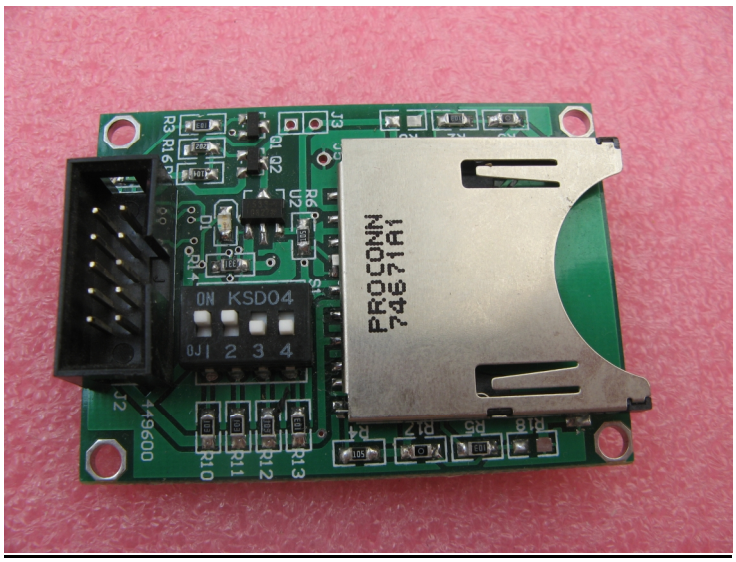
SD 卡读写模块使用说明 V6.00



2008 年 11 月 1 日发布　　2010-12-17 修改

北京安博尔康电子科技有限公司　　发布

1 描述.....1

2 功能.....1

3 通讯协议及命令格式.....1

3.1.1 通讯协议的描述.....1
3.1.2 串口数据底层格式描述.....2
3.2 命令的详细描述.....2
3.2.1 获取系统的状态命令 0x01.....2
3.2.2 创建文件命令 0x02.....2
3.2.3 创建文件夹命令 0x03.....3
3.2.4 保存文件命令 0x04.....3
3.2.5 写文件命令 0x05.....4
3.2.6 打开文件命令 0x06.....4
3.2.7 读文件 0x07.....5
3.2.8 关闭文件命令 0x08.....5
3.2.9 获取文件信息命令 0x09.....6
3.2.11 读 SRAM 0x0a.....6
3.2.12 写 SRAM 命令 0x0b.....7
3.2.13 更改目录命令 0x0c.....7
3.2.14 获取文件或目录名命令 0x0d.....8

4 模块接口说明.....10

附录 1 SD 模硬件接图示例.....11

附录 2 SD 问题解答.....11

附图 模块尺寸.....11

1 描述

SD 卡读写模块是北京安博尔康电子科技有限公司推出的一款模块，此模块整合 SD 卡规范和 FAT 文件格式规范，只要通过本模块规定的通讯协议就可以把数据存储在 SD 卡中的文件中。大家知道 SD 卡规范和 FAT 规范是非常复杂，如果在项目中要单独来写这两个规范的非常费时和费力，而其非常占用系统资源；现在的便携仪采集的数据种类越来越多，数据量越来越大，而其大部分要求在计算机上备份数据或者后期用计算机处理数据；而 SD 卡以其容量大，速度快，接口简单，加之配套的读卡器便宜而发展迅速；这些主观和客观的因素促使项目中迫切要求使用 SD 卡加 FAT 文件系统来存储数据，也促使本模块的诞生，这对电子工程师们来说是一个福音。

我们做到了：

稳定：工作稳定，不挑卡，不 s 机

方便：串口 UART 操作(直接接任何带串口单片机)，指令少，还可以顺序存

---

2 功能

本模块支持 FAT32 文件格式，理论支持 8G 以下 SD 卡。通过命令提供给主机有如下功能：

- 文件的创建（注：文件名只支持 8.3 文件格式：8.3 文件格式文件名不支持中文，文件名长度为最大 8 个字符）。
- 文件的打开（8.3 文件名格式）
- 文件的连续写入和文件的给定起始地址写入
- 文件的连续读取和文件的给定起始地址读取
- 当前打开文件的保存
- 当前文件的关闭
- 文件指针的设置
- 当前打开文件信息的读取，包括文件的大小和当前文件指针值
- 获取系统的状态（有无 SD 卡，是否为 FAT32 文件格式，系统是否繁忙）
- 通过模块上的拨码开关设置串口波特率（2400，9600，19200，57600，115200）

3 通讯协议及命令格式

3.1.1 通讯协议的描述

本模块的通讯协议分为命令发送和命令的应答两部分，其中命令格式由 4 个部分组成：命令识别码（0x55 0xaa），命令号，字节数（参数的个数，占 2 个字节，先发送低位字节，再发送高位字节），参数（根据命令的不同而不同），校验和（除命令识别码和校验和本身，所有发送数据之和的低 8 位数据）。命令格式如图：

0x55 0xaa	命令号	个数(2 字节，低字节在前)	参数（可选）	校验和

应答分为两部分：命令的执行情况（编码将附录 1），数据。数据根据命令的不同而不同。

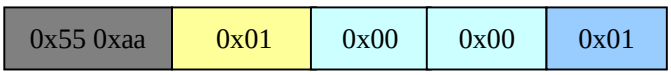
3.1.2 串口数据底层格式描述

模块串口传输一次数据为 10 位：1 位起始位（0），8 位数据（低位在先）及 1 位停止位（1）。在编程是要确认数据模式的真确以确保和模块之间数据的正确交换。

3.2 命令的详细描述

3.2.1 获取系统的状态命令 0x01

获取系统的状态命令是用来获取模块当前的状态。命令格式如下：



应答的数据为 1 个字节得状态信息，各位分别代表不同的状态：

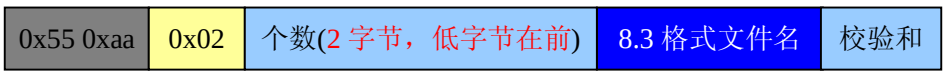
- Bit 0，SD 卡不存在状态，1 表示 SD 卡存在，0 表示 SD 不卡存在；（注意与其他命令不同）
- Bit 1，SD 卡写保护状态，1 表示 SD 卡写保护，0 表示不处在写保护；
- Bit 2，SD 卡初始化成功状态，1 表示 SD 卡初始化成功，0 表示 SD 卡初始化不成功
- Bit 3，文件系统类型，1 表示是 FAT 文件系统，0 表示非 FAT32 文件系统；（注意与其他命令不同）
- Bit 4，文件打开状态，1 表示当前有文件打开，0 表示当期没有文件打开；
- Bit 5，无定义；
- Bit 6，系统忙状态，1 表示系统正处在忙状态，0 表示系统空闲；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，判断第 6，7 位状态，都为 0，代表命令执行成功，状态信息见前 4bit；6、7bit 没有全部为 0，代表命令执行失败，前 4bit 状态信息无效。

例：55 AA 01 00 00 01　获取系统状态

3.2.2 创建文件命令 0x02

创建文件命令提供给主机创建文件的功能。参数为 N 字节 8.3 文件格式的文件名（**字符串格式，即文件名以 0 结尾**），即 8 字节的基本文件名（模块不支持汉字编码，字母不区分大小写），3 字节扩展名。命令格式如下：



应答的数据为 1 个字节的状态信息，各位分别代表不同的状态：

- Bit 0，SD 卡不存在状态，1 表示 SD 卡不存在；
- Bit 1，SD 卡写保护状态，1 表示 SD 卡写保护；
- Bit 2，文件打开状态，1 表示当前有文件打开，创建失败；
- Bit 3，FAT16 根目录满，1 表示根目录满，FAT16 根目录只能创建 32 个文件或文件夹；
- Bit 4，文件名格式，1 表示当前目录下有同名文件或者文件名格式不是 8.3 格式；
- Bit 5，文件系统类型，1 表示不为 FAT32 文件系统；
- Bit 6，系统忙状态，1 表示系统正处在忙状态；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，其中 8 位任何一位不为 0，表示命令执行失败，原因参考位标识。

例如要在 SD 卡中建立文件 12345678.txt,发送的命令数据为：

例：55 AA 02 0D 00 31 32 33 34 35 36 37 38 2E 74 78 74 00 41　创建 12345678.txt 文件

对于 FAT 协议来讲，文件数越多造成得寻址和判断（判断文件重命名）时间越来越长，使建立文件命令执行时间增长，所以在使用命令时一定要判断模块的 BUSY 脚，如果处于 BUSY 状态发送的后续命令就会被模块丢弃，直到不为 BUSY 状态时，后续命令才可被模块执行。

例如：按照状态命令（01）+创建命令（02）+打开命令（06）+写入数据命令（05)+保存命令（04）+关闭命令（08）这样的命令而不判断 BUSY，当创建到大概 800 多个文件时，创建命令时间变长，使得后续打开命令（06）丢失，而写入数据命令（05)+保存命令（04）因没有打开文件（打开命令丢弃没有执行），而执行失败，这样会出现创建好文件后，没有写入数据的现象，如果判断 BUSY 就能解决此问题。

类似问题还有保存命令，关闭命令。

3.2.3 创建文件夹命令 0x03

创建文件命令提供给主机创建文件夹的功能。参数 N 字节的文件夹名（字符串格式，即文件名以 0 结尾，字母不区分大小写），N 不能超过 8 字节。命令格式如下：

0x55 0xaa	0x03	个数(2 字节，低字节在前)	文件夹名	校验和

应答的数据为 1 个字节的状态信息，各位分别代表不同的状态：

- Bit 0，SD 卡不存在状态，1 表示 SD 卡不存在；
- Bit 1，SD 卡写保护状态，1 表示 SD 卡写保护；
- Bit 2，文件打开状态，1 表示当前有文件打开；
- Bit 3，FAT16 根目录满，1 表示根目录满（FAT16 根目录只能创建 32 个文件或文件夹）；
- Bit 4，文件名格式，1 表示当前目录下有同名文件夹或者文件夹名超过 8 字节；
- Bit 5，文件系统类型，1 表示非 FAT32 文件系统；
- Bit 6，系统忙状态，1 表示系统正处在忙状态；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，其中 8 位任何一位不为 0，表示命令执行失败，原因参考位标识。

3.2.4 保存文件命令 0x04

该命令为主机提供保存当前打开文件的功能，为了防止频繁写 SD 卡，每次送入模块的数据先是保存在模块的 512 字节的扇区缓冲中，所以为了防止数据丢失，完成所有数据的传输后，要发送保存文件命令来保存文件。命令格式如下：

0x55 0xaa	0x04	0x00	0x00	0x04

应答的数据为 1 个字节的状态信息，各位分别代表不同的状态：

- Bit 0，SD 卡不存在状态，1 表示 SD 卡不存在；
- Bit 1，SD 卡写保护状态，1 表示 SD 卡写保护；
- Bit 2，文件打开状态，1 表示无文件打开；

- Bit 3，无定义；
- Bit 4，文件系统类型，1 表示不为 FAT32 文件系统；
- Bit 5，系统忙状态，1 表示系统正处在忙状态；
- Bit 6，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，其中 8 位任何一位不为 0，表示命令执行失败，原因参考位标识。

记得文件要保存数据才可以完全写入 sd 卡中，文件信息（大小）等才可以更新

例：55 AA 04 00 00 04　保存文件

3.2.5 写文件命令 0x05

该命令为主机提供向已打开文件中写入数据的功能。每写一个数据文件指针自动加 1，当数据写完，文件指针指向最后一个数据地址加 1 的位置。命令格式如下，其中个数占 2 字节，低字节先发送，起始地址占 4 字节，低字节先发送：

0x55 0xaa	0x05	个数(2 字节)	起始地址（4 个字节）	有效数据	校验和

应答的数据为 1 个字节的状态信息，各位分别代表不同的状态：

- Bit 0，SD 卡不存在状态，1 表示 SD 卡不存在；
- Bit 1，SD 卡写保护状态，1 表示 SD 卡写保护；
- Bit 2，文件打开状态，1 表示无文件打开；
- Bit 3，磁盘状态，1 表示磁盘空间满，写入失败；
- Bit 4，参数个数状态，1 表示参数个数小于 4 个字节；
- Bit 5，文件系统类型，1 表示不为 FAT 文件系统；
- Bit 6，系统忙状态，1 表示系统正处在忙状态；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，其中 8 位任何一位不为 0，表示命令执行失败，原因参考位标识。

例：55 AA 05 0D 00 00 00 00 00 31 32 33 34 35 36 37 38 39 EF　向打开的文件写入 123456789 数据

- **如果你要连续写入文件，只要把写命令中的地址信息置最大值，0xffffffff 就可以了**
- **★★★有效数据≤200 个字节，个数 2 个字节是备用将来扩展**

3.2.6 打开文件命令 0x06

该命令为主机提供打开文件的功能。参数为 N 字节 8.3 文件格式的文件名（字符串格式，即文件名以 0 结尾），即 8 字节的基本文件名（模块不支持汉字编码，字母不区分大小写），3 字节扩展名。命令格式如下，其中个数占 2 字节，低字节先发送：

0x55 0xaa	0x06	个数(2 字节)	8.3 格式文件名	校验和

应答的数据为 1 个字节的状态信息，各位分别代表不同的状态：

- Bit 0，SD 卡不存在状态，1 表示 SD 卡不存在；
- Bit 1，文件打开状态，1 表示当前有文件打开，打开失败，先关闭文件在调用该命令；
- Bit 2，文件名状态，1 表示文件名不是标准的 8.3 文件格式；
- Bit 3，文件存在状态，1 表示无该文件；
- Bit 4，无定义；
- Bit 5，文件系统类型，1 表示不为 FAT 文件系统；

- Bit 6，系统忙状态，1 表示系统正处在忙状态；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，根据第 8 位判断命令执行状态，如果失败，原因参考前几位标识。

例：55 AA 06 0D 00 31 32 33 34 35 36 37 38 2E 74 78 74 00 45　打开 12345678.txt 文件

3.2.7 读文件 0x07

该命令为主机提供向当前打开的文件读取数据的功能。每读取一个数据文件指针自动加 1，当数据读完，文件指针指向最后一个数据地址加 1 的位置。命令格式如下，其中字节个数占 2 字节，低字节先发送，起始地址占 4 字节，低字节先发送，读取字节为本次希望读取的数据个数：

0x55 0xaa	0x07	个数(2 字节)	起始地址（4 个字节）	读取字节（2 个字节）	校验和

例：55 AA 07 06 00 00 00 00 00 09 00 16　读取文件 9 个数据

（55 AA）	（07）	（06 00）	（00 00 00 00）	（09 00）	（16）
帧头	命令	个数	起始地址段	数据个数	校验和

应答的数据分 4 部分，第一部分执行状态，第二部分位为实际读取的数据个数，占 2 字节，低字节先发送，第三部分为实际读取的数据，第四部分位校验和。后 3 部分只有当执行状态的第 8 位为 0 时才有（因为校验失败说明命令执行失败，也就不会有读出数据了），其应答格式如下：

执行状态	数据个数(2 字节)	有效数据	校验和

执行状态为 1 个字节，各位分别代表不同的状态：

- Bit 0，SD 卡不存在状态，1 表示 SD 卡不存在；
- Bit 1，文件打开状态，1 表示无文件打开，读文件失败；
- Bit 2，起始地址，1 表示起始地址超过文件大小；
- Bit 3，文件尾部，1 到了文件尾部，部分数据读取不成功；
- Bit 4，参数个数状态，1 表示参数个数小于 6 个字节；
- Bit 5，文件系统类型，1 表示不为 FAT 文件系统；
- Bit 6，系统忙状态，1 表示系统正处在忙状态；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，其中 8 位任何一位不为 0，表示命令执行失败，原因参考位标识。

数据个数：实际取得的数据个数，占 2 字节，先发送低字节。

有效数据：实际读取的数据

校验和：除校验和本身以外，所有发送数据之和的低 8 位。

3.2.8 关闭文件命令 0x08

该命令为主机提供关闭当前打开的文件的功能。在创建文件、创建文件夹、打开文件之前要求关闭当前打开的文件，才可以执行这些命令，否则返回失败。命令格式如下：

0x55 0xaa	0x08	0x00	0x00	0x08

应答的数据为 1 个字节的状态信息，各位分别代表不同的状态：

- Bit 0，SD 卡不存在状态，1 表示 SD 卡不存在；
- Bit 1，无定义；
- Bit 2，无定义；
- Bit 3，无定义；
- Bit 4，文件系统类型，1 表示不为 FAT 文件系统；
- Bit 5，系统忙状态，1 表示系统正处在忙状态；
- Bit 6，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，其中 8 位任何一位不为 0，表示命令执行失败，原因参考位标识。

3.2.9 获取文件信息命令 0x09

本命令为主机提供了读取当前打开文件的文件指针值和文件大小的功能。其命令格式如下：

0x55 0xaa	0x09	0x00	0x00	0x09

应答的数据分 2 部分，第一部分执行状态，第二部分为读取的文件信息，包括 4 字节的当前文件指针和 4 字节的文件大小，低字节先发送。其应答格式如下：

执行状态	文件指针(4 字节)	文件大小（4 字节）

执行状态为 1 个字节，各位分别代表不同的状态：

- Bit 0，SD 卡不存在状态，1 表示 SD 卡不存在；
- Bit 1，文件打开状态，1 表示当前有文件打开；
- Bit 2，无定义；
- Bit 3，无定义；
- Bit 4，无定义；
- Bit 5，文件系统类型，1 表示不为 FAT 文件系统；
- Bit 6，系统忙状态，1 表示系统正处在忙状态；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，其中 8 位任何一位不为 0，表示命令执行失败，原因参考位标识。

文件指针：当前打开文件的文件指针位置，4 个字节，低字节在前。

文件大小：当前打开文件的文件大小，4 个字节，低字节在前。

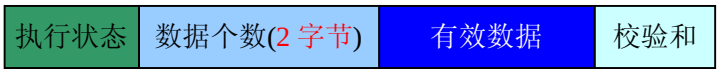
3.2.11 读 SRAM 0x0a

本模块为主机提供了 5K 大小的 SRAM 存储空间，可以为内存紧张的主机提供缓存。本命令为主机提供读取 SRAM 的功能。其命令格式如下：

0x55 0xaa	0x0a	个数(2 字节)	起始地址（2 个字节）	读取个数（2 个字节）	校验和

例：55 AA 0a 04 00 01 00 03 00 12 读 SRAM

应答的数据分 4 部分，第一部分执行状态，第二部分位为实际读取的数据个数，占 2 字节，低字节先发送，第三部分为实际读取的数据，第四部分位校验和。后 3 部分只有当执行状态的第 8 位为 0 时才有，其应答格式如下：



执行状态为 1 个字节，各位分别代表不同的状态：

- Bit 0，起始地址，1 表示起始地址超过 SRAM 大小；
- Bit 1，SRAM 尾部，1 表示要读取的部分数据地址超过了 SRAM 的大小；
- Bit 2，无定义；
- Bit 3，无定义；
- Bit 4，无定义；
- Bit 5，无定义；
- Bit 6，系统忙状态，1 表示系统正处在忙状态；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，其中 8 位任何一位不为 0，表示命令执行失败，原因参考位标识。

数据个数：实际取得的数据个数，占 2 字节，先发送低字节。

有效数据：实际读取的数据

校验和：除校验和本身以外，所有发送数据之和的低 8 位。

3.2.12 写 SRAM 命令 0x0b

该命令为主机提供向以向 5K 大小的 SRAM 写入数据的功能。命令格式如下，其中个数占 2 字节，低字节先发送，起始地址占 2 字节，低字节先发送：

例：55 AA 0b 05 00 01 00 55 AA 55 65　写 SRAM

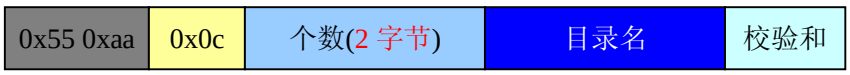
应答的数据为 1 个字节的状态信息，各位分别代表不同的状态：

- Bit 0，起始地址，1 表示起始地址超过 SRAM 大小；
- Bit 1，SRAM 尾部，1 表示只写入了部分数据，其他数据超过了 SRAM 大小；
- Bit 2，参数个数状态，1 表示参数个数小于 2 个字节；
- Bit 3，无定义；
- Bit 4，无定义；
- Bit 5，无定义；
- Bit 6，系统忙状态，1 表示系统正处在忙状态；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答，其中 8 位任何一位不为 0，表示命令执行失败，原因参考位标识。

3.2.13 更改目录命令 0x0c（打开文件夹或返回根目录）

该命令为主机提供更改工作目录的功能。参数为 N 字节目录名（目录名只能为 8 个字符。目录名可以包含路径信息，路径信息以‘\’开头且包含从根目录开始的路径信息，比如“\abc\123\example”，代表在根目录下 abc 目录下的 123 目录下 example 目录；或者不包含路径信息，只包含目录名，例如“example”，指系统在当前目录下的 example 目录。命令格式如下，其中个数占 2 字节，低字节先发送：



应答的数据为 1 个字节的状态信息，各位分别代表不同的状态：

- Bit 0，SD 卡不存在状态，1 表示 SD 卡不存在；
- Bit 1，文件打开状态，1 表示当前有文件打开，更改目录失败，先关闭文件在调用该命令；
- Bit 2，目录名状态，1 目录名不合法，目录名不是以 0 结尾的字符串或者路径信息有误；
- Bit 3，目录存在状态，1 表示要创建的目录不存在；
- Bit 4，无定义；
- Bit 5，文件系统类型，1 表示不为 FAT 文件系统；
- Bit 6，系统忙状态，1 表示系统正处在忙状态；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答为 00 表示执行命令成功，如果失败，原因参考前几位标识。

例：

55 AA 03 09 00 31 32 33 34 35 36 37 37 00 AF//创建文件夹命令 0x03

55 AA 0C 09 00 31 32 33 34 35 36 37 37 00 B8//进入文件夹命令 0x03

55 AA 0C 02 00 5C 00 6A//返回根目录操作

3.2.14 获取文件或目录名命令 0x0d

本命令为主机提供了获取当前目录下给定位置的文件名或者目录名的功能。其中位置信息占 1 个字节，表示要获取第几个文件名或者目录名，类型信息占用 1 个字节，表示将要获取的是文件名（类型值为 0）还是目录名（类型值为 1），其命令格式如下：

0x55 0xaa	0x0d	0x02 0x00	位置（1 字节）	类型（1 字节）	校验和

应答的数据分 4 部分，第一部分执行状态，第二部分位为获取的字符串个数，占 1 字节，第三部分为获得的字符串，第四部分位校验和。后 3 部分只有当执行状态为 0 时才有，其应答格式如下：

执行状态	数据个数(1 字节)	名字字符串	校验和

执行状态为 1 个字节，各位分别代表不同的状态：

- Bit 0，SD 卡不存在状态，1 表示 SD 卡不存在；
- Bit 1，文件打开状态，1 表示有文件打开，获取名失败；
- Bit 2，目录尾部，1 到了目录尾部，当前位置没有目录或者文件了（后面也没有文件或者目录了）；
- Bit 3，无定义；
- Bit 4，无定义；
- Bit 5，文件系统类型，1 表示不为 FAT 文件系统；
- Bit 6，系统忙状态，1 表示系统正处在忙状态；
- Bit 7，校验和状态，1 表示发送命令的校验和不正确；

注：接收到应答不为 0 表示命令执行失败，原因参考位标识。

数据个数：获取的字符串个数，占 1 字节

字符串名：获得的字符串信息

校验和：除校验和本身以外，所有发送数据之和的低 8 位。

例如：在当前目录下只有文件 ABCDEFGH.TXT 文件，那么获取第一个位置文件名将返回 ABCDEFGH.TXT，如果获取第二个位置处的文件将返回 0x04，代表到了目录尾部，没有文件了。

主机->模块：55 AA 0d 02 00 01 00 10　　注：获取第一个位置文件名

模块->主机：00 0C 41 42 43 44 45 46 47 48 2E 54 58 54 5E 注：主机返回 ABCDEFGH.TXT

主机->模块：55 AA 0d 02 00 02 00 11 注：获取第二个位置文件名

模块->主机：04 注：主机返回 0x04 代表到了目录尾部

模块操作数据包例子：

操作说明：下面数据包是在根目录下创建两个文件夹，并在两个文件夹内各创建一个文件。

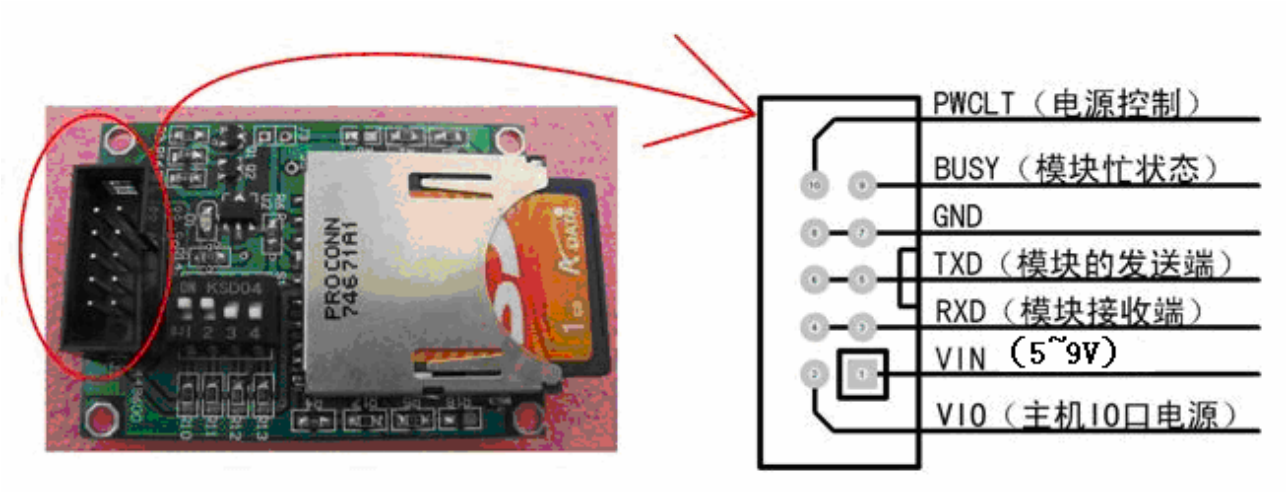
55 AA 01 00 00 01//获取系统的状态命令 0x01
55 AA 03 09 00 31 32 33 34 35 36 37 38 00 B0//创建文件夹命令 0x03
55 AA 0C 09 00 31 32 33 34 35 36 37 38 00 B9//进入文件夹命令 0x03
55 AA 02 0D 00 31 32 33 34 35 36 37 38 2E 74 78 74 00 41//创建文件命令 0x02
55 AA 06 0D 00 31 32 33 34 35 36 37 38 2E 74 78 74 00 45//打开文件命令 0x06
55 AA 05 0D 00 00 00 00 00 31 32 33 34 35 36 37 38 39 EF//写文件命令 0x05
55 AA 05 0D 00 0A 00 00 00 31 32 33 34 35 36 37 38 39 F9//再写文件命令 0x05
55 AA 04 00 00 04//保存文件命令 0x04
55 AA 08 00 00 08//关闭文件命令 0x08
55 AA 0C 02 00 5C 00 6A//返回根目录操作

55 AA 03 09 00 31 32 33 34 35 36 37 37 00 AF//创建文件夹命令 0x03
55 AA 0C 09 00 31 32 33 34 35 36 37 37 00 B8//进入文件夹命令 0x03
55 AA 02 0D 00 31 32 33 34 35 36 37 38 2E 74 78 74 00 41//创建文件命令 0x02
55 AA 06 0D 00 31 32 33 34 35 36 37 38 2E 74 78 74 00 45//打开文件命令 0x06
55 AA 05 0D 00 00 00 00 00 31 32 33 34 35 36 37 38 39 EF//写文件命令 0x05
55 AA 05 0D 00 0A 00 00 00 31 32 33 34 35 36 37 38 39 F9//再写文件命令 0x05
55 AA 04 00 00 04//保存文件命令 0x04
55 AA 08 00 00 08//关闭文件命令 0x08

4 模块接口说明

4.1 主机接口

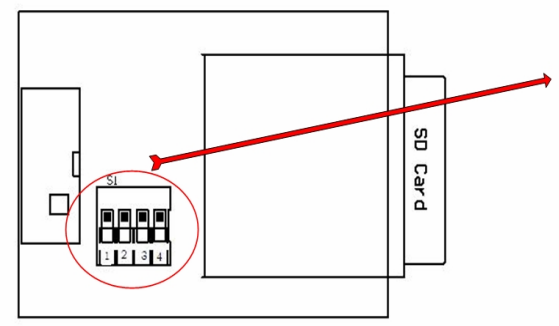
SD 模块图见附录 1， J2 为主机的接口，其引脚排序方式如下：



- VIN：模块供电电源，输入电压为 5V 到 9V 之间；
- VIO：为主机 IO 口电源，主机 IO 电压要与 VIO 一样；（目的是匹配电压，这就是模块适合 3.3V 或 5V 单片机使用）
- RXD：模块串口接收端，连接于主机串口的发送端；
- TXD：模块串口发送端，连接于主机串口的接收端；
- GND：地线端；
- BUSY：模块忙标识硬件端子，输出高电平时表示模块忙，输出低电平表示模块空闲；
- PWCTL：模块电源控制脚，高电平时开启模块电源，低电平时关闭模块电源。

4.2 波特率设置

S1 为 4 位短路电阻（原来为拨码开关，因拨码开关长期使用存在短路情况造成波特率变化，所以改为短路电阻），分别为第 1 位，第 2 位，第 3 位，第 4 位（见波盘开关上的数字标识），用来设置模块的波特率，0 欧电阻焊上为 1，不焊为 0，则 4 个 0 欧电阻焊盘组成波特率表如下：

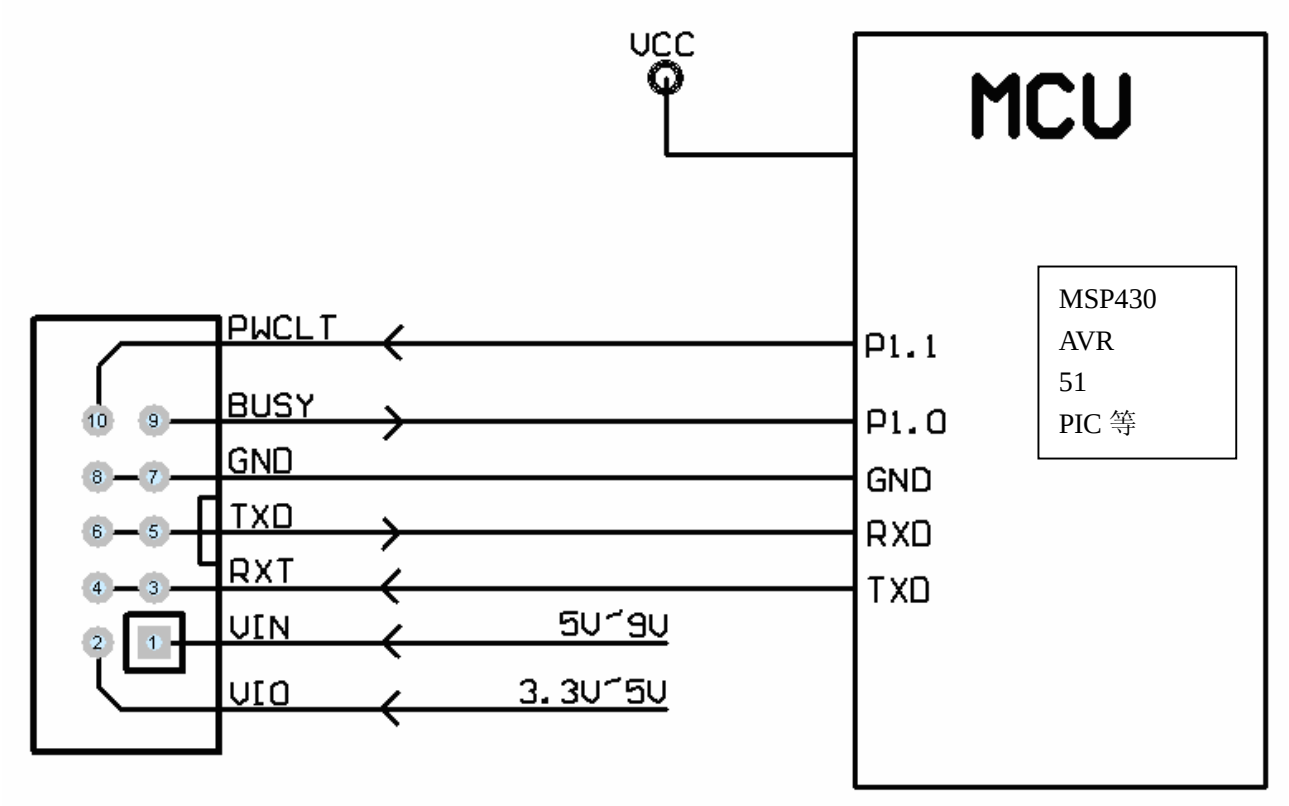


第 1 位	第 2 位	第 3 位	第 4 位	波特率
0	0	0	0	2400
1	0	0	0	9600
0	1	0	0	19200
1	1	0	0	57600
0	0	1	0	115200
模块出厂默认为 19200bps				

注：■如需更改波特率焊接短路电阻到相应位上就可以。

■每次设置好波特率后，要给模块复位，即断开模块电源后在开启模块电源。

附录 1 SD 模块与客户 MCU 接口图



说明：1.VIN 是模块电源，可以和用户单片机一致接 VCC，但要求在 5V~9V 间

2.VIO 是与模块匹配 UART 接口电压的接口，客户可能使用 3.3V 处理器也可能使用 5V 处理器，使用 3.3V 就在 VIO 接 3.3V，使用 5V 就在 VIO 接 5V，这样就能使接口（RXD,TXD,BUSY,PEWCLT）电平一致。可以接 VCC

北京安博尔康电子科技有限公司
联系方式：
网址：www.prog430.com
电话：13261892076
技术 QQ：37955698
邮箱：xzl@prog430.com

附录 2 问答

1、 问：怎样把数据连续写文件尾部？

答：方法 1：要想把文件连续的写入文件尾部，建议做法是用打开命令（0x06）打开文件，再用获取件信息命令（0x09）得到文件大小字段，然后再把文件大小值作为写入文件的起始地址调用写文件命令（0x05）写入数据就可以把数据连续写文件尾部。

方法 2：写数据时候自己建立写入地址表变量，写入数据后自己改变其偏移量，这样每次把该变量作为地址传递给模块就可以连续把数据写入文件尾部了。

方法 3：利用“模块在写入地址大于文件大小时，数据自动写入文件尾部”的特定，每次写入文件的传递给模块的地址信息读写为 0xFFFFFFFF,也即是最大值 4g，一般文件不肯能大于 4g，写没有写入数据都可以自动写入文件尾部，而不用关心起始地址信息。

2、 问：为什么我建立文件是使用的文件名有大小写，而建立文件后在电脑上显示的文件名全部为小写?

答：本模块使用 8.3 文件格式，建立的文件名不分大小写，在调用建立文件命令建立的文件在实际写的文件名都是以大写字母写入 SD 卡中。由于在 XP 系统下，识别 SD 卡中的文件为 8.3 文件格式，且文件名中全部为大写，XP 下自动显示文件名为小写，所以建议工程师们建立文件时使用小写字母或者小写字母和数字的组合方式的文件名。

3、 问：我新建一个文件后，不能向该文件写入数据？

答：新建文件后，模块并没有把该文件打开，而要写数据之前必须先打开文件。另外看你的 SD 卡是不是以 FAT32 格式化得，不要用 FAT 格式化。

4、 问：我向模块写了很多数据，之后我也用文件查询命令查询得到文件大小和地址偏移也改变了且是正确的，但是为什么我断电或者拔出卡后再上电或者插入卡，前面写入的数据没有了，文件大小也没有改变？

答：写入数据后，一定要发送文件保存命令才可以保存数据和文件信息到卡中。用文件查询命令得到的是缓存中的信息，但是这个信息一定要用文件保存命令才可以写入卡中，这样做是为了不频繁读写 SD 卡延长其寿命和提高模块的访问速度，所以建议在写入一定的数据量是建立调用文件保存命令保存数据到 SD 卡中。

5、 问：同一目录下文件数有没有限制？

答：FAT16 根目录下只能存储 32 个文件，按协议 FAT32 根目录下文件个数没有限制

6、 问：一个文件能存多大？

答：文件大小没有限制，但是为了处理和查看方便，数据量很大情况下最好按日期分成不同文件存储，比如每天存一个文件，这样也条理清晰。

7、 问：mini 卡能不能使用？

答：可以

8、 问：SD 卡长期存储数据应该存一个文件好还是多个文件好？

答：最好是建立多个文件，这样对于 SD 卡寿命有好处，但是建立文件多了会影响检索速度，对于建立文件这个命令会有影响，造成命令执行时间很长，所以命令发出后要判断 BUSY 位是否已经置低，再执行其他命令，否在会丢失命令和数据。

9、 文件存储按照什么流程最为合适？

答：SDV600 是内含处理器，整合 FAT 协议，操作起来和芯片一样，它工作状态也是有忙有闲，所以在执行操作时需要检查模块的 busy 标志引脚，客户处理器完全可以去做别的事，必要时判断 busy 进行下一步操作。我们画了个流程供客户参考如附图 1

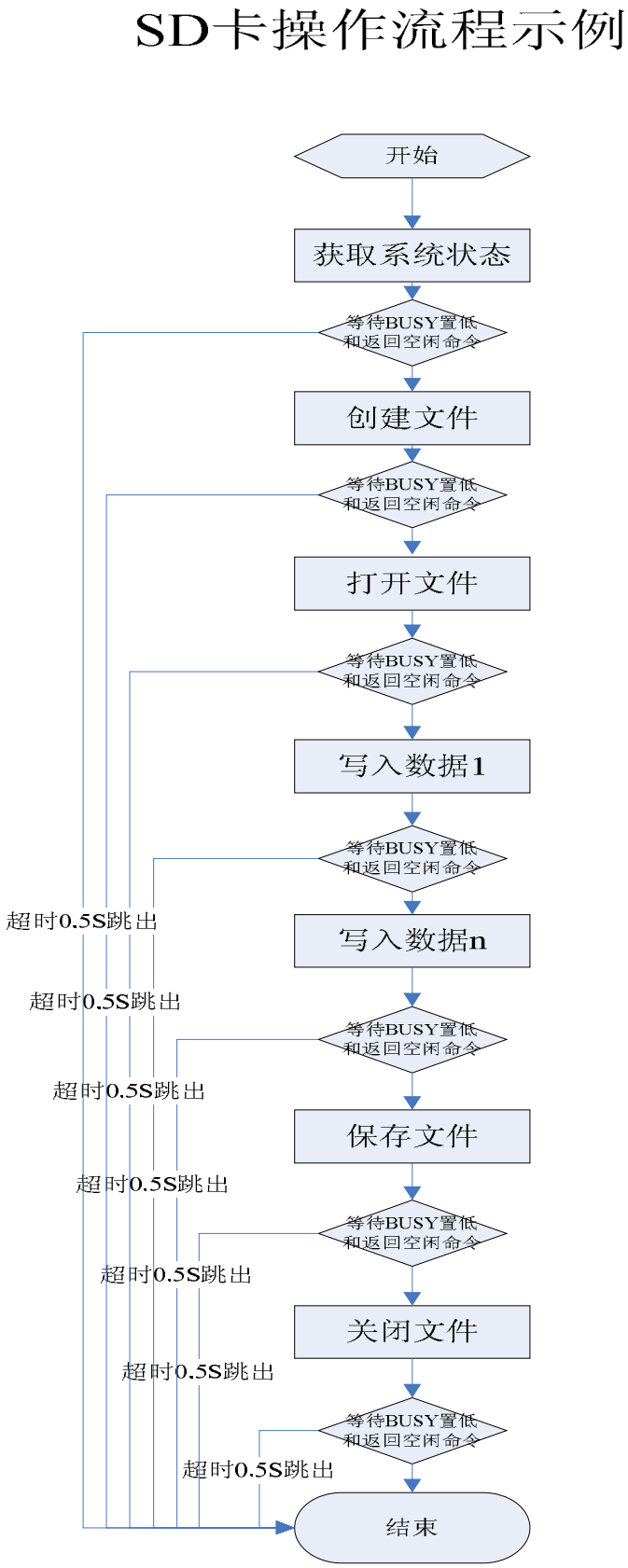
客户可以用我们计算机工具软件练习和测试，计算机测试软件地址如下：

http://www.prog430.com/files/SDV6_test.rar

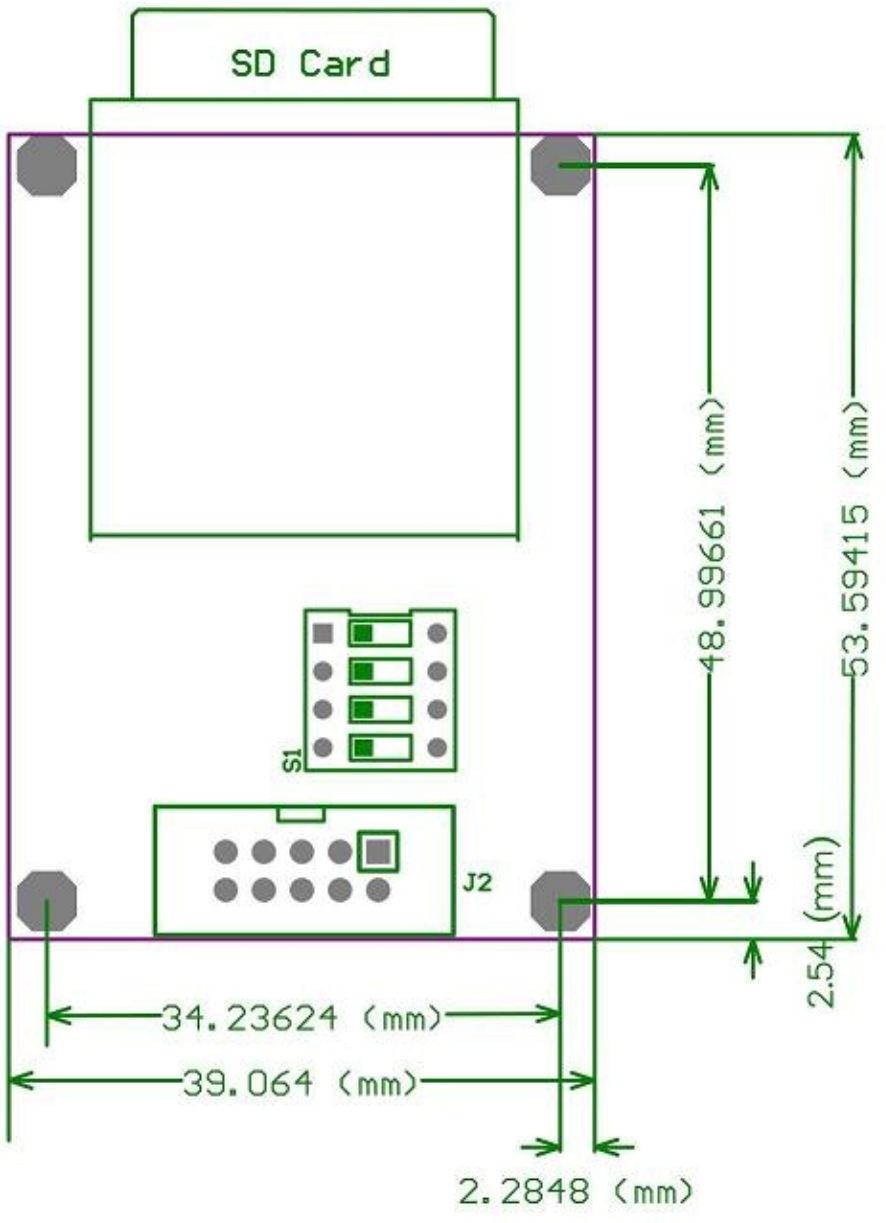
测试辅助模块驱动下载地址：

http://www.prog430.com/files/CP2102.rar

附图 1：模块操作流程示例



附图 2：模块尺寸



SDV600尺寸图