

# SAM4A 大读卡机 快速使用指南

版本 1.05  
2015年8月26日  
苏州市永兴电子有限公司

本手册旨在为用户提供一个快速使用本读卡机的指南，在操作和给本读卡机编程前用户需要了解操作本读卡机的基本规范。

## 1. 硬件操作规范

-  ESD 防护

本产品对高静电势敏感，尽管大部分集成电路带有 ESD 防护，用户还是需要避免电路板接触高静电势，否则会损坏本读卡机。当接触电路板之前，应采取接地带或类似防护措施。

- 加电操作

在电路板处在带电状态时，除了即插即用设备，禁止任何改变电路结构的操作。因为这样的操作很可能造成电路板的损坏。这类可能造成损坏的操作包括但不限于：

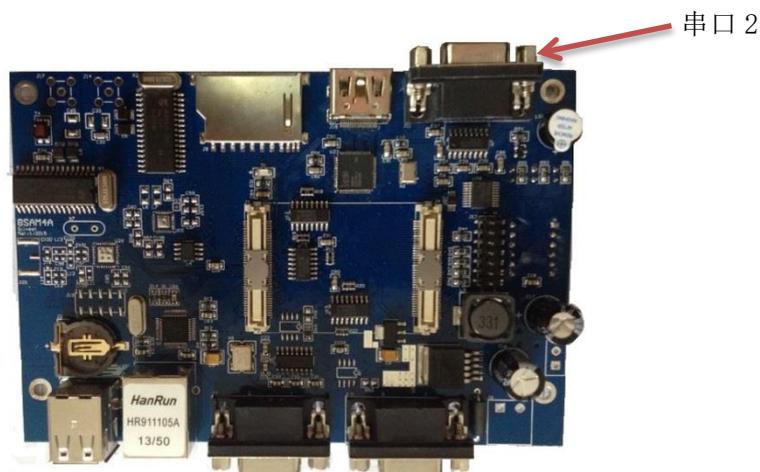
- 1) 在加电的电路板上插入 JTAG 调试器(正确的做法为将电路板断电，等待一段时间直到电容完全放电为止。在 JTAG 调试插座上插入调试器，给电路板重新加电)
- 2) 在加电的电路板上进行焊接操作。
- 3) 在加电的电路板上插入串口。严格的来说，串口是不允许热插拔的，对串口的热插拔有一定可能损坏电路。

- 绝对最大电源电压

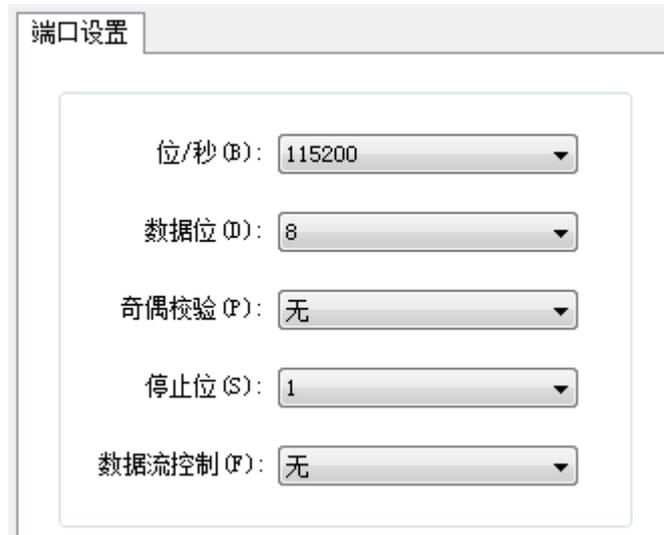
本读卡机的绝对最大电源电压为 20V，任何超出该电源轨的输入电压将损坏读卡机。工作电源电压为 12V，用户需连接一个 12V 直流稳压电源到读卡机的电源插座。该电源需要至少能够提供 0.5A 电流。考虑到浪涌等因素，输入电源范围允许在  $12V \pm 10\%$  以内。

## 2. 串口连接

将串口连接到读卡机的串口 2(插座编号 J11)上，如下图所示。



- 打开串口终端程序。在 Windows 操作系统下, 请使用[超级终端](#)。在 Linux 系统下, 请使用 [cu](#)(参考编程手册第 4.1 节)。串口 2 设置为波特率 115200, 8 位数据位, 1 位停止位, 无奇偶校验, 无数据流控制。



### 3. 电路板加电

- 连接一个 12V 直流稳压电源到读卡机的电源插座。如果电路板正常工作, 电源指示 LED (D6) 此时应该点亮。

### 4. 进入操作系统

在本读卡机加电后, 会自动加载 Uboot 和 Linux 内核, 进而进入 Linux 操作系统。

- 等待操作系统启动完毕, 出现 \$ 提示符

```
Welcome to SAM4A RFID Reader!  
Auto login user: root  
loader.sh : ./hello_world_arm starts now...  
!!!Hello World!!!  
root@SAM4A_RFIDReader:~# _
```

如果用户为首次启动机器, 可能出现 UBI 文件系统的初始化信息。请等待文件系统初始化完成, 单击回车即可。从第二次启动开始, 就不再需要这个初始化过程。

```
UBI: Initializing UBI system, left PEB: 6  
UBI: Initializing UBI system, left PEB: 5  
UBI: Initializing UBI system, left PEB: 4  
UBI: Initializing UBI system, left PEB: 3  
UBI: Initializing UBI system, left PEB: 2  
UBI: Initializing UBI system, left PEB: 1  
UBI: Initializing UBI system, left PEB: 0  
  
root@SAM4A_RFIDReader:~# _
```

注意到系统现在已经自动登录到 sam4a\_rfidreader 用户。在用户的 home 目录下有该读卡机的驱动程序，API 接口库，测试程序等。

```
root@SAM4A_RFIDReader:~# ls
autologin_arm  loader.sh          sam4a_spi.ko      spi_test_arm
hello_world_arm  readme.txt        sam4a_test_arm
```

## 5. 运行测试程序

输入 `./sam4a_test_arm` 执行测试程序

测试程序将给出当前读卡机的各种信息，如下图所示。我们在开发包中提供了测试程序的源代码，位于 SAM4A 开发包 sam4a\_test\_arm 工程内。

```
root@SAM4A_RFIDReader:~# ./sam4a_test_arm
=====
SAM4A RFID Reader Test Program
=====

Part 1 Hardware info:
-----
CPU Architecture: ARM Cortex-A8
CPU ID: AM3352
Board Name: SAM4A RFID Reader
PCB Version: 1.0
Core PCB Version: 1.0
Driver Version: 1.03
Manufacture Date: 2015/4/8
FERAM Available Size: 16 KBytes
UBoot Version: U-Boot 2014.07
UBoot Compile Date: Mar 29 2015
Board UID: 0x0287C5E85842DFC45FD9A878B99A891
  HF Channel 1: MFRC531
  HF Channel 2: MFRC531
```

测试程序接下来将依次测试 GPIO 函数，串口函数，FERAM 函数，LED 函数，日志函数。在测试 GPIO 时，将检查按钮 J21 是否按下。在测试串口函数时，将向串口 0 (J9) 发送 64 字节 'A' 持续 5 秒。

```
Part 2 button,buzzer,GPIO:
-----
```

```
To test button: Press "erase" button(J21)
Press 'y' to test button, 's' to skip:y
Checking button :Button is released!
```

```
GPIO pins are in input mode, GPIO1=0 GPIO2=0
```

```
Part 3 Serial:
-----
```

```
To test serial port: connect to UART0(J9), baudrate
Press 'y' to test serial, 's' to skip:y
Sending character A for 5 seconds...
Sending character A ends now
```

#### Part 4 FERAM:

```
-----  
FERAM Size: 16 KBytes  
Read 64 bytes from address 0x0000: 0x8C5907367B9B94D4  
12F10550FE2C9C8179D5471A48BEC3F1FC15002662F90995EF9F4  
0A3A9  
Write 64 bytes random data from address 0x0000: 0xBFE  
8F368FC68BF46FFF1EB3DEFBBBCAD89289C3CD8033C05BE4D1441  
45BCBC008794921057  
Read 64 bytes data from address 0x0000: 0xBFEA3BF670F  
8BF46FFF1EB3DEFBBBCAD89289C3CD8033C05BE4D1441132CD1E1  
8794921057
```

#### Part 5 LED:

```
-----  
7 segment LED displays "1234", LED blinking...
```

#### Part 6 Log:

```
-----  
Writing to log file /home/root/log/sam.log
```

**执行 SAM 卡测试** 首先插入两张 SAM 卡到卡槽 1 和 2，并输入 y 执行测试。这两张 SAM 要求如下：

卡槽	插座编号	复位波特率	PTS 波特率	符合要求的典型 SAM 卡
SAM1 (SVT)	U15	38400	不执行 PTS	上海地铁交通 PSAM 卡
SAM2 (SJT)	U17	9600	38400	上海地铁手机 SAM 卡

对 SAM 卡的测试主要包括两个命令，Get Challenge 获取随机数命令和 Select File 选择文件命令。下图显示了测试结果，获取随机数命令返回了 4 字节随机数，而选择文件命令返回了 MF 文件 (0x3F00) 的信息。

```
Reset SAM card(Slot2)  
ATR(Slot2,18 bytes): 0x3B7D94000057443751908693F03E9  
Sending get challenge command to SAM card(Slot2)  
Get challenge reply(Slot2,4 bytes): 0x44563243  
SW: 0x9000  
  
Sending select MF file command to SAM card(Slot2)  
  
Get MF(Slot2,46 bytes): 0x6F2C840E315041592E5359532E  
210000000000000056801011F00092100000056  
SW: 0x9000
```

**注意：**不满足要求的 SAM 卡可能测试失败，例如，复位波特率为 9600 的 SAM 卡如果在卡槽 1 中，则不能正常操作。有些卡片不支持 38400 波特率，或是不支持 PTS，这种卡片如果在卡槽 2 中，则不能正常操作。

## ☑ 执行读卡测试

在完成测试 SAM 卡后，将开始非接触卡读写测试。此时，在两个射频通道 1 (SMA 插座 J14) 和射频通道 2 (SMA 插座 J17) 上插入天线插头，通过同轴电缆连接到天线。射频插座的位置定义，请参考《SAM4A 使用手册》第 2.5 节。SMA 插座外形如下图。



测试程序将首先测试 Mifare 卡。它首先读取天线 1 上 Mifare 卡的第 1 个扇区的第一个块，即卡片的 UID。然后将随机数据写入块 2，再读取块 2，以验证写入是否正确。接下来，测试程序读取天线 2 上 Mifare 卡的第 13 块，并写入随机数据再读出以验证是否写入成功。

**注意：**用于测试的 Mifare 卡必须使用空卡，即密钥为 0xFFFFFFFF 的 M1 卡。

### Part 8 RFID:

```
To test M1 card: put 2 Mifare cards on both antennas
Mifare cards use default key: 0xFFFFFFFF
Press 'y' to test Mifare card, 's' to skip:y
```

接下来，测试程序将测试 CPU 卡。它首先对天线 1 上的 CPU 卡进行寻卡操作，然后执行 RATS 命令和 PPS 命令，PPS 设置通信速率为 106KBps。如果卡片 ATS 正确，则测试程序将执行两个命令：Get Challenge 获取随机数命令和 Select File 选择文件命令。通常情况下，两个命令返回数据的最后两个字节为 0x9000，即操作成功。

```
To test CPU card: put 2 CPU cards on both antennas
Press 'y' to test CPU card, 's' to skip:y
```

测试程序最后会测试 Ultralight 卡，在进行寻卡后，Ultralight 卡已经处在 Active 状态。测试程序首先读取前 3 个块的数据，然后尝试进行密钥认证。如果 Ultralight 卡的密钥 (PWD) 为 4 字节 0xFF，则认证成功。测试程序会对块 5 进行读写操作。

```
To test ultralight card: put 2 ultralight cards on both
Press 'y' to test ultralight card, 's' to skip:y
```

测试程序最后测试的是 TypeB 卡，测试程序将尝试对 User Zone1 的前 8 个字节进行读写。

```
To test TypeB card: put 2 TypeB cards on both antennas
Press 'y' to test TypeB card, 's' to skip:y
Request data: 0xFFFFFFFFFFFF22001051
Sending attribute command...
Card is attributed.
Setting user zone
User zone is 1
Read data from user zone 1,addr 0: 0x548FB30C53B9BBE9
Writing random data to addr 0 : 0x684BB0898C50040C
Read data from user zone 1,addr 0: 0x684BB0898C50040C
```

## 6. 开发软件说明

文件 `ti-sdk-am335x-evm-08.00.00.00-Linux-x86-Install.bin` 为德州仪器提供的 AM3352 处理器开发包，包含了 ARM GCC 交叉编译链。用户需要安装此开发包以进行编程。该文件也可以在德州仪器的网站上[下载](#)。

文件 `SAM4A 读卡机软件开发包(版本 X.XX).rar` 为我司提供的软件开发包，其中：

- binary 文件夹包含了编译好的二进制文件，主要包括
  - uBoot 引导程序     `u-boot.bin` `u-boot-spl.bin`
  - Linux 镜像           `zImage.bin`
  - 根文件系统           `rootfs.ubi` `rootfs_standard.ubi`
  - demo 程序           `dwmtest_arm` `hello_world_arm` `sam4a_test_arm`
  - Linux 驱动           `sam4a_spi.ko` `sam4a_uart.ko`
  - 读卡机 API 库       `libsam4a_driver_arm.so`
- document 文件夹包含了用户需要的文档，芯片手册和驱动库手册
- driver 文件夹包含：
  - `sam4a_driver_arm` 文件夹   读卡机 API 库，头文件以及对应的动态库
  - `sam4a_test_arm` 文件夹   demo 程序的源代码和 eclipse 工程文件
- linux-sam4a 文件夹包含 Linux 系统源代码
- script 文件夹包含了 Linux 系统需要的脚本文件
- sysroot 文件夹包含了根文件系统使用的开发环境
- u-boot-sam4a 文件夹包含 uboot 源代码
- utility 文件夹包含
  - `eclipse-cpp-luna-SR2-linux-gtk-x86_64.tar.gz`  
Linux 环境下 Eclipse 开发软件
  - `putty.exe`    终端软件
  - `SocketServer.exe`   套接字测试软件
  - `Tftpd64-4.50-setup.exe`   tftp 服务器安装软件

## 7. 应用程序接口库

libsam4a\_driver\_arm.so 是一个读卡机各个子系统的驱动动态库，用户可以动态连接到该库，并直接调用其函数。出于兼容性考虑，该动态库提供了两组函数声明，用户可以使用任何一组，函数的实现方法是完全相同的。推荐用户使用以 SAM4A\_开头的函数声明。关于该动态库的链接和编译使用，请参考《SAM4A 编程手册》第 3 章。

当前版本	1.05
功能	<b>暂不支持 HDMI 接口 / Zigbee 无线网络 / ISO14443 Type C 卡操作</b>
版本历史 (pre-beta 除外)	
1.00	修改了 RSA 加密函数接口 将需要链接到 RSA 的驱动库单独编译为 libsam4a_driver_rsa_arm.so
1.01	增加了硬件参数函数，可以读取当前硬件版本信息 FERAM 改用 FM25V10 芯片，尺寸增加到 128K 字节
1.02	修改了 GPIO 驱动，去除原 sam4a_gpio.ko 驱动，在 Linux 系统下 GPIO 操作将由 GPIOLib 完成。
1.03	去除看门狗函数，看门狗的功能将由 Linux 内部看门狗驱动完成
1.04	修改了 SAM 卡时序，使其支持大部分卡 修改了 UART 波特率参数，使 SAM 卡支持波特率 56000 添加了对 TypeB 卡的支持 系统自动挂载 USB 设备
1.05	添加了电源管理函数，硬件随机数生成函数 添加了厂商自动测试程序 更新了文件系统，现在支持原生 gcc 编译及远程 gdb 调试功能 更新了文件系统使用的库。glibc 库版本从 2.15 升级到 2.20 版本

**注意：**请参考编程手册 4.3.4 节更新系统，选择并下载需要的文件系统。

从 1.05 版本开始，由于更新了库版本，用户需要严格按照编程手册 3.4 和 3.5 节执行，替换原有工具链的库文件并重新编译程序。否则可能因库版本不匹配，导致程序无法运行或调试。