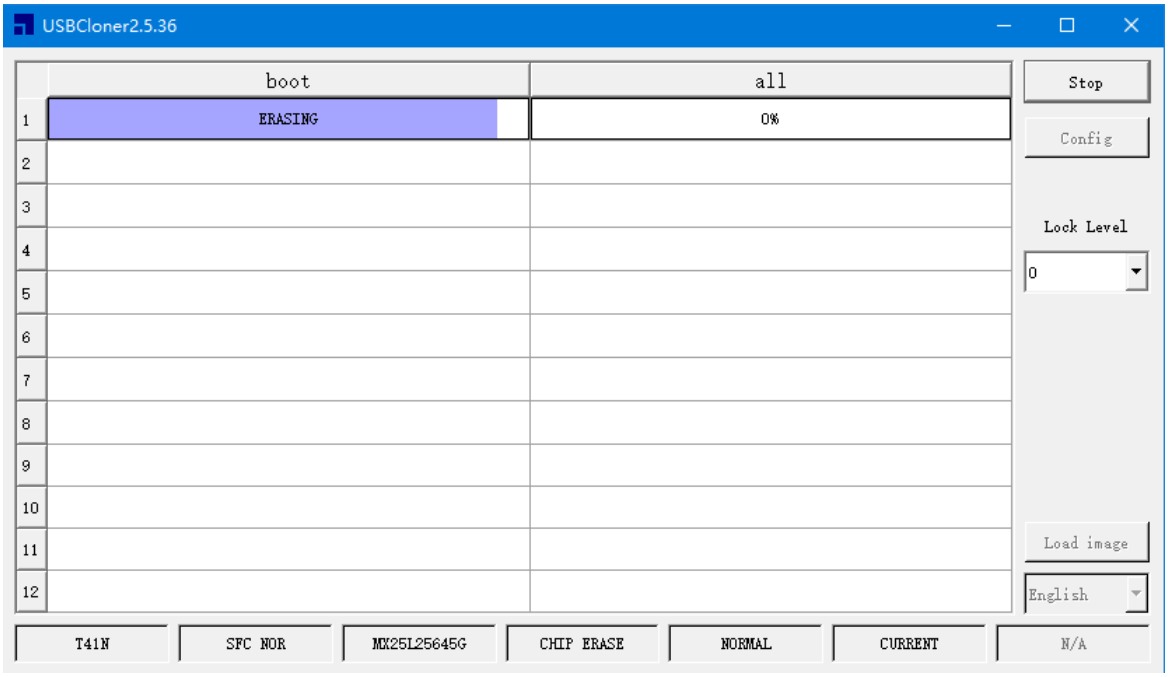
Step 4, Click "**Save**" and then "**Yes**"

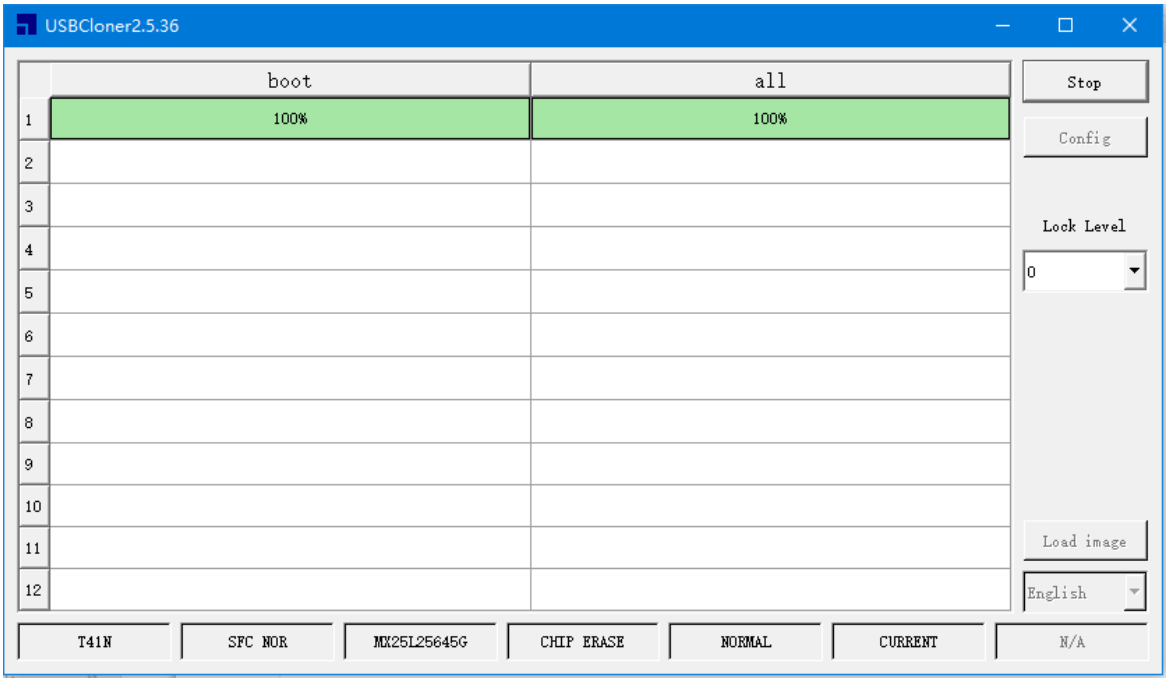


Step 5, Click "**Start**".



Step 6, Connect the device to PC via micro USB, and make sure it's under burning mode (by pressing the **BOOT** key), power the board, it will burn automatically and reboot the system.



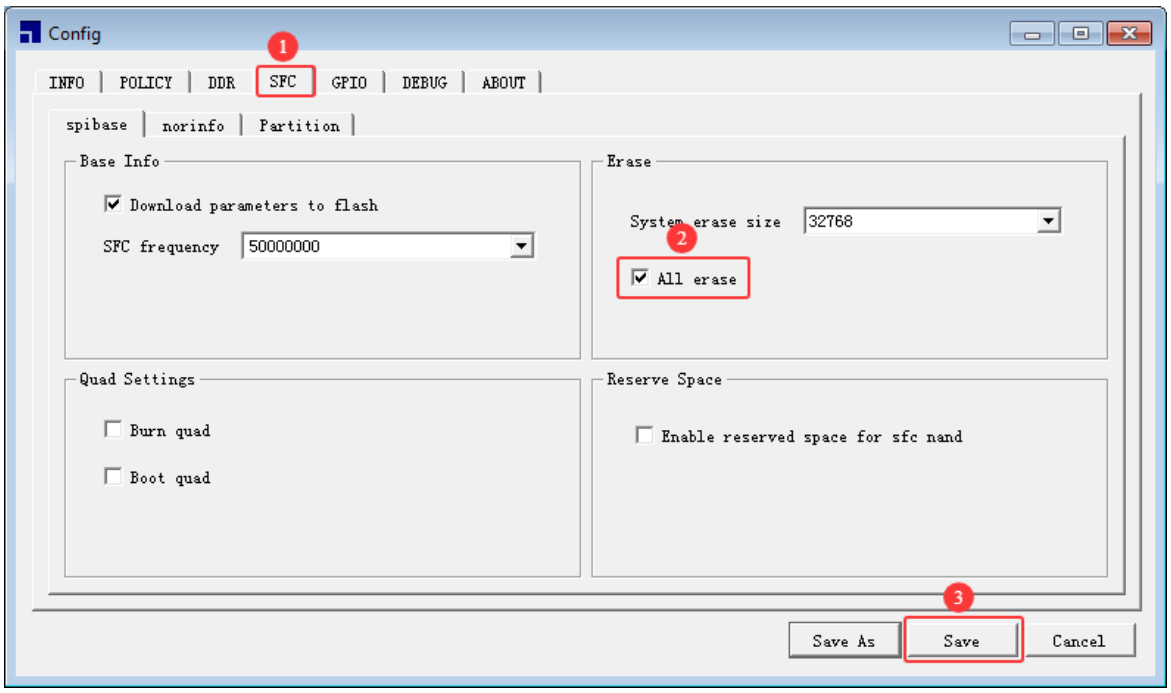


Now, you can see log in serial terminal.

```
[    2.862806] mmc0: SDHCI controller on ingenic-sdhci [13060000.msc] using ADMA
[    2.874972] mmc0: card inserted
[    2.908713] mmc0: Got command interrupt 0x00000001 even though no command operation was in progress.
[    2.932783] mmc1: SDHCI controller on ingenic-sdhci [13070000.msc] using ADMA
[    2.940804] usbcore: registered new interface driver usbhid
[    2.946794] usbhid: USB HID core driver
[    2.951000] Netfilter messages via NETLINK v0.30.
[    2.956142] ip_set: protocol 6
[    2.959577] ip_tables: (C) 2000-2006 Netfilter Core Team
[    2.965383] NET: Registered protocol family 17
[    2.971795] soc_vpu probe success,version:1.0.0-03203fd46d
[    2.978333] input: gpio-keys as /devices/platform/gpio_keys/input/input0
[    2.986111] rtc-ingenic 10003000.rtc: setting system clock to 2020-03-01 15:44:31 UTC (1583077471)
[    2.997793] mmc0: new high speed SDHC card at address b368
[    3.004825] mmcblk0: mmc0:b368 NCard 29.1 GiB
[    3.011048]  mmcblk0: p1
[    3.016468] VFS: Mounted root (squashfs filesystem) readonly on device 31:2.
[    3.028280] ingenic,sdhci 13070000.msc: card claims to support voltages below defined range
[    3.038034] devtmpfs: mounted
[    3.041320] Freeing unused kernel memory: 228K
[    3.058105] random: nonblocking pool is initialized
[    3.066779] mmc1: new high speed SDIO card at address 0001
mdev is ok......
net.core.wmem_max = 26214400
net.core.wmem_default = 26214400
[    4.821245] jffs2: notice: (537) jffs2_build_xattr_subsystem: complete building xattr subsystem, 0 of xdatum
, 0 orphan) and 0 of xref (0 dead, 0 orphan) found.
[    4.845208] Bus Mode Reg after reset: 0x00020101, cnt=0

Ingenic-uc1_1 login: [    7.782927] dwc-mac 134b0000.mac eth0: Link is Up - 100Mbps/Full - flow control rx/tx

Ingenic-uc1_1 login:
```

For details about how to burn image by USBConer, please refer "*USBCloner The Burn tool Quick Guide.pdf*".

Note: If the board has been burned, Do the following to erase all Nor FLASH and the FLASH will be erased before burning.



4.Burn by SD card

If the board does not have uboot, it needs to make a booting SD card first.

4.1 Make a startup SD card

Format the SD card, then burn the booting uboot to the SD card. This step only needs to be performed once, if you do not update the booting uboot, you do not need to perform this step again. The SD card processed by this step can be used as a normal card.

Insert the SD card into the computer (Ubuntu system).

Step 1, Umount SD card

```
# umount /media/user/*
```

Step 2, Repartition the SD card

```
root@linaro-alip:/# sudo fdisk /dev/sda

Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

[  704.795273]  sda: sda1

Command (m for help): o
Created a new DOS disklabel with disk identifier 0x8b4b4398.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-31116287, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-31116287, default 31116287):


Created a new partition 1 of type 'Linux' and of size 14.8 GiB.
Partition #1 contains a vfat signature.

Do you want to remove the signature? [Y]es/[N]o: y

The signature will be removed by a write command.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
[  926.245341]  sda: sda1
Syncing disks.

[  926.259697]  sda: sda1
root@linaro-alip:/# sync
root@linaro-alip:/#
```

Step 3, Execute "**sync**" command, remove the card, then reinsert the card, the PC will re-identify the partition.

Step 4, Format the SD card as a common VFat file system:

```
# mkfs.vfat /dev/sda1    //The repartitioned here is sda1
```

```
root@linaro-alip:/# sync
root@linaro-alip:/# [ 1045.774454] usb 6-1: USB disconnect, device number 2
[ 1048.059220] usb 6-1: new SuperSpeed Gen 1 USB device number 3 using xhci-hcd
[ 1048.081086] usb 6-1: New USB device found, idVendor=05e3, idProduct=0749, bcdDevi
ce=15.35
[ 1048.081145] usb 6-1: New USB device strings: Mfr=3, Product=4, SerialNumber=5
[ 1048.081164] usb 6-1: Product: USB3.0 Card Reader
[ 1048.081180] usb 6-1: Manufacturer: Generic
[ 1048.081195] usb 6-1: SerialNumber: 000000001536
[ 1048.084521] usb-storage 6-1:1.0: USB Mass Storage device detected
[ 1048.085374] usb-storage 6-1:1.0: Quirks match for vid 05e3 pid 0749: 420
[ 1048.085844] scsi host2: usb-storage 6-1:1.0
[ 1048.086838] pwm-backlight backlight1: supply power not found, using dummy regulat
or
[ 1049.114319] scsi 2:0:0:0: Direct-Access     Generic  MassStorageClass 1536 PQ: 0
ANSI: 6
[ 1049.403991] sd 2:0:0:0: [sda] 31116288 512-byte logical blocks: (15.9 GB/14.8 GiB
)
[ 1049.406089] sd 2:0:0:0: [sda] Write Protect is off
[ 1049.407194] sd 2:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't
 support DPO or FUA
[ 1049.428538]  sda: sda1
[ 1049.433070] sd 2:0:0:0: [sda] Attached SCSI removable disk
[ 1049.433579] pwm-backlight backlight1: supply power not found, using dummy regulat
or

root@linaro-alip:/# mkfs.vfat /dev/sda1
mkfs.fat 4.2 (2021-01-31)
root@linaro-alip:/#
```

Step 5, Compile booting u-boot

Select appropriate type of u-boot(2.2.3 uboot compile), burn the file **u-boot-with-spl.bin** to the SD card's 17 KB offset(is dev/sda partition, not dev/sda1):

dd if=u-boot-with-spl-sd.bin of=/dev/sda bs=1024 seek=17

```
root@linaro-alip:/# ls
bin   lib         rockchip-test     srv                   usr
boot  lost+found  root              sys                   var
data  media       run               system                vendor
dev   mnt         sbin              tmp
etc   oem         sdcard            u-boot-with-spl-sd.bin
home  opt         sha256sum.README  udisk
info  proc        sha256sum.txt     userdata
root@linaro-alip:/# dd if=u-boot-with-spl-sd.bin of=/dev/sda bs=1024 seek=17
[ 1451.604425]  sda: sda1
226+1 records in
226+1 records out
232236 bytes (232 kB, 227 KiB) copied, 0.0695728 s, 3.3 MB/s
root@linaro-alip:/#
```

Now, the SD card that can be used for burning has been completed. Now you can use this card to boot the board.

Note: Make sure the SD card 's device node file is sda (it may be sdb or sdc, which can be confirmed by plugging and unplugging the SD card); if the device node is wrong, it may damage the computer's hard disk file system.

4.2 Burn with SD card

1) Put the firmware on SD card. Insert the SD card into the board, power on the board and type any keys in terminal.

2) In uboot environment, execute $ fatls mmc 0 to view the files in SD card.

3) In uboot environment, execute $ fatload mmc 0 mem_addr file_name to transfer the file file_name to

the mem_addr memory of uboot.

For example:

$ fatload mmc 0 0x80600000 u-boot-with-spl-nor.bin

$ fatload mmc 0 0x80640000 uImage

$ fatload mmc 0 0x808c0000 root-uclibc-toolchain720-r519.squashfs

Download u-boot-with-spl-nor.bin, uImage, root-uclibc-toolchain720-r519.squashfs to the memory at 0x806000000, 0x80640000, 0x808c0000.

4) Burn firmware to flash

Write firmware to the board's nor flash, and you can read it directly from the flash after restarting, instead of manually loading it to the memory every time. The burning commands are as follows:

$ sf0 probe;sf0 erase 0x0 0x1000000;sf0 write 0x80600000 0x0 0x1000000

```
*** Warning - bad CRC, using default environment

Data transfer:    serial
Net:   Jz4775-9161
Hit any key to stop autoboot:  0
isvp_t41# fatls mmc 0
  1499136   root-uclibc-toolchain720-r519.squashfs
 28573696   system.jffs2
   233268   u-boot-with-spl-nor.bin
   232236   u-boot-with-spl-sd.bin
  2489065   uimage

5 file(s), 0 dir(s)

isvp_t41# fatload mmc 0 0x80600000 u-boot-with-spl-nor.bin
reading u-boot-with-spl-nor.bin
233268 bytes read in 37 ms (6 MiB/s)
isvp_t41# fatload mmc 0 0x80640000 uImage
reading uImage
2489065 bytes read in 225 ms (10.5 MiB/s)
isvp_t41# fatload mmc 0 0x808c0000 root-uclibc-toolchain720-r519.squashfs
reading root-uclibc-toolchain720-r519.squashfs
1499136 bytes read in 141 ms (10.1 MiB/s)
isvp_t41# sf0 probe;sf0 erase 0x0 0x1000000;sf0 write 0x80600000 0x0 0x1000000
SF: Detected MX25L25645G/35F, flash size: 32MB, manufacturer id: c2

--->probe spend 7 ms
SF: 16777216 bytes @ 0x0 Erased: OK
--->erase spend 70740 ms
SF: 16777216 bytes @ 0x0 Written: OK
--->write spend 20154 ms
isvp_t41#
```

Now, can type "**boot**" command or press "**PPRST**" key to boot system.

5. Burn with TFTP

TFTP burn only the board has been burned with uboot.

Make sure the development board and the PC are connected to the same router are in the same network segment. Install and set up tftp service on PC, put the u-boot and other firmwares in the tftp root directory. The board supports network driver and tftp operation during uboot phase, therefore you can directly download firmware from PC to the memory of uboot via tftp, and then write it to the flash.

The steps are as follows:

1) Install tftp in PC(ubuntu system)

```
$ sudo apt-get update
$ sudo apt-get install xinetd tftpd-hpa
```

2) Config tftp in ubuntu

```
$ sudo vi /etc/xinetd.d/tftp
```

Add the following configuration:

```
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /var/lib/tftpboot
        disable                 = no
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
$ mkdir /home/qinxueqin/tftp      // You can defined your own directory and need to be consistent.
$ chmod 777 /home/qinxueqin/tftp
$ sudo service tftpd-hpa start
$ vi /etc/default/tftpd-hpa
```

Add the following configuration:

```
# /etc/default/tftpd-hpa

        TFTP_USERNAME="tftp"
        TFTP_DIRECTORY="/home/qinxueqin/tftp"
        TFTP_ADDRESS=":69"
        TFTP_OPTIONS="--secure"
$ service tftpd-hpa restart
```

Copy u-boot-with-spl-nor.bin, uImage, root-uclibc-toolchain720-r519.squashfs to /home/qinxueqin/tftp/.

3) Load firmware to memory through tftp command

In uboot phase, execute $ tftpboot mem_addr file_name to transfer the file to the specified memory location of uboot.

For example:

```
$ setenv serverip 192.168.0.130      //ubuntu IP address
$ setenv gatewayip 192.168.0.2
$ setenv ipaddr 192.168.0.11
$ tftpboot 0x80600000 u-boot-with-spl-nor.bin
$ tftpboot 0x80640000 uImage
$ tftpboot 0x808c0000 root-uclibc-toolchain720-r519.squashfs
```

Download u-boot-with-spl-nor.bin, uImage, root-uclibc-toolchain720-r519.squashfs to the memory at 0x806000000, 0x80640000, and 0x808c0000.

```
Hit any key to stop autoboot:  0
isvp_t41# setenv serverip 192.168.0.130
isvp_t41# setenv gatewayip 192.168.0.2
isvp_t41#  setenv ipaddr 192.168.0.11
isvp_t41# tftpboot 0x80600000 u-boot-with-spl-nor.bin
jz_init......
Bus Mode Reg after write: 0x00020100
=====>phy 0:0x0-0x128 found
SPEED:2, DUPLEX:2
=====>found PHY 0
mac phy_id is: 0
############ check_phy_init_ds008 ##############
Link UP
=====>PHY Autonegotiation Complete!
Link is up in FULL DUPLEX mode
Link is with 100M Speed
GMAC init finish
Using Jz4775-9161 device
TFTP from server 192.168.0.130; our IP address is 192.168.0.11
Filename 'u-boot-with-spl-nor.bin'.
Load address: 0x80600000
Loading: #################
         5.6 MiB/s
done
Bytes transferred = 233268 (38f34 hex)
```

```
isvp_t41# tftpboot 0x80640000 uImage
jz_init......
Bus Mode Reg after write: 0x00020100
=====>phy 0:0x0-0x128 found
SPEED:2, DUPLEX:2
=====>found PHY 0
mac phy_id is: 0
############ check_phy_init_ds008 ##############
Link UP
=====>PHY Autonegotiation Complete!
Link is up in FULL DUPLEX mode
Link is with 100M Speed
GMAC init finish
Using Jz4775-9161 device
TFTP from server 192.168.0.130; our IP address is 192.168.0.11
Filename 'uImage'.
Load address: 0x80640000
Loading: ##################################################################
         ##################################################################
         ########################################
         5.7 MiB/s
done
Bytes transferred = 2489065 (25fae9 hex)
```

```
isvp_t41# tftpboot 0x808c0000 root-uclibc-toolchain720-r519.squashfs
jz_init......
Bus Mode Reg after write: 0x00020100
====>phy 0:0x0-0x128 found
SPEED:2, DUPLEX:2
====>found PHY 0
mac phy_id is: 0
########## check_phy_init_ds008 #############
Link UP
====>PHY Autonegotiation Complete!
Link is up in FULL DUPLEX mode
Link is with 100M Speed
GMAC init finish
Using Jz4775-9161 device
TFTP from server 192.168.0.130; our IP address is 192.168.0.11
Filename 'root-uclibc-toolchain720-r519.squashfs'.
Load address: 0x808c0000
Loading: ##################################################################
         ###################################
         5.7 MiB/s
done
Bytes transferred = 1499136 (16e000 hex)
isvp_t41#
```

4) Burn firmware to flash

Write firmware to the board's nor flash, and you can read it directly from the flash after restarting, instead of manually loading it to the memory every time. The burning commands for a 32MB-flash are as follows:

$ sf0 probe;sf0 erase 0x0 0x1000000;sf0 write 0x80600000 0x0 0x1000000

```
isvp_t41# sf0 probe;sf0 erase 0x0 0x1000000;sf0 write 0x80600000 0x0 0x1000000
SF: Detected MX25L25645G/35F, flash size: 32MB, manufacturer id: c2

--->probe spend 7 ms
SF: 16777216 bytes @ 0x0 Erased: OK
--->erase spend 71657 ms
SF: 16777216 bytes @ 0x0 Written: OK
--->write spend 20163 ms
isvp_t41#
```

Note: sf0 probe Before using sf read and sf write, you must call sf probe.

sf0 erase Erase the specified location and length of the flash.

sf0 write Write memory data to flash.

For details about how to burn image, please see "*T41 SDK Installation and Usage Guide.pdf*".

Test System

1. Default user

The user name of system is **root** default, and no password.

```
[    2.962184] ip_set: protocol 6
[    2.965683] ip_tables: (C) 2000-2006 Netfilter Core Team
[    2.971477] NET: Registered protocol family 17
[    2.975272] ingenic,sdhci 13070000.msc: card claims to support voltages below defined range
[    2.986867] soc_vpu probe success,version:1.0.0-03203fd46d
[    2.992845] mmc1: new high speed SDIO card at address 0001
[    3.003635] input: gpio-keys as /devices/platform/gpio_keys/input/input0
[    3.010936] random: nonblocking pool is initialized
[    3.016669] rtc-ingenic 10003000.rtc: setting system clock to 2020-03-01 12:35:51 UTC (1583066151)
[    3.031089] VFS: Mounted root (squashfs filesystem) readonly on device 31:2.
[    3.041148] devtmpfs: mounted
[    3.044478] Freeing unused kernel memory: 228K
mdev is ok......
net.core.wmem_max = 26214400
net.core.wmem_default = 26214400
[    4.829599] jffs2: notice: (536) jffs2_build_xattr_subsystem: complete building xattr subsystem, 0 of xdatum
[    4.856068] Bus Mode Reg after reset: 0x00020101, cnt=0

Ingenic-uc1_1 login: root
Mar  1 12:37:36 login[539]: root login on 'console'
[root@Ingenic-uc1_1:~]#
```

2. Ethernet

Ethernet is not configured for system startup. You can run the following command to configure Ethernet:

1) static IP

```
# ifconfig eth0 xxx.xxx.xxx.xxx up            // IP address
# route add default gw xxx.xxx.xxx.xxx    // gateway IP address
```

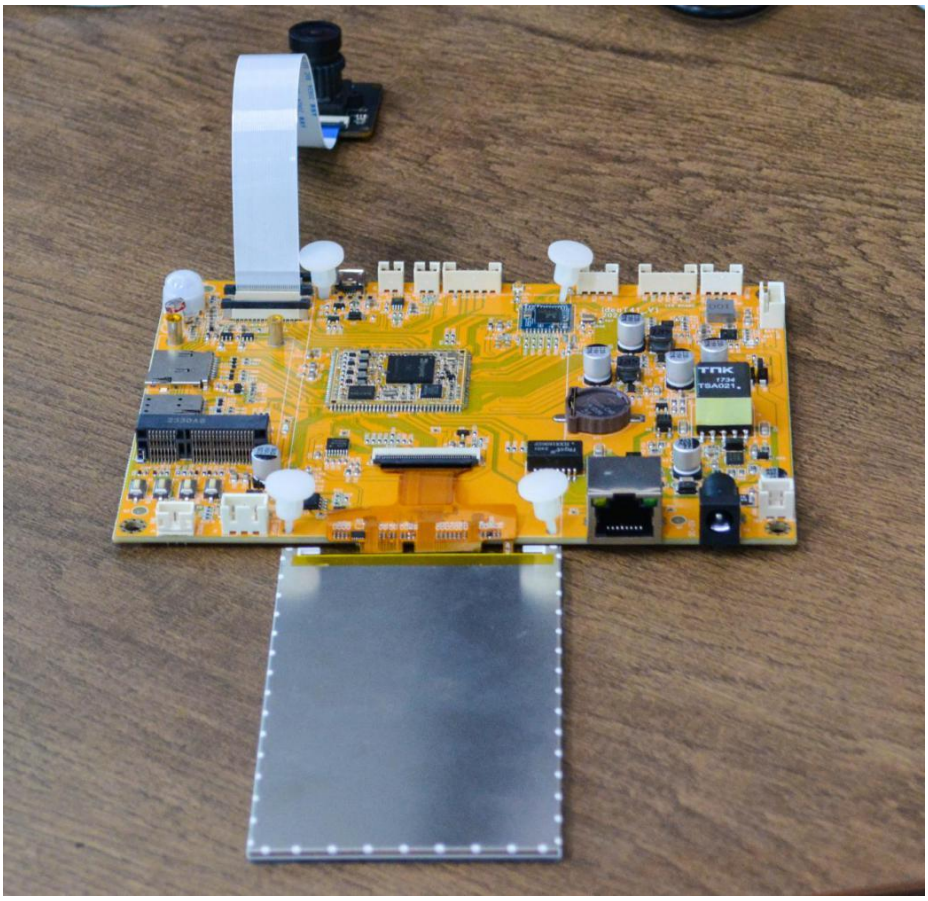
2) DHCP

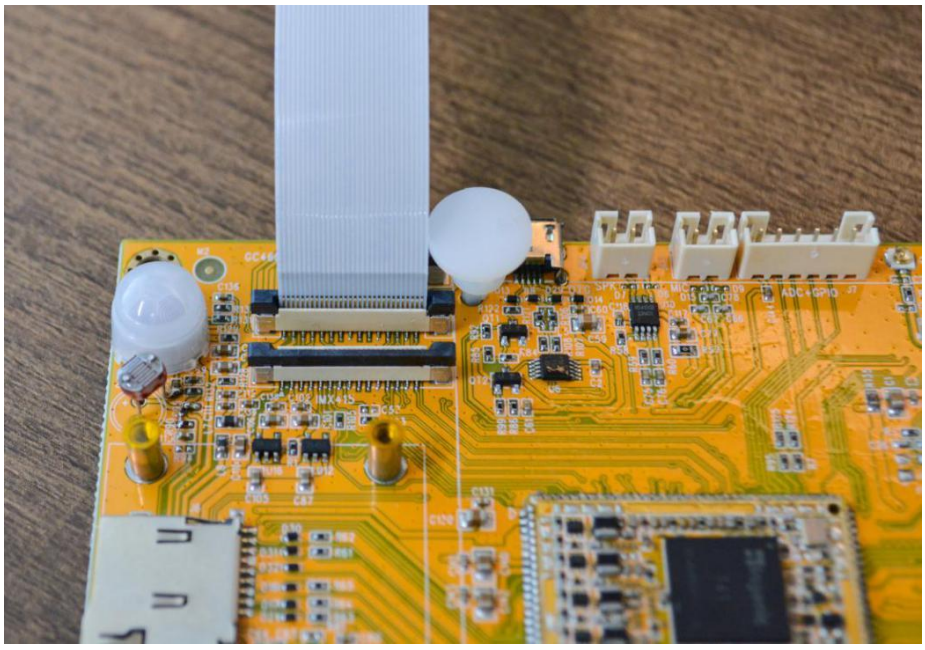
```
# ifconfig eth0 up
# udhcpc eth0
# ping www.boardcon.com
```

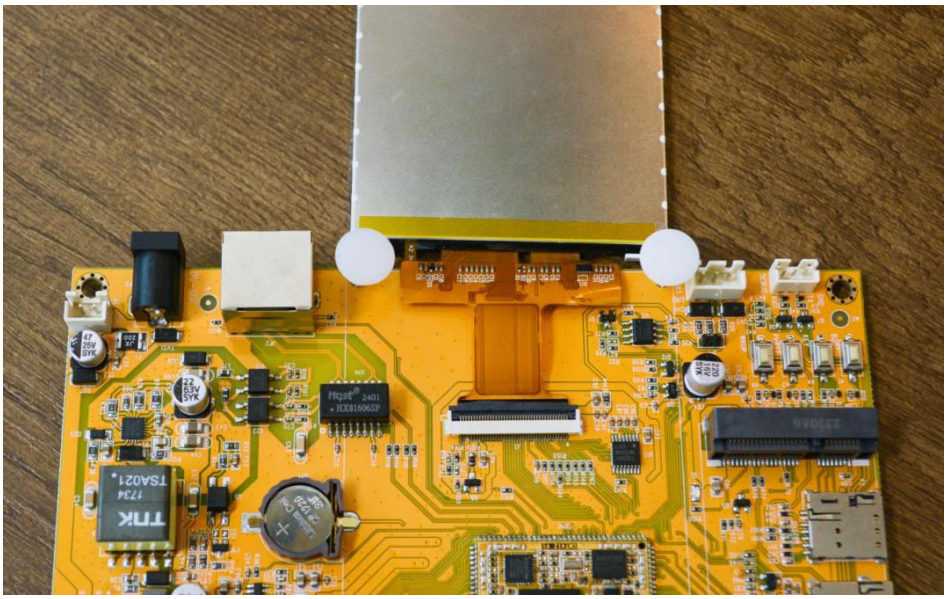
```
[root@Ingenic-uc1_1:~]#
[root@Ingenic-uc1_1:~]# ifconfig eth0 up
[   46.297853] Bus Mode Reg after reset: 0x00020101, cnt=0
[root@Ingenic-uc1_1:~]# [   48.775225] dwc-mac 134b0000.mac eth0: Link is Up - 100Mbps/Full - flow control rx/tx

[root@Ingenic-uc1_1:~]# udhcpc eth0
udhcpc (v1.22.1) started
Sending discover...
Sending select for 192.168.0.187...
Lease of 192.168.0.187 obtained, lease time 86400
deleting routers
adding dns 202.96.134.33
adding dns 202.96.128.86
[root@Ingenic-uc1_1:~]# ping www.boardcon.com
PING www.boardcon.com (67.222.54.196): 56 data bytes
64 bytes from 67.222.54.196: seq=0 ttl=48 time=180.420 ms
64 bytes from 67.222.54.196: seq=1 ttl=48 time=180.166 ms
64 bytes from 67.222.54.196: seq=2 ttl=48 time=180.634 ms
64 bytes from 67.222.54.196: seq=3 ttl=48 time=180.110 ms
```

3. LCD/camera

The LCD and Camera are connected as follows:







When the system boot, the LCD display a green screen by default. Execute the following command to display a preview of the Camera on the LCD:

```
# insmod /lib/modules/tx-isp-t41.ko clk_name="mpll" isp_clk=300000000
# insmod /lib/modules/sensor_sc500ai_t41.ko
# insmod /lib/modules/avpu.ko clk_name="mpll" avpu_clk=650000000
# sample-LCD      //The current streaming video is displayed on the LCD screen
# sample-Encoder-video    //Generate H264 / h265 encoding file in /tmp, it can be played by PotPlayer in windows
```

Note: The camera default use sc500ai connect to camera0. The camera0, camera1, camera2 can not be used at the same time, they are all connected to the same MIPI channel.

4. PIR/KEY

```
# sample-keyevent
```

```
[root@Ingenic-uc1_1:~]# sample-keyevent
Press Enter to exit.
Start read key event.
Keycode: 102, event: Pressed
Keycode: 102, event: Pressed
Keycode: 102, event: Released
Keycode: 183, event: Pressed
Keycode: 183, event: Pressed
Keycode: 158, event: Pressed
Keycode: 158, event: Pressed
Keycode: 158, event: Released
Keycode: 183, event: Released
Keycode: 116, event: Pressed
Keycode: 116, event: Pressed
Keycode: 116, event: Released
Keycode: 183, event: Pressed
Keycode: 183, event: Pressed
Keycode: 183, event: Pressed
```

Note：Treating the PIR sensor as a key, someone near the key is pressed.

158: KEY_BACK (KEY)

183: PIR

102: KEY_HOME （BOOT）

116: KEY_POWER （WKUP）

5. 4G (EC20)

Insert the 4G module. After powering on the board, the module needs to be reset first.

Note: the USB interface used by the 4G module is shared with the micro USB. When using 4G the micro USB needs to be disconnected.

```
# echo 6 > /sys/class/gpio/export
# echo out > /sys/class/gpio/gpio6/direction
# echo 1 > /sys/class/gpio/gpio6/value
# ifconfig eth0 down
```

Wait until the ttyUSB* devices appears, then execute the following command:

```
# pppd call quectel-ppp &
# ping -I ppp0 www.boardcon.com
```

```
[root@Ingenic-uc1_1:~]#  echo 6 > /sys/class/gpio/export
[root@Ingenic-uc1_1:~]#  echo out > /sys/class/gpio/gpio6/direction
[root@Ingenic-uc1_1:~]#  echo 1 > /sys/class/gpio/gpio6/value
[root@Ingenic-uc1_1:~]#
[root@Ingenic-uc1_1:~]# ifconfig eth0 down
[root@Ingenic-uc1_1:~]# [   23.465159] usb 1-1: new high-speed USB device number 2 using dwc2
[   23.676947] option 1-1:1.0: GSM modem (1-port) converter detected
[   23.684301] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB0
[   23.692491] option 1-1:1.1: GSM modem (1-port) converter detected
[   23.700071] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB1
[   23.710529] option 1-1:1.2: GSM modem (1-port) converter detected
[   23.718140] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB2
[   23.728416] option 1-1:1.3: GSM modem (1-port) converter detected
[   23.735805] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB3

[root@Ingenic-uc1_1:~]# pppd call quectel-ppp &
[root@Ingenic-uc1_1:~]# pppd options in effect:
debug           # (from /etc/ppp/peers/quectel-ppp)
nodetach                # (from /etc/ppp/peers/quectel-ppp)
dump            # (from /etc/ppp/peers/quectel-ppp)
noauth          # (from /etc/ppp/peers/quectel-ppp)
user test               # (from /etc/ppp/peers/quectel-ppp)
```

```
local  IP address 10.67.75.103
remote IP address 10.64.64.64
primary   DNS address 120.80.80.80
secondary DNS address 221.5.88.88
Script /etc/ppp/ip-up started (pid 580)
Script /etc/ppp/ip-up finished (pid 580), status = 0x0

[root@Ingenic-uc1_1:~]# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  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:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.67.75.103  P-t-P:10.64.64.64  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:52 (52.0 B)  TX bytes:58 (58.0 B)

[root@Ingenic-uc1_1:~]# ping www.boardcon.com
PING www.boardcon.com (67.222.54.196): 56 data bytes
64 bytes from 67.222.54.196: seq=0 ttl=47 time=320.291 ms
64 bytes from 67.222.54.196: seq=1 ttl=47 time=282.184 ms
64 bytes from 67.222.54.196: seq=2 ttl=47 time=560.013 ms
64 bytes from 67.222.54.196: seq=3 ttl=47 time=518.892 ms
```

6. SD card

The system does not automatically mount the SD card, it needs to be mounted manually.

mount /dev/mmcblk0p1 /mnt/

ls /mnt/

```
[root@Ingenic-uc1_1:~]# [  498.130436] mmc0: card inserted
[  498.133806] mmc0: card inserted
[  498.179840] mmc0: Got command interrupt 0x00000001 even though no command operation was in progress.
[  498.494629] mmc0: new high speed SDHC card at address b368
[  498.511334] mmcblk0: mmc0:b368 NCard 29.1 GiB
[  498.518511]  mmcblk0: p1

[root@Ingenic-uc1_1:~]# mount /dev/mmcblk0p1 /mnt/
[  512.466773] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
[root@Ingenic-uc1_1:~]# ls /mnt/
1080P30-Frozen-Clip-Let It Go.mp4
2.wav
3588
4K30-H264-????.mp4
4K60-H265-GUGUDAN-Wonderland.2160p.UHDTV.H265.ts
4K60-H265-I.O.I-Dream.Girl.2160p.UHDTV.H265.ts
```

7. Light sensor

Cover or illuminate the light sensor to observe the voltage changes.

sample-adc

```
[root@Ingenic-uc1_1:mnt]# ./sample-adc
### adc value is : 8mV ###
result time = 17086 usec !!
### adc value is : 23mV ###
result time = 19847 usec !!
### adc value is : 1199mV ###
result time = 19901 usec !!
### adc value is : 1799mV ###
result time = 19895 usec !!
### adc value is : 1799mV ###
result time = 19905 usec !!
### adc value is : 1799mV ###
result time = 19894 usec !!
### adc value is : 12mV ###
result time = 19905 usec !!
### adc value is : 10mV ###
result time = 19901 usec !!
### adc value is : 22mV ###
result time = 19901 usec !!
```

8. Audio

IdeaT41 supports a MIC and a speaker. Test as following:

```
# insmod /lib/modules/audio.ko
# cd /tmp       // The /tmp directory is readable and writable, as it needs to generate files for recording.
# sample-Ai     // Press Enter to continue. The ai_record.pcm recording file will be generated in /tmp
# cp ai_record.pcm ao_play.pcm
# sample-Ao     // Ensure the current directory has the ao_play.pcm file. Press Enter twice to play
```

```
[root@Ingenic-uc1_1:~]# insmod /lib/modules/audio.ko
[ 1199.455195] @@@@@ inner codec power up@@@@@@
[ 1199.855197] @@@@ audio driver ok(version H20230311a) @@@@@
[root@Ingenic-uc1_1:~]# cd /tmp
[root@Ingenic-uc1_1:tmp]# sample-Ai
[INFO] Test 1: Start audio record test.
[INFO]        : Can create the ai_record.pcm file.
[INFO]        : Please input any key to continue.

Audio In GetVol    vol : 60
fopen error file /etc/webrtc_profile.ini: No such file or directory
fopen error file /etc/webrtc_profile.ini: No such file or directory
fopen error file /etc/webrtc_profile.ini: No such file or directory

[root@Ingenic-uc1_1:tmp]# ls
ai_record.pcm  resolv.conf    setir
[root@Ingenic-uc1_1:tmp]# cp ai_record.pcm ao_play.pcm
[root@Ingenic-uc1_1:tmp]# sample-Ao
[INFO] Test Ao basic:
[INFO]  : Ao play file: ./ao_play.pcm is exist.
[INFO]  : Please input any key to continue:

fopen error file /etc/webrtc_profile.ini: No such file or directory
fopen error file /etc/webrtc_profile.ini: No such file or directory
[INFO] Test : Audio Play Pause test.
[INFO]          : Please input any key to continue:

[root@Ingenic-uc1_1:tmp]#
```

9. wifi

devmem 0x1000006C 32 0x4011800B

insmod /lib/modules/8723ds.ko

vi /system/wpa_supplicant.conf

Add the following configuration information:

```
ctrl_interface=/system/var/run/wpa_supplicant
ap_scan=1
update_config=1
network={
        ssid="xxxx"           // “xxxx” is the WiFi SSID to connect to
        psk="xxxxxxx"       // "xxxxxxx" is the password
        key_mgmt=WPA-PSK
      }
```

ifconfig wlan0 down

killall wpa_supplicant

killall udhcpc

ifconfig wlan0 up

wpa_supplicant -Dnl80211 -iwlan0 -c /system/wpa_supplicant.conf &

udhcpc -i wlan0

ping -I wlan0 www.boardcon.com

```
[root@Ingenic-uc1_1:~]# udhcpc -i wlan0
udhcpc (v1.22.1) started
Sending discover...
Sending select for 192.168.0.196...
Lease of 192.168.0.196 obtained, lease time 86400
deleting routers
adding dns 202.96.134.33
adding dns 202.96.128.86
[root@Ingenic-uc1_1:~]# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:402 (402.0 B)  TX bytes:402 (402.0 B)

wlan0     Link encap:Ethernet  HWaddr C4:3C:B0:43:36:F8
          inet addr:192.168.0.196  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:115 errors:0 dropped:41 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14183 (13.8 KiB)  TX bytes:1028 (1.0 KiB)

[root@Ingenic-uc1_1:~]# ping www.boardcon.com
PING www.boardcon.com (67.222.54.196): 56 data bytes
64 bytes from 67.222.54.196: seq=0 ttl=48 time=320.408 ms
64 bytes from 67.222.54.196: seq=1 ttl=48 time=240.096 ms
64 bytes from 67.222.54.196: seq=2 ttl=48 time=349.873 ms
```

If you are already connected to WIFI and need to change another ssid and password, you can execute the following command:

wpa_cli -i wlan0 -p /system/var/run/wpa_supplicant scan

wpa_cli -i wlan0 -p /system/var/run/wpa_supplicant scan_result

wpa_cli -i wlan0 -p /system/var/run/wpa_supplicant set_network 0 ssid '"xxxx"'

```
# wpa_cli -i wlan0 -p /system/var/run/wpa_supplicant set_network 0 psk 'xxxxxx"'
# wpa_cli -i wlan0 -p /system/var/run/wpa_supplicant enable_network 0
# udhcpc -i wlan0
```

10. Bluetooth

Execute the following commands to test the bluetooth:

```
# insmod /lib/modules/hci_uart.ko
# echo 81 > /sys/class/gpio/export    // Reset the bluetooth model
# echo out > /sys/class/gpio/gpio81/direction
# echo 0 > /sys/class/gpio/gpio81/value
# echo 1 > /sys/class/gpio/gpio81/value
# rtk_hciattach -n -s 115200 ttyS0 rtk_h5 &
# hciconfig hci0 up
# hcitool scan
```

```
[   34.227110] Bluetooth: __hci_uart_flush: hdev 80c33000 tty 81b84600
[   34.233743] rtk_btcoex: Close BTCOEX
[   34.237544] rtk_btcoex: -x

[root@Ingenic-uc1_1:~]# hciconfig
hci0:   Type: Primary  Bus: UART
        BD Address: C4:3C:B0:43:36:F9  ACL MTU: 1021:8  SCO MTU: 255:12
        DOWN
        RX bytes:1067 acl:0 sco:0 events:30 errors:0
        TX bytes:859 acl:0 sco:0 commands:30 errors:0

[root@Ingenic-uc1_1:~]# hciconfig hci0 up
[   47.069941] rtk_btcoex: Open BTCOEX
[   47.315156] Bluetooth: hu 80d46a80 retransmitting 1 pkts
[   47.322661] rtk_btcoex: BTCOEX hci_rev 0xaaa8
[   47.327369] rtk_btcoex: BTCOEX lmp_subver 0x2df5
[root@Ingenic-uc1_1:~]#  hcitool scan
Scanning ...
[   66.334715] rtk_btcoex: hci (periodic)inq start
[   66.585148] Bluetooth: hu 80d46a80 retransmitting 1 pkts
[   76.834924] rtk_btcoex: inquiry complete
        40:EC:99:71:77:20       n/a
        30:4F:00:36:98:04       OPPO K9 Pro 5G
        8C:67:94:3D:EF:A9       vivo X60t Pro+
        88:68:4B:74:39:44       OPPO K10 5G
[root@Ingenic-uc1_1:~]#
```

11. Test UART2

Short circuit RX and TX pins of UART2, then execute the following commands:

```
# com /dev/ttyS2 115200 8 0 1
```

```
[root@Ingenic-uc1_1:~]# com /dev/ttyS2 115200 8 0 1
port = /dev/ttyS2
baudrate = 115200
cs = 8
parity = 0
stopb = 1
123456789
RECV: 123456789
qwedsazcxxxxxxxxxx
RECV: qwedsazcxxxxxxxxxx
```

12. RTC

Alarm clock

sample-alarm

Set time

date -s "2024-05-20 12:00:00"

hwclock -w

hwclock

```
[root@Ingenic-uc1_1:~]# sample-alarm
Now: 2024.5.20 12:01:26
ALarm Time: 2024.5.20 12:01:36
Wait Alarm event
Alarm timeout...
Now: 2024.5.20 12:01:37
[root@Ingenic-uc1_1:~]# date -s "2024-05-20 12:00:00"
Mon May 20 12:00:00 UTC 2024
[root@Ingenic-uc1_1:~]# hwclock -w
[root@Ingenic-uc1_1:~]# hwclock
Mon May 20 12:00:12 2024  0.000000 seconds
[root@Ingenic-uc1_1:~]# hwclock
Mon May 20 12:00:17 2024  0.000000 seconds
[root@Ingenic-uc1_1:~]# hwclock
Mon May 20 12:00:18 2024  0.000000 seconds
[root@Ingenic-uc1_1:~]# hwclock
Mon May 20 12:00:19 2024  0.000000 seconds
```

13. Browsing video in PC

13.1 setup the board

Make sure the development board and the PC are connected to the same router are in the same network segment. Reboot the board, set up the network, Use Ethernet as an example:

ifconfig eth0 up

udhcpc -i eth0

And then execute the following commands:

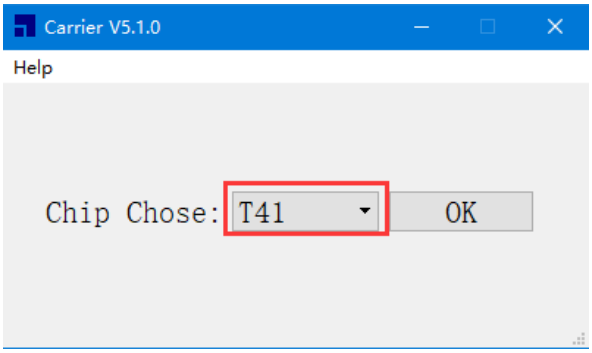
cd /system/init/ // Make sure to enter the directory, otherwise an error will occur.

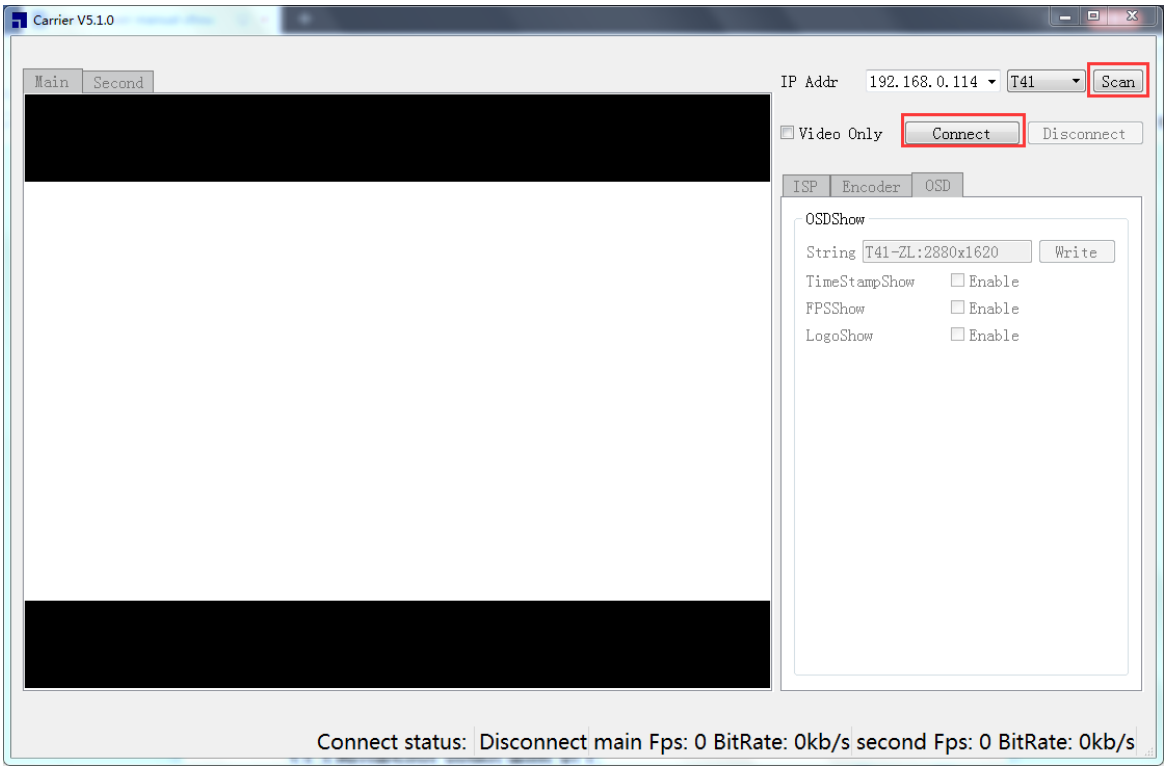
./start.sh start_param

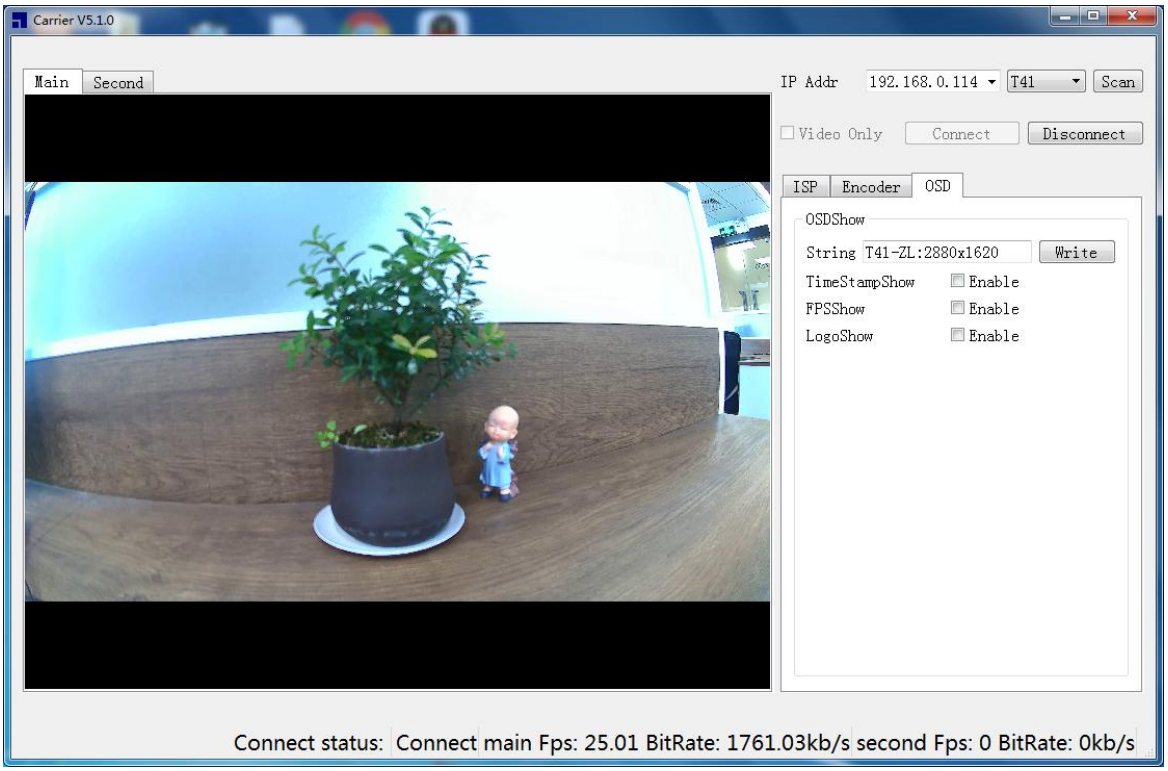
```
Function usage:
        showfps: Show FPS and bitrate infomation
        osd: Switch on/off timestamp OSD
        snap: Snapshot a JPEG picture
        display: Display on LCD
Initializing...
[   63.105336] -----sc500ai_detect: 870 ret = 0, v = 0xce
[   63.110932] -----sc500ai_detect: 878 ret = 0, v = 0x1f
[   63.116400] sc500ai chip found @ 0x30 (i2c0)
 sensor drv version H20230720a Create framechan0 OK!
[   63.126728]  Create framechan1 OK!
[   63.130772]  Create framechan2 OK!
[   63.181872] Ivdc init! direct_mode is 1
---- FPGA board is ready ----
  Board UID : 30AB6E51
  Board HW ID : 71000260
  Board rev.  : 62D42E77
  Board date  : 20211214
-----------------------------
chn : 0, direct_mode: 1
chn : 2, direct_mode: 1
> ...done initializing
Play this video stream using the URL:
        rtsp://192.168.0.114:8554/main
Play this video stream using the URL:
        rtsp://192.168.0.114:8554/second
```

13.2 Browsing video with Carrier

Double click Carrier.exe, The interface shown as following, select **T41** and click **OK** to enter the main page. After entering the main page, click the **Scan** button to search for devices in the same network segment, if an IP address is displayed in "**IP Addr**", the board sis detected, Click **Connect** to connect the device. After successful connection, the real-time screen of the device will be displayed on the left.



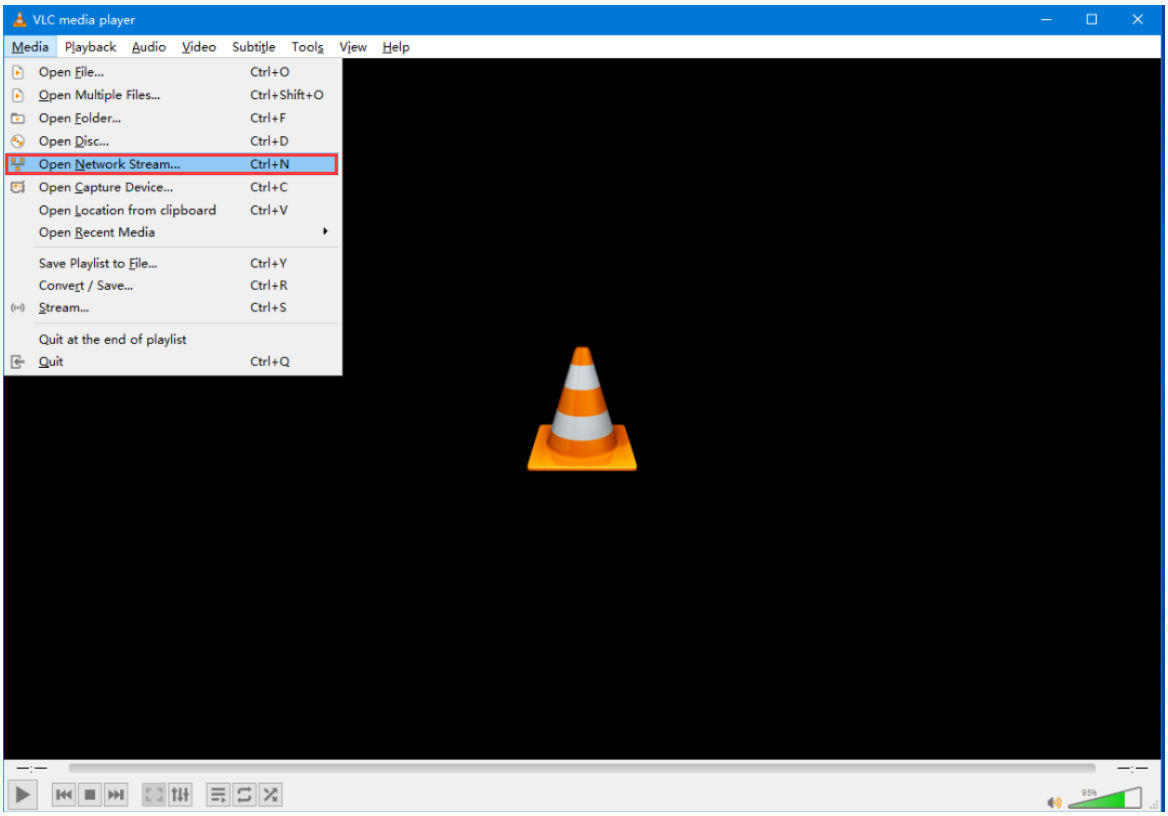


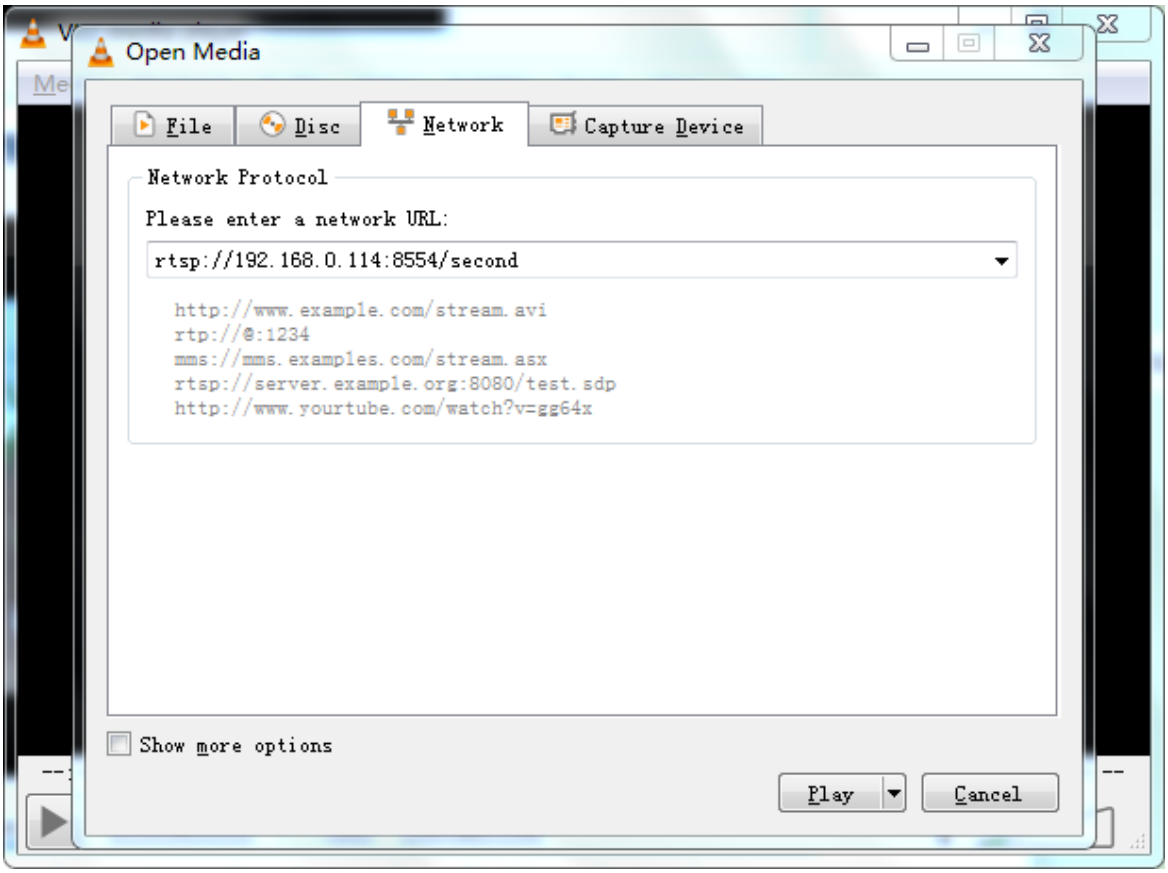


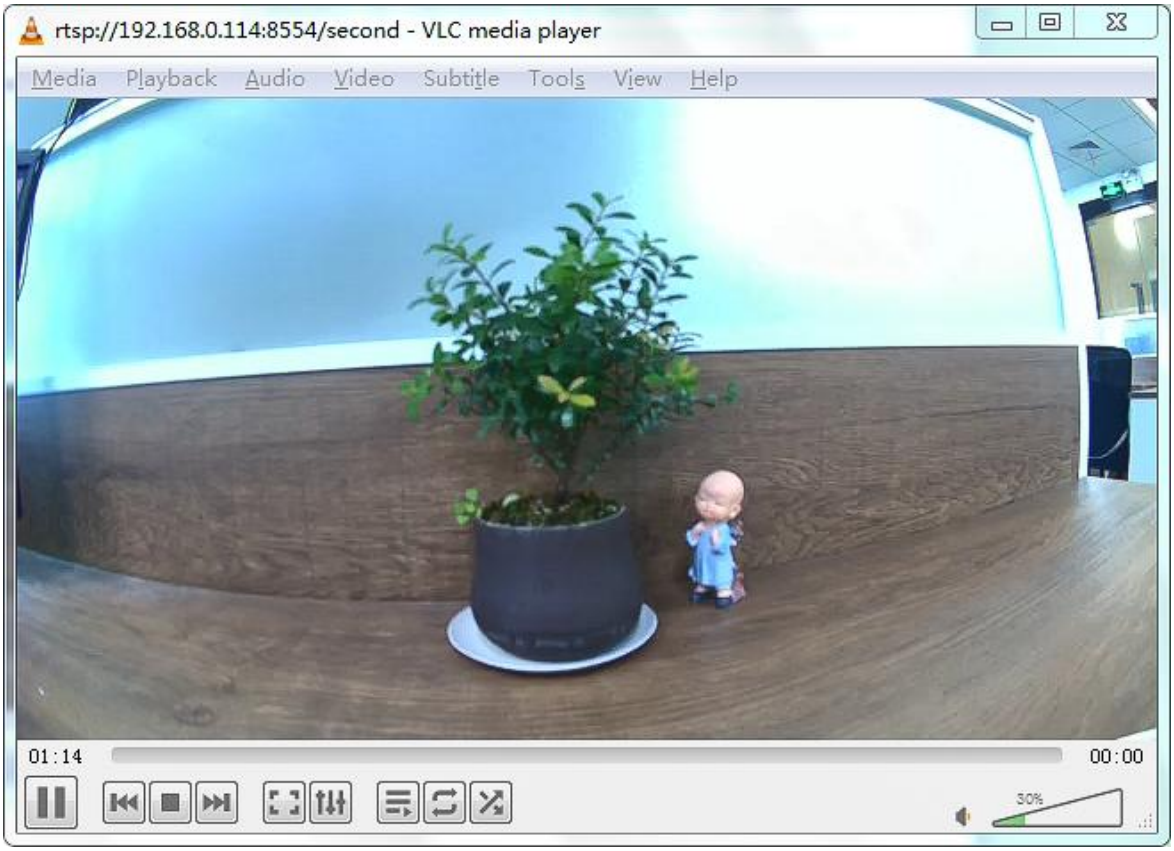
For more information on using Carrier, please refer to "*Ingenic Carrier Tool Usage Guide.pdf*".

13.3 Browsing video with VLC

Media -> Open Network Streaming -> Network -> input URL -> Play


VLC media player
Media Playback Audio Video Subtitle Tools View Help
Open File... Ctrl+O
Open Multiple Files... Ctrl+Shift+O
Open Folder... Ctrl+F
Open Disc... Ctrl+D
Open Network Stream... Ctrl+N
Open Capture Device... Ctrl+C
Open Location from clipboard Ctrl+V
Open Recent Media
Save Playlist to File... Ctrl+Y
Convert / Save... Ctrl+R
Stream... Ctrl+S
Quit at the end of playlist
Quit Ctrl+Q


Open Media
File
Disc
Network
Capture Device
Network Protocol
Please enter a network URL:
rtsp://192.168.0.114:8554/second
http://www.example.com/stream.avi
rtp://@:1234
mms://mms.examples.com/stream.asx
rtsp://server.example.org:8080/test.sdp
http://www.yourtube.com/watch?v=gg64x
Show more options
Play
Cancel


rtsp://192.168.0.114:8554/second - VLC media player
Media Playback Audio Video Subtitle Tools View Help
01:14
00:00
30%