SigmaStar Camera
The Enviroment Setup Guideline

V1.1

REVISION HISTORY

Revision No.	Description	Date
1.0	• Initial release	19/12/2018
1.1	• Add spi nand note.	22/01/2019

TABLE OF CONTENTS

1. THE EVB CONNECTIONS

1.1. SSC009A EVB:

Power: DC 12V,
Debug UART: TTL level，baud rate 115200

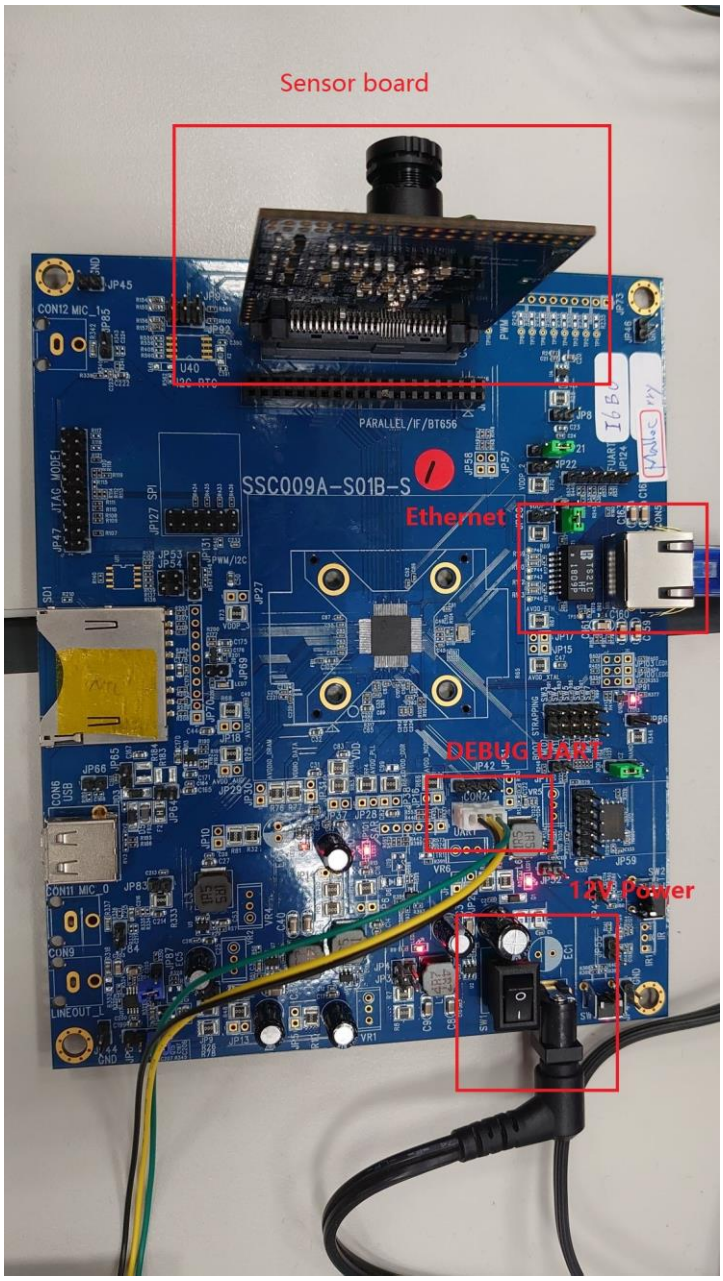

Figure 1: Pic 1-1

Jumper set for booting from SPI-NOR：

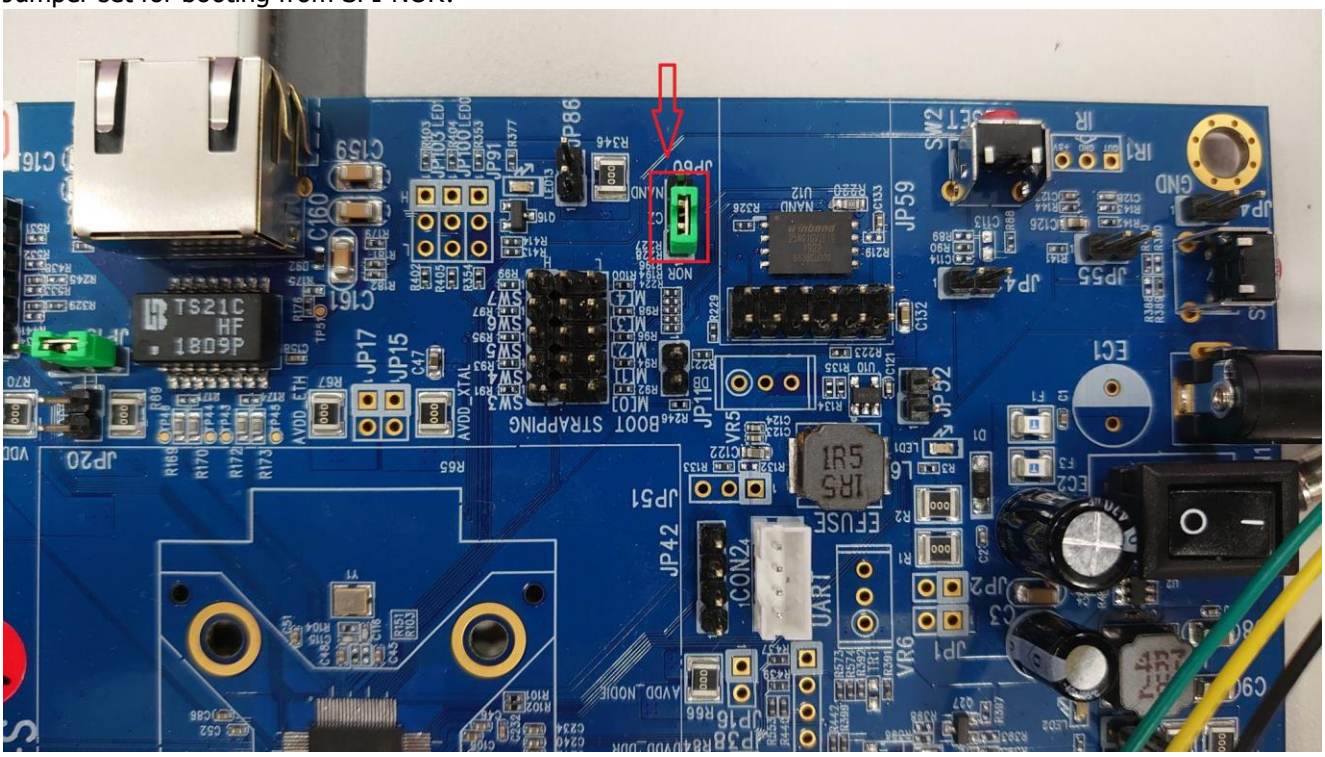

Figure 2: pic 1-2

Jumper set for booting from SPI-NAND：

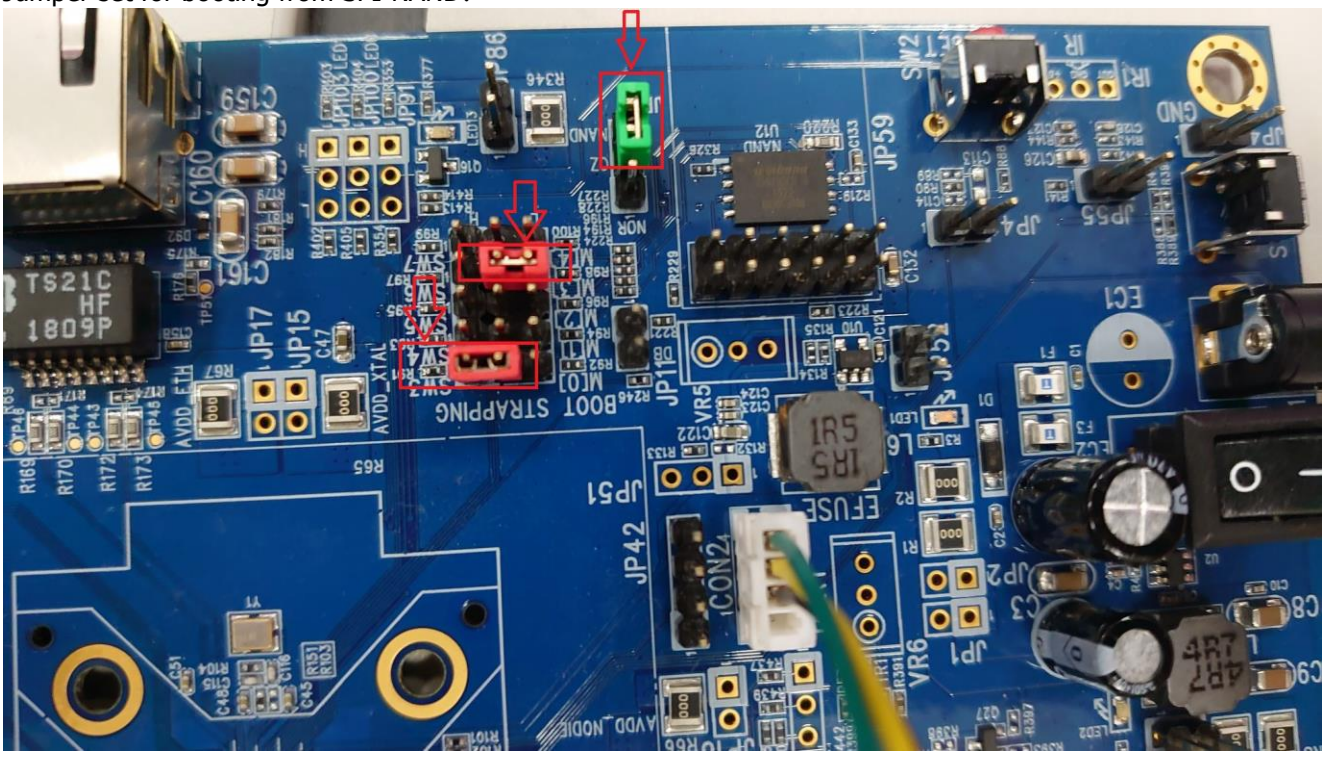

Figure 3: pic 1-3

1.2. SSC009B：

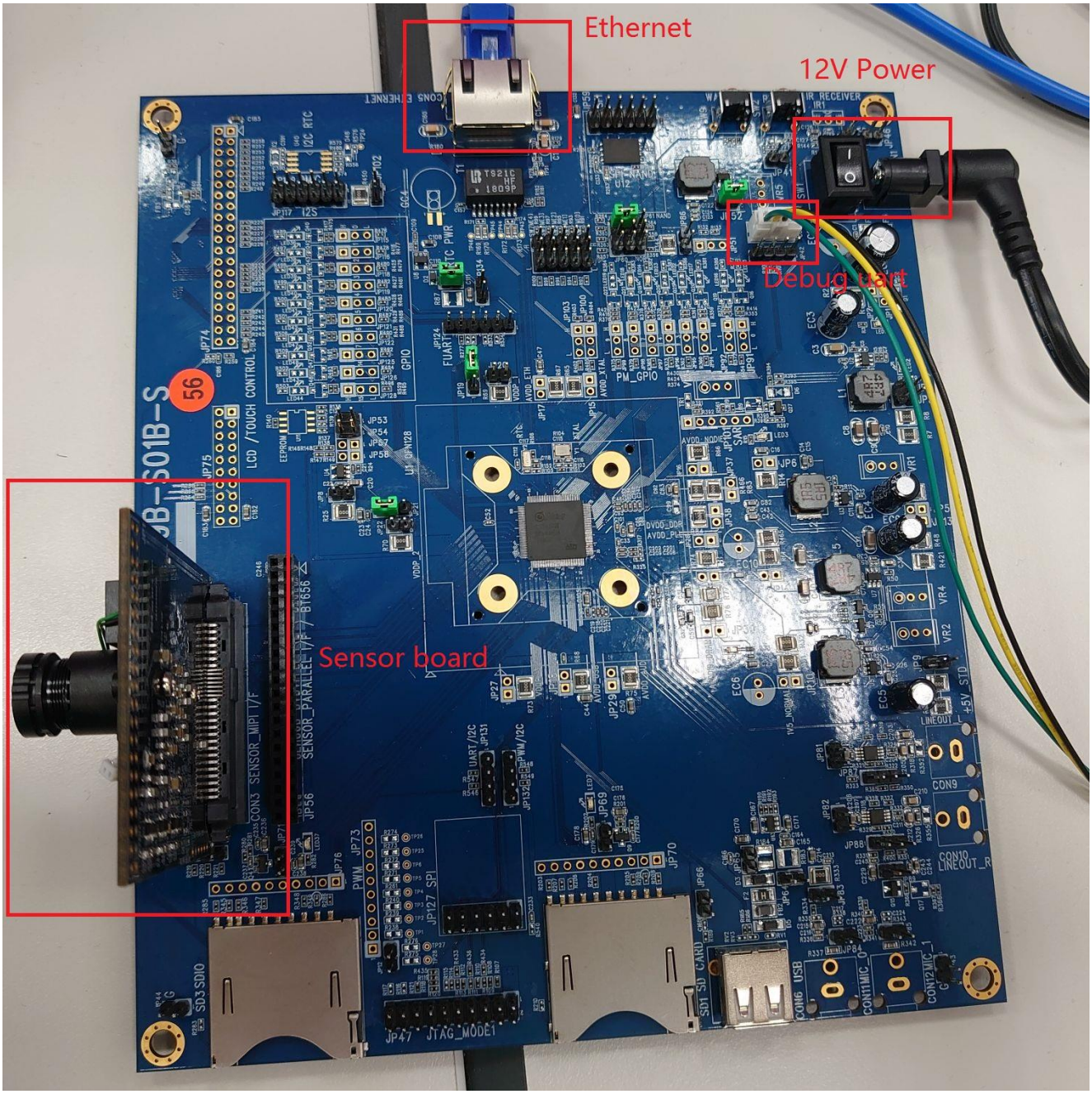

Figure 4: pic 1-4

Jumper set for booting from SPI-NOR：

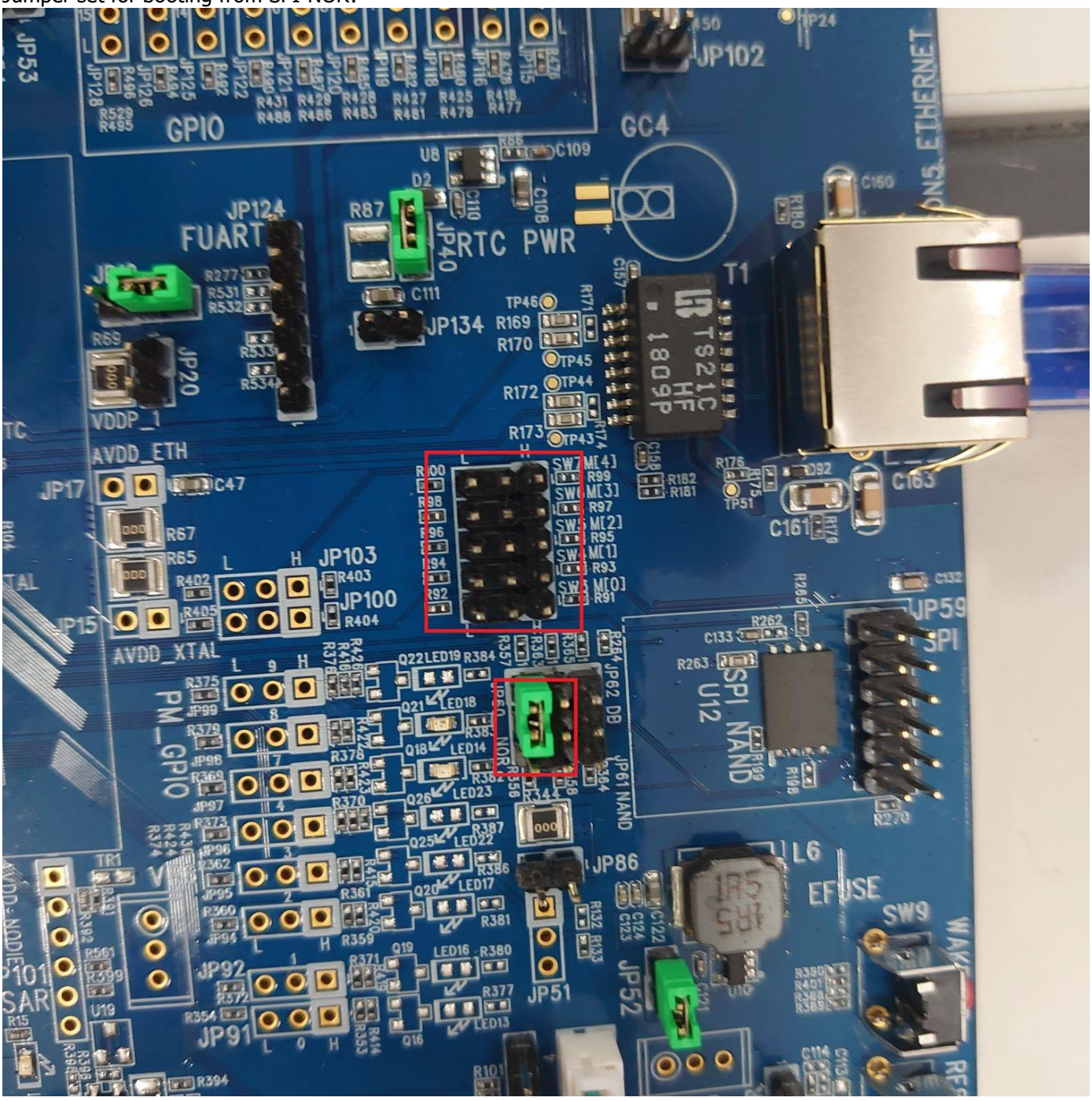

Figure 4: pic 1-5

Jumper set for booting from SPI-NAND

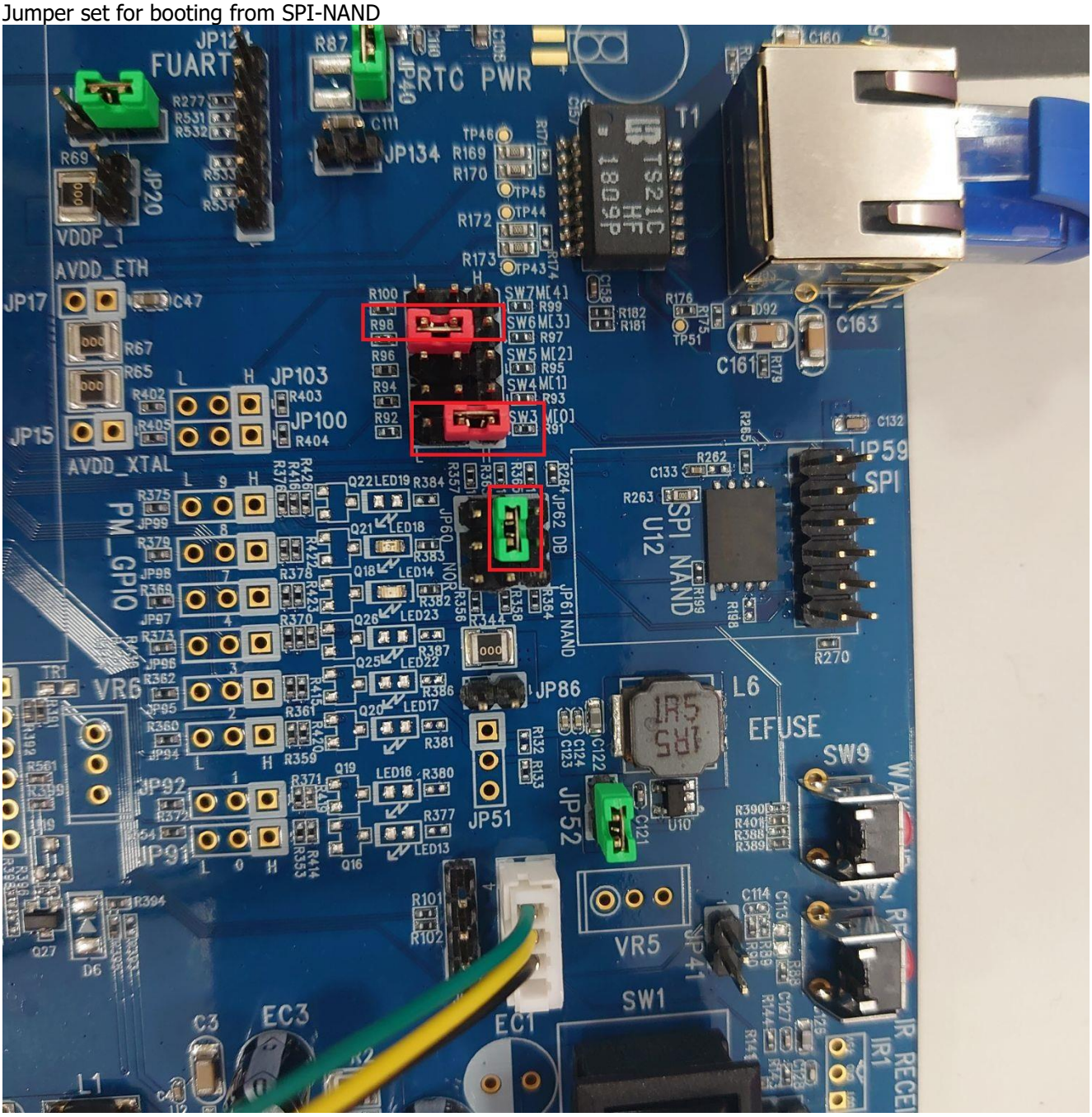

Figure 5: pic 1-6

2. THE COMPILING SETUP

Use cross-compiling tool to develop/debug software. Make the connection between your host and target device by UART and Ethernet, as showed below:

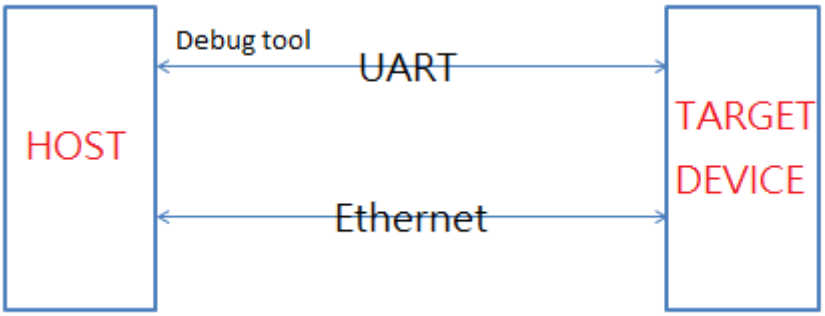

Figure 6: pic 2-1

2.1. Install Linux Server

Recommend to use Ubuntu 16.04。

2.2. Install the cross compiler toolchain

Recommend to use

- gcc-sigmastar-9.1.0-2019.11-x86_64_arm-linux-gnueabihf.tar.xz for compiling glibc version of sources;
- arm-buildroot-linux-uclibcgnueabihf-4.9.4-uclibc-1.0.31.tar.xz for compiling uClibc version of sources.

These toolchain(s) were provided in the SDK. If you didn't have it, please contact our FAE.

3. COMPILING

SSC335 Series platform supports booting from SPI NOR and SPI NAND, so you need to choose the corresponding configure files in the SDK for compiling to generate the proper images. Then, you can utilize script file to pack the images for download (to the target device)

3.1.1 Compiling for Uboot

- **SPI-NOR package**

```
#declare -x ARCH="arm"
# declare -x CROSS_COMPILE="$compiler"
        //exp: uclibc "arm-buildroot-linux-uclibcgnueabihf"
# make infinity6b0_defconfig;
#make clean;
#make
```

- **SPI-NAND package**

```
#declare -x ARCH="arm"
# declare -x CROSS_COMPILE="$compiler"
        //exp: uclibc "arm-buildroot-linux-uclibcgnueabihf"
# make infinity6b0_spinand_defconfig;
#make clean;
#make
```

- **Get image**

```
# cp u-boot.xz.img.bin ${ your_release_path }   // for spi-nor
# cp u-boot_spinand.xz.img.bin ${ your_release_path }      // for spi-nand
```

3.1.2 Compiling for Kernel

- **SPI-NOR Kernel (ASIC)**

CHIP	Glibc compiler	Uclibc compiler	Kernel make config	DTS
QFN88 64MB: SSC335 QFN88 128MB: SSC335D	Linaro Glibc8.2.1 arm-linux-gnueabihf-	Buildroot Uclibc 4.9.4 arm-buildroot-linux-uclibcgnueabihf-	infinity6b0_ssc009a_s01a_defconfig	infinity6b0-ssc009a-s01a.dts
QFN128 128MB: SSC337DE	Linaro Glibc8.2.1 arm-linux-gnueabihf-	Buildroot Uclibc 4.9.4 arm-buildroot-linux-uclibcgnueabihf-	infinity6b0_ssc009b_s01a_defconfig	infinity6b0-ssc009b-s01a.dts

Table 1

- **SPI-NAND Kernel (ASIC)**

CHIP	Glibc compiler	Uclibc compiler	Kernel make config	DTS
QFN88 64MB: SSC335 QFN88 128MB: SSC335D	Linaro Glibc8.2.1 arm-linux-gnueabihf-	Buildroot Uclibc 4.9.4 arm-buildroot-linux-uclibcgnueabihf-	infinity6b0_ssc009a_s01a_spinand_defconfig	infinity6b0-ssc009a-s01a.dts
QFN128 128MB: SSC335DE	Linaro Glibc8.2.1 arm-linux-gnueabihf-	Buildroot Uclibc 4.9.4 arm-buildroot-linux-uclibcgnueabihf-	infinity6b0_ssc009b_s01a_spinand_defconfig	infinity6b0-ssc009b-s01a.dts

Table 2

Please refer to the table(s) above to choose the corresponding configuration files what you need for compiling:

```
# declare -x ARCH="arm"
# declare -x CROSS_COMPILE="$compiler"
      //exp: uclibc "arm-buildroot-linux-uclibcgnueabihf"
# make xxx_kernel_make_config
      //exp: make infinity6b0_ssc009b_s01a_spinand_defconfig
# make clean;
# make
```

- **Get image**

```
# cp arch/arm/boot/uImage.xz ${ your_release_path }
```

3.1.3 Compiling for SDK(ALKAID)

- **SPI-NOR flash package**

CHIP	Glibc	Uclibc
QFN88 64MB: SSC335	nor.glibc-squashfs.009a.64.qfn88	nor.uclibc-squashfs.009a.64.qfn88
QFN88 128MB: SSC335D	nor.glibc-squashfs.009a.128.qfn88	nor.uclibc-squashfs.009a.128.qfn88
QFN128 128MB: SSC337DE	nor.glibc-squashfs.009b.128.qfn128	nor.uclibc-squashfs.009b.128.qfn128

3Table 3

- **SPI-NAND flash package**

CHIP	Glibc	Uclibc
QFN88 64MB: SSC335	spinand.glibc-squashfs.009a.64.qfn88	spinand.uclibc-squashfs.009a.64.qfn88
QFN88 128MB: SSC335D	spinand.glibc-squashfs.009a.128.qfn88	spinand.uclibc-squashfs.009a.128.qfn88

QFN128 128MB: SSC337DE	spinand.glibc-squashfs.009b.128.qfn128	spinand.uclibc-squashfs.009b.128.qfn128

Table 4

Please refer to the table(s) above to choose the corresponding configuration files what you need for compiling:

```
# cd $/{Alkaid}/project
# . /setup_config.sh xxx_alkaid_build_config
//exp: . /setup_config.sh ./configs/ipc/i6/nor.glibc-squashfs.009a.128.qfn88
# make image
```

- **Get image**

```
# cd ${Alkaid}/project/image/output/images
```

4. DOWNLOAD THE IMAGE TO THE TARGET DEVICE

4.1.1 Download code by uboot

- **Run tftp (FTP server) on PC (Host)**

Step1. Use tftp with the image path：SDK\project\image\output\images\, and set your networking well on host.

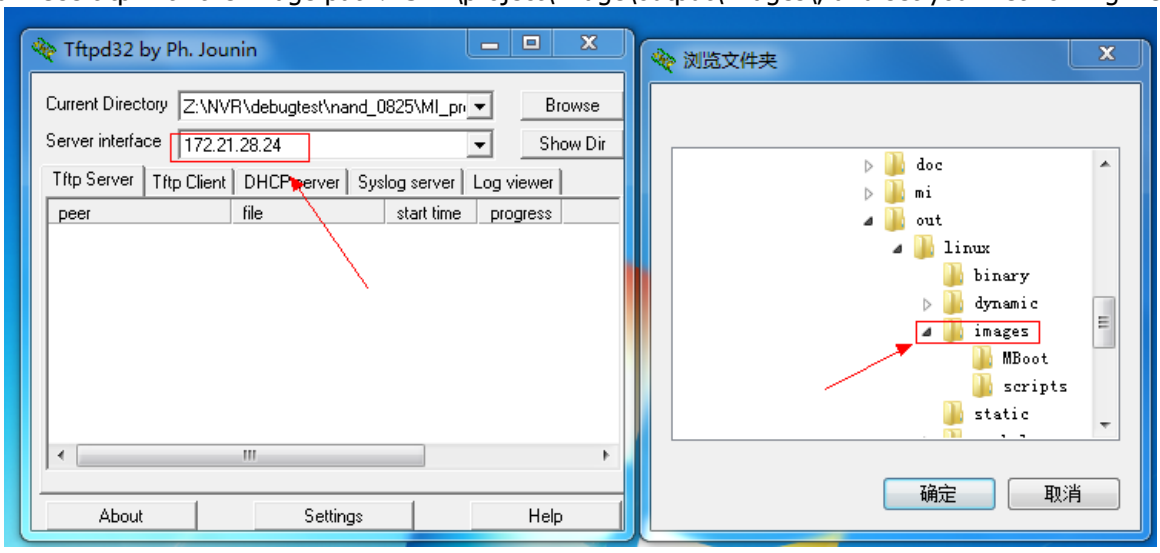

Figure 7: pic 4-1

Step 2. Make the connection between your host and target device correctly, like Figure 6: pic 2-1

- **Run tftp (FTP Client) on EVB (Target Device)**

Step 1. Power on your device and keep pressing "Enter" via UART console (e.g. HyperTerminal) to enter the bootloader command line.

■ At the first time of code download, please set up IP address:

```
# set -f gatewayip 192.168.1.1
# set -f ipaddr 192.168.1.127          //set FTP Client (EVB/Target Device) IP
# set -f netmask 255.255.255.0
# set -f serverip 192.168.1.100        //set FTP server (PC/Host) IP
# saveenv
```

Note：

1. For the download process working, please ensure the PC(Host) and EVB (target device) are in the same subnet of Ethernet.
2. Please use static IP address assignment to avoid IP changing during the development.

■ Run the command below at the UBOOT console to trigger the code downloading:

```
# estar (OR: estar auto_update.txt)
```

4.1.2 Download uboot to the target device by ISP Tool

Use this method when Uboot is not available on your EVB (target device).

4.1.2.1. SPI-NOR-Flash

- **Default Partition layout**

No	range	size
IPL	0x00000000, 0x00010000	64KB
IPL_CUST	0x00010000, 0x00020000	64KB
MXPT	0x00020000, 0x00030000	64KB
UBOOT	0x00030000, 0x0004F000	124KB
UBOOT_ENV	0x0004F000, 0x00050000	4KB
BOOT	0x00000000, 0x00050000	320KB
KERNEL	0x00050000, 0x00250000	2048KB
ROOTFS	0x00250000, 0x00650000	4096KB
MISERVICE	0x00650000, 0x00950000	3072KB
CUSTOMER	0x00950000, 0x01000000	6848KB

Table 5

- **Burning code by ISP tool**

	offset	Binary 放置目录
IPL.bin	0x0000	${ALKAID}\project\image\output\images\IPL.bin
IPL_CUST.bin	0x10000	${ALKAID}\project\image\output\images\IPL_CUST.bin
MXP_SF.bin	0x20000	${ALKAID}\project\image\output\images\MXP_SF.bin
u-boot.xz.img.bin	0x30000	${ALKAID}\project\image\output\images\u-boot.xz.img.bin

Table 6

- **Burning Steps**
  - Step1. Run ISP tool and close the UART terminal (or the 'Connect' could be blocked by UART terminal app.)
  - Step2. Choose SPI tab, and click 'More' for the type of 'SPI'.

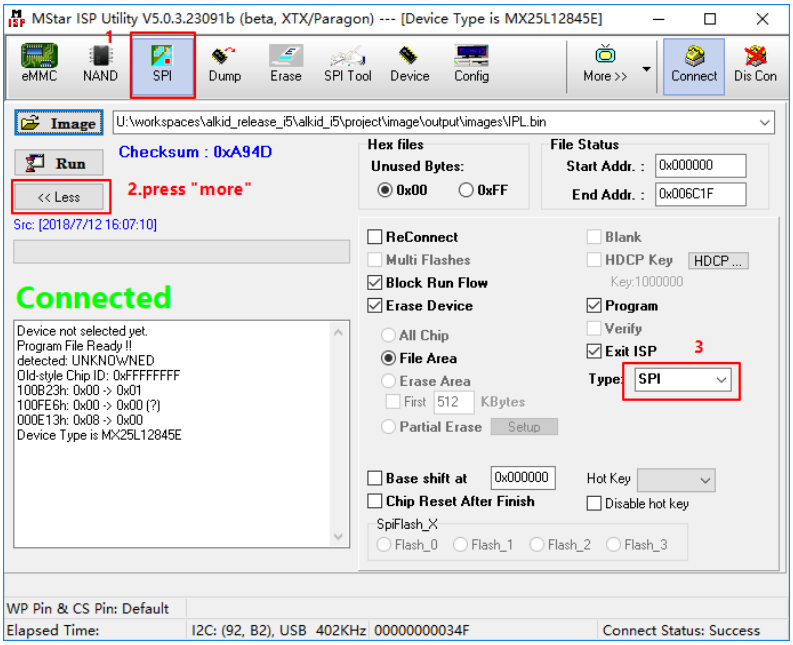

Figure 8: pic 4-2

- Step3. Load the image file and click 'Connect'.

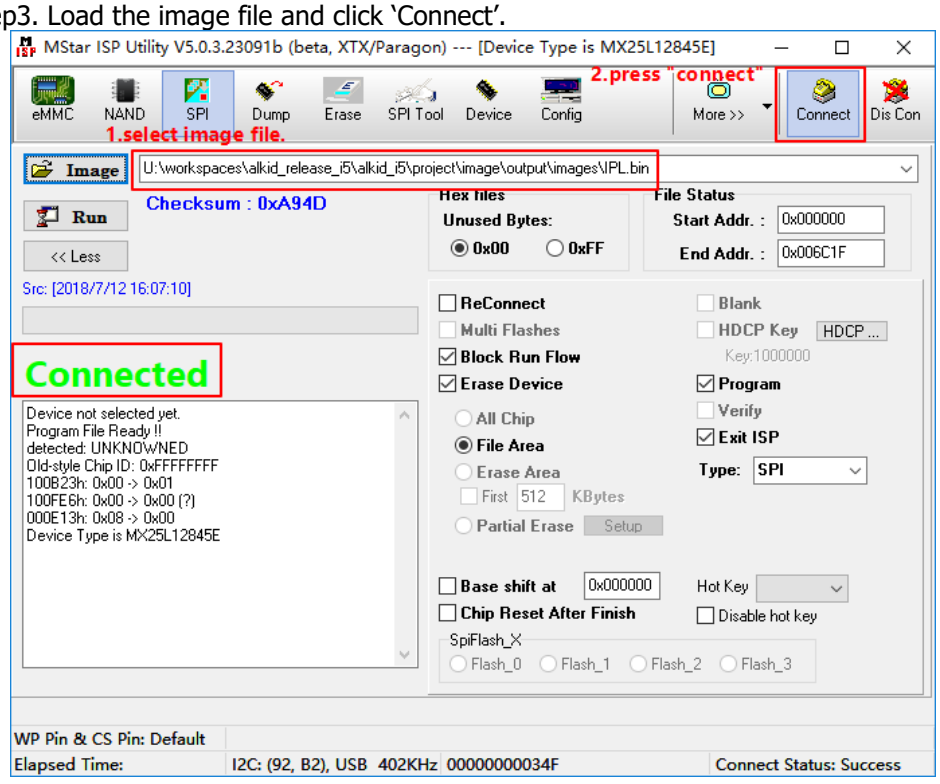

Figure 9: pic 4-3

- Step4. Load the image "IPL.bin", and click 'Run'.

Note：need to select 'erase file area'。

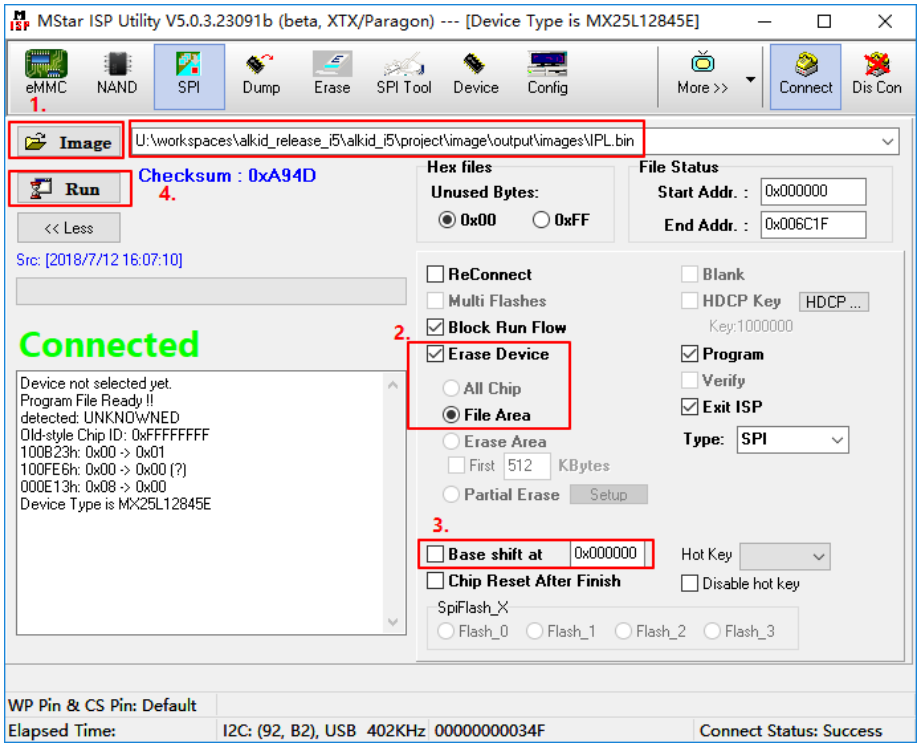

Figure 10: pic 4-4

- Step5. Load the image "IPL_CUST.bin", and cancel 'Erase Device' option. Set 'Base shift 'at 0x10000.

Note: Base shift address might be different by versions. Please check table 1-2.

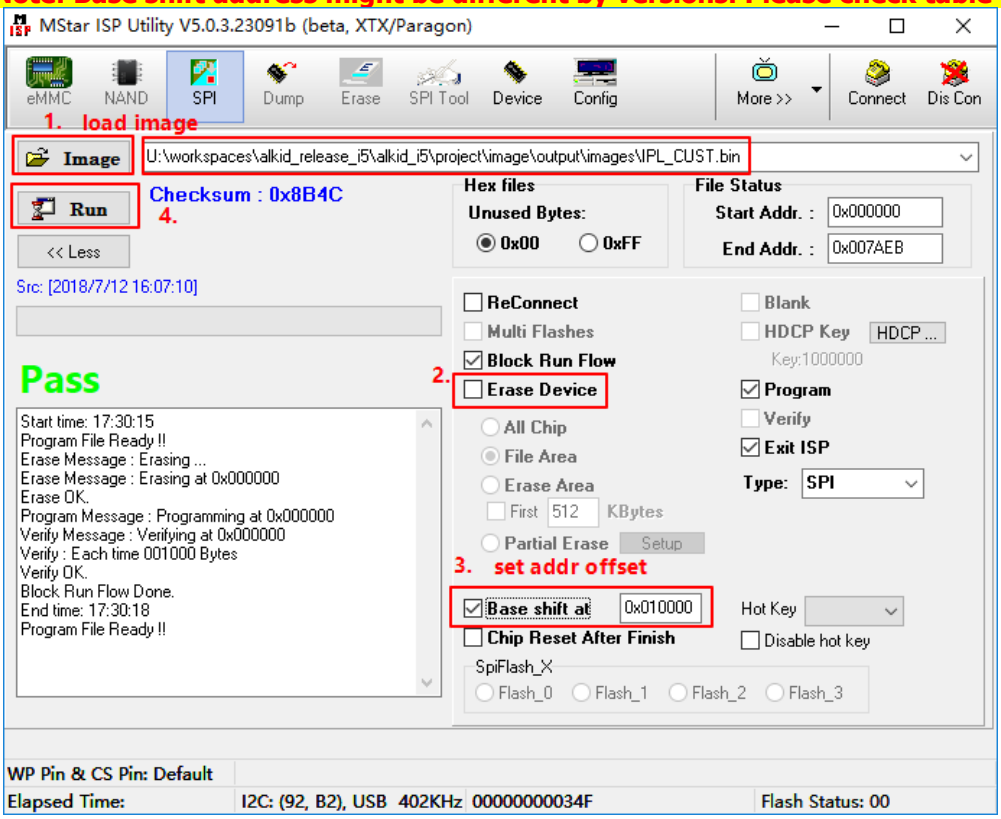

Figure 11: pic 4-5

- Step6. Load image "MXP_SF.bin", and set 'Base shift 'at 0x20000。

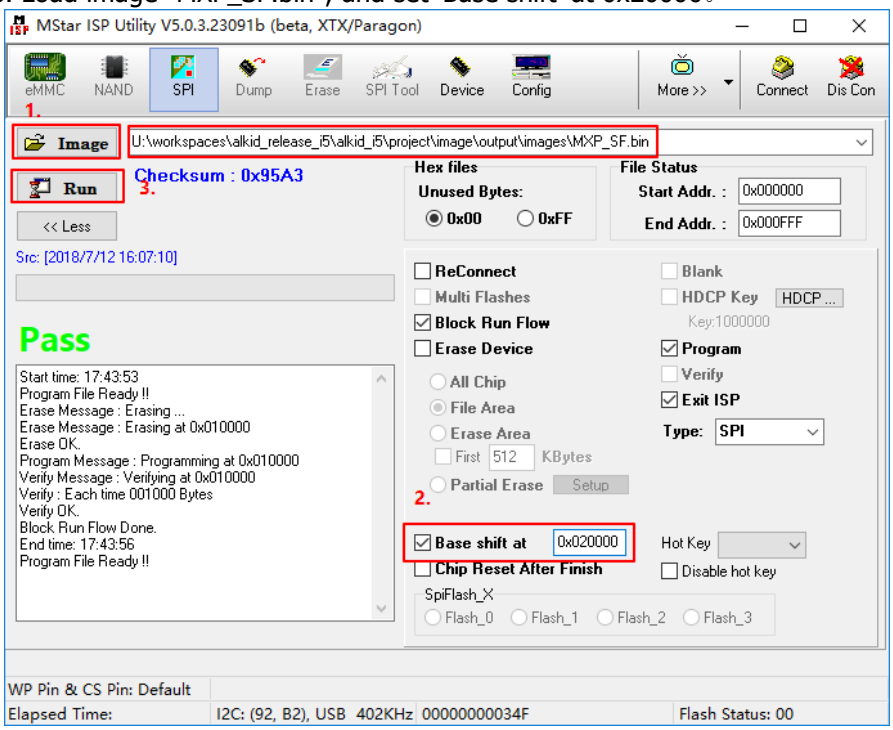

Figure 12: pic 4-6

- Step7. Load image "u-boot.xz.img.bin" , and set 'Base shift 'at 0x30000。

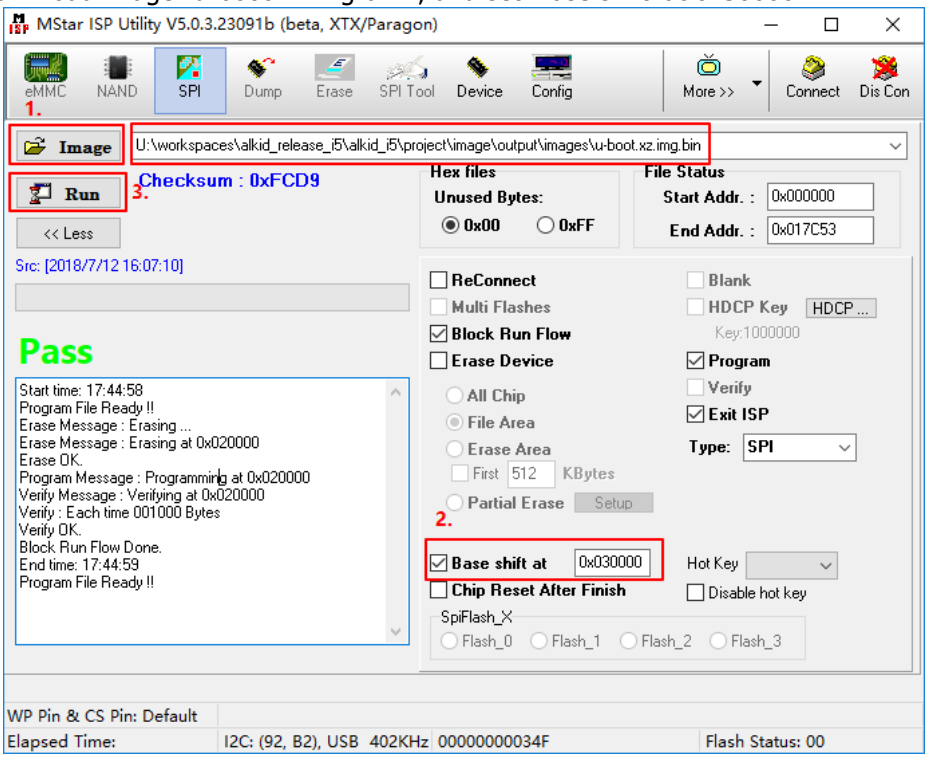

Figure 13: pic 4-7

- Step8. Reboot the EVB (target device), and close the tool.

4.1.2.2. SPI-NAND Flash

- **Default Partition layout**

No	range	size
CIS	0x00000000-0x0020000	128KB
IPL0	0x00140000-0x00200000	768KB
IPL_CUST0	0x00200000-0x00260000	384KB
IPL_CUST1	0x00260000-0x002c0000	384KB
UBOOT0	0x002c0000-0x00320000	384KB
UBOOT1	0x00320000-0x00380000	384KB
ENV	0x00380000-0x003c0000	256KB
KERNEL	0x003c0000-0x008c0000	5120KB
RECOVERY	0x008c0000-0x00dc0000	5120KB
rootfs	0x00dc0000-0x013c0000	6144KB
UBI	0x013c0000-0x008000000	110848KB

Table 7

- **Burning code by ISP tool**
  - **ISP Tool Version**

    Please ensure the ISP Tool version is V5.0.3.23091b(beta)。

    ISP Tool could be found in the SDK, under the folder of "Tool".

  - **Images list**

	Offset	**Image 所在目录**
GCIS.bin	0x000000	project\image\output\images\ GCIS.bin
IPL.bin	0x140000	project\image\output\images\IPL.bin
IPL_CUST.bin	0x200000	project\image\output\images\IPL_CUST.bin
u-boot_spinand.xz.img.bin	0x2C0000	project\image\output\images\ u-boot_spinand.xz.img.bin

Table 8

  - **Burning Steps**
    - Step1. Run ISP tool and close the UART terminal (or the 'Connect' could be blocked by UART terminal app.)

Figure 6: pic 1-6

    - Step2. Choose SPI tab, and click 'More' for the type of 'SPINAND'。

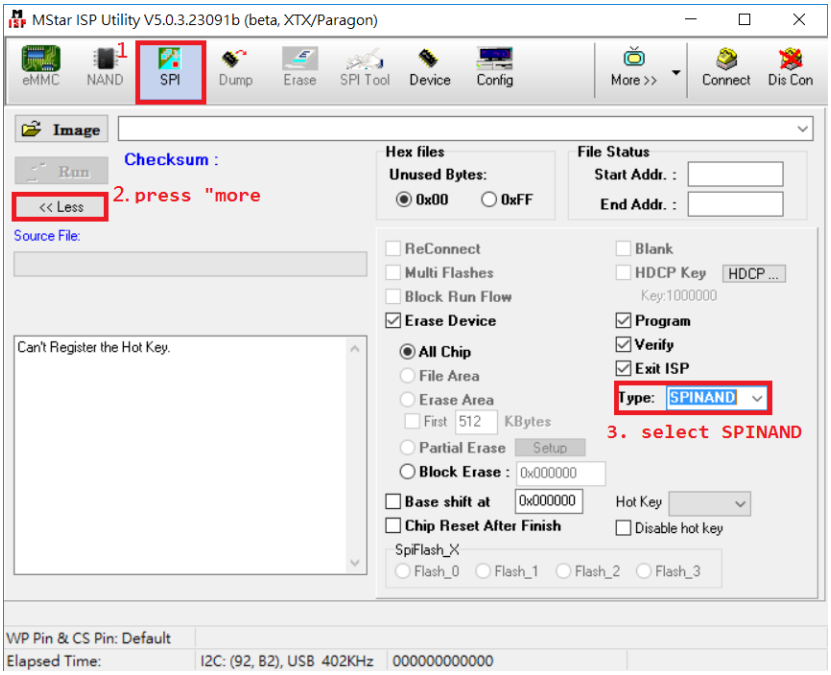

Figure 14: pic 4-8

- Step3. Load the image file and click 'Connect'。

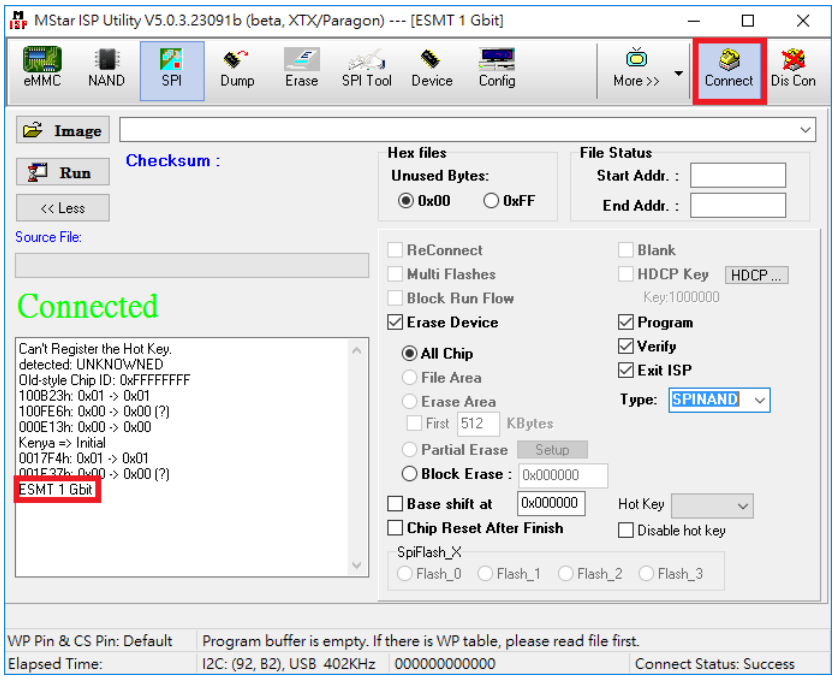

Figure 15: pic 4-9

- Step4. Load the image "GCIS.bin", and click 'Run'
  Note：Need to select 'erase all chip'

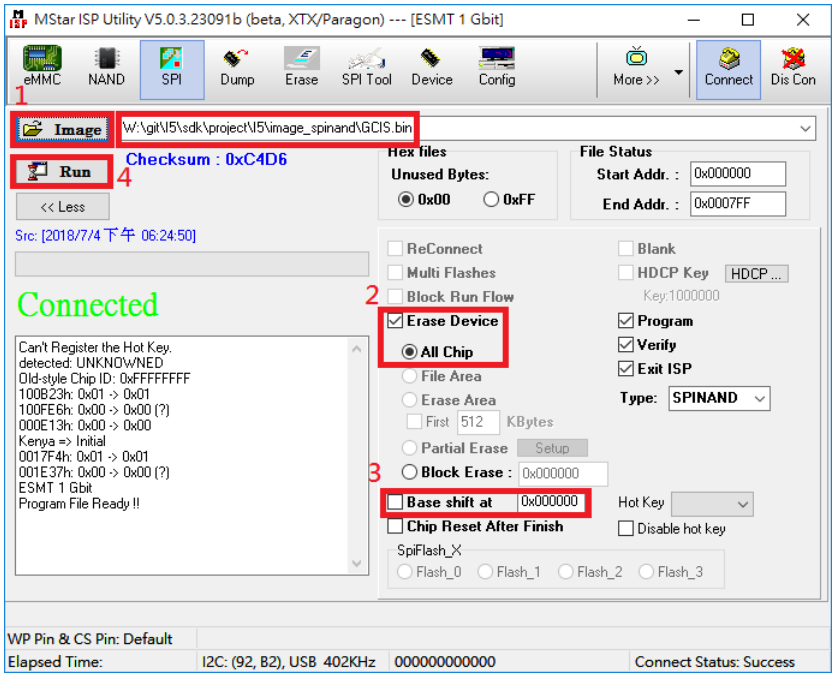

Figure 16: pic 4-10

- Step5. Load the image "IPL.bin", and cancel 'Erase Device' option. Set 'Base shift 'at 0x140000.

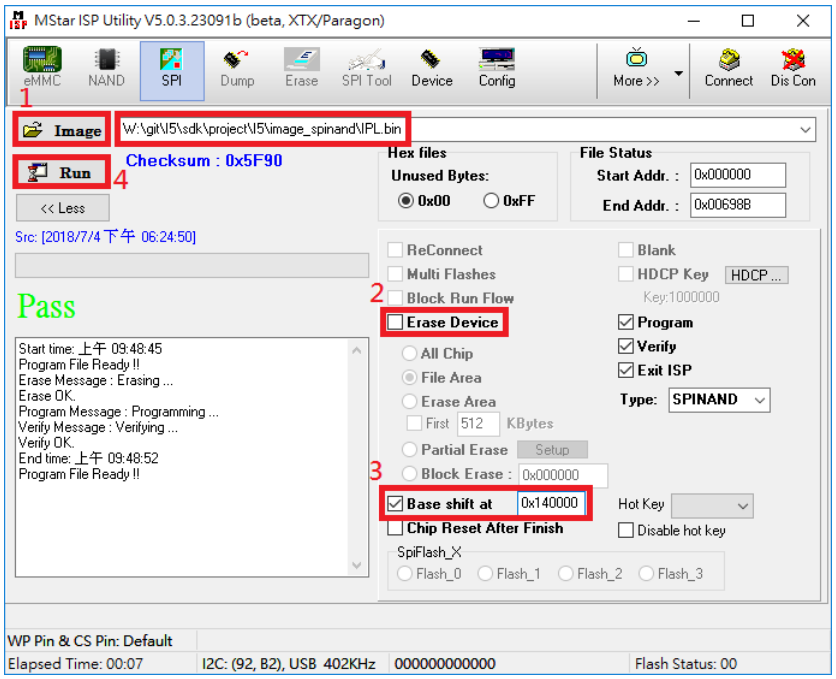

Figure 17: pic 4-11

- Step6. Load the image "IPL_CUST.bin", and set 'Base shift 'at 0x200000。

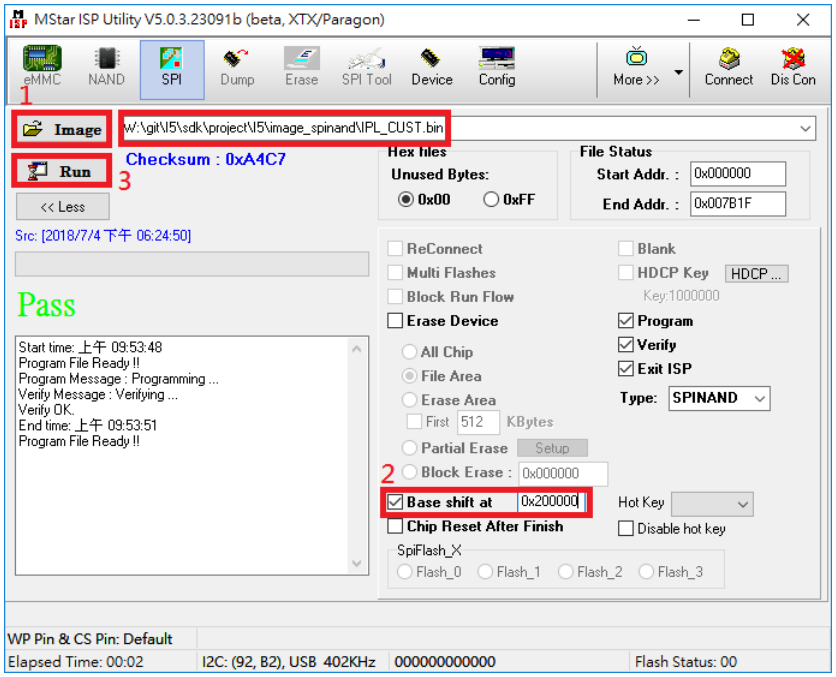

Figure 18: pic 4-12

- Step7. Load the image "u-boot_spinand.xz.img.bin", and set 'Base shift 'at 0x2C0000.

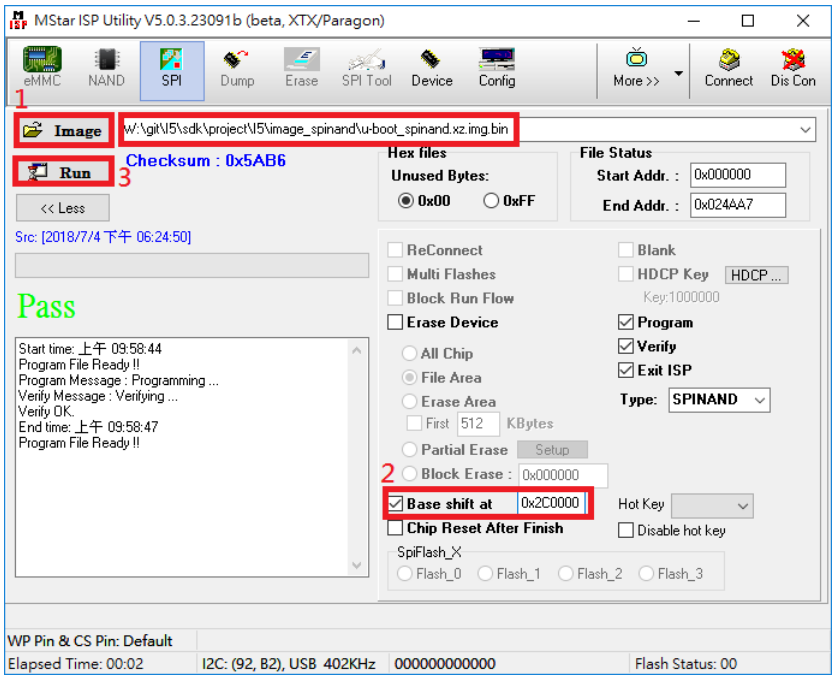

Figure 19: pic 4-13

- Step8. Reboot the EVB (target device), and close the tool.