	文件编号：	版本：V1.1
	项目文件	

身份证阅读器 EMP5811A 通讯接口

深圳市雄帝科技股份有限公司

2020 年 12 月

文件修订记录

版本号	生成日期	作 者	修订内容
V1.0	2020/12/10	李伟春	初始版本

目录

目录

- 1. 概述 1
 - 1.1 应用范围 1
- 2. 通信协议 1
 - 2.1 物理层协议 1
 - 2.2 命令格式 1
 - 2.2.1 发送格式 1
 - 2.2.2 接收格式 2
 - 2.2.3 命令码和参数 2
 - 2.2.4 状态字 3
 - 2.2.5 参考示例 3
- 3. 身份证数据说明 5
 - 3.1.1 身份证信息结构： 5
 - 3.1.2 文字信息 5
 - 3.1.3 民族参考 6
 - 3.1.4 性别代码对照表 6
 - 3.1.5 参考示例 6

1. 概述

EMP5811A 外置身份证阅读器的接口说明，包括通信协议和通信数据定义。

1.1 应用范围

本规则适应用于拥有 RS232C 通信接口的主机。

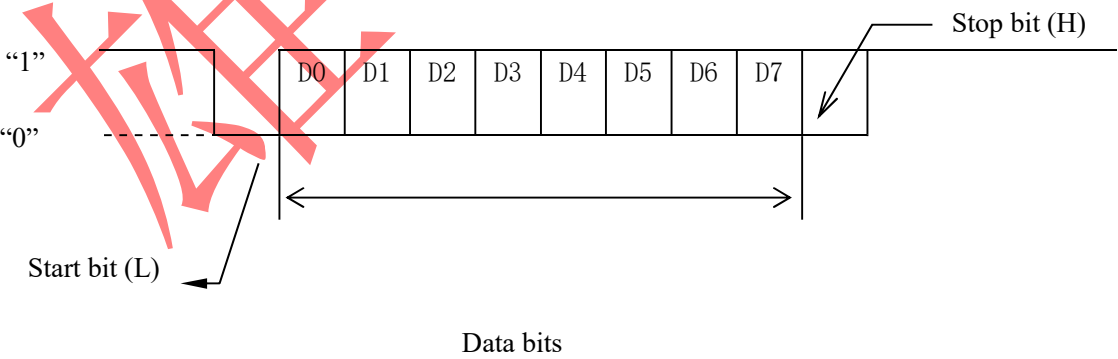
2. 通信协议

- 除文档中特别说明外，协议中其它未指定的数据格式都为 16 进制
- 除文档中特别说明外，协议中数据串都为低字节低地址（在前）

2.1 物理层协议

通讯接口类型：标准 RS232C，异步半双工，三线式（RX、TX、GND）

Symbol	Value	Description
BPS（Transfer Speed）	115200bps	通讯波特率
Data Bits	8	8 位数据位
Parity	NO	无校验位
Stop Bit	1	1 位停止位
Flow Control	NO	无流控制



2.2 命令格式

2.2.1 发送格式

EMP5811A 协议包主设备发送命令格式

定义	preamble	Len1	Len2	CMD	Para	Data	CHK_SUM
长度	5	1	1	1	1	N	1
数据	0xAA, 0xAA, 0xAA, 0x96, 0x69						
含义	帧头	有效数据长度。是 CMD, Para, Data, CHK_SUM 各字节数之和, Len1 为高 8 位, Len2 为低 8 位		命令码和参数。参考 2.2.3。		数据域	校验和。除帧头和校验和之外的数据逐字节按位异或的结果

2.2.2 接收格式

EMP5811A 协议包主设备接收命令格式

定义	preamble	Len1	Len2	SW1	SW2	SW3	Data	CHK_SUM
长度	5	1	1	1	1	1	N	1
数据	0xAA, 0xAA, 0xAA, 0x96, 0x69							
含义	帧头	有效数据长度。是 SW1, SW2, SW3, Data, CHK_SUM 各字节数之和, Len1 为高 8 位, Len2 为低 8 位		状态字。参考 2.2.4			数据域	校验和。除帧头和校验和之外的数据逐字节按位异或的结果

2.2.3 命令码和参数

CMD	Para	含义	Data	CHK_SUM	示例
0x2B	0x00	打开射频	0x00	0x2F	AA AA AA 96 69 00 04 2B 00 00 2F
0x2B	0x00	关闭射频	0x01	0x2E	AA AA AA 96 69 00 04 2B 00 01 2E
0x2B	0x00	关开打开射频	0x02	0x2D	AA AA AA 96 69 00 04 2B 00 02 2D
0x20	0x01	寻找身份证	NULL	0x22	AA AA AA 96 69 00 03 20 01 22
0x20	0x02	选定身份证	NULL	0x21	AA AA AA 96 69 00 03 20 02 21
0x30	0x01	读取身份证	NULL	0x32	AA AA AA 96 69 00 03 30 01 32
0x2B	0x01	寻 B 卡	NULL	0x29	AA AA AA 96 69 00 03 2B 01 29
0xC0	0x01	开启红灯	0x00	0xC5	AA AA AA 96 69 00 04 C0 01 00 C5

0xC0	0x01	开启绿灯	0x01	0xC4	AA AA AA 96 69 00 04 C0 01 01 C4
0xC0	0x02	关闭红灯	0x00	0xC6	AA AA AA 96 69 00 04 C0 02 00 C6
0xC0	0x02	关闭绿灯	0x01	0xC7	AA AA AA 96 69 00 04 C0 02 01 C7
0x11	0xFF	复位身份证模块	NULL	0xED	AA AA AA 96 69 00 03 11 FF ED
0xF5	0x06	获取版本号	NULL	0xF0	AA AA AA 96 69 00 03 F5 06 F0

2.2.4 状态字

SW1	SW2	SW3	含义
0x00	0x00	0x90	操作成功
0x00	0x00	0xA0	接收数据帧超时
0x00	0x01	0xA0	接收数据帧长度错
0x00	0x02	0xA0	接收数据帧校验错
0x00	0x03	0xA0	接收数据帧参数错
0xFF	0xFF	0xFF	不支持的命令
0x00	0x00	0x9F	寻找证/卡成功
0x00	0x00	0x80	寻找证/卡失败
		0x81	选取证/卡失败
		0x41	读证/卡操作失败
0x00	0x00	0x24	无法识别的错误
0x00	0x00	0x21	越权操作
0x00	0x00	0x40	无法识别的卡类型

2.2.5 参考示例

注意（1）：每条命令发送完数据，需等待数据的最大超时时间可参考括号红色字体

注意（2）：每条返回的数据可以通过判断 SW1,SW2,SW3 是不是等于 00 00 90 或者 00 00 9F 来判断是否操作成功。

1. 复位身份证模块（50ms）

Send: AA AA AA 96 69 00 03 11 FF ED

Recv: AA AA AA 96 69 00 04 00 00 90 94

2. 关闭射频（50ms）

Send: AA AA AA 96 69 00 04 2B 00 01 2E

Recv: AA AA AA 96 69 00 04 00 00 90 94

延时 5ms

3. 打开射频（50ms）

Send: AA AA AA 96 69 00 04 2B 00 00 2F

Recv: AA AA AA 96 69 00 04 00 00 90 94

延时 15ms

4. 寻 B 卡（200ms）

Send: AA AA AA 96 69 00 03 2B 01 29

Recv: AA AA AA 96 69 00 10 00 00 90 **50 00 00 00 00** D1 03 86 0C 00 80 80 88

是身份证的话返回的帧含有 50 00 00 00 00

5. 寻找身份证 (200ms)

Send: AA AA AA 96 69 00 03 20 01 22

Recv: AA AA AA 96 69 00 08 00 00 9F 00 00 00 00 97

6. 选定身份证 (200ms)

Send: AA AA AA 96 69 00 03 20 02 21

Recv: AA AA AA 96 69 00 0C 00 00 90 00 00 00 00 00 00 00 9C

7.读取身份证 (2000ms)

Send: AA AA AA 96 69 00 03 30 01 32

Recv: AA AA AA 96 69 05 08 00 00 90 01 00 04 00 4E 67 1F 4F 25 66 20 00 20 00 20 00 20 00 20
00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 31 00 30 00 31 00 31 00 39 00 39 00 35 00 31 00 32 00 30
00 34 00 7F 5E 1C 4E 01 77 E0 60 1C 4E BF 53 14 7A 71 5C 47 95 03 83 8C 54 51 67 D4 59 03 83 2F 6E
51 67 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
20 00 20 00 20 00 34 00 34 00 31 00 33 00 32 00 33 00 31 00 39 00 39 00 35 00 31 00 32 00 30 00 34 00
32 00 33 00 31 00 34 00 E0 60 1C 4E BF 53 6C 51 89 5B 40 5C 20 00 20 00 20 00 20 00 20 00 20 00 20
00 20 00 20 00 32 00 30 00 31 00 35 00 30 00 38 00 32 00 38 00 32 00 30 00 32 00 35 00 30 00 38 00 32
00 38 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20
00 20 00 20 00 57 4C 66 00 7E 00 32 00 00 FF 85 1D 51 51 51 3E 71 0D D5 64 F3 7D 2C 9F C7 48 2B
DC E3 91 58 82 4D 5E 15 05 A3 D2 A3 C0 92 DE 1B B4 FF 17 BB 72 69 ED B2 3F 68 0C 4B 5F 16 C6
62 0C 08 12 B7 95 D2 EC DE A1 9D CD 2A 79 98 0B D4 E9 02 8E 11 F1 C9 1C AA F4 2E 5E 1D 38 48
D1 AE D6 52 51 51 5A 3E 8B 62 47 B7 40 91 35 B6 0C 13 9E BE D3 6B 26 CC 33 B4 30 E8 6F B8 3E
CE FB 57 8B 32 EC ED D8 99 3E 63 DA 6E F7 F3 5C 09 E1 A8 43 C0 FF AA 86 1E DF 87 6E 37 DA 7D
19 25 7D AC 6C 4A C6 96 B6 3B 61 36 07 A5 CF 74 AE 51 45 CA 77 54 16 6E 50 11 61 36 1C 64 B9 F7
4B F9 B8 05 1D CB DA F4 8C E4 47 40 89 D6 37 88 B0 4E F2 C8 1B 3F F7 E7 5E 0A 1C 03 85 D5 28
D3 FC 2A B0 EF 5B 9B 3C FC 09 36 56 80 38 32 12 F8 6B B1 87 90 D0 40 DA A6 0D F0 59 CB DC 84
0A 39 6A 5E 01 01 42 12 C1 35 4F 43 BE E3 B5 84 8E B5 02 D7 55 C3 EF F2 E4 19 73 0A 50 98 A0 1E
5C 65 90 41 E8 F5 33 B5 70 75 3C 89 9D E3 22 72 1B 66 0C 33 B9 EE A4 74 EE B2 F6 DD 0E 42 E0 E6
68 CF 0B 5A FD 88 63 53 24 B7 90 D2 4E 41 1E 5C C3 DA A4 F8 74 D3 A3 91 6D 02 24 13 23 9C 5D
D2 EA A8 96 74 C0 F5 61 37 48 69 00 12 3A 8B 92 51 74 9B 6F 79 26 B9 40 50 FA 80 E0 9F 59 AA E2
86 BE DD EF DF 07 38 C2 E1 B4 CB F2 72 AF 55 25 89 AB 18 4F D4 CB 8E A2 8E 40 75 3C 3E E6 EA
DF 03 6B 50 E0 0D B2 70 3B AC DD AA 11 DD 40 03 A2 7C 9F 7C A1 9C 20 FC 96 EA 06 04 60 54 0F
1E 02 57 FC 54 48 83 F2 1B 4C AA 92 3F D5 E8 40 0D 9F BC 4A CC 60 BA DE 79 A7 45 FA EE FC 81
C0 01 F6 30 81 69 E4 CD 85 F6 ED EC 6A 36 4E 44 6D 56 45 AD 1C 3B 75 9F DA 1F A1 42 51 DF 91
3D 37 C0 E7 6F EE 0C A6 83 7C C4 5E 6D E1 45 EB DB 52 0A 5E 46 2C 61 C0 D4 DD 22 0E 22 B0 0F

0D 24 DF 82 4B 79 2A 11 9E 98 7A 45 6D D2 B7 18 1E 3C E0 1C AD 5F 4B 4E 93 05 4F C0 1B 01 1F
EE EB C8 EC B3 90 A6 63 72 76 29 1F 9B F7 4D 92 03 27 C8 FA 72 0F 9B 62 36 37 0A D2 AD 83 41
1B A4 44 54 CE DE 8A 83 72 2F 30 2B E9 38 97 85 9E F7 55 63 52 A3 75 A4 0A 18 CC 64 9A 34 75 9C
53 E5 BA 07 05 84 F2 2C 06 EA DF C4 D0 34 60 06 20 C1 38 D5 3C 72 7E 6E 9F 64 B7 E9 F5 F6 DE
CD 17 6F A9 8D A8 05 45 DE 79 1D A1 3D 8D 9F 6B DE 60 27 7B DA 8D 64 DE C4 E9 75 54 AF 36 2F
9F E3 37 AC C2 67 9F FA 20 66 D0 92 C3 C4 8E F7 A4 AB DE D8 2B A2 79 6D FA FA AC 1A 3A A9
C3 12 63 7B 60 8D 73 A6 10 9A BF 0F BC 57 30 5B 35 FA AC 21 89 47 22 8F BA 86 55 D0 B4 8C 52
49 E2 7E 44 B8 F7 07 F5 0B C5 9A 73 BF B8 AF 0A 4B 38 AA 95 E8 AA CB F1 45 13 89 FF 09 FE 1B
CF FF 03 6A FC 17 FF 3C 7F 16 2B CC A1 A3 73 25 77 7E 26 94 EE 7F E2 D3 A3 1F 02 A5 34 F0 40 23
09 3C 9F 12 88 C0 3F C5 42 6F 25 2E 19 1F 17 28 6B 27 B5 20 F2 94 F5 B6 31 23 1C C7 C9 E8 F6 47 36
FD C4 49 2B 22 E5 09 05 79 91 BC EC 21 0D 3F DC 1E 1C BA 95 6D 11 BB C4 80 B9 7F C8 96 5A 3E
D0 D8 5E F7 82 2D 36 94 98 76 E8 1C 33 6B 04 E5 3B 7D CC 98 C4 89 7F 5E 47 AD 55 C4 AA 08 27
5E AA 15 9B B6 C0 82 DA 88 BD CD 14 B9 0B 6A D3 44 E4 0D 82 A9 A0 3A 0A B6 9C CF 53 1A BE
C1 9A C3 48 54 9A 7F BD 07 B3 2A 07 E9 52 3A D0 E5 F8 5A 3E 1C 6E 61 32 85 C5 58 AF 2A B2 0C
AD D7 87 D1 B5 41 56 3B 47 CD 46 EA 04 07 40 4C 5E 6A 80 38 8E 1D 43 49 29 63 41 96 69 17 5C DE
1E 06 7F 97 5C 71 6D 46 B9 2E 75 06 EF 5E F2 E5 BC 1D A9 7F F5 0E 47 2B 2A CE BC B5 13 73

3. 身份证数据说明

以下对读取身份证（最后一条命令）获得的信息进行解析。

3.1.1 身份证信息结构：

AA AA AA 96 69 05 08 00 00 90 01 00 04 00 + (256 字节文字信息) + (1024 字节
预留信息) + (1 字节 CRC)

3.1.2 文字信息

文字信息采用 GB 13000 的 UCS-2 进行存储，各项目分配如下：

项目	长度（字节）	说明
姓名	30	汉字
性别	2	代码
民族	4	代码
出生	16	年月日：YYYYMMDD
住址	70	汉字和数字
公民身份号码	36	数字

签发机关	30	汉字
有效期起始日期	16	年月日: YYYYMMDD
有效期截止日期	16	年月日: YYYYMMDD 有效期为长期时存储“长期”
备用	36	

3.1.3 民族参考

民族代码对照表

01	汉	15	土家	29	柯尔克孜	43	乌孜别克
02	蒙古	16	哈尼	30	土	44	俄罗斯
03	回	17	哈萨克	31	达斡尔	45	鄂温克
04	藏	18	傣	32	仫佬	46	德昂
05	维吾尔	19	黎	33	羌	47	保安
06	苗	20	傈僳	34	布朗	48	裕固
07	彝	21	佤	35	撒拉	49	京
08	壮	22	畲	36	毛难	50	塔塔尔
09	布依	23	高山	37	仡佬	51	独龙
10	朝鲜	24	拉祜	38	锡伯	52	鄂伦春
11	满	25	水	39	阿昌	53	赫哲
12	侗	26	东乡	40	普米	54	门巴
13	瑶	27	"纳西	41	塔吉克	55	珞巴
14	白	28	景颇	42	怒	56	基诺
97	其他	98	外国血统中国籍 人士				

3.1.4 性别代码对照表

0	未知	2	女
1	男	9	未说明

3.1.5 参考示例

假如最后一条命令成功读取的数据存放在 buf 中，可参考如下解析数据（c++）。

```
qstrList << "汉" << "蒙古" << "回" << "藏" << "维吾尔" << "苗" << "彝" << "壮" << "布依" << "朝
```

鲜"<<"满"<<"侗"<<"瑶"<<"白"<<"土家"<<"哈尼"<<"哈萨克"<<"傣"<<"黎"<<"傈僳"
"<<"佤"<<"畲"<<"高山"<<"拉祜"<<"水"<<"东乡"<<"纳西"<<"景颇"<<"柯尔克孜"<<"
土"<<"达斡尔"<<"仫佬"<<"羌"<<"布朗"<<"撒拉"<<"毛难"<<"仡佬"<<"锡伯"<<"阿昌"
"<<"普米"<<"塔吉克"<<"怒"<<"乌孜别克"<<"俄罗斯"<<"鄂温克"<<"德昂"<<"保安"
"<<"裕固"<<"京"<<"塔塔尔"<<"独龙"<<"鄂伦春"<<"赫哲"<<"门巴"<<"珞巴"<<"基诺"
"<<"其他"<<"外国血统中国籍人士";

```
char buf[2048];
```

```
QString TempStr,result;
```

```
unsigned short U16buf[20];
```

```
#define offset_head 14
```

```
for ( i = 0 ; i < (30/2) ; i++ )
```

```
{
```

```
    U16buf[i] = buf[offset_head + 2*i] + (buf[offset_head + 2*i+1] << 8);
```

```
}
```

```
TempStr = QString::fromUtf16(U16buf,i);
```

```
str= "姓名:      "+TempStr +"\r\n";
```

```
result.append(str);
```

```
U16buf[0] = buf[offset_head + 30 + 0] + (buf[offset_head + 30 + 1] << 8);
```

```
TempStr = QString::fromUtf16(U16buf,1);
```

```
if(TempStr.contains("1",Qt::CaseSensitive))
```

```
    TempStr = "男";
```

```
else if(TempStr.contains("2",Qt::CaseSensitive))
```

```
    TempStr = "女";
```

```
str= "性别:      "+TempStr +"\r\n";
```

```
result.append(str);
```

```
U16buf[0] = buf[offset_head + 32 + 0] + (buf[offset_head + 32 + 1] << 8);
```

```
U16buf[1] = buf[offset_head + 34 + 0] + (buf[offset_head + 34 + 1] << 8);
```

```
TempStr = QString::fromUtf16(U16buf,2);
```

```
if(TempStr.toInt() > 56 || TempStr.toInt() <= 0) str= "民族:      未知\r\n";
```

```
else str= "民族:      "+ qstrList.at(TempStr.toInt()-1) +"\r\n";
```

```
result.append(str);
```

```
for ( i = 0 ; i < (16/2) ; i++ )
{
    U16buf[i] = buf[offset_head + 36 + 2*i ] + (buf[offset_head + 36 + 2*i+1] << 8);
}
TempStr = QString::fromUtf16(U16buf,i);
str= "出生时间:  "+TempStr+"\r\n";
result.append(str);

for ( i = 0 ; i < (70/2) ; i++ )
{
    U16buf[i] = buf[offset_head + 52 + 2*i ] + (buf[offset_head + 52 + 2*i+1] << 8);
}
TempStr = QString::fromUtf16(U16buf,i);
str= "住址:      "+TempStr+"\r\n";
result.append(str);

for ( i = 0 ; i < (36/2) ; i++ )
{
    U16buf[i] = buf[offset_head + 122 + 2*i ] + (buf[offset_head + 122 + 2*i+1] << 8);
}
TempStr = QString::fromUtf16(U16buf,i);
str= "身份证号码: "+TempStr+"\r\n";
result.append(str);

for ( i = 0 ; i < (30/2) ; i++ )
{
    U16buf[i] = buf[offset_head + 158 + 2*i ] + (buf[offset_head + 158 + 2*i+1] << 8);
}
TempStr = QString::fromUtf16(U16buf,i);
str= "签发机关:  "+TempStr+"\r\n";
result.append(str);

for ( i = 0 ; i < (16/2) ; i++ )
```

```
{
    U16buf[i] = buf[offset_head + 188 + 2*i] + (buf[offset_head + 188 + 2*i+1] << 8);
}
TempStr = QString::fromUtf16(U16buf,i);
TempStr = TempStr + "-";
for ( i = 0 ; i < (16/2) ; i++)
{
    U16buf[i] = buf[offset_head + 204 + 2*i] + (buf[offset_head + 204 + 2*i+1] << 8);
}
TempStr = TempStr + QString::fromUtf16(U16buf,i);
str= "签发日期: "+TempStr +"\r\n";
result.append(str);
```