



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR122L Serial NFC Reader with LCD



Application Programming Interface V1.03



Table of Contents

| | | |
|--------------------|---|-----------|
| 1.0. | Introduction | 3 |
| 2.0. | Features | 4 |
| 3.0. | Application Programming Interface Overview | 5 |
| 3.1. | Reader | 5 |
| 3.1.1. | Define Documentation | 5 |
| 3.1.2. | Function Documentation | 5 |
| 3.2. | LCD | 10 |
| 3.2.1. | Define Documentation | 10 |
| 3.2.2. | Function Documentation | 10 |
| 3.3. | LED | 18 |
| 3.3.1. | Function Documentation | 18 |
| 3.4. | Buzzer..... | 20 |
| 3.4.1. | Function Documentation | 20 |
| 3.5. | Card | 21 |
| 3.5.1. | Function Documentation | 21 |
| Appendix A. | Data Structures..... | 24 |
| Appendix A.1. | _ACR122_TIMEOUTS Struct Reference | 24 |
| Appendix A.2. | _ACR122_LED_CONTROL Struct Reference | 24 |
| Appendix B. | Error Codes returned by High-level APIs..... | 25 |
| Appendix C. | Standard Program Flow | 26 |

List of Figures

| | | |
|-------------------|---|----------|
| Figure 1 : | ACR122L Library Architecture | 3 |
|-------------------|---|----------|



1.0. Introduction

This API document describes the use of ACR122L interface software to facilitate application development with the ACR122L reader. This interface software is supplied in the form of 32-bit and 64-bit DLL (Dynamic Link Library) which can be programmed using popular development tools like Java, Delphi, Visual Basic, Visual C++, Visual C# and Visual Basic .NET.

ACR122L can be connected to the PC via the RS-232 interface.

The architecture of the ACR122L library can be visualized as the following diagram:

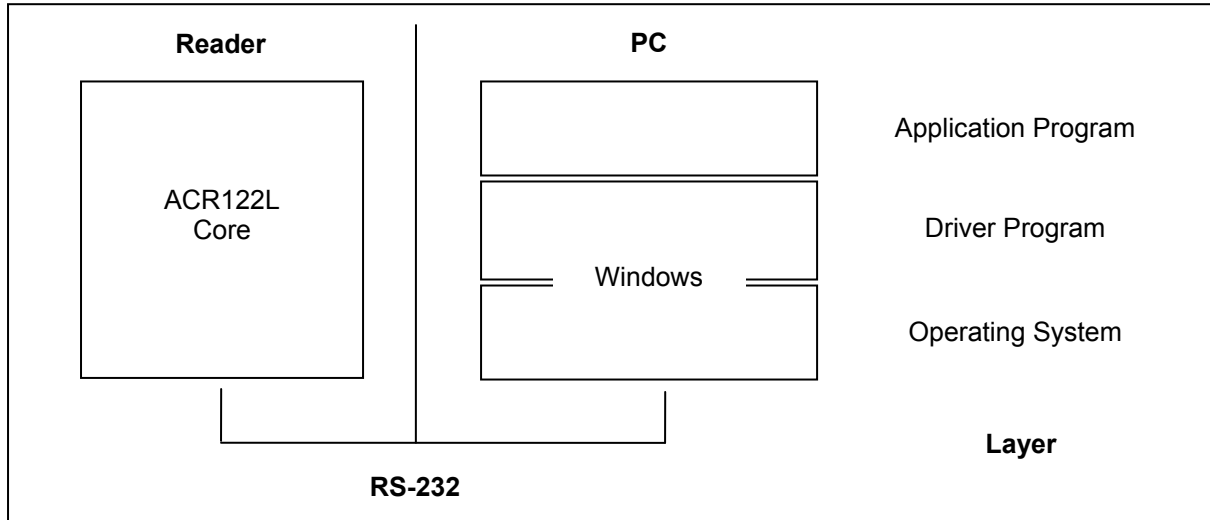


Figure 1: ACR122L Library Architecture



2.0. Features

- Serial RS-232 Interface: Baud Rate = 115200 bps, 8-N-1
- 7 V DC adaptor for power supply
- CCID-like frame format (Binary format)
- Smart Card Reader:
 - Read/Write speed of up to 424 kbps
 - Built-in antenna for contactless tag access, with card reading distance of up to 50 mm (depending on tag type)
 - Supports ISO 14443 Part 4 Type A and B cards, Mifare, FeliCa and all four types of NFC (ISO/IEC 18092) tags
 - Built-in anti-collision feature (only one tag is accessed at any time)
 - Three ISO 7816-compliant SAM slots
- Built-in Peripherals:
 - Two-line graphic LCD with interactive operability (i.e. scroll up and down, left and right, etc.) and multi-language support (i.e. Chinese, English, Japanese and several European languages)
 - Four user-controllable LEDs
 - User-controllable buzzer
- Compliant with the following standards:
 - ISO 14443
 - CE
 - FCC
 - VCCI
 - RoHS



3.0. Application Programming Interface Overview

The ACR122L DLL is a set of high-level functions provided for the application software to use. It supplies a consistent API (Application Programming Interface) for the application to operate on the ACR122L and the corresponding presented card. The DLL communicates with the ACR122L via the communication port facilities provided by the operating system.

The ACR122L API defines a common way of accessing the ACR122L. Application programs invoke the ACR122L through the interface functions and perform operations on the presented card.

The header file ACR122.h is available for the program developer, which contains all the function prototypes and macros described below.

3.1. Reader

3.1.1. Define Documentation

3.1.1.1. ACR122_GetFirmwareVersion and ACR122_GetFirmwareVersionA

ACR122_GetFirmwareVersion will be mapped to ACR122_GetFirmwareVersionW() function if Unicode is defined. Otherwise, it will be mapped to ACR122_GetFirmwareVersionA() function.

```
#define ACR122_GetFirmwareVersion ACR122_GetFirmwareVersionA
```

3.1.1.2. ACR122_Open and ACR122_OpenA

ACR122_Open will be mapped to ACR122_OpenW() function if UNICODE is defined. Otherwise, it will be mapped to ACR122_OpenA() function.

```
#define ACR122_Open ACR122_OpenA
```

3.1.2. Function Documentation

3.1.2.1. ACR122_OpenA

This function is used to open the reader and return a handle value as a reference.

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

| Parameter | Description | |
|----------------|--|--|
| [in] portName | Port name. "\\.\COM1" means that the reader is connected to COM1 in Windows. | |
| [out] phReader | Pointer to the HANDLE variable. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |



3.1.2.2. ACR122_OpenW

This function is used to open a reader and return a handle value as reference.

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

| Parameter | Description | |
|---------------------|--|--|
| [in] portName | Port name. "\\.\COM1" means that the reader is connected to COM1 in Windows. | |
| [out] phReader | Pointer to the HANDLE variable. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.1.2.3. ACR122_Close

This function is used to close the reader and release the resources.

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

| Parameter | Description | |
|---------------------|--|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.1.2.4. ACR122_GetNumSlots

This function is used to retrieve the number of slots.

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

| Parameter | Description |
|-----------------|---|
| [in] hReader | A reference value returned from ACR122_Open () function. |
| [out] pNumSlots | Pointer to a DWORD variable in which the number of slots is returned. |



| Parameter | Description | |
|---------------------|---------------|--|
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.1.2.5. ACR122_GetBaudRate

This function is used to retrieve the baud rate of reader.

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

| Parameter | Description | |
|---------------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [out] pBaudRate | Pointer to a DWORD variable in which the baud rate is returned. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.1.2.6. ACR122_SetBaudRate

This function is used to set the communication baud rate of reader. The reader supports 9600 bps and 115200 bps.

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

| Parameter | Description | |
|---------------------|--|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] baudRate | Baud rate must be 9600 bps or 115200 bps. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |



3.1.2.7. ACR122_GetTimeouts

This function is used to retrieve the timeout parameters for status and response operations of the reader.

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

| Parameter | Description | |
|---------------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open() function. | |
| [out] pTimeouts | Pointer to a PACR122_TIMEOUTS structure in which the timeout information is returned. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

Note: For PACR122_TIMEOUTS, please see **Appendix A.1 _ACR122_TIMEOUTS Struct Reference.**

3.1.2.8. ACR122_SetTimeouts

This function is used to set the timeout parameters for status and response operations on the reader.

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

| Parameter | Description | |
|---------------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open() function. | |
| [in] pTimeouts | Pointer to a PACR122_TIMEOUTS structure that contains the new timeout values. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

Note: For PACR122_TIMEOUTS, please see **Appendix A.1 _ACR122_TIMEOUTS Struct Reference.**



3.1.2.9. ACR122_GetFirmwareVersionA

This function is used to retrieve the firmware version in ANSI string of the slot.

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

| Parameter | Description | |
|------------------------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] slotNum | Slot number. | |
| [out] firmwareVersion | A pointer to the buffer that receives the firmware version returned from the reader. | |
| [in,out] pFirmwareVersionLen | The length in number of bytes of the firmware version parameter and receives the actual number of bytes received from the reader. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.1.2.10. ACR122_GetFirmwareVersionW

This function is used to retrieve the firmware version in Unicode string of the slot.

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

| Parameter | Description | |
|------------------------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] slotNum | Slot number. | |
| [out] firmwareVersion | A pointer to the buffer that receives the firmware version returned from the reader. | |
| [in,out] pFirmwareVersionLen | The length in number of bytes of the firmware version parameter and receives the actual number of bytes received from the reader. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |



3.2. LCD

3.2.1. Define Documentation

ACR122_DisplayLcdMessage will be mapped to ACR122_DisplayLcdMessageW () function if Unicode is defined. Otherwise, it will be mapped to ACR122_DisplayLcdMessageA () function.

```
#define ACR122_DisplayLcdMessage ACR122_DisplayLcdMessageA
```

3.2.2. Function Documentation

3.2.2.1. ACR122_DisplayLcdMessageA

This function is used to display LCD message on the reader in ANSI mode.

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

| Parameter | Description | |
|---------------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] row | Row number must be from 0 to 1. | |
| [in] col | Column number must be from 0 to 15. | |
| [in] message | Message for display. The length of message must be less than or equal to 16 characters. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.2.2.2. ACR122_DisplayLcdMessageW

This function is used to display LCD message on the reader in Unicode mode.

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

| Parameter | Description |
|--------------|---|
| [in] hReader | A reference value returned from ACR122_Open() function. |
| [in] row | Row number must be from 0 to 1. |
| [in] col | Column number must be from 0 to 15. |



| Parameter | Description | |
|---------------------|---|--|
| [in] message | Message for display. The length of message must be less than or equal to 16 characters. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.2.2.3. ACR122_DisplayLcdMessageExA

This function is used to display the LCD message on the reader in Ex (ANSI) mode.

```

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

```

| Parameter | Description | |
|---------------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] row | Row number must be from 0 to 1. | |
| [in] col | Column number must be from 0 to 15. | |
| [in] TableIndex | Select which fonts table be used, from 0 to 2. | |
| [in] BoldMode | 1 – Bold Mode, 0 – Normal mode | |
| [in] message | Message for display. The length of message must be less than or equal to 16 characters. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.2.2.4. ACR122_DisplayLcdMessageExW

This function is used to display the LCD message on the reader in Ex (Unicode).

```

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

```



| Parameter | Description | |
|---------------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] row | Row number must be from 0 to 1. | |
| [in] col | Column number must be from 0 to 15. | |
| [in] TableIndex | Select which fonts table be used, from 0 to 2. | |
| [in] BoldMode | 1 – Bold Mode, 0 – Normal mode | |
| [in] message | Message for display. The length of message must be less than or equal to 16 characters. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.2.2.5. ACR122_DisplayLcdMessageGBA

This function is used to display the LCD message on the reader in GB (ANSI) mode using Simplified Chinese font.

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

| Parameter | Description | |
|---------------------|--|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] row | Row number must be from 0 to 1. | |
| [in] col | Column number must be from 0 to 7. | |
| [in] BoldMode | 1 – Bold Mode, 0 – Normal mode | |
| [in] message | Message for display. The length of message must be less than or equal to 8 characters. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |



3.2.2.6. ACR122_DisplayLcdMessageGBW

This function is used to display the LCD message on the reader in GB (Unicode) mode using Simplified Chinese font.

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

| Parameter | Description | |
|---------------------|--|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] row | Row number must be from 0 to 1. | |
| [in] col | Column number must be from 0 to 7. | |
| [in] BoldMode | 1 – Bold Mode, 0 – Normal mode | |
| [in] message | Message for display. The length of message must be less than or equal to 8 characters. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.2.2.7. ACR122_DrawLcd

This function is used to display the LCD message on the reader in Graphic mode.

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

| Parameter | Description | |
|---------------------|--|---------------------------------------|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] lineIndex | The line which the LCD Display starts update. | |
| [in] pixelBuffer | Buffer with pixel data to display, the length should not be larger than 128 bytes. | |
| [in] pixelBufferLen | Length of pixel buffer in bytes. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |



| Parameter | Description |
|-----------|--|
| Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.2.2.8. ACR122_StartLcdScrolling

This function is used to set the scrolling feature of the current LCD display.

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

| Parameter | Description |
|---------------------|---|
| [in] hReader | A reference value returned from ACR122_Open () function. |
| [in] row | Row number must be 0. |
| [in] col | Column number must be 0. |
| [in] Hrange | Horizontal range (Unit: 8 pixels). The value must be 15. |
| [in] Vrange | Vertical range (Unit: 1 pixel). The value must be 31. |
| [in] Speed | Bit0 - Bit3: Number of pixels in pre-scrolling. Bit4 - Bit7: Scrolling period. |
| [in] Direction | Scrolling direction, from 0 to 3 |
| Return Value | ERROR_SUCCESS The operation completed successfully. |
| | Failure An error code. See Windows API error codes and ACR122 error codes. |

[in] Speed

| Value (Bit 4 – Bit 7) | Scrolling Period |
|-----------------------|------------------|
| 0 | 1 Unit |
| 1 | 3 Units |
| 2 | 5 Units |
| 3 | 7 Units |
| 4 | 17 Units |
| 5 | 19 Units |
| 6 | 21 Units |



| Value (Bit 4 – Bit 7) | Scrolling Period |
|-----------------------|------------------|
| 7 | 23 Units |
| 8 | 129 Units |
| 9 | 131 Units |
| 10 | 133 Units |
| 11 | 135 Units |
| 12 | 145 Units |
| 13 | 147 Units |
| 14 | 149 Units |
| 15 | 151 Units |

[in] Direction

| Value | Meaning |
|-------|--------------------|
| 0 | From left to right |
| 1 | From right to left |
| 2 | From top to bottom |
| 3 | From bottom to top |

3.2.2.9. ACR122_PauseLcdScrolling

This function is used to pause the LCD scrolling set previously.

```
DWORD WINAPI ACR122_PauseLcdScrolling ( HANDLE hReader )
```

| Parameters | Description |
|---------------------|---|
| [in] hReader | A reference value returned from ACR122_Open() function. |
| Return Value | ERROR_SUCCESS The operation completed successfully. |
| | Failure An error code. See Windows API error codes and ACR122 error codes. |

3.2.2.10. ACR122_StopLcdScrolling

This function is used to stop the LCD scrolling set previously. The LCD display will return to normal display position.

```
DWORD WINAPI ACR122_StopLcdScrolling ( HANDLE hReader )
```

| Parameter | Description |
|--------------|---|
| [in] hReader | A reference value returned from ACR122_Open() function. |



| Parameter | Description | |
|--------------|---------------|--|
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.2.2.11. ACR122_ClearLcd

This function clears the LCD display of the reader.

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

| Parameters | Description | |
|--------------|--|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

Note: For ACR122L firmware versions 307 and above, using the *ACR122L_ClearLCD* function successively with other LCD functions requires the application to handle an additional 100 ms time delay.

3.2.2.12. ACR122_EnableLcdBacklight

This function is used to enable/disable the LCD backlight of the reader.

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

| Parameter | Description | |
|--------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] enabled | Set to TRUE to enable backlight. Otherwise, set to FALSE. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |



3.2.2.13. ACR122_SetLcdContrast

This function is used to set the LCD contrast level of the reader.

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

| Parameter | Description |
|---------------------|---|
| [in] hReader | A reference value returned from ACR122_Open() function. |
| [in] level | Contrast level. The value must be from 0x00 to 0x0F |
| Return Value | ERROR_SUCCESS The operation completed successfully. |
| | Failure An error code. See Windows API error codes and ACR122 error codes. |



3.3. LED

3.3.1. Function Documentation

3.3.1.1. ACR122_EnableLed

This function is used to enables/disable the LED control to the application. By default, LED is controlled by the firmware. Before calling `ACR122_SetLedStatesWithBeep()` and `ACR122_SetLedStates()`, the application needs to call this function in order to control the LED.

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

| Parameters | Description | |
|--------------|--|--|
| [in] hReader | A reference value returned from <code>ACR122_Open()</code> function. | |
| [in] enabled | Set to <code>TRUE</code> to enable LED. Otherwise, set to <code>FALSE</code> . | |
| Return Value | <code>ERROR_SUCCESS</code> | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.3.1.2. ACR122_SetLedStates

This function is used to turn on/off the LEDs on the reader. LED0, LED1, LED2 and LED3 can be controlled.

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

| Parameters | Description | |
|----------------|--|--|
| [in] hReader | A reference value returned from <code>ACR122_Open()</code> function. | |
| [in] states | A pointer to the array of states. Possible values are <code>ACR122_LED_STATE_OFF</code> and <code>ACR122_LED_STATE_ON</code> . | |
| [in] numStates | Number of states must be 4. | |
| Return Value | <code>ERROR_SUCCESS</code> | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |



3.3.1.3. ACR122_SetLedStatesWithBeep

This function is used to control LED0, LED1 and buzzer operation on the reader.

```

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

```

| Parameters | Description | | | | | | | | |
|--------------------------|---|--|---------|------------------------|------------------------------|--------------------------|---|--------------------------|---|
| [in] hReader | A reference value returned from ACR122_Open () function. | | | | | | | | |
| [in] controls | A pointer to the ACR122_LED_CONTROL data structure. | | | | | | | | |
| [in] numControls | Number of controls must be 2. | | | | | | | | |
| [in] t1 | T1 in milliseconds. The value must be from 0 to 25500. | | | | | | | | |
| [in] t2 | T2 in milliseconds. The value must be from 0 to 25500 | | | | | | | | |
| [in] numTimes | Number of times. The values must be from 0 to 255. | | | | | | | | |
| [in] buzzerMode | A bitmask of buzzer mode. Possible values may be combined with the OR operation. <table border="1" data-bbox="616 1115 1337 1352"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>ACR122_BUZZER_MODE_OFF</td> <td>The buzzer will not turn on.</td> </tr> <tr> <td>ACR122_BUZZER_MODE_ON_T1</td> <td>The buzzer will turn on during T1 duration.</td> </tr> <tr> <td>ACR122_BUZZER_MODE_ON_T2</td> <td>The buzzer will turn on during T2 duration.</td> </tr> </tbody> </table> | Value | Meaning | ACR122_BUZZER_MODE_OFF | The buzzer will not turn on. | ACR122_BUZZER_MODE_ON_T1 | The buzzer will turn on during T1 duration. | ACR122_BUZZER_MODE_ON_T2 | The buzzer will turn on during T2 duration. |
| Value | Meaning | | | | | | | | |
| ACR122_BUZZER_MODE_OFF | The buzzer will not turn on. | | | | | | | | |
| ACR122_BUZZER_MODE_ON_T1 | The buzzer will turn on during T1 duration. | | | | | | | | |
| ACR122_BUZZER_MODE_ON_T2 | The buzzer will turn on during T2 duration. | | | | | | | | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. | | | | | | | |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. | | | | | | | |

Note: For PACR122_LED_CONTROL, please see **Appendix A.2 _ACR122_LED_CONTROL Struct Reference.**



3.4. Buzzer

3.4.1. Function Documentation

3.4.1.1. ACR122_Beep

This function is used to control the buzzer on the reader to generate the beep sound and does not return control to its caller until the sound finishes.

```

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

```

| Parameters | Description | |
|------------------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] buzzerOnDuration | Buzzer ON duration in milliseconds. The value must be from 0 to 25500. | |
| [in] buzzerOffDuration | Buzzer OFF duration in milliseconds. The value must be from 0 to 25500. | |
| [in] numTimes | Number of times. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |



3.5. Card

3.5.1. Function Documentation

3.5.1.1. ACR122_DirectTransmit

This function is used to send tag command and then receives response from the reader.

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

| Parameter | Description | |
|-------------------------|---|--|
| [in] hReader | A reference value returned from ACR122_Open () function. | |
| [in] sendBuffer | A pointer to the actual data to be written to the card. | |
| [in] sendBufferLen | The length in number of bytes of the sendBuffer parameter. | |
| [in] recvBuffer | A pointer to any data returned from the card. | |
| [in,out] pRecvBufferLen | The length in number of bytes of the recvBuffer parameter and receives the actual number of bytes received from the card. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.5.1.2. ACR122_ExchangeApdu

This function is used to send APDU command, and then receives APDU response from the card.

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

| Parameter | Description |
|-----------------|--|
| [in] hReader | A reference value returned from ACR122_Open () function. |
| [in] slotNum | Slot number. |
| [in] sendBuffer | A pointer to the actual data to be written to the card. |



| Parameter | Description | |
|-------------------------|--|--|
| [in] sendBufferLen | The length in number of bytes of the <code>sendBuffer</code> parameter. | |
| [out] recvBuffer | A pointer to any data returned from the card. | |
| [in,out] pRecvBufferLen | The length in number of bytes of the <code>recvBuffer</code> parameter and receives the actual number of bytes received from the card. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.5.1.3. ACR122_PowerOffIcc

This function is used to power off the card in the slot.

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

| Parameter | Description | |
|---------------------|--|--|
| [in] hReader | A reference value returned from <code>ACR122_Open()</code> function. | |
| [in] slotNum | Slot number. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |

3.5.1.4. ACR122_PowerOnIcc

This function is used to power on the card in the slot and then return the ATR string from the card.

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

| Parameter | Description | |
|--------------|--|--|
| [in] hReader | A reference value returned from <code>ACR122_Open()</code> function. | |
| [in] slotNum | Slot number. | |



| Parameter | Description | |
|---------------------|--|--|
| [out] atr | A pointer to the buffer that receives the ATR string returned from the card. | |
| [in,out] pAttrLen | The length in number of bytes of the atr parameter and receives the actual number of bytes received from the card. | |
| Return Value | ERROR_SUCCESS | The operation completed successfully. |
| | Failure | An error code. See Windows API error codes and ACR122 error codes. |



Appendix A. Data Structures

Appendix A.1. `_ACR122_TIMEOUTS` Struct Reference

This data structure is used in `ACR122_GetTimeouts()` and `ACR122_SetTimeouts()` function.

- `DWORD _ACR122_TIMEOUTS::numResponseRetries`
Number of response retries.
Default is 1.
- `DWORD _ACR122_TIMEOUTS::numStatusRetries`
Number of status retries.
Default is 1.
- `DWORD _ACR122_TIMEOUTS::responseTimeout`
Response timeout in milliseconds.
Default is 10000 ms.
- `DWORD _ACR122_TIMEOUTS::statusTimeout`
Status timeout in milliseconds.
Default is 2000 ms.

Appendix A.2. `_ACR122_LED_CONTROL` Struct Reference

This data structure is used in `ACR122_SetLedStatesWithBeep()` function.

- `BOOL _ACR122_LED_CONTROL::blinkEnabled`
Enable blink.
Set to `TRUE` to enable blink. Otherwise, set to `FALSE`.
- `DWORD _ACR122_LED_CONTROL::finalState`
Final state.
Possible values are `ACR122_LED_STATE_OFF` and `ACR122_LED_STATE_ON`.
- `DWORD _ACR122_LED_CONTROL::initialBlinkingState`
Initial blinking state.
Possible values are `ACR122_LED_STATE_OFF` and `ACR122_LED_STATE_ON`.
- `BOOL _ACR122_LED_CONTROL::updateEnabled`
Enable update.
Set to `TRUE` to update the state. Otherwise, set to `FALSE` to keep the state unchanged.



Appendix B. Error Codes returned by High-level APIs

- ACR122_ERROR_NO_MORE_HANDLES ((DWORD) 0x20000001L)
The handle is invalid.
- ACR122_ERROR_UNKNOWN_STATUS ((DWORD) 0x20000002L)
Reader unknown error.
- ACR122_ERROR_OPERATION_FAILURE ((DWORD) 0x20000003L)
Operation failed.
- ACR122_ERROR_OPERATION_TIMEOUT ((DWORD) 0x20000004L)
Timeout operation.
- ACR122_ERROR_INVALID_CHECKSUM ((DWORD) 0x20000005L)
Checksum calculation error
- ACR122_ERROR_INVALID_PARAMETER ((DWORD) 0x20000006L)
Incorrect parameter input.



Appendix C. Standard Program Flow

```
// 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;
}
```