

德科物联

二代身份证&港澳证（护照）识别模块

DK26-SAMV

Datasheet

文档修订历史：

版本	日期	原因
V1.0.0	2019-5-4	创建文档
V1.1.1	2020-12-5	修改部分描述
V1.1.2	2021-04-28	增加 2D 图、接线端子规格说明

免责声明：

本文档提供有关深圳市德科物联技术有限公司产品的信息。本文档并未以暗示、禁止反言或其他形式转让本公司或任何第三方的专利、商标、版权或所有权或其下的任何权利或许可。除德科物联在其产品的销售条款和条件中声明的责任之外，本公司概不承担任何其它责任，并且，德科物联对其产品的销售或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。若不按手册要求连接或操作产生的问题，本公司免责。德科物联可能随时对产品规格及产品描述作出修改，恕不另行通知。对于本公司产品可能包含某些设计缺陷或错误，一经发现将收入勘误表，并因此可能导致产品与已出版的规格有所差异。如客户索取，可提供最新的勘误表。

在订购产品之前，请您与本公司联系，以获取最新的规格说明。

# 目录

1	概述.....	1
2	产品开发特性.....	1
2.1	电气参数.....	1
2.2	应用开发支持.....	1
3	硬件接口.....	2
3.1	模块尺寸与接口.....	2
4	UART 串口通信协议.....	4
4.1	软件流程.....	4
4.2	通讯传输字节格式.....	5
4.3	数据帧格式.....	5
4.4	命令码.....	6
4.4.1	通用指令.....	7
4.4.2	Mifare 卡操作指令.....	12
4.4.3	Ultralight 卡操作指令.....	16
4.4.4	ISO14443-A CPU 卡操作指令.....	18
4.4.5	身份证操作指令.....	19
4.4.6	ISO15693 卡片操作指令.....	21
5	模块反馈.....	23
6	本地身份证解析通讯协议.....	25
6.1	解析流程.....	25
6.2	通讯传输字节格式.....	25
6.3	数据帧格式.....	26
6.4	命令码及应答码.....	26
6.5	解析命令.....	27
7	护照读取.....	28
	第一步 发送激活认证护照指令（HEX 发送接收）：.....	28
	第二步 发送读取护照文件指令.....	28
8	常见故障分析.....	29
8.1	打开串口失败.....	29
8.2	与模块无法通信.....	29
8.3	读取不到卡片.....	29
9	技术支持.....	30

# 1 概述

DK26-SAMV 是一款支持对二代身份证、护照、港澳通行证等法定证件识别的模块，同时，还支持多种高频 IC 卡、NFC 标签卡片的识别。

此模块搭载了经公安部授权的本地 SAM 安全模组，支持识别二代身份证信息，同时还支持对出入境证件如护照、港澳通行证的识别。

模块采用 UART 通讯接口（可选 USB），兼容性好，通用性强。用户通过 UART 向模块发送简单指令可实现对卡片进行读/写操作，简单易用，用户可使用该模块高效快速开发高频读写功能产品。

DK26-SAMV 支持 ISO14443A、ISO14443B、IOS15693、ISO18092、Felica 等协议的卡片类型，支持 APDU 指令通道。

DK26-SAMV 模块采用天线-主控分离化设计，使天线设计更加灵活方便，支持定制天线。

# 2 产品开发特性

## 2.1 电气参数

【支持卡片】:	支持二代身份证的识别；支持护照、港澳通行证的识别；同时还支持 mifare one（M1）、S50、S70、UL:NTAG2xx 系列、复旦 F08、I-CODE-X 等类型标签、城市一卡通、银行卡的识别等
【身份证解析】	搭载本地 SAM 模块，可以读出身份证明文照片信息
【护照读取】	该模块具有护照及港澳通行证的读取功能，可读出证件中的身份信息与照片信息
【电压范围】:	5V（±0.5V）
【支持协议】:	ISO14443A、ISO14443B、IOS15693、ISO18092、Felica
【工作温度】:	-20~85℃
【通信接口】:	UART（预留 USB 接口）
【波特率】:	115200bps（默认）
【模块尺寸】:	主体模块：外形尺寸（52*51*10 mm），天线模块 4.8*2.8mm
【读卡距离】:	20mm~60mm（视标签与天线而定）
【看门狗】	内置硬件看门狗，杜绝死机；

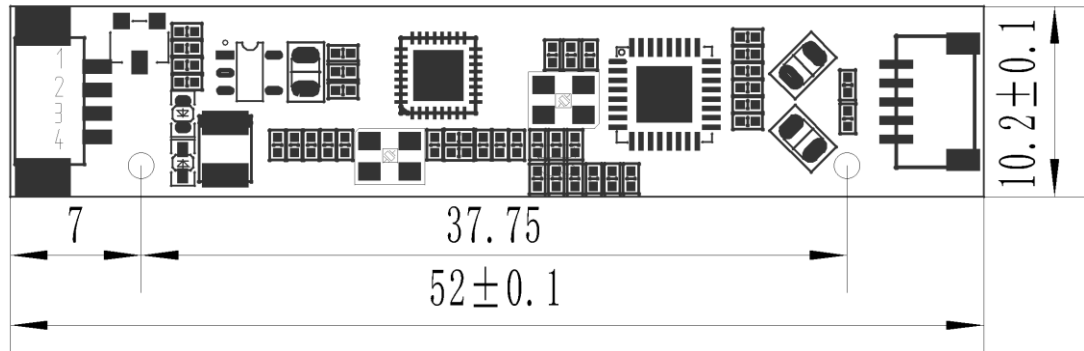
## 2.2 应用开发支持

提供安卓、Linux、windows 的 demo 支持，提供 SDK 以及 demo 的源码供设计参考



## 3 硬件接口

### 3.1 模块尺寸与接口



产品实物图（电路板面）

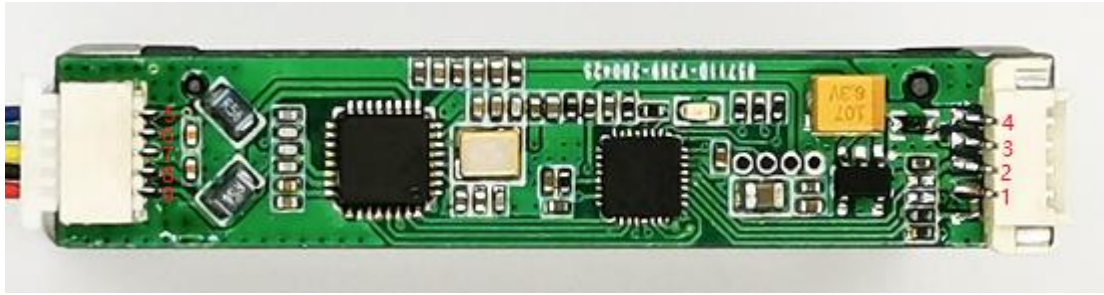


产品实物图（正面）

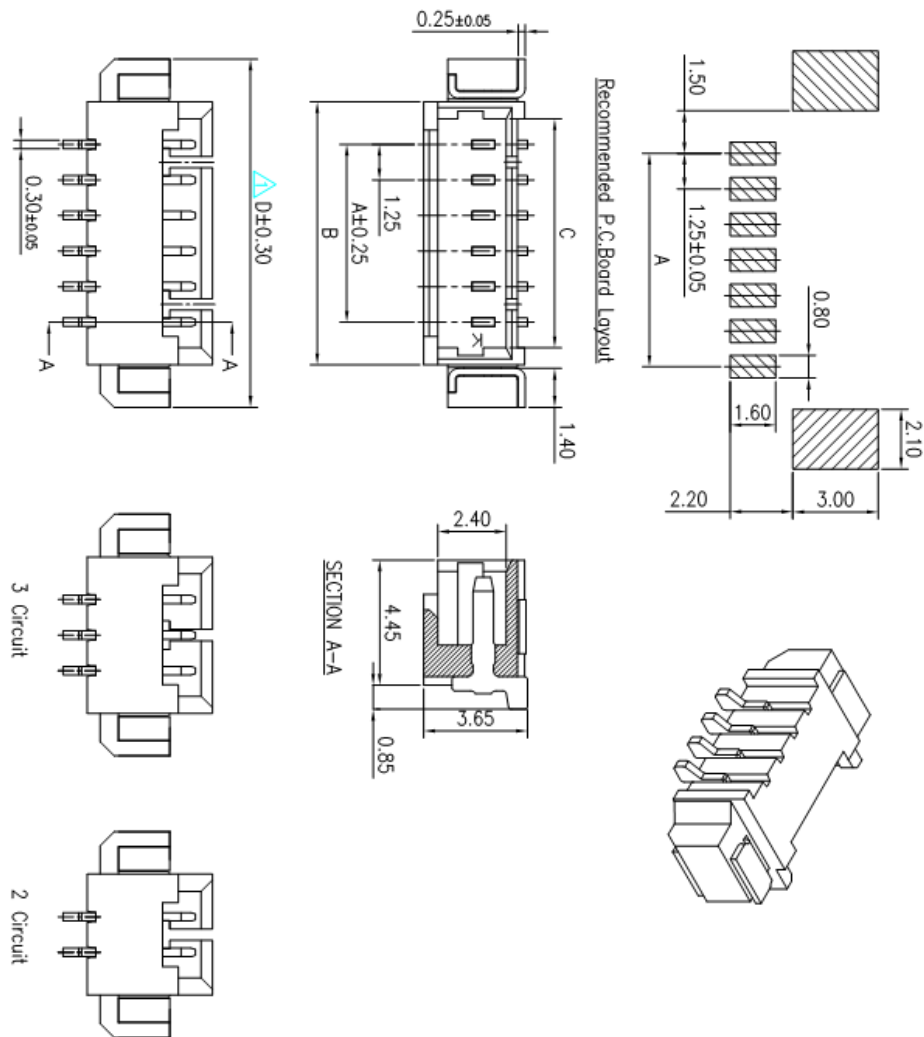


产品实物图（反面）





### 1.25 接线端子规格



接口定义:

DK26-SAMV 模块引脚说明		
引脚号	功能	说明
1	GND	电源地
2	RX/D+	模块串口接收/USB D+
3	TX/D-	模块串口发送/USB D-
4	5V	5V 电源
5	RX1	天线接收 1
6	TX1	天线发送 1
7	GND	天线地
8	TX2	天线发送 2
9	RX2	天线接收 2

## 4 UART 串口通信协议

### 4.1 软件流程

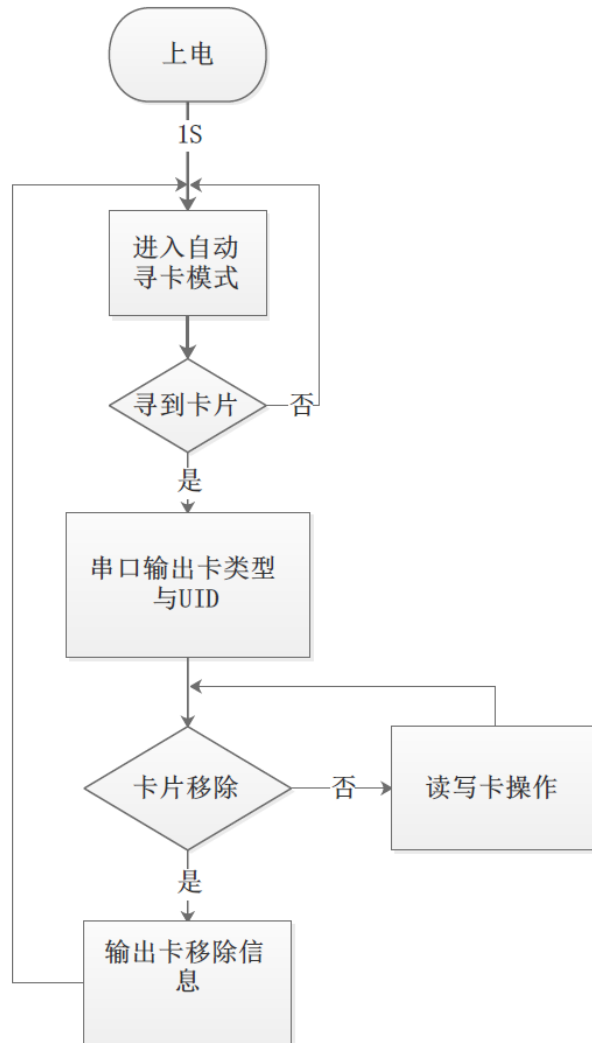
模块上电 1 秒钟后将会进入自动寻卡模式，寻到卡片自动从串口输出：卡片类型、卡片 UID 号，输出格式为（16 进制）：

**0xAA + 1 字节长度 + 0x01 + 1 字节卡片类型 + n 字节卡片 UID**

寻到卡片时模块的 STATUS 脚是高电平。当卡片拿开时，模块的 STATUS 脚会变化低电平，并从串口输出卡片离开指令，输出格式为（16 进制）：

**AA 01 EA**

软件流程如下图所示：



## 4.2 通讯传输字节格式

- 1 ) 通讯标准: UART
- 2 ) 波特率: 115200 bps
- 3 ) 起始位: 1bit
- 4 ) 数据位: 8 bits
- 5 ) 停止位: 1 bit
- 6 ) 校验: 无

## 4.3 数据帧格式

帧头 (0xAA) + 长度 (1Byte) + 命令 (1Byte) + 【地址 (1Byte)】 + 【数据 (N Byte)】

注: 【】表示指令中不一定会存在此部分, 即地址和数据不一定在所有的指令中都存在。



帧头：固定为 0xAA

长度：长度 = 命令长度（1Byte）+ 【地址长度（1Byte）】 + 【数据长度（N Byte）】

命令：命令分为卡片操作命令、错误反馈命令、模块配置命令

地址：卡片地址（M1 卡每个块数据 16 Byte，UL 卡每个块数据 4 Byte，ISO14443-B 卡每个块数据 4 Byte）

数据：在对应命令处有详细说明。

## 4.4 命令码

指令类型	命令码	说明
<a href="#">通用指令</a>	0x01	获取卡片 UID
	0x02	获取卡片类型
	0x95	自动寻卡开关
	0xA0	修改模块波特率
	0xA1	配置系统参数指令
	0xA2	读取系统参数指令
	0xB0	获取模块固件版本号
	0xB1	获取模块硬件版本号
<a href="#">Mifare 卡操作指令</a>	0x03	向模块写入需要验证的密钥(A 密钥)
	0x04	Mifare 卡读块
	0x05	Mifare 卡写块
	0x06	Mifare 卡增减值初始化
	0x07	Mifare 卡增值
	0x08	Mifare 卡减值
	0x0B	向模块写入需要验证的密钥(B 密钥)
	0x0C	设置模块使用密钥的类型
<a href="#">Ultralight 卡操作指令</a>	0x09	Ultralight 卡读块
	0x1C	Ultralight 卡读多个块
	0x1D	Ultralight 卡写多个块
<a href="#">ISO14443-A CPU 卡操作指令</a>	0x15	ISO14443-A CPU 卡片激活指令
	0x17	ISO14443-A CPU 卡 APDU 指令接口
	0x18	卡片断电指令、关闭天线指令接口
<a href="#">身份证卡片操作</a>	0x14	身份证激活指令
	0x16	身份证 APDU 指令接口
	0x18	卡片断电指令、关闭天线指令接口
<a href="#">ISO15693 卡片操作</a>	0x90	ISO15693 卡读单个块
	0x91	ISO15693 读多个块
	0x92	ISO15693 写单个块
	0x93	ISO15693 写多个块
	0x94	ISO15693 锁住块
<a href="#">模块反馈</a>	0xE0	卡类型错误反馈指令
	0xE1	未寻到卡错误反馈指令

	0xE2	密钥不匹配错误反馈指令
	0xE3	读块失败错误指令
	0xE4	写块失败错误指令
	0xE5	M1 卡值初始化失败错误指令
	0xE6	M1 卡增值失败错误指令
	0xE7	M1 卡减值失败错误指令
	0xFE	ACK 确认命令
	0xFF	NACK 否认命令

#### 4.4.1 通用指令

##### a) 获取卡片 UID

命令码 0x01

上位机->模块指令格式:

帧头	长度	指令码
0xAA	0x01	0x01

模块->上位机指令格式:

未寻到卡时返回:

帧头	长度	指令码
0xAA	0x01	0xE1

寻到卡时返回:

帧头	长度	指令码	卡片 UID
0xAA	UID 长度 + 1	0x01	4 - 8 字节

卡片 UID 长度:

卡片类型	UID 长度 Byte
ISO14443-A CPU 卡	4
ISO14443-B 协议卡	8
Mifare 卡 (M1)	4
ISO15693 协议卡	8
Ultralight 卡 (NFC 标签)	7
DF 卡	7

例:

向模块发送: AA 01 01

返回: AA 05 01 16 AB E1 C5

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 青色表示数据

本指令中，数据为卡片的 UID，本指令适用于所有的卡类型，其中 M1 卡 4 Byte UID、UL 卡 7 Byte UID，ISO14443-B 卡 8 Byte UID。

## b) 获取卡片类型（M1 或者 UL）

命令码 0x02

上位机->模块指令格式：

帧头	长度	指令码
0xAA	0x01	0x02

模块->上位机指令格式：

未寻到卡时返回：

帧头	长度	指令码
0xAA	0x01	0xE1

寻到卡时返回：

帧头	长度	指令码	卡片类型
0xAA	0x02	0x02	1 字节

卡片类型：

0x00	未定义
0x01	Mifare 卡
0x02	Ultralight 卡（NFC 标签）
0x03	ISO14443-B
0x04	ISO14443-A CPU 卡
0x05	ISO15693 协议卡

例：

向模块发送：AA 01 02

返回：AA 02 02 01

注：红色表示帧头，黄色表示指令长度，绿色表示指令，青色表示数据

本指令中，数据为卡片的类型，01 表示 M1 卡，02 表示 UL 卡，03 表示 ISO14443-B，04 表示 ACPU 卡，05 表示 ISO15693 卡

## c) 获取模块软硬件版本号

命令码 0xB0

上位机->模块指令格式：

帧头	长度	指令码
0xAA	0x01	0xB0

模块->上位机指令格式：

操作成功时返回：

帧头	长度	指令码	版本号
0xAA	0x02	0xB0	1 字节

例：

向模块发送: AA 01 B0

返回: AA 02 B0 20

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 青色表示数据

本指令中, 数据为版本信息

#### d) 打开/关闭自动寻卡指令

命令码 0x95

上位机->模块指令格式:

帧头	长度	指令码	打开/关闭自动寻卡	寻卡间隔	寻卡参数
0xAA	0x04	0x95	1 字节	1 字节	1 字节

说明:

打开/关闭自动寻卡: 0 为关闭自动寻卡, 非 0 为打开自动寻卡

**寻卡间隔:** 非低功耗自动寻卡间隔, 仅在 LP 为 0 时 (低功耗寻卡关闭) 有效。寻卡间隔转换关系公式为: 寻卡间隔 = SD \* 10 毫秒。默认值为 0x14, 对应寻卡间隔为 200 毫秒。

寻卡参数: 如下表所示

BIT	含义
Bit 0 – Bit 1	0: 未定义 1: 将 CPU 卡当成 M1 卡处理 2: 将 CPU 卡当成 CPU 卡处理 3: 未定义
Bit 2	0: 卡片离开不返还 AA 01 EA 1: 卡片离开时返回 AA 01 EA
Bit 3	0: 识别到身份证后, 不自动输出身份证明文 1: 识别到身份证后自动输出身份证明文
Bit 4	0: 不返回卡片类型 1: 返回卡片类型
Bit 5	0: 不返回银行卡卡号 1: 返回银行卡卡号
Bit 6	0: 不返回身份证特殊序列号 1: 返回身份证特殊序列号
Bit 7	0: 使用 106kbit/速率读取 CPU 卡 1: 根据卡片支持的速率选择最高速率读取 CPU 卡

模块->上位机指令格式:

操作成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE

例:

向模块发送: AA 04 95 FF FF 02

返回: AA 01 FE

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 青色表示数据

**注意：**打开或关闭自动寻卡后状态会保存到 EEPROM 中，掉电后重新上电模块会恢复当前状态。

### e) 配置模块波特率指令

模块配置命令中有串口波特率配置命令，串口波特率配置命令在发送命令后，下次上电或复位后生效，模块默认波特率为 115200。模块支持的波特率有 4800、9600、14400、19200、28800、38400、57600、115200 共 8 种速率。其对应的编码（编码在命令中使用）如下表所示：

波特率	对应编码
4800	1
9600	2
14400	3
19200	4
28800	5
38400	6
57600	7
115200	8

命令码 0xA0

上位机->模块指令格式：

帧头	长度	指令码	波特率编码
0xAA	0x02	0xA0	1 字节

模块->上位机指令格式：

操作成功时返回：

帧头	长度	指令码
0xAA	0x01	0xFE

例：配置串口波特率为 19200

向模块发送：AA 02 A0 04

返回(ACK)：AA 01 FE

注：红色表示帧头，黄色表示指令长度，绿色表示指令，青色表示数据

本指令中，数据为波特率对应的编码

### f) 配置系统参数指令

系统参数决定模块上电后的默认状态，配置后的参数将会保存到 EEPROM，掉电保存。

命令码 0xA1

上位机->模块指令格式：

帧头	长度	指令码	波特率	LPWSD	SD	SP	AUTO	LPW
0xAA	0x07	0xA1	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节

**波特率：**波特率编码请参考 [e\)配置波特率指令](#)，默认值为 0x08（对应 115200 波特

率)

**LPWSD:** 低功耗自动寻卡间隔编码, 仅在 LPW 为非 0 时 (低功耗寻卡开启) 有效。寻卡间隔编码如下表所示(默认值为 0x17, 对应 800 毫秒):

寻卡间隔	对应编码
10 毫秒	0x00
20 毫秒	0x01
30 毫秒	0x02
40 毫秒	0x03
50 毫秒	0x04
60 毫秒	0x05
70 毫秒	0x06
80 毫秒	0x07
100 毫秒	0x10
200 毫秒	0x11
300 毫秒	0x12
400 毫秒	0x13
500 毫秒	0x14
600 毫秒	0x15
700 毫秒	0x16
800 毫秒	0x17

注意: 寻卡间隔越小, 模块功耗越大。

**SD:** 非低功耗自动寻卡间隔, 仅在 LPW 为 0 时 (低功耗寻卡关闭) 有效。寻卡间隔转换关系公式为: 寻卡间隔 = SD \* 10 毫秒。默认值为 0x14, 对应寻卡间隔为 200 毫秒。

**SP:** 寻卡参数。请参考 [d\) 打开/关闭自动寻卡](#) 中的寻卡参数。默认值为 0x76。

**AUTO:** 自动寻卡开关。非 0 为开启自动寻卡, 0 为关闭自动寻卡。默认值为 0xFF, 开启自动寻卡。

**LPW:** 低功耗寻卡开关。非 0 为开启低功耗寻卡, 0 为关闭低功耗寻卡。默认值为 0xFF。

模块->上位机指令格式:

操作成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE

## g) 读取系统参数指令

命令码 0xA2

上位机->模块指令格式:

帧头	长度	指令码
0xAA	0x01	0xA2

模块->上位机指令格式:

操作成功时返回:

帧头	长度	指令码	波特率	LPWSD	SD	SP	AUTO	LPW
0xAA	0x07	0xA2	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节

#### 4.4.2 Mifare 卡操作指令

##### a) 写入 M1 卡 KEY A 密钥到模块

命令码 0x03

上位机->模块指令格式:

帧头	长度	指令码	密钥
0xAA	0x07	0x03	6 字节

模块->上位机指令格式:

操作成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE

例:

向模块发送: AA 07 03 ff ff ff ff ff

返回 (ACK): AA 01 FE

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 青色表示数据

本指令中, 数据为 M1 卡 KEY A 密钥, 共 6 byte。另外本指令指示保存密钥到模块中, 并不验证密钥。只有在读写 M1 卡的操作, 并且设置了模块密钥类型为 TYPE A (使用 M1\_SET\_KEY\_TYPE 指令设置) 才会使用此密钥去验证, 模块默认密钥为 ff ff ff ff

##### b) M1 卡读块指令

命令码 0x04

上位机->模块指令格式:

帧头	长度	指令码	块地址
0xAA	0x02	0x04	1 字节

模块->上位机指令格式:

未寻到卡时返回:

帧头	长度	指令码
0xAA	0x01	0xE1

操作成功时返回:

帧头	长度	指令码	块地址	块数据
0xAA	0x12	0x04	1 字节	16 字节

例: M1 卡读块 1 指令

向模块发送: AA 02 04 01

返回: AA 12 04 01 3E 9C 00 00 C1 63 FF FF 3E 9C 00 00 01 FE 01 FE

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 灰色表示地址, 青色表示数据

本指令中, 数据为 M1 卡一个块的数据, 共 16 byte

### c) M1 卡写块指令

命令码 0x05

上位机->模块指令格式:

帧头	长度	指令码	块地址	块数据
0xAA	0x12	0x05	1 字节	16 字节

模块->上位机指令格式:

未寻到卡时返回:

帧头	长度	指令码
0xAA	0x01	0xE1

操作成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE

例: M1 卡写块 4 指令

向模块发送: AA 12 05 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

返回 (ACK): AA 01 FE

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 灰色表示地址, 青色表示数据

本指令中, 数据为 M1 卡一个块的数据, 共 16 Byte

### d) M1 卡电子钱包初始化指令

命令码 0x06

上位机->模块指令格式:

帧头	长度	指令码	块地址	值
0xAA	0x06	0x06	1 字节	4 字节

模块->上位机指令格式:

未寻到卡时返回:

帧头	长度	指令码
0xAA	0x01	0xE1

操作成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE



例：M1 卡使用块 4 作为电子钱包，并且初始值为 1 元

向模块发送：AA 06 06 04 01 00 00 00

返回(ACK)：AA 01 FE

注：红色表示帧头，黄色表示指令长度，绿色表示指令，灰色表示地址 青色表示数据

本指令中，数据为 M1 卡电子钱包初始值，共 4 Byte，低 8 位在前

#### e) M1 卡电子钱包充值指令

命令码 0x07

上位机->模块指令格式：

帧头	长度	指令码	块地址	值
0xAA	0x06	0x07	1 字节	4 字节

模块->上位机指令格式：

未寻到卡时返回：

帧头	长度	指令码
0xAA	0x01	0xE1

操作成功时返回：

帧头	长度	指令码
0xAA	0x01	0xFE

例：M1 卡给块 4 的电子钱包充值 2 元

向模块发送：AA 06 07 04 02 00 00 00

返回(ACK)：AA 01 FE

注：红色表示帧头，黄色表示指令长度，绿色表示指令，灰色表示地址 青色表示数据

本指令中，数据为 M1 卡电子钱包要充值的数额，共 4 Byte，低 8 位在前

#### f) M1 卡电子钱包扣款指令

命令码 0x08

上位机->模块指令格式：

帧头	长度	指令码	块地址	值
0xAA	0x06	0x08	1 字节	4 字节

模块->上位机指令格式：

未寻到卡时返回：

帧头	长度	指令码
0xAA	0x01	0xE1

操作成功时返回：

帧头	长度	指令码
0xAA	0x01	0xFE

例：M1 卡给块 4 的电子钱包扣款 2 元

向模块发送：AA 06 08 04 02 00 00 00

返回(ACK)：AA 01 FE

注：红色表示帧头，黄色表示指令长度，绿色表示指令，灰色表示地址，青色表示数据

本指令中，数据为 M1 卡电子钱包要扣的数额，共 4 Byte，低 8 位在前

### g) 写入 M1 卡 KEY B 密钥到模块

命令码 0x0B

上位机->模块指令格式：

帧头	长度	指令码	密钥
0xAA	0x0B	0x03	6 字节

模块->上位机指令格式：

操作成功时返回：

帧头	长度	指令码
0xAA	0x01	0xFE

例：

向模块发送：AA 07 0B ff ff ff ff ff

返回 (ACK)：AA 01 FE

注：红色表示帧头，黄色表示指令长度，绿色表示指令，青色表示数据

本指令中，数据为 M1 卡 KEY B 密钥，共 6 byte。另外本指令指示保存密钥到模块中，并不验证密钥。只有在读写 M1 卡的操作，并且设置了模块密钥类型为 TYPE B（使用 M1\_SET\_KEY\_TYPE 指令设置）才会使用此密钥去验证，模块默认密钥为 ff ff ff ff ff ff

### h) 设置模块使用密钥的类型

命令码 0x0C

上位机->模块指令格式：

帧头	长度	指令码	密钥类型
0xAA	0x02	0x0C	1 字节

模块->上位机指令格式：

操作成功时返回：

帧头	长度	指令码
0xAA	0x01	0xFE

例：

向模块发送：AA 02 0C 0A

返回：AA 01 FE

注：红色表示帧头，黄色表示指令长度，绿色表示指令，青色表示数据

本指令中，数据为要设置的密钥类型，0x0A 表示 KEY A，0x0B 表示 KEY B，默认为 KEY A

### 4.4.3 Ultralight 卡操作指令

#### a) UL 卡读块指令

命令码 0x09

上位机->模块指令格式：

帧头	长度	指令码	块地址
0xAA	0x02	0x09	1 字节

模块->上位机指令格式：

未寻到卡时返回：

帧头	长度	指令码
0xAA	0x01	0xE1

读卡成功时返回：

帧头	长度	指令码	块地址	块数据
0xAA	0x06	0x90	1 字节	4 字节

例：UL 卡读块 1 指令

向模块发送：AA 02 09 01

返回：AA 06 09 01 3E 9C 00 00

注：红色表示帧头，黄色表示指令长度，绿色表示指令，灰色表示地址，青色表示数据

本指令中，数据为 UL 卡一个块的数据，共 4 Byte

#### b) UL 卡写块指令

命令码 0x0A

上位机->模块指令格式：

帧头	长度	指令码	块地址	块数据
0xAA	0x06	0x0A	1 字节	4 字节

模块->上位机指令格式：

未寻到卡时返回：

帧头	长度	指令码
0xAA	0x01	0xE1

写卡成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE

例: UL 卡写块 4 指令

向模块发送: AA 06 0A 04 00 01 02 03

返回 (ACK): AA 01 FE

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 灰色表示地址, 青色表示数据

本指令中, 数据为 UL 卡一个块的数据, 共 4 Byte

### c) UL 卡读多个块指令

命令码 0x1C

上位机->模块指令格式:

帧头	长度	指令码	起始块	结束块
0xAA	0x03	0x1C	1 字节	1 字节 (必须大于起始块)

模块->上位机指令格式:

未寻到卡时返回:

帧头	长度	指令码
0xAA	0x01	0xE1

读卡成功时返回:

帧头	长度	指令码	地址	数据
0xAA	块数量 x4 + 2 字节	0x1C	1 字节	块数量 x4 字节

例: UL 卡读取 0 到 0x30 块 (读出数据量为:  $(0x30 - 0) \times 4 = 192$  字节)

向模块发送: AA 03 1C 00 30

返回 (ACK): AA C6 1C 00 192 字节数据

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 灰色表示地址, 青色表示数据

### d) UL 卡写多个块指令

命令码 0x1D

上位机->模块指令格式:

帧头	长度	指令码	起始块	要写的的数据
0xAA	n	0x1D	1 字节	最大 240 字节

模块->上位机指令格式:

未寻到卡时返回:

帧头	长度	指令码
0xAA	0x01	0xE1

写卡成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE

例: UL 卡写两个块

向模块发送: AA 0A 1D 04 30 30 30 30 30 30 30 30 30

返回 (ACK): AA 01 FE

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 灰色表示地址, 青色表示数据

#### 4.4.4 ISO14443-A CPU 卡操作指令

##### a) ISO14443-A 卡激活指令

命令码 0x15

上位机->模块指令格式:

帧头	长度	指令码
0xAA	0x01	0x15

模块->上位机指令格式:

操作成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE

例:

向模块发送: AA 01 15

返回: AA 01 FE

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令

卡片激活后, 如果移开卡片 (卡片断电), 会返回 AA 01 EA。

##### b) ISO14443-A PDU 指令接口

命令码 0x17

上位机->模块指令格式:

帧头	长度	指令码	APDU 指令
0xAA	n+1 字节	0x17	n 字节

模块->上位机指令格式:

操作成功时返回:

帧头	长度	指令码	APDU 指令
0xAA	n+1 字节	0x17	n 字节

例:

向模块发送: AA 06 17 00 84 00 00 08

返回: AA 0B 17 3E 9C 00 08 1D 82 11 C1 90 00

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 青色表示数据

本指令中, 数据为接收到的 PDU 指令及指令的返回数据。

### c) 卡片断电指令

命令码 0x18

上位机->模块指令格式:

帧头	长度	指令码
0xAA	0x01	0x18

模块->上位机指令格式:

操作成功时返回:

帧头	长度	指令码
0xAA	0x01	0xEA

例:

向模块发送: AA 01 18

返回: AA 01 EA

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令

## 4.4.5 身份证操作指令

### a) ISO14443-B 卡激活指令

命令码 0x14

上位机->模块指令格式:

帧头	长度	指令码
0xAA	0x01	0x14

模块->上位机指令格式:

操作成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE

例:

向模块发送: AA 01 14

返回: AA 01 FE

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令

卡片激活后, 如果移开卡片 (卡片断电), 会返回 AA 01 EA

## b) ISO14443-B PDU 指令接口

命令码 0x16

上位机->模块指令格式:

帧头	长度	指令码	APDU 指令
0xAA	n+1 字节	0x16	n 字节

模块->上位机指令格式:

操作成功时返回:

帧头	长度	指令码	APDU 指令
0xAA	n+1 字节	0x16	n 字节

例:

向模块发送: AA 06 16 00 84 00 00 08

返回: AA 0B 16 3E 9C 00 08 1D 82 11 C1 90 00

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 青色表示数据

本指令中, 数据为接收到的 PDU 指令及指令的返回数据。

## c) 卡片断电指令

命令码 0x18

上位机->模块指令格式:

帧头	长度	指令码
0xAA	0x01	0x18

模块->上位机指令格式:

操作成功时返回:

帧头	长度	指令码
0xAA	0x01	0xEA

例:

向模块发送: AA 01 18

返回: AA 01 EA

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令

## 4.4.6 ISO15693 卡片操作指令

### a) 读取 ISO15693 卡片单个块数据

命令码 0x90

上位机->模块指令格式:

帧头	长度	指令码	块地址
0xAA	0x02	0x90	1 字节

模块->上位机指令格式:

未寻到卡时返回:

帧头	长度	指令码
0xAA	0x01	0xE1

读卡成功时返回:

帧头	长度	指令码	块地址	块数据
0xAA	0x06	0x90	1 字节	4 字节

例:

向模块发送: AA 02 90 00

返回: AA 06 90 00 3E 9C 00 00

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 灰色表示地址, 青色表示数据

数据表示读到的数据, 一个块 4 字节

### b) 读取 ISO15693 卡片多个块数据

命令码 0x91

上位机->模块指令格式:

帧头	长度	指令码	块起始地址	块数量
0xAA	0x03	0x91	1 字节	1 字节

模块->上位机指令格式:

未寻到卡时返回:

帧头	长度	指令码
0xAA	0x01	0xE1

读卡成功时返回:

帧头	长度	指令码	块地址	块数据
0xAA	块数据长度+2 字节	0x91	1 字节	块数量 x 4 字节

例:

向模块发送: AA 03 91 00 04

返回: AA 0A 91 00 3E 9C 00 3E 9C 00 00 3E 9C 00 00 3E 9C 00 00 00

注: 红色表示帧头, 黄色表示指令长度, 绿色表示指令, 灰色表示地址, 青色表示数据



其中向模块发送的数据表示要读取的块数量（每个块4个字节）

c) 写一个块数据到 ISO15693 卡片中

命令码 0x92

上位机->模块指令格式:

帧头	长度	指令码	块地址	块数据
0xAA	0x06	0x92	1 字节	4 字节

模块->上位机指令格式:

未寻到卡时返回:

帧头	长度	指令码
0xAA	0x01	0xE1

写卡成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE

例：

向模块发送: AA 06 92 00 01020304

返回 (ACK): AA 01 FE

注：红色表示帧头，黄色表示指令长度，绿色表示指令，灰色表示地址，青色表示数据

数据表示要写入卡片的数据，一个块 4 个字节

d) ISO15693 写多个块

命令码 0x93

上位机->模块指令格式:

帧头	长度	指令码	起始地址	块数量	要写的块数据
0xAA	n+3	0x93	1 字节	1 字节	块数量 x4 字节

模块->上位机指令格式:

未寻到卡时返回:

帧头	长度	指令码
0xAA	0x01	0xE1

写卡成功时返回:

帧头	长度	指令码
0xAA	0x01	0xFE

例：

向模块发送: AA 06 92 00 01020304

AA 13 93 00 04 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30

返回 (ACK): AA 01 FE

注：红色表示帧头，黄色表示指令长度，绿色表示指令，灰色表示地址，青色表示数据

地址后面的 04 表示要写入 4 个块的数据，后面的数据必须是 4x4=16 字节

#### e) 锁住 ISO15693 的一个块（变成只读，且不能修改）

命令码 0x94

上位机->模块指令格式：

帧头	长度	指令码	块地址
0xAA	0x02	0x94	1 字节

模块->上位机指令格式：

未寻到卡时返回：

帧头	长度	指令码
0xAA	0x01	0xE1

操作成功时返回：

帧头	长度	指令码
0xAA	0x01	0xFE

例：

向模块发送：AA 02 94 00

返回（ACK）：AA 01 FE

注：红色表示帧头，黄色表示指令长度，绿色表示指令，灰色表示地址

此条指令将块 0 锁住，变成只读，不能写入数据，一旦锁住将永远不能修改！

## 5 模块反馈

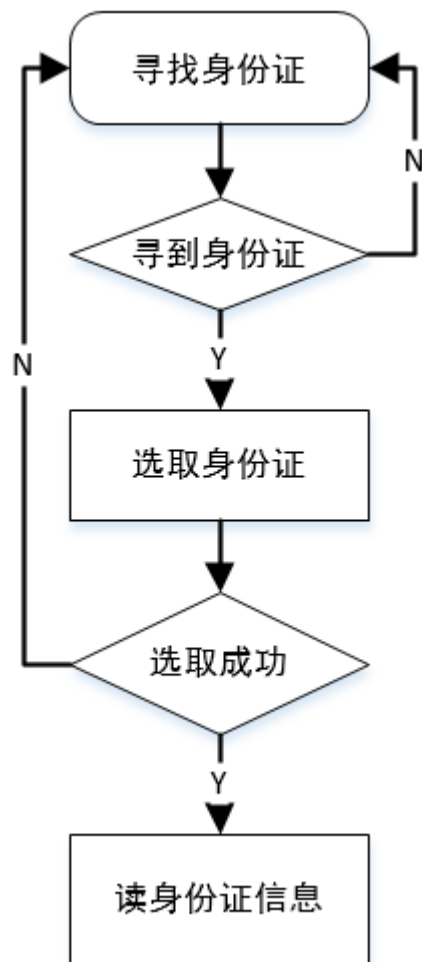
错误反馈指令用于在操作卡片时将遇到的错误类型反馈给上位机，使用者可参考错误代码，快速定位错误源。

错误标记	错误类型	反馈指令	详细说明
EER_TAG_TYPE	卡类型错误	AA 01 E0	发送卡片操作指令与当前检测到的卡片类型不一致时返回
EER_NO_FINE_TAG	未寻到卡错误	AA 01 E1	发送卡片操作指令没有寻到卡时返回
EER_KEY_NO_AUTH	M1 卡密钥不匹配错误	AA 01 E2	M1_READ_BLOCK、M1_WRITE_BLOCK、M1_VALUE_INIT、M1_VALUE_ADD、M1_VALUE_SUB 时，验证密钥失败。此时需使用 M1_SAVE_KEY 指令先把正确的密码

			保存到模块，再使用上述指令。
EER_READ_BLOCK	读块错误	AA 01 E3	使用 M1_READ_BLOCK、UL_READ_BLOCK 时读块不成功时，返回该指令。
EER_WRITE_BLOCK	写块错误	AA 01 E4	使用 M1_WRITE_BLOCK、UL_WRITE_BLOCK 写块不成功。
EER_VALUE_INIT	M1 卡电子钱包充值错误	AA 01 E5	M1_VALUE_INIT 操作卡片失败时返回
EER_VALUE_ADD	M1 卡电子钱包充值	AA 01 E6	使用 M1_VALUE_ADD 操作卡片失败时返回
EER_VALUE_SUB	M1 卡电子钱包扣款错误	AA 01 E7	使用 M1_VALUE_SUB 操作卡片电子钱包扣款失败时返回

## 6 本地身份证解析通讯协议

### 6.1 解析流程



### 6.2 通讯传输字节格式

- 1 ) 通讯标准: UART
- 2 ) 波特率: 115200 bps
- 3 ) 起始位: 1bit
- 4 ) 数据位: 8 bits
- 5 ) 停止位: 1 bit
- 6 ) 校验: 无

## 6.3 数据帧格式

上位机->模块命令格式:

帧头	长度高 8 位	长度低 8 位	命令码	参数	数据	异或和校验
5byte	1byte	1byte	1byte	1byte	n byte	1byte

帧头: 固定为 0xAA 0xAA 0xAA 0x96 0x69

长度: 长度 = 命令码长度 (1Byte) + 参数长度 (1Byte) + 数据长度 (n Byte) + 异或和校验长度 (1Byte)

命令码: 1Byte, 具体见表 1

参数: 1Byte

数据: 帧数据内容, 可以为零字节

异或和校验: 数据帧中除帧头和校验和之外的数据逐字节按位亦或的结果。

模块->上位机应答格式:

帧头	长度高 8 位	长度低 8 位	应答码	数据	异或和校验
5byte	1byte	1byte	3byte	n byte	1byte

帧头: 固定为 0xAA 0xAA 0xAA 0x96 0x69

长度: 长度 = 状态码长度 (3Byte) + 数据长度 (n Byte) + 异或和校验长度 (1Byte)

应答码: 3Byte, 具体见表 2

数据: 帧数据内容, 可以为零字节

异或和校验: 数据帧中除帧头和校验和之外的数据逐字节按位亦或的结果。

## 6.4 命令码及应答码

表 1 命令集

指令类型	命令码	参数	说明
解析命令	0x20	0x01	寻找卡片指令
	0x20	0x02	选取卡片命令
	0x30	0x01	读取明文信息命令

表 2 应答码

指令类型	应答码 1	应答码 2	应答码 3	数据	说明
应答指令	0x00	0x00	0x90	和具体命令有关，可能为空	操作成功
	0x00	0x00	0x9F	证/卡芯片管理号	寻找身份证成功
	0x00	0x00	0x10	-	接收上位机数据的校验和错
	0x00	0x00	0x11	-	接收上位机数据的长度错
	0x00	0x00	0x21	-	接收上位机命令错，包括命令钟的各种数值或逻辑搭配错误
	0x00	0x00	0x24	-	无法识别的错误
	0x00	0x00	0x41	-	读身份证操作失败
	0x00	0x00	0x80	-	寻找身份证失败
	0x00	0x00	0x81	-	选择身份证失败

## 6.5 解析命令

### 1、发送指令寻找身份证

AA AA AA 96 69 00 03 20 01 22

激活成功返回

AA AA AA 96 69 00 08 00 00 9F xx xx xx xx 97

激活失败返回

AA AA AA 96 69 00 04 00 00 80 84

### 2、发送命令选取身份证

AA AA AA 96 69 00 03 20 02 21

选取成功返回

AA AA AA 96 69 00 0C xx xx xx xx xx xx xx xx 9C

选取失败返回

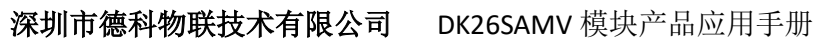
AA AA AA 96 69 00 04 00 00 81 85

### 3、读取身份证信息

AA AA AA 96 69 00 03 30 01 32

读取成功返回数据格式为

AA AA AA 96 69 + xx xx + 00 00 90 + data[1]...data[n] + 亦或和校验



文字信息 长度高 8 位	文字信息 长度低 8 位	照片信息 长度高 8 位	照片信息 长度高低 位	文字信息（可 变）（不大于 256 字节）	照片信息（可变） （不大于 1024 字节）
1byte	1byte	1byte	1byte	n byte	m byte

AA AA AA 96 69 00 04 00 00 41 45

## 7 护照读取

### 第一步 发送激活认证护照指令 (HEX 发送接收):

认证失败返回: AA 01 FF

[illegible]

**AA 19 30 45 31 33 36 31 31 32 31 33 30 39 30 31 30 31 32 39 32 33 30 33 30 33 37**

## 第二步 发送读取护照文件指令

第一步成功返回 AA 01 FE 并且护照没有拿开过的情况下，可以发送读取护照文件指令：

AA 02 31 00

其中最后一个字节 00 代表读取哪个文件。文件编号从 0~16 分别对应护照的: EF.COM、EF.DG1、EF.DG2...EF.DG16。文件结构如下图所示:

具体文件结构及存储格式请参考文档《护照文件结构及存储格式中文版.pdf》。

此部分可以参考我司提供的 demo 以及 demo 源码流程进行设计;

## 8 常见故障分析

### 8.1 打开串口失败

如果使用演示软件连接模块时,提示“拒绝访问”,原因通常为该串口已被其它软件(如串口助手、门禁管理软件等使用串口的软件)打开占用,请退出占用该串口的软件,或者换另外一个串口,再次尝试。

### 8.2 与模块无法通信

如果能打开串口,但与模块通信失败,则请逐一检查确认以下各项:

- 1) 软件上所选的串口是否为主机连接模块的串口,如果使用 USB—UART 转换器,请在系统设备管理器中的“端口 (COM 和 LPT)”项中查看所虚拟出来的串口号;
- 2) 主机和模块之间的通信数据线是否接触良好;串口的 TXD 和 RXD 数据线是否连接正确,是否调反;主机的 GND 和模块的 GND 是否连接在一块。如果连线正确,是否将 TXD 和 RXD 线接反,请将其对调再次尝试;
- 3) 通信参数设置是否正确,正确的串口通信参数应为默认波特率 115200;
- 4) 发送的数据是否有错误,特别需要注意校验和是否有误;
- 5) 检查是否为 16 进制发送接收(勾选 Hex)。

### 8.3 读取不到卡片

如果主机能读取到模块返回的信息(表明主机和模块通信已正常),但读取不到卡片的序列号,那么请检查是否是以下情况造成的:

- 1) 卡片距读卡天线是否过远,试把卡片靠近天线再读取;
- 2) 卡片或模块天线周围是否有大面积金属,大面积金属会严重影响读卡,请把模块移到没有金属的环境再读取;
- 3) 卡片协议是否为模块所支持的协议类型;卡片是否为 ID 卡等,由于 ID 卡工作频率和原理相差甚大,本模块不能支持此类卡片。



## 9 技术支持

德科物联作为专业从事非接触式智能卡读写技术及相关应用的解决方案供应商和制品制造商，始终以为客户提供最及时、最全面的服务为宗旨。

如需您在研发的过程当中遇到技术方面的问题需要我们协助的，请随时联系我司，联系方式如下：

深圳市德科物联技术有限公司

[www.derkiot.com](http://www.derkiot.com)

email: [weiting@derkiot.com](mailto:weiting@derkiot.com)

技术支持 QQ: 460403381