



SigmaStar Camera **开发环境搭建**



© 2019 SigmaStar Technology Corp. All rights reserved.

SigmaStar Technology makes no representations or warranties including, for example but not limited to, warranties of merchantability, fitness for a particular purpose, non-infringement of any intellectual property right or the accuracy or completeness of this document, and reserves the right to make changes without further notice to any products herein to improve reliability, function or design. No responsibility is assumed by SigmaStar Technology arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

SigmaStar is a trademark of SigmaStar Technology Corp. Other trademarks or names herein are only for identification purposes only and owned by their respective owners.



REVISION HISTORY

Revision No.	Description	Date
1.0	<ul style="list-style-type: none">Initial release	19/12/2018
1.1	<ul style="list-style-type: none">Add spi nand note.	22/01/2019



TABLE OF CONTENTS

REVISION HISTORY	i
TABLE OF CONTENTS.....	ii
1. 板子硬件连接.....	1
1.1. SSC009A 主板示意图:	1
1.2. SSC009B 主板示意图:	4
2. 准备编译环境.....	7
2.1. 安装 Linux 服务器	7
2.2. 安装交叉编译工具.....	7
3. 编译.....	8
3.1.1 编译 boot.....	8
3.1.2 编译 kernel	8
3.1.3 编译 SDK(ALKAID)	9
4. 烧写.....	11
4.1.1 Burning code by uboot	11
4.1.2 Burning uboot by ISP Tool	11

1.1. SSC009A 主板示意图:

调试串口： TTL 电平，特率 115200

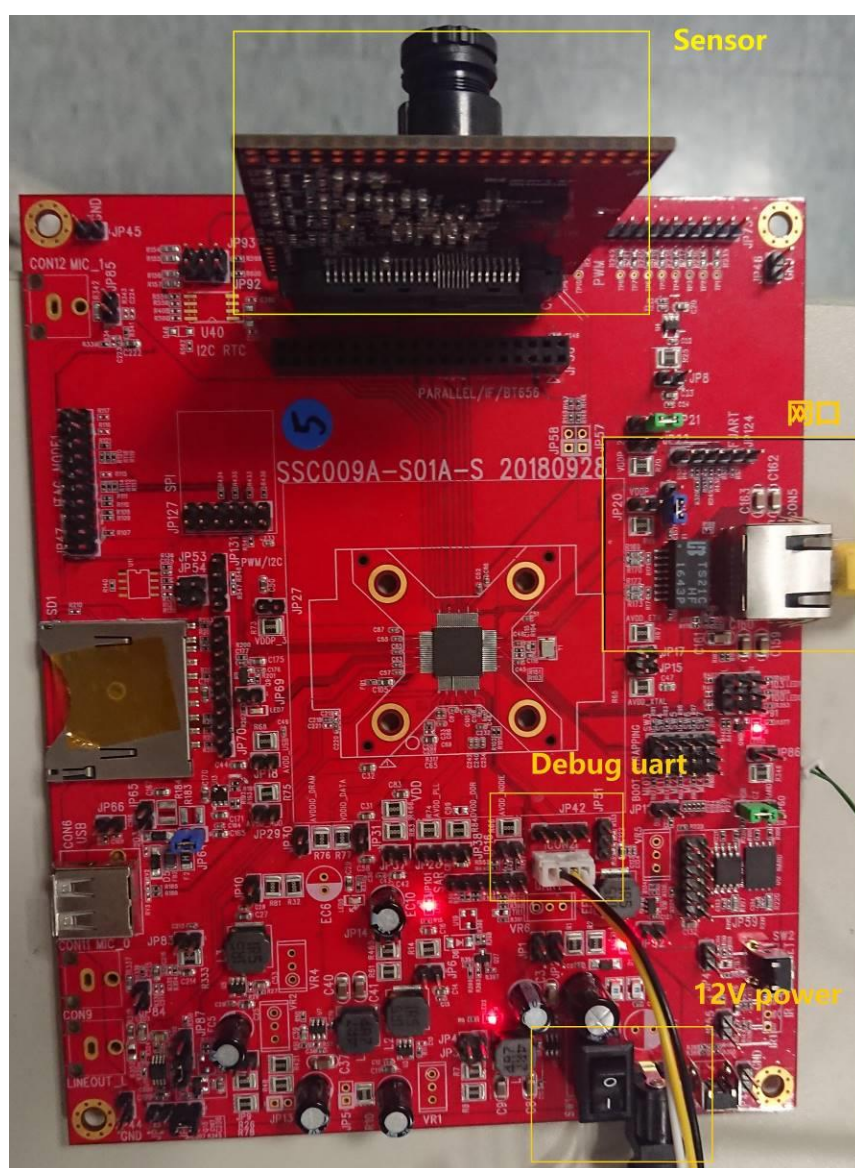


Figure 1: 图 1-1



Security Level: Confidential A

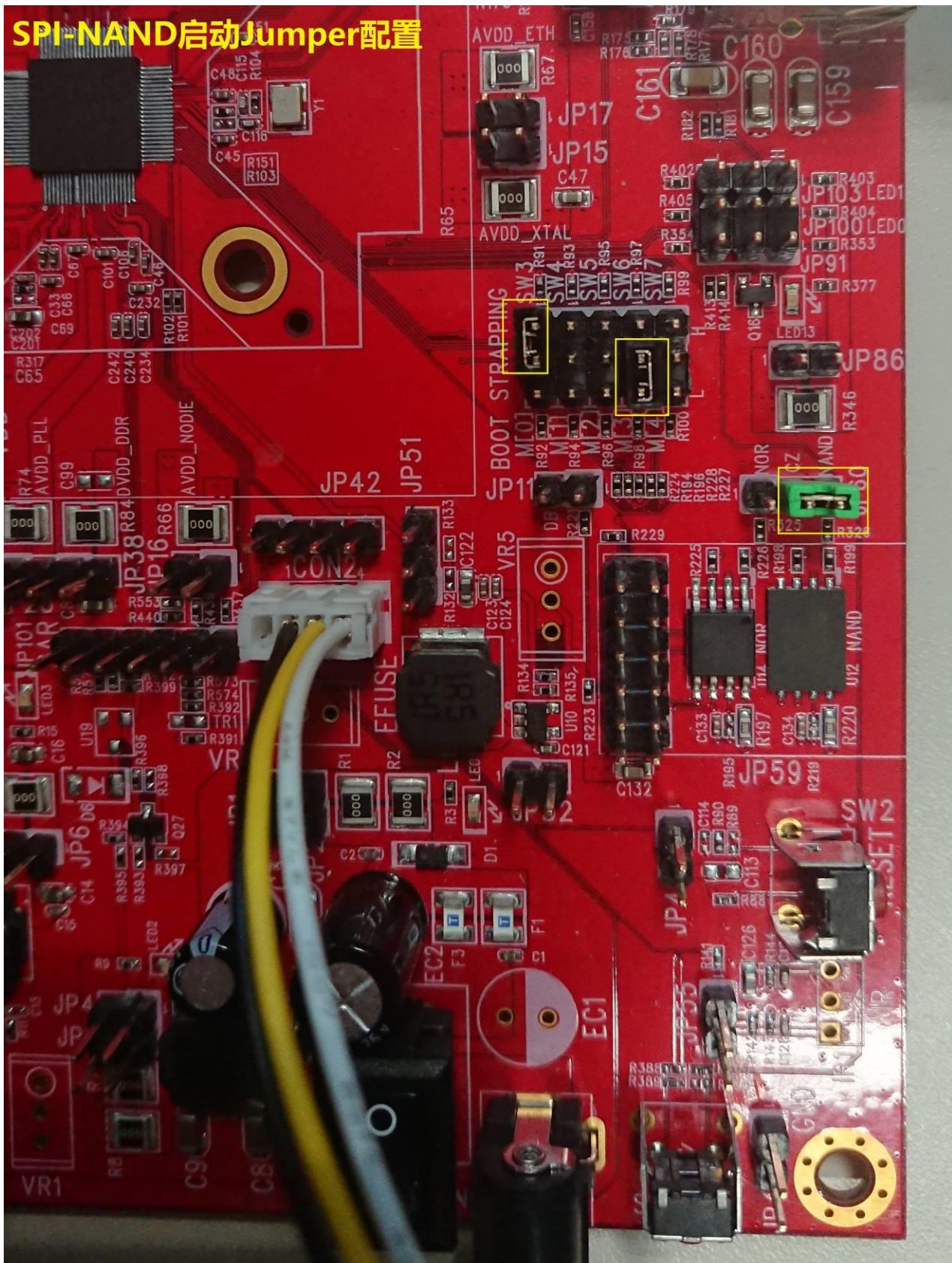


Figure 3 图 1-3

Sensor

Debug uart

12V power

3/6/2019

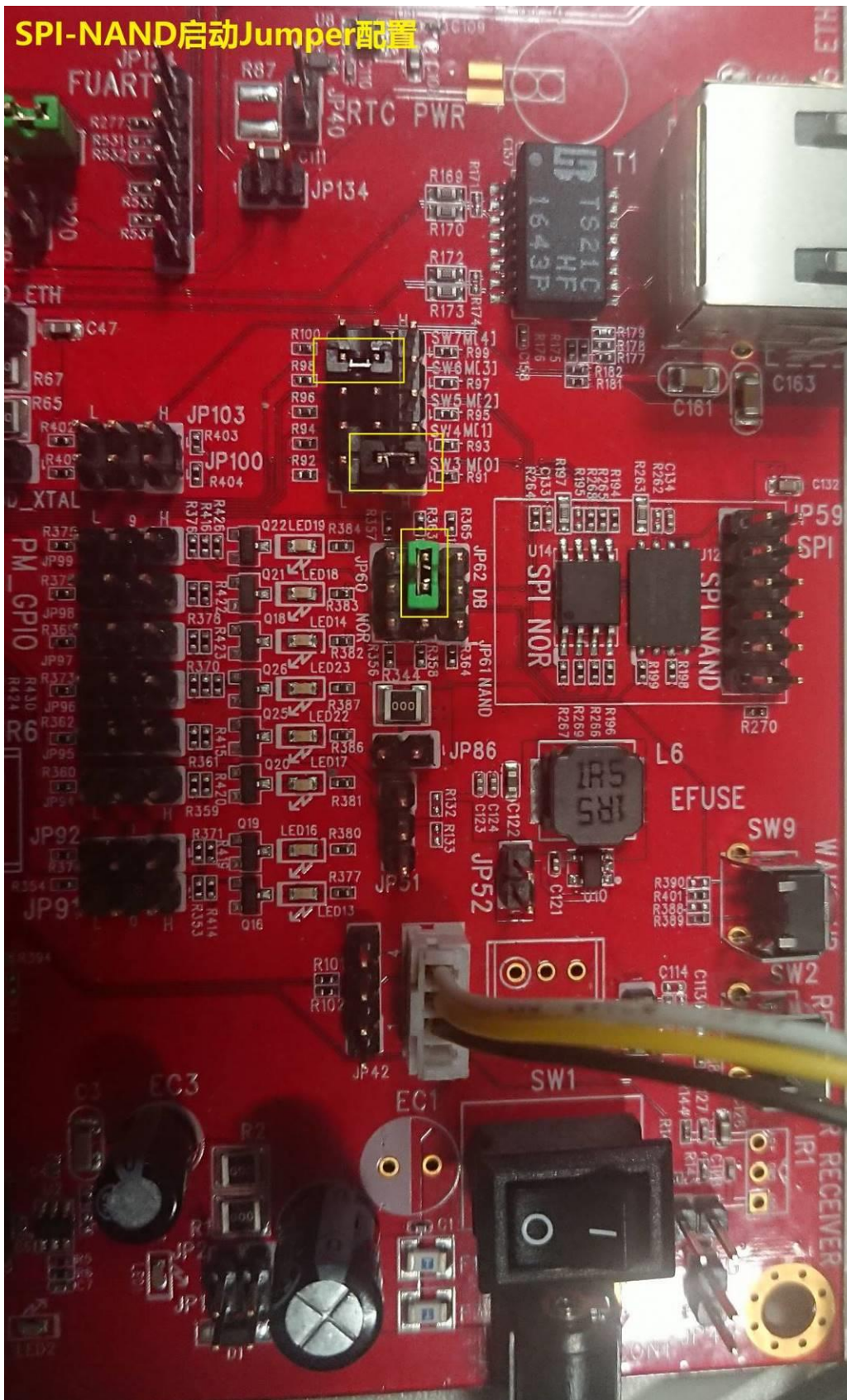


Figure 4 图 1-5

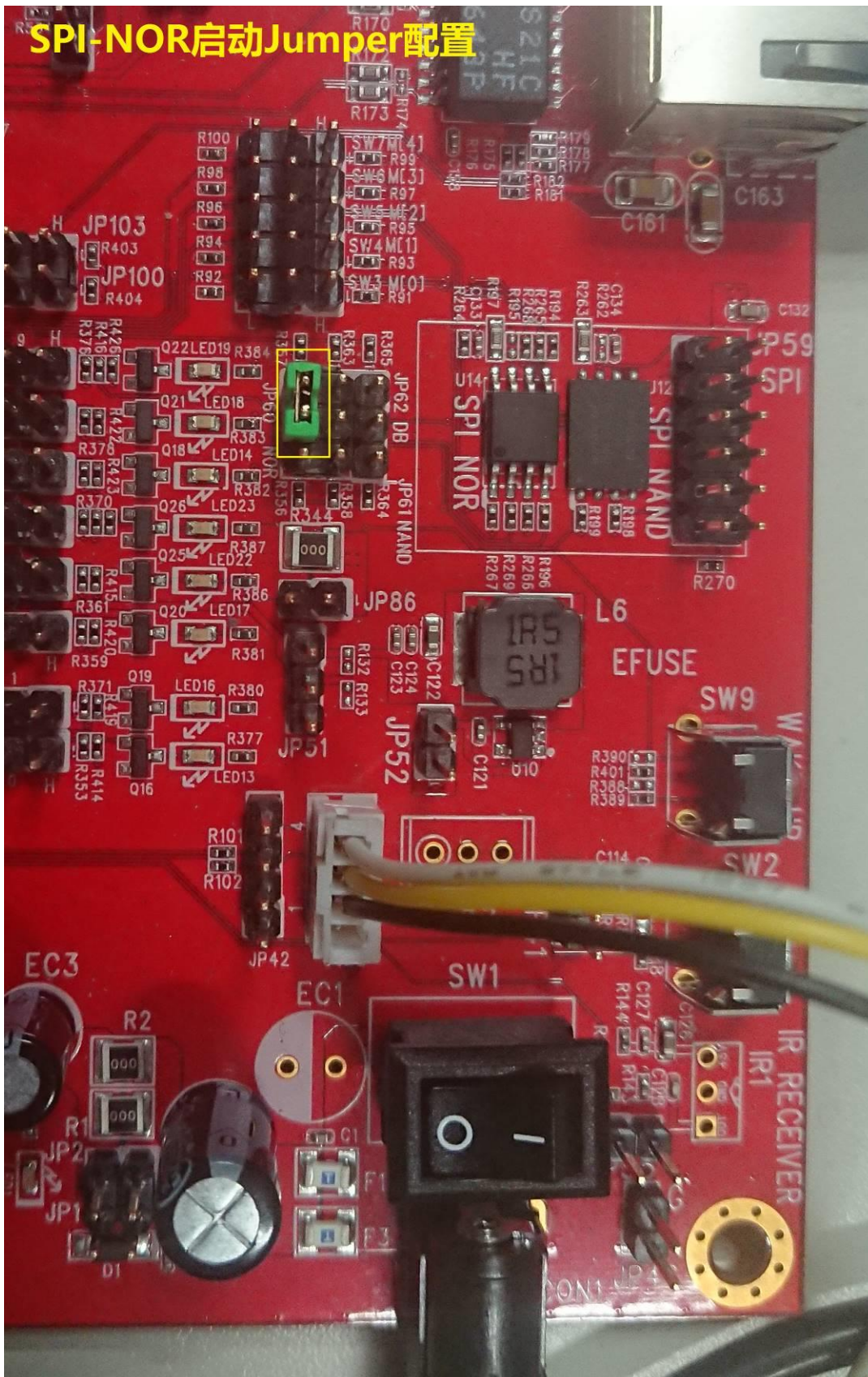


Figure 5 图 1-6



2. 准备编译环境

通常我们会以交叉编译的方式进行开发和调试，即“宿主机+目标机”的形式。而宿主机和目标机的连接我们一般采用串口连接或网络连接，如下图所示：

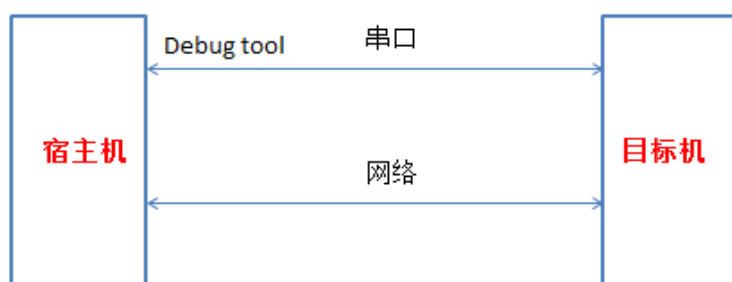


Figure 6: 图 2-1 交叉编译连接方式

注：我们提供的 debug tool 作用为读寄存器和烧写 Mboot。

2.1. 安装 Linux 服务器

建议使用 Ubuntu 16.04。

2.2. 安装交叉编译工具

我们提供编译工具链 gcc-linaro-arm-linux-gnueabi-4.8-2014.04_linux 用于编译 glibc 版本的 mi 和 kernel, boot, arm-buildroot-linux-uclibc-gnueabi-4.9.4 用于 uclibc 版本，工具链一般打包在 SDK 开发包中，或者请找 FAE 索取。



3. 编译

本芯片支持 nor flash 和 spi nand flash 两种启动方式，因此在 SDK 中编译有所区分，通过不同的配置文件来实现，SDK 中包含了最新的 boot 和 kernel 的 image，并且脚本打包成可烧录的 image。

3.1.1 编译 boot

- **SPI-NOR package**
#declare -x ARCH="arm"
#declare -x CROSS_COMPILE="arm-linux-gnueabihf-"
make infinity6_defconfig;
#make clean;
#make
- **Get image**
cp u-boot.xz.img.bin \${ your_release_path } // 选择 spi-nor 时

3.1.2 编译 kernel

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

CHIP	Glibc compiler	Uclibc compiler	Kernel make config	DTS
QFN88 64MB: SSC323 (No Eth) SSC325 SSC327 QFN88 128MB: SSC325D	Linaro Glibc4.8.3 arm-linux-gnueabi hf-	Buildroot Uclibc 4.9.4 arm-buildroot-linux-uc libcgnueabi-hf-	infinity6_defconfig	infinity6-ssc009a-s 01a.dts
QFN128 128MB: SSC327DE QFN128 256MB SSC327Q	Linaro Glibc4.8.3 arm-linux-gnueabi hf-	Buildroot Uclibc 4.9.4 arm-buildroot-linux-uc libcgnueabi-hf-	infinity6-ssc009b-s01 a_defconfig	infinity6-ssc009b-s 01a.dts

Table 1

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

CHIP	Glibc compiler	Uclibc compiler	Kernel make config	DTS
------	----------------	-----------------	--------------------	-----



QFN88 64MB: SSC323 (No Eth) SSC325 SSC327 QFN88 128MB: SSC325D SSC327D	Linaro Glibc4.8.3 arm-linux-gnueabi f-	Buildroot Uclibc 4.9.4 arm-buildroot-linux-uc libcgnueabi-hf-	infinity6_spinand_defc onfig	infinity6-ssc009a-s 01a.dts
QFN128 128MB: SSC325DE SSC327DE QFN128 256MB SSC327Q	Linaro Glibc4.8.3 arm-linux-gnueabi f-	Buildroot Uclibc 4.9.4 arm-buildroot-linux-uc libcgnueabi-hf-	infinity6_ssc009b_s01 a_spinand_defconfig	infinity6-ssc009b-s 01a.dts

Table 2

注：请参考表格和你的芯片版本做对应的编译。

```
# declare -x ARCH="arm"
# declare -x CROSS_COMPILE="$compiler"
//exp: uclibc "arm-buildroot-linux-uclibcgnueabi-hf"
# make xxx_kernel_make_config
//exp: make infinity6_ssc009b_s01a_defconfig
# make clean;
# make
```

- **Get image**

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

3.1.3 编译 SDK(ALKOID)

- **SPI-NOR flash package**

CHIP	Glibc	Uclibc
QFN88 64MB: SSC323 (No Eth) SSC325 SSC327	nor.glibc-squashfs.009a.64.qfn88	nor.uclibc-squashfs.009a.64.qfn88
QFN88 128MB: SSC325D	nor.glibc-squashfs.009a.128.qfn88	nor.uclibc-squashfs.009a.128.qfn88
QFN128 128MB: SSC327DE	nor.glibc-squashfs.009b.128.qfn128	nor.uclibc-squashfs.009b.128.qfn128
QFN128 256MB SSC327Q	nor.glibc-squashfs.009b.256.qfn128	nor.uclibc-squashfs.009b.256.qfn128

Table 3

- **SPI-NAND flash package**

CHIP	Glibc	Uclibc
QFN88 64MB:	Ubifs version:	NA



SSC323 (No Eth) SSC325 SSC327	spinand.glibc.009a.64.qfn88 Squashfs version: spinand.glibc-squashfs.009a.64.qfn88	
QFN88 128MB: SSC325D	Ubifs version: spinand.glibc.009a.128.qfn88 Squashfs version: spinand.glibc-squashfs.009a.128.qfn88	NA
QFN128 128MB: SSC327DE	Ubifs version: spinand.glibc.009b.128.qfn128 Squashfs version: spinand.glibc-squashfs.009b.128.qfn128	NA
QFN128 256MB SSC327Q	Ubifs version: spinand.glibc.009b.256.qfn128 Squashfs version: spinand.glibc-squashfs.009b.256.qfn128	NA

Table 4

注：请参考表格和你的芯片版本做对应的编译。

```
# 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. 烧写

4.1.1 Burning code by uboot

● Run tftp (FTP server) on PC

Step1. 使用 tftp 工具指向 image path: SDK\project\image\output\images\, 并选择正确的网卡。

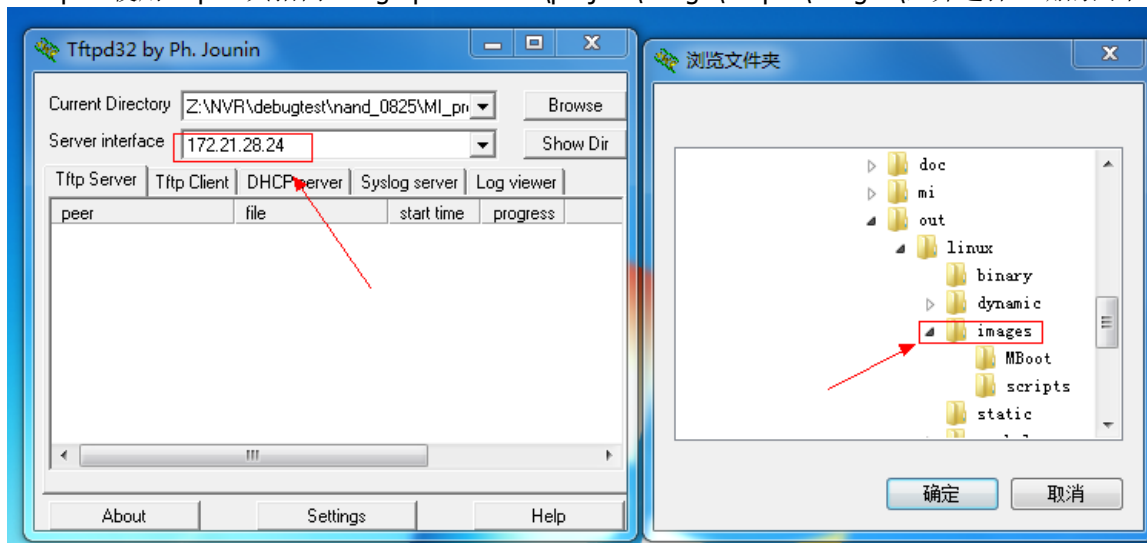


Figure 7: 图 4-1

Step 2. 连接板子的网口到 PC 端, 连接 debug 串口工具到 PC 端, 并检查连接的正确性。

● Run tftp (FTP Client) on EVB

Step 1. 板子开机, 长按回车, 进入 bootloader command line. Mstar 的 boot loader。

■ 首次烧录请设置 IP: (除非 ip 设置变更或者更换 flash)

set -f gatewayip 192.168.1.1

set -f ipaddr 192.168.1.127

//设定 FTP Client (EVB 板子) 使用的 IP

set -f netmask 255.255.255.0

set -f serverip 192.168.1.100

//设定 FTP server (PC) 的 IP

saveenv

注:

1. 为了保证烧录顺利, 请保证 PC 和开发板处于同一网段。
2. 请采用静态方式固定分配 ip。防止烧录时 ip 地址跳变。
3. 您也可以使用独立网卡使 PC 端直连开发板, 固定该网卡的内网 ip 地址, 并按上述方法设定开发板。

■ 在 UBOOT console 下执行以下指令即可自动透过 ethernet 烧录。

estar (OR: estar auto_update.txt)

4.1.2 Burning uboot by ISP Tool

本方式适用于空机烧录, 或者 uboot 已经损坏导致无法通过 uboot 升级的场合。



4.1.2.1. SPI-NOR-Flash

● Default Partition layout

No	range	size
IPL	0x00000000, 0x00010000	64KB
IPL_CUST	0x00010000, 0x0000F000	60KB
KEY_CUST	0x0001F000, 0x00001000	64KB
MXPT	0x00020000, 0x00001000	64KB
UBOOT	0x00030000, 0x0001F000	124KB
UBOOT_ENV	0x0004F000, 0x00001000	64KB
BOOT	0x00000000, 0x00050000	320KB
KERNEL	0x00050000, 0x00200000	2048KB
ROOTFS	0x00250000, 0x00400000	4096KB
NVRSERVICE	0x00650000, 0x00300000	3072KB
CUSTOMER	0x00950000, 0x006B0000	6848KB

Table 5

● Burning code by ISP tool

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

Table 6

● Burning Steps

- Step1. 执行 ISP tool。并且关闭 UART terminal，否则无法正常‘Connect’。
- Step2. 选择 SPI tab, 点击‘More’ 并且选择类型为‘SPI’。

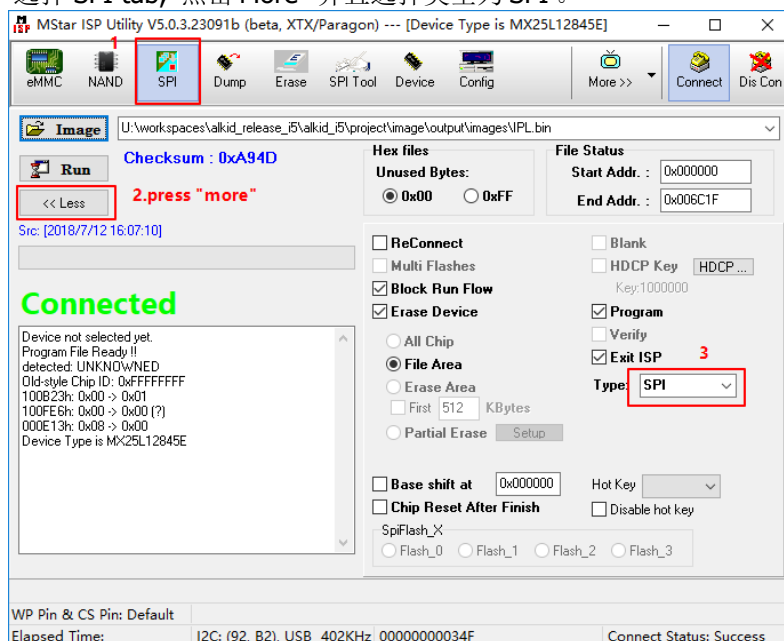


Figure 8: 图 4-2

- Step3. 加载烧录文件并点击'Connect'。

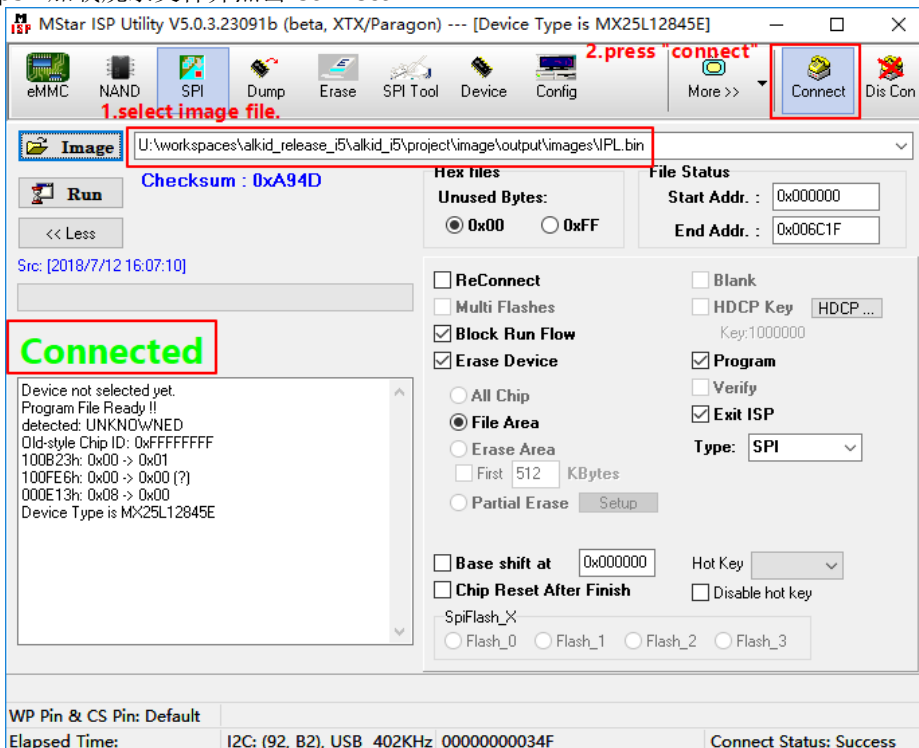


Figure 9: 图 4-3

- Step4. 加载 image "IPL.bin", 并点击'Run'。

注: 需要勾选'erase file area'。

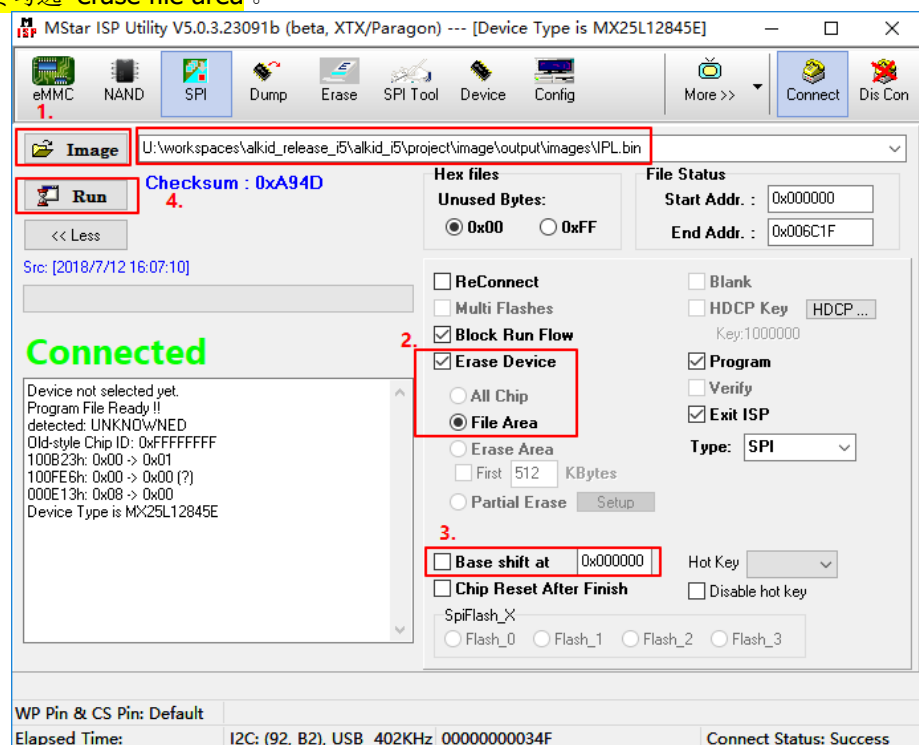


Figure 10 图 4-4



- Step5. 加载 image "IPL_CUST.bin", 取消'Erase Device'选项。设置'Base shift 'at 0x10000。

注:可能随着版本变化, Base shift 的地址以表格 1-2 数据为准。

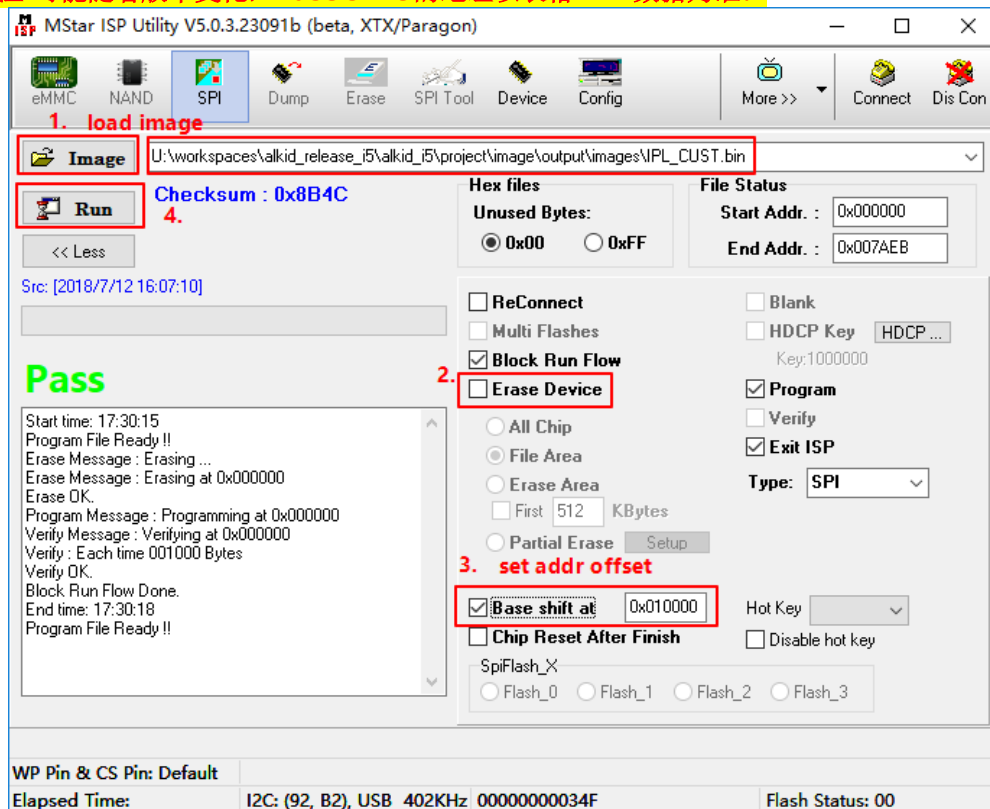


Figure 11: 图 4-5

- Step6. 加载 image "MXP_SF.bin", 设置'Base shift 'at 0x20000。

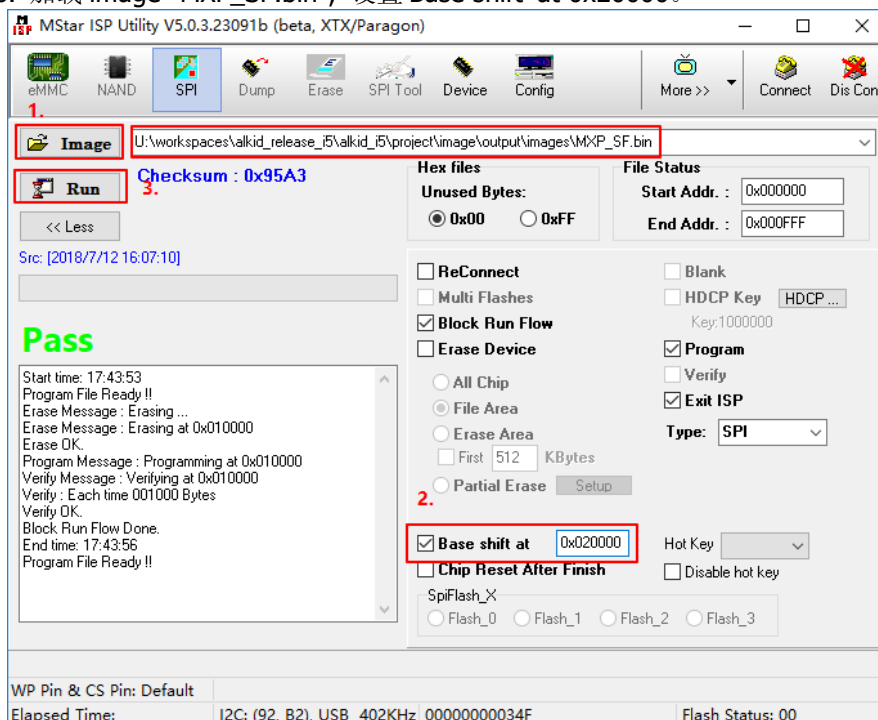


Figure 12: 图 4-6



- Step7. 加载 image “u-boot.xz.img.bin”，设置‘Base shift’at 0x30000。

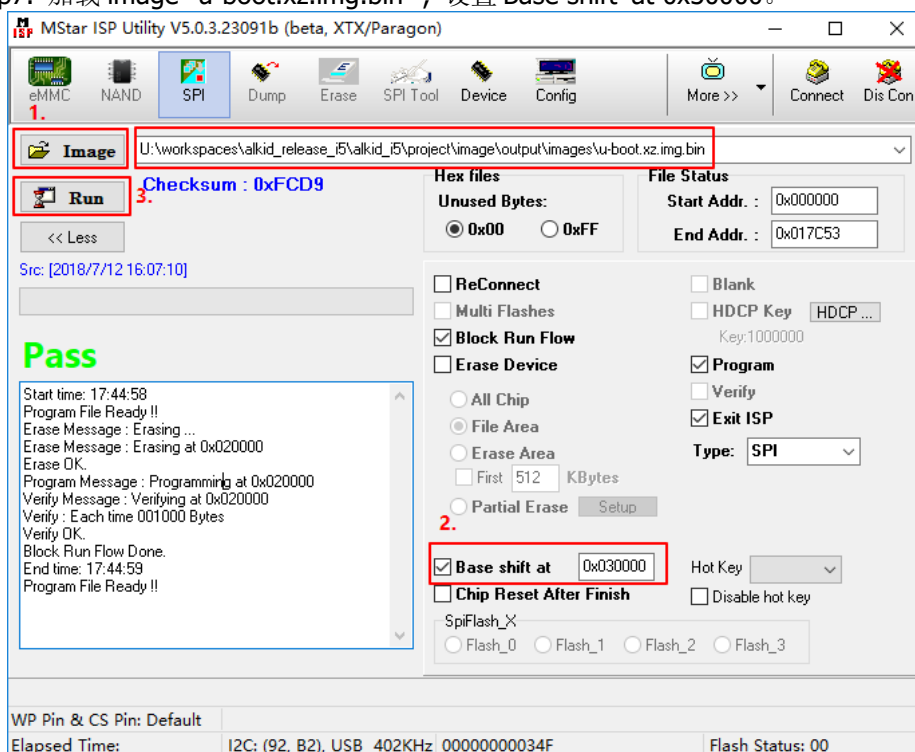


Figure 13: 图 4-7

- Step8. 重启 EVB 板子，关闭工具。

4.1.2.2. SPI-NAND Flash

● Default Partition layout

No	range	size
CIS	0x00000000-0x00200000	128KB
IPL0	0x00140000-0x001a0000	384KB
IPL1	0x001a0000-0x00200000	384KB
IPL_CUST0	0x00200000-0x00260000	384KB
IPL_CUST1	0x00260000-0x002c0000	384KB
UBOOT0	0x002c0000-0x00380000	768KB
UBOOT1	0x00380000-0x00440000	768KB
ENV	0x00440000-0x004a0000	384KB
LOGO	0x004a0000-0x005c0000	1152KB
KERNEL	0x005c0000-0x00ac0000	5120KB
RECOVERY	0x00ac0000-0x00fc0000	5120KB
UBI	0x00fc0000-0x008000000	114944KB

Table 7



● Burning code by ISP tool

■ ISP Tool Version

请确定 ISP Tool 版本为 V5.0.3.23091b(beta)。

ISP Tool 会在首次版本发布的时候一起打包在 Tool 目录下。

■ Images list

	Offset	Image 所在目录
GCIS.bin	0x000000	project\image\output\images\ GCIS.bin
IPL.bin	0x60000	project\image\output\images\IPL.bin
IPL_CUST.bin	0xC0000	project\image\output\images\IPL_CUST.bin
u-boot_spinand.xz.img.bin	0x120000	project\image\output\images\ u-boot_spinand.xz.img.bin

Table 8

■ Burning Steps

- Step1. 执行 ISP tool。并且关闭 UART terminal，否则可能无法正常'Connect'。

ISP_Tool_5.0.3.23091b_beta_release.exe

Figure 6: 图 1-6

- Step2. 选择 SPI tab, 点击'More' 并且选择类型为'SPINAND'。

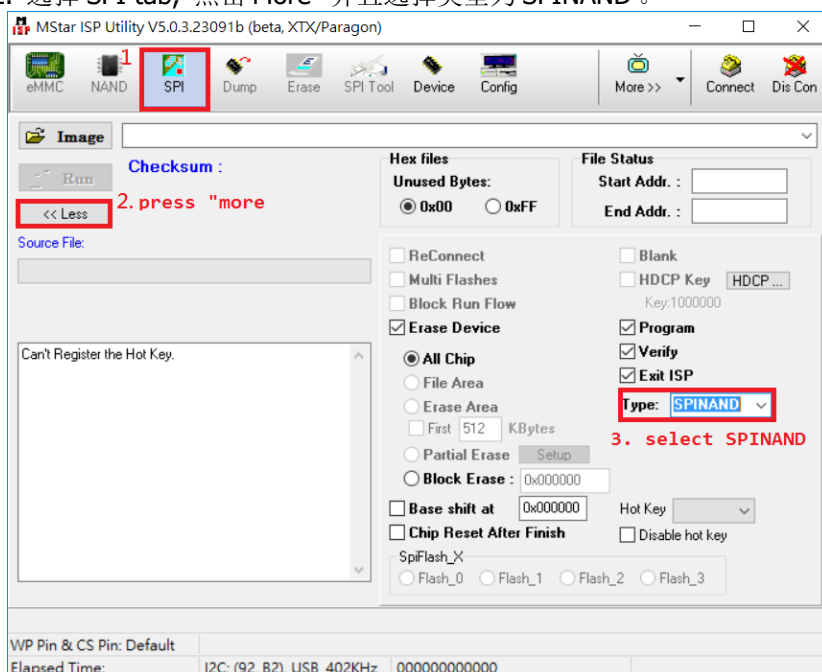


Figure 14: 图 4-8

- Step3. 加载烧录文件并点击'Connect'。

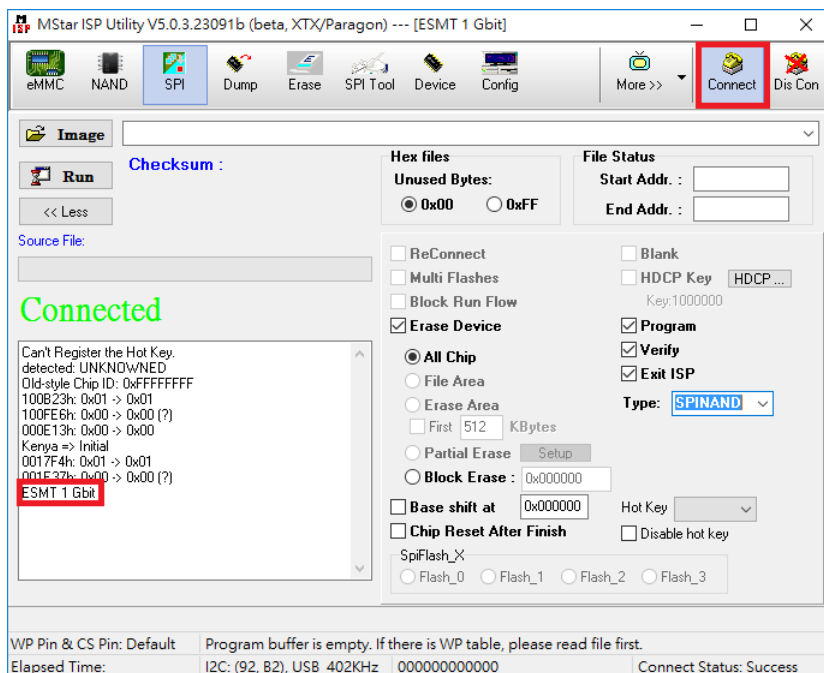


Figure 15: 图 4-9

- Step4. 加载 image "GCIS.bin", 并点击 'Run'

注: 需要勾选 'erase all chip'

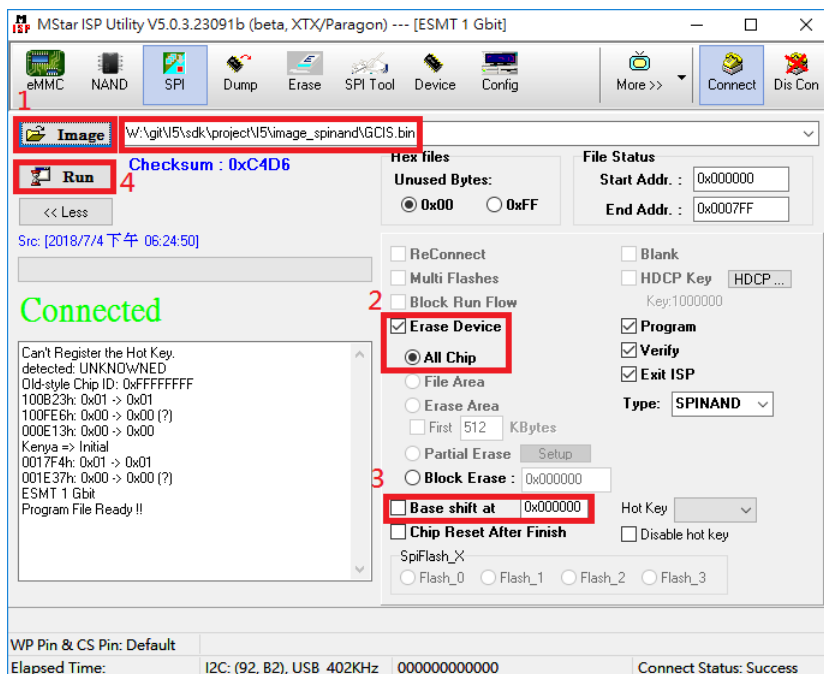


Figure 16: 图 4-10

- Step5. 加载 image "IPL.bin", 取消 'Erase Device' 选项, 设置 'Base shift' at 0x140000.

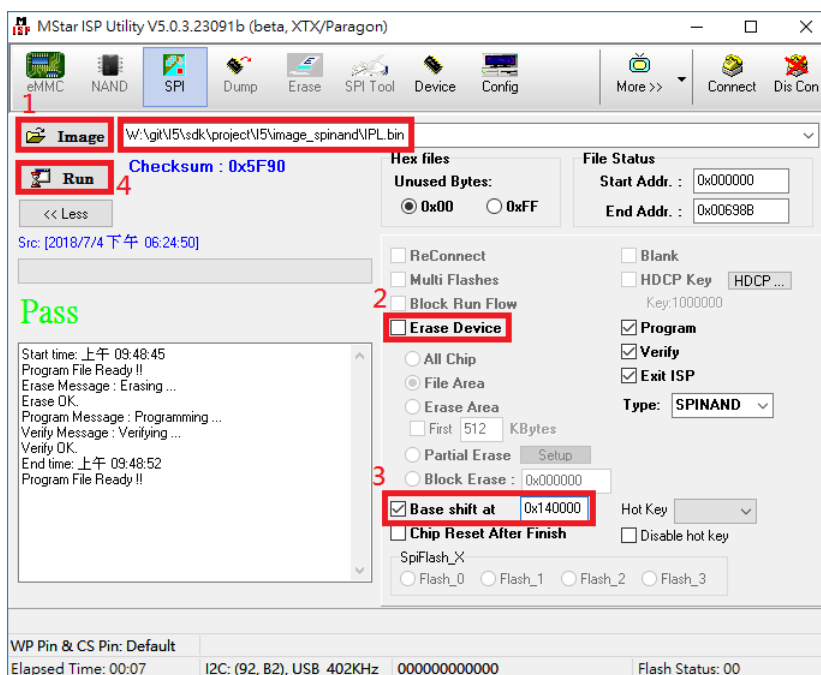


Figure 17: 图 4-11

- Step6. 加载 image "IPL_CUST.bin", 设置'Base shift' at 0x200000。

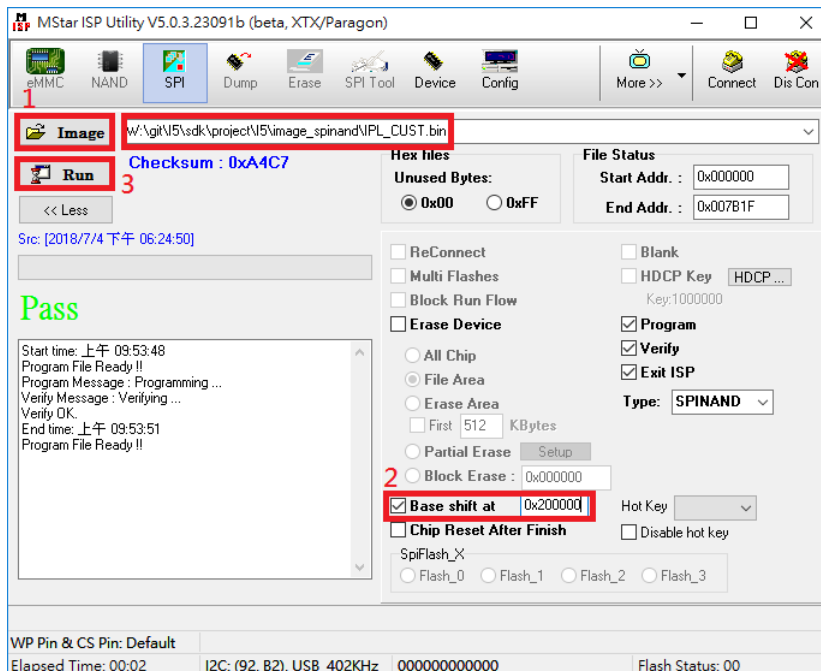


Figure 18: 图 4-12

- Step7. 加载 image "u-boot_spinand.xz.img.bin", 设置'Base shift' at 0x2C0000。

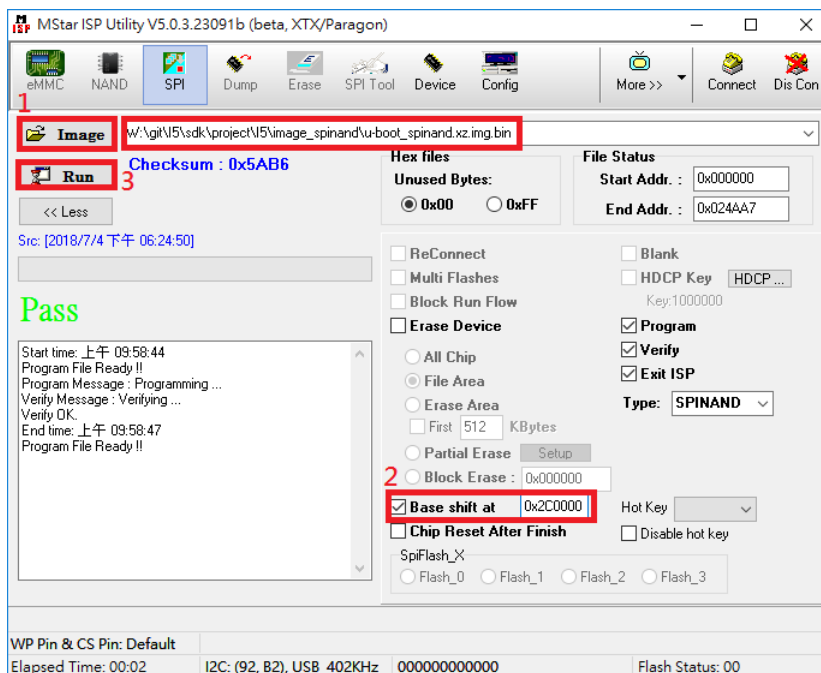


Figure 19: 图 4-13

- Step8. 重启 EVB 板子，关闭工具。