



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR122L

带液晶显示屏的 NFC 读写器 (串口)

应用程序编程接口 V1.03





目录

1.0.	简介	3
2.0.	特性	4
3.0.	应用程序编程接口概述	5
3.1.	读写器	5
3.1.1.	预定义文档	5
3.1.2.	函数文档	5
3.2.	LCD	10
3.2.1.	预定义文档	10
3.2.2.	函数文档	10
3.3.	LED	18
3.3.1.	函数文档	18
3.4.	蜂鸣器	20
3.4.1.	函数文档	20
3.5.	卡片	21
3.5.1.	函数文档	21
	附录 A.数据结构	24
	附录 A.1 _ACR122_TIMEOUTS 结构引用	24
	附录 A.2 _ACR122_LED_CONTROL 结构引用	24
	附录 B.高阶 API 返回的错误代码	25
	附录 C.标准的程序流	26

图目录

图 1	: ACR122L 库架构设计	3
------------	------------------------------	----------

1.0. 简介

本手册介绍了如何使用 ACR122L 软件接口进行 ACR122L 读写器应用程序的开发。此软件接口以 32 位和 64 位动态链接库 (DLL) 的形式提供，可以通过当前流行的开发工具进行编程，其中包括 Java、Delphi、Visual Basic、Visual C++、Visual C#和 Visual Basic .NET。

ACR122L 可以通过 RS-232 接口与电脑进行连接。

ACR122L 库的架构设计可视化如下图：

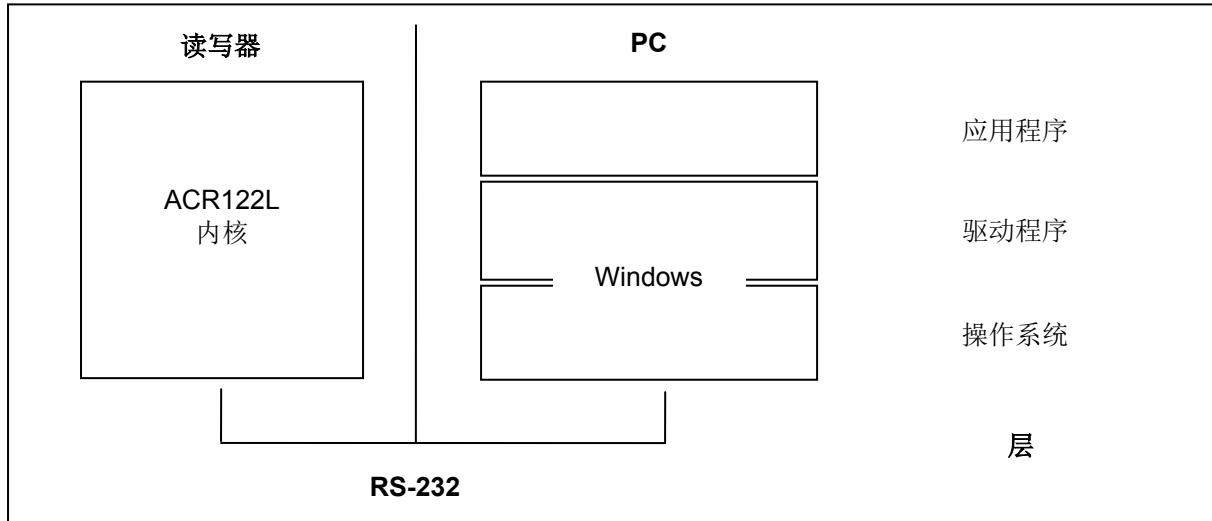


图1：ACR122L 库架构设计



2.0. 特性

- RS-232 串行接口：波特率 = 9600 or 115200 bps, 8-N-1
- 7V 直流适配器供电
- 仿 CCID 架构（二进制格式）
- 智能卡读写器：
 - 读写速率高达 424 kbps
 - 内置天线用于读写非接触式标签，读取智能卡的距离可达 50 mm（视标签的类型而定）
 - 支持 ISO 14443 第 4 部分 A 类和 B 类卡、MIFARE 卡、FeliCa 卡和全部 4 种 NFC（ISO/IEC 18092）标签
 - 内建防冲突特性（任何时候都只能访问 1 张标签）
 - 3 个符合 ISO 7816 标准的 SAM 卡槽
- 内置外围设备：
 - 2 行图形液晶显示屏，可进行交互操作（如上下、左右滚动）并支持多种语言（中文、英语、日语和多种欧洲语言）
 - 4 个用户可控的 LED 指示灯
 - 1 个用户可控的蜂鸣器
- 符合下列标准：
 - ISO 14443
 - CE
 - FCC
 - VCCI
 - RoHS



3.0. 应用程序编程接口概述

ACR122L 的 DLL 是一套供应用软件使用的高阶函数。它提供了一致的 API（应用程序编程接口）供各个程序调用，既可以操作 ACR122L，也可以操作相应的卡片。DLL 通过操作系统提供的通信端口与 ACR122L 进行通信。

ACR122L API 定义了访问 ACR122L 的通用方式。应用程序可以通过接口函数启用 ACR122L 并对出示的卡片进行操作。

我们提供了头文件“ACR122.h”供程序开发人员使用，其中包含此文档介绍的所有函数原型和宏。

3.1. 读写器

3.1.1. 预定义文档

3.1.1.1. ACR122_GetFirmwareVersion 和 ACR122_GetFirmwareVersionA

如果定义了 Unicode，ACR122_GetFirmwareVersion 函数将扩展为 ACR122_GetFirmwareVersionW() 函数。否则将扩展为 ACR122_GetFirmwareVersionA() 函数。

```
#define ACR122_GetFirmwareVersion ACR122_GetFirmwareVersionA
```

3.1.1.2. ACR122_Open 和 ACR122_OpenA

如果定义了 UNICODE，ACR122_Open 函数将扩展为 ACR122_OpenW() 函数。否则将扩展为 ACR122_OpenA() 函数。

```
#define ACR122_Open ACR122_OpenA
```

3.1.2. 函数文档

3.1.2.1. ACR122_OpenA

此函数用于打开读写器并返回一个句柄值作为引用。

```
DWORD WINAPI ACR122_OpenA ( LPCSTR portName,
                           LPHANDLE phReader
                           )
```

参数	说明
[in] portName	端口名称“\\.\COM1”表示读写器连接到 Windows 的 COM1。
[out] phReader	指向 HANDLE 变量的指针
返回值	ERROR_SUCCESS 操作成功完成。
	Failure 错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码



3.1.2.2. ACR122_OpenW

此函数用于打开读写器并返回一个句柄值作为引用。

```
DWORD WINAPI ACR122_OpenW ( LPCWSTR portName,
                             LPHANDLE phReader
                             )
```

参数	说明	
[in] portName	端口名称"\\.\COM1" 表示读写器连接到 Windows 的 COM1。	
[out] phReader	指向 HANDLE 变量的指针	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.1.2.3. ACR122_Close

此函数用于关闭读写器并释放资源。

```
DWORD WINAPI ACR122_Close ( HANDLE hReader
                             )
```

参数	说明	
[in] hReader	ACR122_Open () 函数返回的引用值	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.1.2.4. ACR122_GetNumSlots

此函数用于获取卡槽数。

```
DWORD WINAPI ACR122_GetNumSlots ( HANDLE hReader,
                                   LPDWORD pNumSlots
                                   )
```

参数	说明	
[in] hReader	ACR122_Open () 函数返回的引用值	
[out] pNumSlots	指向一个 DWORD 变量的指针，该变量用于接收返回的卡槽数	
返回值	ERROR_SUCCESS	操作成功完成。



参数	说明
Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.1.2.5. ACR122_GetBaudRate

此函数用于获取读写器的通信波特率。

```
DWORD WINAPI ACR122_GetBaudRate ( HANDLE hReader,
                                  LPDWORD pBaudRate
                                )
```

参数	说明	
[in] hReader	ACR122_Open() 函数返回的引用值	
[out] pBaudRate	指向一个 DWORD 变量的指针，该变量用于接收返回的通信波特率。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.1.2.6. ACR122_SetBaudRate

此函数用于设定读写器的通信波特率。ACR122L 支持 9600 bps 和 115200 bps 的通信波特率。

```
DWORD WINAPI ACR122_SetBaudRate ( HANDLE hReader,
                                   DWORD baudRate
                                 )
```

参数	说明	
[in] hReader	ACR122_Open() 函数返回的引用值	
[in] baudRate	波特率必须为 9600 bps 或 115200 bps。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码



3.1.2.7. ACR122_GetTimeouts

此函数用于获取读写器状态操作与应答操作的超时参数。

```
DWORD WINAPI ACR122_GetTimeouts ( HANDLE hReader,
                                  PACR122_TIMEOUTS pTimeouts
                                )
```

参数	说明	
[in] hReader	ACR122_Open () 函数返回的引用值	
[out] pTimeouts	指向一个 ACR122_TIMEOUTS 结构体的指针，该结构体用于接收返回的超时信息。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

注：关于 PACR122_TIMEOUTS，请参阅 附录 A.1 - _ACR122_TIMEOUTS 结构引用。

3.1.2.8. ACR122_SetTimeouts

此函数用于设定读写器状态操作与应答操作的超时参数。

```
DWORD WINAPI ACR122_SetTimeouts ( HANDLE hReader,
                                   const PACR122_TIMEOUTS pTimeouts
                                 )
```

参数	说明	
[in] hReader	ACR122_Open () 函数返回的引用值。	
[in] pTimeouts	指向一个 ACR122_TIMEOUTS 结构体的指针，该结构体内包含了新的超时值	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

注：关于 PACR122_TIMEOUTS，请参阅 附录 A.1 - _ACR122_TIMEOUTS 结构引用。



3.1.2.9. ACR122_GetFirmwareVersionA

此函数用于获取卡槽的固件版本号（以 ANSI 字符串形式表示）。

```
DWORD WINAPI ACR122_GetFirmwareVersionA ( HANDLE hReader,
                                         DWORD slotNum,
                                         LPSTR firmwareVersion,
                                         LPDWORD pFirmwareVersionLen
                                         )
```

参数	说明	
[in] hReader	ACR122_Open() 函数返回的引用值。	
[in] slotNum	卡槽号。	
[out] firmwareVersion	指向缓冲区的指针，该缓冲区用于接收从读写器返回的固件版本。	
[in,out] pFirmwareVersionLen	固件版本参数的长度（字节数），并且接收实际从读写器收到的字节数。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.1.2.10. ACR122_GetFirmwareVersionW

此函数用于获取卡槽的固件版本号（以 Unicode 字符串形式表示）。

```
DWORD WINAPI ACR122_GetFirmwareVersionW ( HANDLE hReader,
                                           DWORD slotNum,
                                           LPWSTR firmwareVersion,
                                           LPDWORD pFirmwareVersionLen
                                           )
```

参数	说明	
[in] hReader	ACR122_Open() 函数返回的引用值	
[in] slotNum	卡槽号。	
[out] firmwareVersion	指向缓冲区的指针，该缓冲区用于接收从读写器返回的固件版本。	
[in,out] pFirmwareVersionLen	固件版本参数的长度（字节数），并且接收实际从读写器收到的字节数。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码



3.2. LCD

3.2.1. 预定义文档

如果定义了 Unicode, ACR122_DisplayLcdMessage 函数将扩展为 ACR122_DisplayLcdMessageW() 函数。否则将扩展为 ACR122_DisplayLcdMessageA() 函数。

```
#define ACR122_DisplayLcdMessage ACR122_DisplayLcdMessageA
```

3.2.2. 函数文档

3.2.2.1. ACR122_DisplayLcdMessageA

此函数用于 ANSI 模式下字符消息在读写器 LCD 上的显示。

```
DWORD WINAPI ACR122_DisplayLcdMessageA ( HANDLE hReader,
                                         DWORD row,
                                         DWORD col,
                                         LPCSTR message
                                         )
```

参数	说明
[in] hReader	ACR122_Open() 函数返回的引用值
[in] row	行号 (0 - 1)。
[in] col	列号 (0 - 15)。
[in] message	要显示的消息。消息的长度必须小于等于 16 个字符。
返回值	ERROR_SUCCESS 操作成功完成。
	Failure 错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.2.2.2. ACR122_DisplayLcdMessageW

此函数用于 Unicode 模式下字符消息在读写器 LCD 上的显示。

```
DWORD WINAPI ACR122_DisplayLcdMessageW ( HANDLE hReader,
                                           DWORD row,
                                           DWORD col,
                                           LPCWSTR message
                                           )
```

参数	说明
[in] hReader	ACR122_Open() 函数返回的引用值。
[in] row	行号 (0 - 1)。
[in] col	列号 (0 - 15)。
[in] message	要显示的消息。消息的长度必须小于等于 16 个字符。



参数	说明	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.2.2.3. ACR122_DisplayLcdMessageExA

此函数用于 Ex (ANSI) 模式字符消息在读写器 LCD 上的显示。

```
DWORD WINAPI ACR122_DisplayLcdMessageExA ( HANDLE hReader,
                                           DWORD row,
                                           DWORD col,
                                           DWORD TableIndex,
                                           DWORD BoldMode,
                                           LPCSTR message
                                           )
```

参数	说明	
[in] hReader	ACR122_Open () 函数返回的引用值	
[in] row	行号 (0 - 1)。	
[in] col	列号 (0 - 15)。	
[in] TableIndex	选择要使用的字库 (0 - 2)	
[in] BoldMode	1 - 加粗, 0 - 常规	
[in] message	要显示的消息。消息的长度必须小于等于 16 个字符。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.2.2.4. ACR122_DisplayLcdMessageExW

此函数用于 Ex (Unicode) 模式字符消息在读写器 LCD 上的显示。

```
DWORD WINAPI ACR122_DisplayLcdMessageExW ( HANDLE hReader,
                                           DWORD row,
                                           DWORD col,
                                           DWORD TableIndex,
                                           DWORD BoldMode,
                                           LPCWSTR message
                                           )
```

参数	说明
[in] hReader	ACR122_Open () 函数返回的引用值
[in] row	行号 (0 - 1)。



参数	说明	
[in] col	列号 (0 - 15)。	
[in] TableIndex	选择要使用的字库 (0 - 2)	
[in] BoldMode	1 – 加粗, 0 – 常规	
[in] message	要显示的消息。消息的长度必须小于等于 16 个字符。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.2.2.5. ACR122_DisplayLcdMessageGBA

此函数用于 GB (ANSI) 模式字符消息在读写器 LCD 上的显示 (简体中文)。

```
DWORD WINAPI ACR122_DisplayLcdMessageGBA ( HANDLE hReader,
                                           DWORD row,
                                           DWORD col,
                                           DWORD BoldMode,
                                           LPCSTR message
                                           )
```

参数	说明	
[in] hReader	ACR122_Open () 函数返回的引用值	
[in] row	行号 (0 - 1)。	
[in] col	列号 (0 - 7)。	
[in] BoldMode	1 – 加粗, 0 – 常规	
[in] message	要显示的消息。消息的长度必须小于等于 8 个字符。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.2.2.6. ACR122_DisplayLcdMessageGBW

此函数用于 GB (Unicode) 模式字符消息在读写器 LCD 上的显示 (简体中文)。

```
DWORD WINAPI ACR122_DisplayLcdMessageGBA ( HANDLE hReader,
                                           DWORD row,
                                           DWORD col,
                                           DWORD BoldMode,
                                           LPCWSTR message
                                           )
```



参数	说明
[in] hReader	ACR122_Open () 函数返回的引用值
[in] row	行号 (0 - 1)。
[in] col	列号 (0 - 7)。
[in] BoldMode	1 – 加粗, 0 – 常规
[in] message	要显示的消息。消息的长度必须小于等于 8 个字符。
返回值	ERROR_SUCCESS 操作成功完成。
	Failure 错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.2.2.7. ACR122_DrawLcd

此函数用于图形模式字符消息在读写器 LCD 上的显示。

```

DWORD WINAPI ACR122_DrawLcd ( HANDLE hReader,
                              DWORD lineIndex,
                              const LPBYTE pixelBuffer,
                              DWORD pixelBufferLen
                              )

```

参数	说明
[in] hReader	ACR122_Open () 函数返回的引用值
[in] lineIndex	设置从 LCD 显示屏上的哪一行开始更新
[in] pixelBuffer	缓冲区, 含有待显示的像素数据, 长度不得超过 128 字节
[in] pixelBufferLen	像素数据缓冲区的字节长度
返回值	ERROR_SUCCESS 操作成功完成。
	Failure 错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码



3.2.2.8. ACR122_StartLcdScrolling

此函数用于设置当前 LCD 显示屏的字符滚动功能。

```
DWORD WINAPI ACR122_StartLcdScrolling ( HANDLE hReader,
                                       DWORD row,
                                       DWORD col,
                                       DWORD HRange,
                                       DWORD VRange,
                                       DWORD Speed,
                                       DWORD Direction
                                       )
```

参数	说明	
[in] hReader	ACR122_Open () 函数返回的引用值	
[in] row	行号必须为 0。	
[in] col	列号必须是 0。	
[in] Hrange	水平范围（单位：8 像素）。值必须为 15。	
[in] Vrange	垂直范围（单位：1 像素）。值必须为 31。	
[in] Speed	Bit 0 – Bit 3: 滚动前的像素数。 Bit 4 – Bit 7: 滚动周期。	
[in] Direction	滚动方向，从 0 到 3	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

[in] Speed

值 (Bit 4 – Bit 7)	滚动周期
0	1 Unit
1	3 Units
2	5 Units
3	7 Units
4	17 Units
5	19 Units
6	21 Units
7	23 Units
8	129 Units
9	131 Units
10	133 Units
11	135 Units
12	145 Units



值 (Bit 4 – Bit 7)	滚动周期
13	147 Units
14	149 Units
15	151 Units

[in] Direction

值	含义
0	从左至右
1	从右至左
2	从上至下
3	从下至上

3.2.2.9. ACR122_PauseLcdScrolling

此函数用于暂停之前设置的 LCD 字符滚动功能。

DWORD WINAPI ACR122_PauseLcdScrolling (HANDLE hReader)

参数	说明
[in] hReader	ACR122_Open () 函数返回的引用值
返回值	ERROR_SUCCESS 操作成功完成。
	Failure 错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.2.2.10. ACR122_StopLcdScrolling

此函数用于停止之前设置的 LCD 字符滚动功能，之后 LCD 显示屏恢复到正常显示状态。

DWORD WINAPI ACR122_StopLcdScrolling (HANDLE hReader)

参数	说明
[in] hReader	ACR122_Open () 函数返回的引用值
返回值	ERROR_SUCCESS 操作成功完成。
	Failure 错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码



3.2.2.11. ACR122_ClearLcd

此函数用于清除 LCD 上显示的内容。

```
DWORD WINAPI ACR122_ClearLcd ( HANDLE hReader )
```

参数	说明	
[in] hReader	ACR122_Open () 函数返回的引用值	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

注：若 ACR122L 的固件版本为 307 及以上，在执行 ACR122L_ClearLCD 命令后立刻使用其他功能的情况下，应用需要处理额外 100 ms 的延迟时间。

3.2.2.12. ACR122_EnableLcdBacklight

此函数用于开启/关闭读写器 LCD 的背光。

```
DWORD WINAPI ACR122_EnableLcdBacklight ( HANDLE hReader,
                                          BOOL enabled
                                          )
```

参数	说明	
[in] hReader	ACR122_Open () 函数返回的引用值	
[in] enabled	设置为 TRUE 可开启背光，否则设置为 FALSE。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.2.2.13. ACR122_SetLcdContrast

此函数用于设置读写器 LCD 的对比度。

```
DWORD WINAPI ACR122_SetLcdContrast ( HANDLE hReader,
                                       DWORD level
                                       )
```

参数	说明	
[in] hReader	ACR122_Open () 函数返回的引用值。	
[in] level	对比度，区间范围是 00h 至 0Fh。	
返回值	ERROR_SUCCESS	操作成功完成。



参数	说明
Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码



3.3. LED

3.3.1. 函数文档

3.3.1.1. ACR122_EnableLed

此函数用于启用/禁止应用对 LED 进行控制。默认情况下，LED 通过固件进行控制。在调用 ACR122_SetLedStatesWithBeep() 和 ACR122_SetLedStates() 之前，应用需要调用此函数才能够控制 LED。

```
DWORD WINAPI ACR122_EnableLed ( HANDLE hReader,
                                BOOL   enabled
                                )
```

参数	说明	
[in] hReader	ACR122_Open() 函数返回的引用值	
[in] enabled	设置为 TRUE 可开启 LED 控制功能，否则设置为 FALSE。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.3.1.2. ACR122_SetLedStates

此函数用于开启/关闭读写器的 LED。通过它可以控制 LED0、LED1、LED2 和 LED3。

```
DWORD WINAPI ACR122_SetLedStates ( HANDLE hReader,
                                   DWORD *states,
                                   DWORD numStates
                                   )
```

参数	说明	
[in] hReader	ACR122_Open() 函数返回的引用值	
[in] states	指向状态数组的指针。可能的值是 ACR122_LED_STATE_OFF 和 ACR122_LED_STATE_ON。	
[in] numStates	状态数量必须为 4。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码



3.3.1.3. ACR122_SetLedStatesWithBeep

此函数用于控制读写器 LED0、LED1 和蜂鸣器的操作。

```
DWORD WINAPI ACR122_SetLedStatesWithBeep ( HANDLE           hReader,
                                           PACR122_LED_CONTROL controls,
                                           DWORD           numControls,
                                           DWORD           t1,
                                           DWORD           t2,
                                           DWORD           numTimes,
                                           DWORD           buzzerMode
                                           )
```

参数	说明
[in] hReader	ACR122_Open() 函数返回的引用值
[in] controls	指向 ACR122_LED_CONTROL 数据结构的指针。
[in] numControls	控制器的数量必须为 2。
[in] t1	T1 周期，单位为毫秒，值必须从 0 至 25500。
[in] t2	T2 周期，单位为毫秒，值必须从 0 至 25500。
[in] numTimes	次数。值必须从 0 至 255。
[in] buzzerMode	蜂鸣器状态掩码。可能值可以结合 OR 操作。

值	含义
ACR122_BUZZER_MODE_OFF	蜂鸣器不会开启。
ACR122_BUZZER_MODE_ON_T1	蜂鸣器会在 T1 周期内开启。
ACR122_BUZZER_MODE_ON_T2	蜂鸣器会在 T2 周期内开启。

返回值	说明
ERROR_SUCCESS	操作成功完成。
Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

注: 关于 PACR122_LED_CONTROL, 请参阅附录 A 附录 A.2_ACR122_LED_CONTROL 结构引用。



3.4. 蜂鸣器

3.4.1. 函数文档

3.4.1.1. ACR122_Beep

此函数用于控制蜂鸣器发出蜂鸣声，并且控制权直到声音结束了才会回到调用者手中。

```
DWORD WINAPI ACR122_Beep ( HANDLE hReader,
                          DWORD buzzerOnDuration,
                          DWORD buzzerOffDuration,
                          DWORD numTimes
                          )
```

参数	说明	
[in] hReader	ACR122_Open() 函数返回的引用值	
[in] buzzerOnDuration	蜂鸣器鸣响时间，单位为毫秒，值必须从 0 至 25500。	
[in] buzzerOffDuration	蜂鸣器关闭时间，单位为毫秒，值必须从 0 至 25500。	
[in] numTimes	次数。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.5. 卡片

3.5.1. 函数文档

3.5.1.1. ACR122_DirectTransmit

此函数用于发送标签命令，以及接收由读写器返回的响应。

```

DWORD WINAPI ACR122_DirectTransmit ( HANDLE           hReader,
                                     const LPBYTE     sendBuffer,
                                     DWORD             sendBufferLen,
                                     LPBYTE           recvBuffer,
                                     LPDWORD          pRecvBufferLen
                                     )
    
```

参数	说明	
[in] hReader	ACR122_Open() 函数返回的引用值	
[in] sendBuffer	指向实际要写入卡片的数据的指针	
[in] sendBufferLen	sendBuffer 参数的长度（字节数）	
[in] recvBuffer	指向卡片返回的数据的指针	
[in,out] pRecvBufferLen	recvBuffer 参数的长度（字节数），并且接收实际从卡片收到的字节数。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.5.1.2. ACR122_ExchangeApdu

此函数用于发送 APDU 命令，以及接收由卡片返回的 APDU 响应。

```

DWORD WINAPI ACR122_ExchangeApdu ( HANDLE           hReader,
                                    DWORD             slotNum,
                                    const LPBYTE     sendBuffer,
                                    DWORD             sendBufferLen,
                                    LPBYTE           recvBuffer,
                                    LPDWORD          pRecvBufferLen
                                    )
    
```

参数	说明
[in] hReader	ACR122_Open() 函数返回的引用值
[in] slotNum	卡槽号。
[in] sendBuffer	指向实际要写入卡片的数据的指针
[in] sendBufferLen	sendBuffer 参数的长度（字节数）



参数	说明	
[out] recvBuffer	指向卡片返回的数据的指针	
[in, out] pRecvBufferLen	参数的长度（字节数），并且接收实际从卡片收到的字节数。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.5.1.3. ACR122_PowerOffIcc

此函数用于对卡槽中的 ICC 进行下电。

```
DWORD WINAPI ACR122_PowerOffIcc ( HANDLE hReader,
                                  DWORD slotNum
                                )
```

参数	说明	
[in] hReader	ACR122_Open() 函数返回的引用值	
[in] slotNum	卡槽号。	
返回值	ERROR_SUCCESS	操作成功完成。
	Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码

3.5.1.4. ACR122_PowerOnIcc

此函数用于对卡槽中的卡片进行上电，同时返回卡片的 ATR 字符串。

```
DWORD WINAPI ACR122_PowerOnIcc ( HANDLE hReader,
                                  DWORD slotNum,
                                  LPBYTE atr,
                                  LPDWORD pAttrLen
                                )
```

参数	说明	
[in] hReader	ACR122_Open() 函数返回的引用值	
[in] slotNum	卡槽号。	
[out] atr	指向缓冲区的指针，该缓冲区用于接收从卡片返回的 ATR 字符串。	
[in, out] pAttrLen	atr 参数的长度（字节数），并且接收实际从卡片收到的字节数。	
返回值	ERROR_SUCCESS	操作成功完成。



参数	说明
Failure	错误代码。请参阅 Windows API 错误代码和 ACR122 错误代码



附录A. 数据结构

附录A.1 **_ACR122_TIMEOUTS** 结构引用

_ACR122_TIMEOUTS 数据结构主要用于 **ACR122_GetTimeouts()** 和 **ACR122_SetTimeouts()** 函数。

- `DWORD _ACR122_TIMEOUTS::numResponseRetries`
应答重试次数。
默认为 1 次。
- `DWORD _ACR122_TIMEOUTS::numStatusRetries`
状态重试次数。
默认为 1 次。
- `DWORD _ACR122_TIMEOUTS::responseTimeout`
应答超时值，单位为毫秒。
默认为 10000 毫秒。
- `DWORD _ACR122_TIMEOUTS::statusTimeout`
状态超时值，单位为毫秒。
默认为 2000 毫秒。

附录A.2 **_ACR122_LED_CONTROL** 结构引用

此 **_ACR122_LED_CONTROL** 数据结构主要用于 **ACR122_SetLedStatesWithBeep()** 函数。

- `BOOL _ACR122_LED_CONTROL::blinkEnabled`
启用闪烁。
设置为 `TRUE` 可开启 LED 闪烁功能，否则设置为 `FALSE`。
- `DWORD _ACR122_LED_CONTROL::finalState`
最终状态。
可能的值是 `ACR122_LED_STATE_OFF` 和 `ACR122_LED_STATE_ON`。
- `DWORD _ACR122_LED_CONTROL::initialBlinkingState`
初始闪烁状态。
可能的值是 `ACR122_LED_STATE_OFF` 和 `ACR122_LED_STATE_ON`。
- `BOOL _ACR122_LED_CONTROL::updateEnabled`
启用更新。
设置为 `TRUE` 可以更新状态。否则设置为 `FALSE`。



附录B. 高阶 API 返回的错误代码

- ACR122_ERROR_NO_MORE_HANDLES ((DWORD) 0x20000001L)
句柄无效。
- ACR122_ERROR_UNKNOWN_STATUS ((DWORD) 0x20000002L)
读写器出现未知错误。
- ACR122_ERROR_OPERATION_FAILURE ((DWORD) 0x20000003L)
操作失败。
- ACR122_ERROR_OPERATION_TIMEOUT ((DWORD) 0x20000004L)
超时操作。
- ACR122_ERROR_INVALID_CHECKSUM ((DWORD) 0x20000005L)
校验和计算错误。
- ACR122_ERROR_INVALID_PARAMETER ((DWORD) 0x20000006L)
不正确的参数输入。



附录C. 标准的程序流

```
// ACR122L Test Program.cpp :Defines the entry point for the console
application.
//
#include "stdafx.h"
#include "acr122.h"
#include "conio.h"

int _tmain(int argc, _TCHAR* argv[])
{
    DWORD Error;
    DWORD i;

    HANDLE hReader;

    TCHAR Temp[255];

    DWORD LED_Status[4];
    ACR122_LED_CONTROL LED_CTRL[2];

    BYTE *cmd;

    BYTE PollCmd[] = {0xD4,0x4A,0x01,0x00};
    BYTE GetChallenge[] = {0x80,0x84,0x00,0x00,0x08};
    BYTE RecBuff[255];

    DWORD RecLen;

    TCHAR DisplayMessage1[] = _T("ACR122L TESTING ");
    TCHAR DisplayMessage2[] = _T("123456789ABCDEF@");

    TCHAR *FW_Ver;
    DWORD FW_Len;

    FW_Ver = Temp;
    FW_Len = 255;

    printf("1) Open COM1\n");
    printf("=====\n");
    Error = ACR122_Open(_T("\\\\.\\COM1"), &hReader);
    printf("Error Code:%d\n", Error);
    printf("Press any key to continuou\n");
    getch();
    printf("\n");

    printf("2) Test Firmware Version Slot 0\n");
    printf("=====\n");
    Error = ACR122_GetFirmwareVersion(hReader, 0, FW_Ver, &FW_Len);

    printf("Error Code:%d\n", Error);

    if(Error == 0)
    {
        printf("Firmware Code:");
        for (i=0; i<FW_Len; i++)
            printf("%s", Temp+i);
        printf("\n");
    }
    printf("Press any key to continuou\n");
}
```



```
getch();
printf("\n");

printf("3) Test Firmware Version Slot 1\n");
printf("=====\n");
Error = ACR122_GetFirmwareVersion(hReader,1,FW_Ver,&FW_Len);

printf("Error Code:%d\n",Error);

if(Error == 0)
{
    printf("Firmware Code:");
    for (i=0;i<FW_Len;i++)
        printf("%s",Temp+i);
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("4) Test Firmware Version Slot 2\n");
printf("=====\n");
Error = ACR122_GetFirmwareVersion(hReader,2,FW_Ver,&FW_Len);

printf("Error Code:%d\n",Error);

if(Error == 0)
{
    printf("Firmware Code:");
    for (i=0;i<FW_Len;i++)
        printf("%s",Temp+i);
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("5) Display First Row LCD Message\n");
printf("=====\n");
Error = ACR122_DisplayLcdMessage(hReader,0,0,DisplayMessage1);
printf("Error Code:%d\n",Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("6) Display Second Row LCD Message\n");
printf("=====\n");
Error = ACR122_DisplayLcdMessage(hReader,1,0,DisplayMessage2);
printf("Error Code:%d\n",Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("7) Turn On Backlight\n");
printf("=====\n");
Error = ACR122_EnableLcdBacklight(hReader,true);
printf("Error Code:%d\n",Error);
printf("Press any key to continuou\n");
getch();
printf("\n");
```



```
printf("8) Turn Off Backlight\n");
printf("=====\n");
Error = ACR122_EnableLcdBacklight(hReader, false);
printf("Error Code:%d\n", Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("9) Clear LCD\n");
printf("=====\n");
Error = ACR122_ClearLcd(hReader);
printf("Error Code:%d\n", Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

Error = ACR122_EnableLed(hReader, true);

printf("10) Turn ON LED0 \n");
printf("=====\n");

LED_Status[0] = ACR122_LED_STATE_ON;
LED_Status[1] = ACR122_LED_STATE_OFF;
LED_Status[2] = ACR122_LED_STATE_OFF;
LED_Status[3] = ACR122_LED_STATE_OFF;

Error = ACR122_SetLedStates(hReader, LED_Status, 4);
printf("Error Code:%d\n", Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("11) Turn ON LED1 \n");
printf("=====\n");

LED_Status[0] = ACR122_LED_STATE_OFF;
LED_Status[1] = ACR122_LED_STATE_ON;
LED_Status[2] = ACR122_LED_STATE_OFF;
LED_Status[3] = ACR122_LED_STATE_OFF;

Error = ACR122_SetLedStates(hReader, LED_Status, 4);
printf("Error Code:%d\n", Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("12) Turn ON LED2 \n");
printf("=====\n");

LED_Status[0] = ACR122_LED_STATE_OFF;
LED_Status[1] = ACR122_LED_STATE_OFF;
LED_Status[2] = ACR122_LED_STATE_ON;
LED_Status[3] = ACR122_LED_STATE_OFF;

Error = ACR122_SetLedStates(hReader, LED_Status, 4);
printf("Error Code:%d\n", Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("13) Turn ON LED3 \n");
```



```
printf("=====\n");

LED_Status[0] = ACR122_LED_STATE_OFF;
LED_Status[1] = ACR122_LED_STATE_OFF;
LED_Status[2] = ACR122_LED_STATE_OFF;
LED_Status[3] = ACR122_LED_STATE_ON;

Error = ACR122_SetLedStates(hReader, LED_Status, 4);
printf("Error Code:%d\n", Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("14) Turn ON All LEDs \n");
printf("=====\n");

LED_Status[0] = ACR122_LED_STATE_ON;
LED_Status[1] = ACR122_LED_STATE_ON;
LED_Status[2] = ACR122_LED_STATE_ON;
LED_Status[3] = ACR122_LED_STATE_ON;

Error = ACR122_SetLedStates(hReader, LED_Status, 4);
printf("Error Code:%d\n", Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("15) Turn OFF All LEDs \n");
printf("=====\n");

LED_Status[0] = ACR122_LED_STATE_OFF;
LED_Status[1] = ACR122_LED_STATE_OFF;
LED_Status[2] = ACR122_LED_STATE_OFF;
LED_Status[3] = ACR122_LED_STATE_OFF;

Error = ACR122_SetLedStates(hReader, LED_Status, 4);
printf("Error Code:%d\n", Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("16) Buzzer Test\n");
printf("=====\n");
Error = ACR122_Beep(hReader, 500, 500, 2);
printf("Error Code:%d\n", Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("17) Set LED with Buzzer \n");
printf("=====\n");
LED_Status[0] = ACR122_LED_STATE_OFF;
LED_Status[1] = ACR122_LED_STATE_OFF;
LED_Status[2] = ACR122_LED_STATE_OFF;
LED_Status[3] = ACR122_LED_STATE_OFF;

LED_CTRL[0].blinkEnabled = true;
LED_CTRL[0].finalState = ACR122_LED_STATE_OFF;
LED_CTRL[0].initialBlinkingState = ACR122_LED_STATE_ON;
LED_CTRL[0].updateEnabled = true;
```



```
LED_CTRL[1].blinkEnabled = true;
LED_CTRL[1].finalState = ACR122_LED_STATE_OFF;
LED_CTRL[1].initialBlinkingState = ACR122_LED_STATE_OFF;
LED_CTRL[1].updateEnabled = true;

Error
ACR122_SetLedStatesWithBeep(hReader, LED_CTRL, 2, 500, 500, 3, ACR122_BUZZER_M
ODE_ON_T1);
printf("Error Code:%d\n", Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

Error = ACR122_EnableLed(hReader, false);

printf("18) Direct Command - Poll Type A Command \n");
printf("=====\n");
cmd = PollCmd;
RecLen = 255;
Error = ACR122_DirectTransmit(hReader, cmd, 4, RecBuff, &RecLen);
printf("Error Code:%d\n", Error);

if(Error == 0)
{
    printf("Data Return:");
    for (i=0; i<RecLen; i++)
    {
        printf("%02X", RecBuff[i]);
        printf(" ");
    }
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("19) Activate SAM A\n");
printf("=====\n");
RecLen = 255;
Error = ACR122_PowerOnIcc(hReader, 0, RecBuff, &RecLen);
printf("Error Code:%d\n", Error);

if(Error == 0)
{
    printf("Data Return:");
    for (i=0; i<RecLen; i++)
    {
        printf("%02X", RecBuff[i]);
        printf(" ");
    }
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("20) Activate SAM B\n");
printf("=====\n");
RecLen = 255;
Error = ACR122_PowerOnIcc(hReader, 1, RecBuff, &RecLen);
printf("Error Code:%d\n", Error);
```



```
if(Error == 0)
{
    printf("Data Return:");
    for (i=0;i<RecLen;i++)
    {
        printf("%02X",RecBuff[i]);
        printf(" ");
    }
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("21) Activate SAM C\n");
printf("=====\n");
RecLen = 255;
Error = ACR122_PowerOnIcc(hReader,2,RecBuff,&RecLen);
printf("Error Code:%d\n",Error);

if(Error == 0)
{
    printf("Data Return:");
    for (i=0;i<RecLen;i++)
    {
        printf("%02X",RecBuff[i]);
        printf(" ");
    }
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("22) Get Challenge SAM A\n");
printf("=====\n");
cmd = GetChallenge;
RecLen = 255;
Error = ACR122_ExchangeApdu(hReader,0,cmd,5,RecBuff,&RecLen);
printf("Error Code:%d\n",Error);

if(Error == 0)
{
    printf("Data Return:");
    for (i=0;i<RecLen;i++)
    {
        printf("%02X",RecBuff[i]);
        printf(" ");
    }
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("23) Get Challenge SAM B\n");
printf("=====\n");
cmd = GetChallenge;
RecLen = 255;
Error = ACR122_ExchangeApdu(hReader,1,cmd,5,RecBuff,&RecLen);
```



```
printf("Error Code:%d\n",Error);
if(Error == 0)
{
    printf("Data Return:");
    for (i=0;i<RecLen;i++)
    {
        printf("%02X",RecBuff[i]);
        printf(" ");
    }
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("24) Get Challenge SAM C\n");
printf("=====\n");
cmd = GetChallenge;
RecLen = 255;
Error = ACR122_ExchangeApdu(hReader,2,cmd,5,RecBuff,&RecLen);
printf("Error Code:%d\n",Error);

if(Error == 0)
{
    printf("Data Return:");
    for (i=0;i<RecLen;i++)
    {
        printf("%02X",RecBuff[i]);
        printf(" ");
    }
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("25) Deactivate SAM A\n");
printf("=====\n");
Error = ACR122_PowerOffIcc(hReader,0);
printf("Error Code:%d\n",Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("26) Deactivate SAM B\n");
printf("=====\n");
Error = ACR122_PowerOffIcc(hReader,1);
printf("Error Code:%d\n",Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("27) Deactivate SAM C\n");
printf("=====\n");
Error = ACR122_PowerOffIcc(hReader,2);
printf("Error Code:%d\n",Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("28) Change Baud Rate to 115200\n");
```




```
printf("=====\n");
Error = ACR122_SetBaudRate(hReader,115200);
printf("Error Code:%d\n",Error);
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("29) Test Firmware Version Slot 0\n");
printf("=====\n");
Error = ACR122_GetFirmwareVersion(hReader,0,FW_Ver,&FW_Len);

printf("Error Code:%d\n",Error);

if(Error == 0)
{
    printf("Firmware Code:");
    for (i=0;i<FW_Len;i++)
        printf("%s",Temp+i);
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("30) Test Firmware Version Slot 1\n");
printf("=====\n");
Error = ACR122_GetFirmwareVersion(hReader,1,FW_Ver,&FW_Len);

printf("Error Code:%d\n",Error);

if(Error == 0)
{
    printf("Firmware Code:");
    for (i=0;i<FW_Len;i++)
        printf("%s",Temp+i);
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

printf("31) Test Firmware Version Slot 2\n");
printf("=====\n");
Error = ACR122_GetFirmwareVersion(hReader,2,FW_Ver,&FW_Len);

printf("Error Code:%d\n",Error);
if(Error == 0)
{
    printf("Firmware Code:");
    for (i=0;i<FW_Len;i++)
        printf("%s",Temp+i);
    printf("\n");
}
printf("Press any key to continuou\n");
getch();
printf("\n");

ACR122_Close(hReader);

return 0;
}
```