



深圳市英蓓特科技有限公司

MaaXBoard

(EM-MC-SBC-IMX8M)

Linux 软件开发指导

V1.1

版权声明

- ◆ MaaXBoard 单板机(型号:EM-MC-SBC-IMX8M)及其相关知识产权由深圳市英蓓特科技有限公司所有。
- ◆ 本文档由深圳市英蓓特科技有限公司版权所有，并保留一切权利，在未经英蓓特公司书面许可的情况下，不得以任何形式来修改、分发或复制本文档的任何部分。

免责声明

- ◆ 产品附带提供的程序源代码、软件、资料文档等，深圳市英蓓特有限公司不提供任何类型的担保；不论是明确的，还是隐含的，包括但不限于合适特定用途的保证，全部的风险，由使用者来承担。

通过认证

- ◆ MaaXBoard 单板机已经获得 CE, FCC 证书。

版本	描述	作者	日期
V1.0	初稿	Sandy	20190411

目录

版本记录	3
第 1 章 环境搭建	6
1.1 搭建开发环境	6
1.2 配置编译环境	6
第 2 章 编译	7
2.1 U-Boot	7
2.1.1 获取 U-Boot 源码	7
2.1.2 编译 U-Boot 镜像	7
2.2 Kernel	8
2.2.1 获取内核源码	8
2.2.2 编译镜像	8
2.2.3 编译模块	8
第 3 章 烧写	9
3.1 分区	9
3.1.1 对 SD 卡进行分区	9
3.2 烧写 U-Boot	9
3.2.1 烧写到 SD 卡	9
3.3 烧写 Kernel	9
3.3.1 烧写镜像到 SD 卡	9
3.4 复制 uEnv.txt	10
3.5 烧写 File system	10
3.5.1 烧写 File system 到 SD 卡	10
3.5.2 复制模块文件	10
3.6 读取完整镜像	11
3.6.1 读取 SD 卡的镜像	11
第 4 章 用户自定义模块	12
4.1 Hello world 的编译和测试	12
4.1.1 测试代码	12
4.1.2 makefile 文件（板载编译和交叉编译）	12

4.1.3	编译和测试	12
第 5 章	附录	14
5.1	硬件	14
第 7 章	联系方式	17

1.1 搭建开发环境

搭建开发环境需要：

- ◆ 硬件：至少 20GB 磁盘空间，2GB 运行内存
- ◆ 软件：Ubuntu 64 bit 操作系统，14.04 LTS 或以上的 LTS 版本 (Ubuntu Desktop 或 Ubuntu Server)

也可以用虚拟机来运行 Ubuntu 64 bit 操作系统。

启动 Ubuntu 系统后，运行下列命令安装开发需要的软件。

```
sudo apt install git
sudo apt install kpartx
sudo apt install zlib1g-dev
sudo apt install device-tree-compiler
```

注意：如果提示安装失败或找不到指定的安装包，先执行 `sudo apt update` 再试一遍。

1.2 配置编译环境

将发布文件夹的 02Linux 的全部内容拷贝到 Linux 环境下的 \$HOME 目录下(解压 rar 文件)，编译工具 gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz 在 \$HOME/03LinuxTools 目录下，用如下命令解压：

```
$tar -Jxvf gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz
```

导入环境变量：

```
$export
CROSS_COMPILE=$HOME/03LinuxTools/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu/bin/
aarch64-linux-gnu-
$export ARCH=arm64
```

注意：每次编译前都要配置环境变量。

2.1 U-Boot

2.1.1 获取 U-Boot 源码

Uboot 源码在\$HOME/01LinuxSourceCode 目录下, 解压 u-boot*.tar.gz :

```
$ cd $HOME/01LinuxSourceCode
$ tar -zxvf u-boot*.tar.gz
```

2.1.2 编译 U-Boot 镜像

```
$ make em_sbc_imx8m_defconfig
$ make
$ cp -f spl/u-boot-spl.bin tools/imx-boot/iMX8M/
$ cp -f u-boot-nodtb.bin tools/imx-boot/iMX8M/
$ cp -f arch/arm/dts/em-sbc-imx8m.dtb tools/imx-boot/iMX8M/
$ cd tools/imx-boot/
$ make clean
$ make SOC=iMX8M flash_dds_em
$ cd ../../
$ cp -f ./tools/imx-boot/iMX8M/flash.bin u-boot.imx
```

编译完成后在\$HOME/01LinuxSourceCode/u-boot 目录下生成 u-boot.imx, 复制到 output/目录备用

2.2 Kernel

2.2.1 获取内核源码

内核源码存在\$HOME/01LinuxSourceCode 目录下,解压 linux*.tar.gz

```
$ tar -zxvf linux*.tar.gz
```

2.2.2 编译镜像

```
$ cd $HOME/01LinuxSourceCode/linux
$ make distclean
$ make em-sbc-imx8m_defconfig
$ make
```

编译完成后在

- ◆ \$HOME/01LinuxSourceCode/linux/arch/arm64/boot 目录下生成 Image
- ◆ \$HOME/01LinuxSourceCode/linux/arch/arm64/boot/dts/freescale/目录下生成
 - ◆ em-sbc-imx8m.dtb
 - ◆ em-sbc-imx8m-dcss-dsi.dtb
 - ◆ em-sbc-imx8m-dcss-lvds.dtb
 - ◆ em-sbc-imx8m-usb0-device.dtb

复制这些文件到 output 目录备用

2.2.3 编译模块

Linux Kernel 的镜像编译之后,需要编译一些模块文件:

```
$ make modules_install INSTALL_MOD_PATH=~/.output/modules/
```

编译后在~/.output/modules/生成 lib 文件夹,这个文件夹的内容会在后续烧写时使用。

MaaXBoard 标准版搭载 SD 卡,另有 eMMC 版本可控客户定制。关于 eMMC 版 MaaXBoard 的烧写过程,请参考《MaaXBoard EMMC 烧写指导》

3.1 分区

从空白的 SD 卡(或 eMMC)开始烧写之前,需要对 SD 卡或 eMMC 进行分区。如果你的 SD 卡(或 eMMC)已经烧写过出厂镜像,则可以跳过分区步骤。

3.1.1 对 SD 卡进行分区

将 SD 卡连接到 Linux 系统,如果系统自动挂载,如果 SD 卡被自动挂载,还需要输入下列命令取消挂载(sdb 用 Linux 系统下 SD 卡的实际编号代替)

```
$ sudo umount /dev/sdb*
```

对 SD 卡进行分区

```
$ echo -e "o\nn\np\n1\n20480\n+256M\na\n\t\nc\n\n\np\n2\n544768\n\nw\n" | sudo fdisk /dev/sdb  
$ sudo mkfs.vfat /dev/sdb1  
$ sudo mkfs.ext4 /dev/sdb2
```

3.2 烧写 U-Boot

3.2.1 烧写到 SD 卡

(sdb 用 Linux 系统下 SD 卡的实际编号代替)

```
$ sudo dd if=u-boot.imx of=/dev/sdb bs=1k seek=33 conv=fsync
```

3.3 烧写 Kernel

3.3.1 烧写镜像到 SD 卡

将 Image 和 dtb 文件拷贝到 SD 卡的第一分区即可。

3.4 复制 uEnv.txt

uEnv.txt 是系统启动的配置文件

参照烧写 Kernel 文件的过程，将这两个文件复制到 SD 卡的第一分区。

uEnv.txt 内容：

```
fdt_file=em-sbc-imx8m.dtb
#fdt_file=em-sbc-imx8m-dcss-lvds.dtb
#fdt_file=em-sbc-imx8m-dcss-dsi.dtb
#fdt_file=em-sbc-imx8m-usb0-device.dtb
console=ttyMXC0,115200 console=tty1 fbcon=rotate:0
```

实际使用时根据启动的需求更改 fdt_file 参数的数值。

3.5 烧写 File system

3.5.1 烧写 File system 到 SD 卡

用 dd 命令烧写 file system 的 img 文件到 SD 卡的第二分区，如：

```
dd if= debian-9.x-arm64-2019-02-28.img of=/dev/sdb2
```

3.5.2 复制模块文件

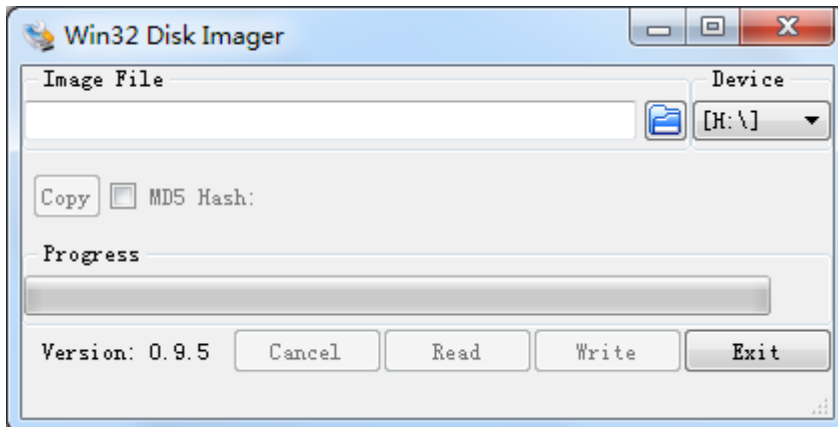
在之前的操作中，我们把模块文件编译到了/output/modules/目录，现在需要复制他们到文件系统中，执行下列命令：

```
$ mkdir ~/fs_mountpoint
$ sudo mount /dev/sdb2 ~/fs_mountpoint
$ sudo cp -af ~/output/modules/* ~/fs_mountpoint/
$ sudo sync
$ sudo umount /dev/sdb2
```

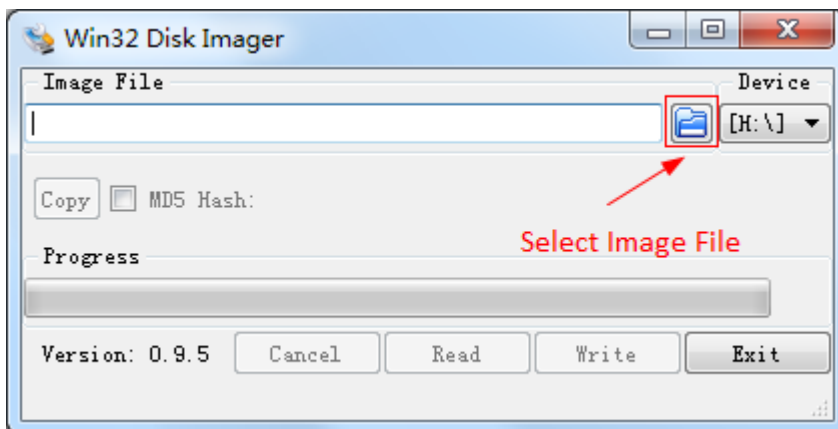
3.6 读取完整镜像

3.6.1 读取 SD 卡的镜像

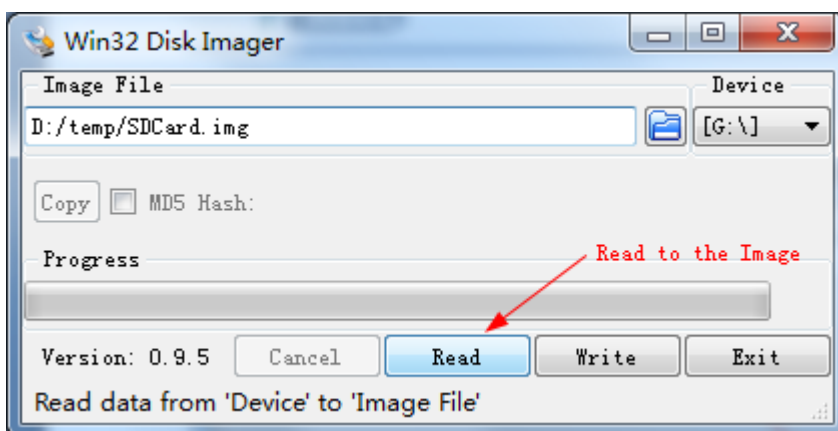
1. 将 SD 卡连接到电脑，运行 Win32 Disk Imager



2. 选择镜像文件的存放地址，如：D:/temp/SDCard.img



3. 点击 Read 将 SD 卡的内容读取到镜像文件中：



执行成功后，你会获得一个完整的 SD 镜像。

4.1 Hello world 的编译和测试

本节将提供一个简单示例的编写和编译方法，供开发者参考。

4.1.1 测试代码

```
embest@compiler:~/compile/test $ cat hello_world.c
#include <stdio.h>

void main(void)
{
    printf("hello world!\n");
}
```

4.1.2 makefile 文件（板载编译和交叉编译）

```
#cross compile tools path :
CROSS_COMPILE:=/home/embest/tool/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-
ARCH:= arm64
CC:=$(CROSS_COMPILE)gcc
LD:=$(CROSS_COMPILE)ld
#c files
objects = hello_world.o
# output executable file
hello_world: $(objects)
    $(CC) -O2 -static -o hello_world $(objects)
clean:
    @rm -vf *hello_world *.o *~
```

4.1.3 编译和测试

编译时，执行 `make` 命令即可

```
embest@compiler:~/compile/test$ make
/home/embest/tool/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-gcc
-c -o hello_world.o hello_world.c
```

```
/home/embest/tool/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-gcc  
-O2 -static -o hello_world hello_world.o
```

编译成功后，将得到两个文件，.o 是过程文件，hello_world 是可执行文件

```
embest@compiler:~/compile/test$ ls hello_world*  
hello_world hello_world.c hello_world.o  
embest@compiler:~/compile/test$ file hello_world  
hello_world: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), statically linked, for  
GNU/Linux 3.7.0, BuildID[sha1]=c5fa2e21b75b6a8afdaa0e4f581bfa9c807bd9bb, with debug_info,  
not stripped
```

5.1 硬件

详细硬件介绍请参考《MAAXBOARD 硬件用户手册》

6.1 技术支持

英蓓特科技对所销售的产品提供一年的免费技术支持服务，技术支持服务范围：

- ◆ 提供英蓓特科技嵌入式平台产品的软硬件资源；
- ◆ 帮助用户正确地编译和运行我们提供的源代码；
- ◆ 用户在按照本公司提供的产品文档操作的情况下，如本公司的嵌入式软硬件产品出现异常问题，我们将提供技术支持；
- ◆ 帮助用户判定是否存在产品故障。
- ◆ 以下情况不在我们的免费技术支持服务范围内，但我们将根据情况酌情处理：
 - ◆ 用户自行开发中遇到的软硬件问题；
 - ◆ 用户自行修改嵌入式操作系统遇到的问题；
 - ◆ 用户自己的应用程序遇到的问题；
 - ◆ 用户自行修改本公司提供的软件代码遇到的问题。

6.2 保修服务

- ◆ 产品自出售之日起，在正常使用状况下为印刷电路板提供 12 个月的免费保修服务；
- ◆ 以下情况不属于免费服务范围，英蓓特科技将酌情收取服务费用：
 - ◆ 无法提供产品有效购买凭证、产品识别标签撕毁或无法辨认，涂改标签或标签与实际产品不符；
 - ◆ 未按用户手册操作导致产品损坏的；

- ◆ 因天灾 (水灾、火灾、地震、雷击、台风等) 或零件之自然耗损或遇不可抗力导致的产品外观及功能损坏;
- ◆ 因供电、磕碰、房屋漏水、动物、潮湿、杂 / 异物进入板内等原因导致的产品外观及功能损坏;
- ◆ 用户擅自拆焊零件或修改而导致不良或授权非英蓓特科技认可的人员及机构进行产品的拆装、维修, 变更产品出厂规格及配置或扩充非英蓓特科技公司销售或认可的配件及由此引致的产品外观及功能损坏;
- ◆ 用户自行安装软件、系统或软件设定不当或由电脑病毒等造成的故障;
- ◆ 未经授权渠道购得此产品者;
- ◆ 非英蓓特科技对用户做出的超出保修服务范围的承诺 (包括口头及书面等) 由承诺方负责兑现, 英蓓特科技恕不承担任何责任。
- ◆ 保修期内由用户发到我们公司的运费由用户承担, 由我们公司发给用户的运费由我们承担; 保修期外的全部运输费用由用户承担。
- ◆ 若板卡需要维修, 请联系技术支持服务部。
- ◇ 英蓓特科技公司对于未经本公司许可私自寄回的产品不承担任何责任。

- ◆ 电话: +86-755-33190846/33190847/33190848

- ◆ 邮箱:
 - ◆ 技术支持: support@embest-tech.com

 - ◆ 销售: chinasales@embest-tech.com

- ◆ 传真: +86-755-25616057

- ◆ 网站: <http://www.embest-tech.cn>

- ◆ 地址: 深圳市南山区留仙大道 4093 号南山云谷创新产业园山水楼 4 楼 B