

# **ZM12xUE 系列接口函数说明**

## **（USB HID 接口）**

声明：此开发接口只适合在 windows32 操作系统上使用，且只适合支持 C,C++,C#编译器的开发工具。

**注：**

- **此文档为全功能文档，部分产品仅支持其中部分功能**

更新日期:2013-7-12

# 目录

1. 读卡器设置接口函数.....	3
1.1. “查询版本”函数 .....	3
1.2. “设置 LED”函数 .....	3
1.3. “时间设置”函数 .....	3
1.4. “获取时间”函数 .....	4
1.5. “正确蜂鸣设置”函数 .....	4
1.6. “错误蜂鸣设置”函数 .....	4
2. 非接触式卡接口函数.....	6
2.1. “开场”函数 .....	6
2.2. “关场”函数 .....	6
2.3. “复位”函数 .....	6
2.4. “COS 命令”函数 .....	7
2.5. 获取卡片 UID 函数.....	7
2.6. Mifare“写块”函数 .....	8
2.7. Mifare “读块”函数 .....	8
2.8. Mifare “写金额”函数 .....	8
2.9. Mifare “读金额”函数 .....	9
2.10. Mifare “充值”函数 .....	9
2.11. Mifare “消费”函数 .....	10
3. PSAM 卡接口函数.....	11
3.1. “冷复位”函数 .....	11
3.2. “热复位”函数 .....	11
3.3. “下电”函数 .....	11
3.4. “APDU 应用命令”函数 .....	12
4. “15693 卡片”接口函数 .....	13
4.1. “获取卡片 UID”函数 .....	13
4.2. “读取卡片数据”函数 .....	13
4.3. “向卡片写入数据”函数 .....	13
4.4. “读取指定块数据”函数 .....	14
4.5. “向指定块写数据”函数 .....	14
4.6. “锁块”函数 .....	14
4.7. “写 AFI”函数.....	15
4.8. “锁 AFI”函数.....	15
4.9. “写 DSFID”函数.....	15
4.10. “锁 DSFID”函数.....	16
4.11. “获取卡片系统信息”函数 .....	16
附录：返回值对照.....	17

# 1. 读卡器设置接口函数

## 1.1. “查询版本”函数

函数：

```
int Reader_Version(unsigned char *resp,int *resp_len)
```

功能：

查询读卡器的版本号。

参数说明：

**\*resp:** 输出参数。指向存放响应数据的存储区的指针。

**\*resp\_len:** 输出参数。指向存放响应数据长度的存储区的指针。

返回值：

如果命令执行成功，则返回 0，返回值小于 0 则为状态码，其含义见表一。

## 1.2. “设置 LED”函数

函数：

```
int LED(unsigned char *data,int data_len,unsigned char point)
```

功能：

设置 LED 屏显示的数

参数说明：

**\*data:** 输入参数。指向存放要设置的数据的存储区的指针。

**data\_len:** 要设置的数据的长度。

**Point:** 小数点位置。0x00 = 无小数点；0x01~0x06 分别对应小数点从右至左的位置。

返回值：

如果命令执行成功，则返回 0，返回值小于 0 则为状态码，其含义见表一。

## 1.3. “时间设置”函数

**注：此函数需要定制。**

函数：

```
int Adjust_Clock(unsigned char *times,int times_len)
```

功能：

设置读卡器的时间。

参数说明：

**\*times:** 输入参数。指向存放要设置的时间的存储区的指针；年、月、日、时、分、秒共 7 字节。

times\_len: 要设置的数据的长度。

返回值:

如果命令执行成功, 则返回 0, 返回值小于 0 则为状态码, 其含义见表一。

## 1.4. “获取时间”函数

**注：此函数需要定制。**

函数:

```
int Get_time(unsigned char *resp,int *resp_len)
```

功能:

获取读卡器的当前时间。

参数说明:

\*resp: 输出参数。指向存放响应数据的存储区的指针。

\*resp\_len: 输出参数。指向存放响应数据长度的存储区的指针。

返回值:

如果命令执行成功, 则返回 0, 返回值小于 0 则为状态码, 其含义见表一。

## 1.5. “正确蜂鸣设置”函数

函数:

```
int buzzer_setting(unsigned char buzzer_duration, unsigned char silent_duration, int repeat_time, int flag=0)
```

功能:

该函数设置读卡器读到卡片且卡片有信号时的蜂鸣器响应

参数说明:

buzzer\_duration: 蜂鸣的鸣叫时间, 单位是 ms

silent\_duration: 蜂鸣的不鸣叫时间, 单位是 ms

repeat\_time : 鸣叫次数

flag: 标志位, 用来判断蜂鸣器是设置完成后开始鸣叫还是设置完成时不鸣叫, 下次刷卡时鸣叫; flag 只能设置为 0 或者 1, 1 为设置完成后开始鸣叫; 0 为设置完成后不响, 下次刷卡的时候鸣叫。

返回值:

如果命令执行成功, 则返回 0, 返回值小于 0 则为状态码, 其含义见表一。

## 1.6. “错误蜂鸣设置”函数

函数:

```
int buzzer_failsetting(unsigned char buzzer_duration,unsigned char silent_duration,int repeat_time,int flag=0)
```

功能:

该函数设置读卡器没有读到卡片或者卡片无信息返回时的蜂鸣器响应。

参数说明:

**buzzer\_duration:** 蜂鸣的鸣叫时间, 单位是 ms

**silent\_duration :** 蜂鸣的不鸣叫时间, 单位是 ms

**repeat\_time :** 蜂鸣的次数

**flag:** 标志位, 用来判断蜂鸣器是设置完成后开始鸣叫还是设置完成时不鸣叫, 下次刷卡时鸣叫; **flag** 只能设置为 0 或者 1, 1 为设置完成后开始鸣叫; 0 为设置完成后不响, 下次刷卡的时候鸣叫。

返回值:

如果命令执行成功, 则返回 0, 返回值小于 0 则为状态码, 其含义见表一。

## 2. 非接触式卡接口函数

### 2.1. “开场”函数

函数:

```
int IDD_PowerOn(void)
```

功能:

给读卡器开场。

参数说明:

无

返回值:

如果开场成功，则返回 0；返回值小于 0 则为状态码，其含义见表一。

### 2.2. “关场”函数

函数:

```
int IDD_PowerOff(void)
```

功能:

给读卡器关场。

参数说明:

无

返回值:

如果关场成功，则返回 0；返回值小于 0 则为状态码，其含义见表一。

### 2.3. “复位”函数

函数:

```
int IDD_Reset(unsigned char *resp,int *resp_len)
```

功能:

对非接触式 CPU 卡进行复位。

参数说明:

\*resp: 输出函数。指向存放响应数据的存储区的指针。

\*resp\_len: 输出函数。指向存放响应数据长度的存储区的指针。

返回值:

如果复位成功，则返回 0，返回值小于 0 则为状态码，其含义见表一。

注：复位成功数据解析见下表：

卡片类型	UID 长度	UID	ATS 长度	ATS
0x0A：A 协议卡； 0x0B：B 协议卡	1 字节	4~10 字节	1 字节	不定长

例如，CPU 卡返回值：0A04019939780D0D780091020088008801993978

0x0A = A 协议卡片；

0x04 = UID 为 4 字节；

0x01993978 = 4 字节的卡片 UID；

0x0D = ATS 长度（16 进制数）；

0x0D780091020088008801993978 = ATS 内容。

## 2.4. “COS 命令”函数

函数：

```
int IDD_COS(unsigned char *COS,int COS_len,unsigned char *resp,int *resp_len)
```

功能：

向读卡器发送 COS 命令。

参数说明：

\*COS：输入函数。指向存放 COS 命令数据的存储区的指针。

COS\_len：输入函数。COS 命令的数据长度。

\*resp：输出函数。指向存放响应数据的存储区的指针。

\*resp\_len：输出函数。指向存放响应数据长度的存储区的指针。

返回值：

如果命令执行成功，则返回 0，返回值小于 0 则为状态码，其含义见表一。

## 2.5. 获取卡片 UID 函数

函数：

```
int find_14443(unsigned char* uid,int* uid_len);
```

功能：

获取 14443-A 卡片的 UID。

参数说明：

\*uid：输出函数。卡片 UID 数据的存储区的指针。

\*uid\_len：输出函数。UID 长度。

返回值：

如果命令执行成功，则返回 0，返回值小于 0 则为状态码，其含义见表一。

## 2.6. Mifare“写块”函数

函数:

```
int write_block(int block,int page,unsigned char pswtype,unsigned char *psw,unsigned char *src_data,int src_len);
```

功能:

向 M1 卡写信息

参数说明:

block: M1 卡块号; (S50 块号为 0x00~0x03)。

page: M1 卡扇区号; (S50 扇区号 0x00~0x0F)。

pswtype: 密钥类型; 参数说明: A 密钥: 0x0A; B 密钥: 0x0B;

\*psw: 输入参数, 密钥值; 6 字节的十六进制数, 默认密钥 0xFFFFFFFFFFFF;

\*src\_data: 输入参数, 写入的信息数据, 每次最多写入 16 字节的数据;

src\_len: 输入参数, 取\*src\_data 信息的长度, 最多 16 字节。

返回值:

如果命令执行成功, 则返回 0, 返回值小于 0 则为状态码, 其含义见表一。

**注: 块号 3 为密钥存储块, 密钥存储块的改变将导致扇区密码改变, 因此请勿随便更改密钥存储块信息; 如需更改密钥, 详情请参考 S50 卡片手册**

## 2.7. Mifare “读块”函数

函数:

```
int read_block(int page, int block, unsigned char pswtype, unsigned char *psw, unsigned char *des_data, int* des_len);
```

功能:

读取 M1 卡块信息

参数说明:

page: M1 卡扇区号; (S50 扇区号 0x00~0x0F)。

block: M1 卡块号; (S50 块号为 0x00~0x03)。

pswtype: 密钥类型; 参数说明: A 密钥: 0x0A; B 密钥: 0x0B;

\*psw: 输入参数, 密钥值; 6 字节的十六进制数, 默认密钥 0xFFFFFFFFFFFF;

\*des\_data: 输出参数, 返回读取的信息

\*des\_len: 输出参数, 返回信息的长度

返回值:

如果命令执行成功, 则返回 0, 返回值小于 0 则为状态码, 其含义见表一。

## 2.8. Mifare “写金额”函数

函数:

```
int write_account(int page, int block, unsigned char pswtype, unsigned char *psw, LONG
```



account);

功能:

对 M1 卡指定扇区下某块进行初始化

参数说明:

page: M1 卡扇区号; (S50 扇区号 0x00~0x0F)。

block: M1 卡块号; (S50 块号为 0x00~0x03)。

pswtype: 密钥类型; 参数说明: A 密钥: 0x0A; B 密钥: 0x0B;

\*psw: 输入参数, 密钥值; 6 字节的十六进制数, 默认密钥 0xFFFFFFFFFFFF;

account: 初始化的数据

返回值:

如果命令执行成功, 则返回 0, 返回值小于 0 则为状态码, 其含义见表一。

## 2.9. Mifare “读金额”函数

函数:

```
int read_account(int page, int block, unsigned char pswtype, unsigned char *psw, LONG*  
account);
```

功能:

获取 M1 卡指定扇区下某块的数据

参数说明:

page: M1 卡扇区号; (S50 扇区号 0x00~0x0F)。

block: M1 卡块号; (S50 块号为 0x00~0x03)。

pswtype: 密钥类型; 参数说明: A 密钥: 0x0A; B 密钥: 0x0B;

\*psw: 输入参数, 密钥值; 6 字节的十六进制数, 默认密钥 0xFFFFFFFFFFFF;

\*account: 输出参数, 获取的数据

返回值:

如果命令执行成功, 则返回 0, 返回值小于 0 则为状态码, 其含义见表一。

## 2.10. Mifare “充值”函数

函数:

```
int add_account(int page, int block, unsigned char pswtype, unsigned char *psw, long  
addAccount);
```

功能:

对 M1 卡钱包进行充值操作

参数说明:

page: M1 卡的扇区号; (S50 扇区号 0x00~0x0F)。

block: M1 卡的块号; (S50 块号为 0x00~0x03)。

pswtype: 密钥类型; 参数说明: A 密钥: 0x0A; B 密钥: 0x0B;

\*psw: 输入参数, 密钥值; 6 字节的十六进制数, 默认密钥 0xFFFFFFFFFFFF;

addAccount: 充值金额

返回值:

如果命令执行成功, 则返回 0, 返回值小于 0 则为状态码, 其含义见表一。

## 2.11. Mifare “消费”函数

函数:

```
int sub_account(int page, int block, unsigned char pswtype, unsigned char *psw, long subAccount);
```

功能:

对 M1 卡钱包进行消费操作

参数说明:

page: M1 卡扇区号; (S50 扇区号 0x00~0x0F)。

block: M1 卡块号; (S50 块号为 0x00~0x03)。

pswtype: 密钥类型; 参数说明: A 密钥: 0x0A; B 密钥: 0x0B;

\*psw: 输入参数, 密钥值; 6 字节的十六进制数, 默认密钥 0xFFFFFFFFFFFF;

subAccount: 消费金额

返回值:

如果命令执行成功, 则返回 0, 返回值小于 0 则为状态码, 其含义见表一。

## 3. PSAM 卡接口函数

**注：仅 ZM124UE 支持此功能**

### 3.1. “冷复位”函数

函数：

```
int ICC_C_Reset(unsigned char *resp,int *resp_len)
```

功能：

对读卡器上的 PSAM 卡进行冷复位。

参数说明：

\*resp: 输出参数，指向存放响应数据的存储区的指针。

\*resp\_len: 输出参数，指向存放响应数据长度的存储区的指针。

返回值：

如果对 PSAM 卡冷复位成功，则返回 0；返回值小于 0 则为状态码，其含义见表一。

### 3.2. “热复位”函数

函数：

```
int ICC_H_Reset(unsigned char *resp,int *resp_len)
```

功能：

对读卡器的 PSAM 卡进行热复位。

参数说明：

\*resp: 输出参数，指向存放响应数据的存储区的指针。

\*resp\_len: 输出参数，指向存放响应数据长度的存储区的指针。

返回值：

如果对 PSAM 卡冷复位成功，则返回 0；返回值小于 0 则为状态码，其含义见表一。

**注意：执行热复位的前提是冷复位必须执行成功。**

### 3.3. “下电”函数

函数：

```
int ICC_Power_Off(void)
```

功能：

对 PASM 卡进行下电处理。

参数：

无

返回值：

如果对 PASM 卡下电成功，则返回 0；返回值小于 0 则为状态码，其含义见表一。

### 3.4. “APDU 应用命令”函数

函数：

```
int ICC_Application(int Lenth_of_Command_APDU, int *Rsep_len,unsigned char  
*Command_APDU, unsigned char *Resp_APDU)
```

功能：

向 PSAM 卡发送 APDU 应用命令。

参数说明：

Lenth\_of\_Command\_APDU: APDU 命令数据的长度。

\*Rsep\_len: 输出参数，指向存放响应数据长度的存储区的指针。

\*Command\_APDU: 输入参数，指向存放 APDU 命令数据的存储区的指针。

\*Resp\_APDU: 输出参数，指向存放响应数据的存储区的指针。

返回值：

如果 APDU 命令执行成功，则返回 0；返回值小于 0 则为状态码，其含义见表一。

## 4. “15693 卡片”接口函数

**注：仅 ZM121UE 支持此功能**

### 4.1. “获取卡片 UID”函数

函数：

```
int RFID15693_FindUid(unsigned char *resp)
```

功能：

获取卡片的 UID 号

参数说明：

**\*resp:** 输出参数，指向存放获取的 UID 数据的存储区的指针。

返回值：

如果命令执行成功，则返回值>0 表示 UID 号的长度；返回值小于 0 则为状态码，其含义见表一。

### 4.2. “读取卡片数据”函数

函数：

```
int RFID15693_ReadFile(int blocknum, unsigned char *resp,unsigned char *fl)
```

功能：

读取设置的卡片块数的所有数据。

参数说明：

**blocknum:** 设置的卡片块数。

**\*resp:** 输出参数，指向存放读取数据的存储区的指针。

**\*fl:** 输出参数，锁定位状态，指向存放标识数据的存储区的指针。

返回值：

如果命令执行成功，则返回值>0 表示读取的长度；返回值小于 0 则为状态码，其含义见表一。

### 4.3. “向卡片写入数据”函数

函数：

```
int RFID15693_WriteFile(unsigned char *src_data, int src_len)
```

功能：

向卡片写入指定长度的数据

参数说明：

**\*src\_data:** 输入参数，指向存放写入数据的存储区的指针。

**src\_len:** 要写入数据的长度。

返回值:

如果命令执行成功，则返回 0，返回值小于 0 则为状态码，其含义见表一。

## 4.4. “读取指定块数据”函数

函数:

```
int RFID15693_ReadSingle(int block,unsigned char *resp)
```

功能:

读取卡片指定的块数据。

参数说明:

**block:** 指定的块号。

**\*resp:** 输出参数，指向存放读取数据的存储区的指针。

返回值:

如果命令执行成功，则返回值>0 表示读取的长度；返回值小于 0 则为状态码，其含义见表一。

## 4.5. “向指定块写数据”函数

函数:

```
int RFID15693_WriteSingle(int block,unsigned char *src_data,int src_len)
```

功能:

向卡片指定的块写数据。

参数说明:

**block:** 指定的块号。

**\*src\_data:** 输入参数，指向存放写入数据的存储区的指针。

**src\_len:** 写入数据的长度。

返回值:

如果函数执行成功，则返回值为 0；返回值小于 0 为状态码，其含义见表一。

## 4.6. “锁块”函数

函数:

```
int RFID15693_LockBlock(int block)
```

功能:

锁住指定的块，使其不能读写数据。

参数说明:

**block:** 指定的块号。

返回值:

如果函数执行成功，则返回值为 0；返回值小于 0 为状态码，其含义见表一。

## 4.7. “写 AFI”函数

函数:

```
int RFID15693_WriteAFI(unsigned char afi)
```

功能:

向卡片写入 AFI 值。

参数说明:

**afi:**写入的 AFI 值

返回值:

如果函数执行成功，则返回值为 0；返回值小于 0 为状态码，其含义见表一。

## 4.8. “锁 AFI”函数

函数:

```
int RFID15693_LockAFI()
```

功能:

锁住 AFI，使其不能读写操作。

参数说明:

无。

返回值:

如果函数执行成功，则返回值为 0；返回值小于 0 为状态码，其含义见表一。

## 4.9. “写 DSFID”函数

函数:

```
int RFID15693_WriteDSFID(unsigned char dsfid)
```

功能:

向卡片写入 DSFID 值。

参数说明:

**dsfid:**写入的 DSFID 值。

返回值:

如果函数执行成功，则返回值为 0；返回值小于 0 为状态码，其含义见表一。

## 4.10. “锁 DSFID”函数

函数:

```
int RFID15693_LockDSFID()
```

功能:

锁住 DSFID，使其不能读写操作。

参数说明:

无。

返回值:

如果函数执行成功，则返回值为 0；返回值小于 0 为状态码，其含义见表一。

## 4.11. “获取卡片系统信息”函数

函数:

```
int RFID15693_GetSystemInfo(unsigned char *uid_buff, unsigned char *dsfid, unsigned char*afi)
```

功能:

获取卡片的 uid 号，dsfid 值，afi 值。

参数说明:

\*uid\_buff: 输出参数，指向存放获取 UID 号数据的存储区的指针。

\*dsfid: 输出参数，获取的 dsfid 值。

\*afi: 输出参数，获取的 afi 值。

返回值:

如果函数执行成功，则返回值为 0；返回值小于 0 为状态码，其含义见表一。



# 附录：返回值对照

表一：

应用编程的标识符	返回值	含义
IFD_OK	0	执行成功
IFD_ICC_TypeError	-1	卡片类型不对
IFD_ICC_NoExist	-2	无卡
IFD_ICC_NoPower	-3	有卡未上电
IFD_ICC_NoResponse	-4	卡片无应答
IFD_ICC_BCCError	-5	BCC 校验错误
IFD_ICC_TimeOut	-6	接收超时
IFD_ICC_RunFail	-7	执行失败
IFD_ICC_SiteFail	-8	卡片位置错误
IFD_ICC_SetFail	-9	设置失败
IFD_ConnectError	-11	读卡器连接错
IFD_UnConnected	-12	未建立连接(没有执行打开设备函数)
IFD_BadCommand	-13	(动态库)不支持该命令
IFD_ParameterError	-14	(发给动态库的)命令参数错
IFD_CheckSumError	-15	信息校验和出错
IFD_ICC_PowerFail	-16	卡片上电失败
IFD_ICC_ResetFail	-17	卡片复位失败
IFD_ICC_PowerOffFail	-18	卡片下电失败