

Roger RUD-3 Library

1.1.0.2

Generated by Doxygen 1.7.4

Thu Sep 27 2012 09:25:07



# Contents

<b>1</b>	<b>Roger RUD-3 Library.</b>	<b>1</b>
1.1	Requirements: . . . . .	1
1.2	Includes: . . . . .	1
<b>2</b>	<b>Roger RUD-3 Library Version</b>	<b>3</b>
<b>3</b>	<b>Roger Find Readers</b>	<b>5</b>
<b>4</b>	<b>Roger Communication</b>	<b>7</b>
<b>5</b>	<b>Module Documentation</b>	<b>9</b>
5.1	Function - Roger RUD-3 Library Version . . . . .	9
5.1.1	Function Documentation . . . . .	9
5.1.1.1	rud3LibVer . . . . .	9
5.2	Functions support readers search . . . . .	9
5.2.1	Function Documentation . . . . .	10
5.2.1.1	rud3GetDeviceCount . . . . .	10
5.2.1.2	rud3GetSerialNumber . . . . .	10
5.2.1.3	rud3GetIndex . . . . .	11
5.2.1.4	rud3GetDescriptionBySerialNumber . . . . .	11
5.2.1.5	rud3GetDescriptionByIndex . . . . .	12
5.2.1.6	rud3GetPort . . . . .	12
5.2.1.7	rud3GetDeviceType . . . . .	13
5.3	Functions supporting communication . . . . .	13
5.3.1	Function Documentation . . . . .	14
5.3.1.1	rud3Open . . . . .	14
5.3.1.2	rud3Close . . . . .	14

5.3.1.3	rud3Write	15
5.3.1.4	rud3Read	15
5.4	Flags returned by library functions	16
5.4.1	Define Documentation	16
5.4.1.1	RUD3_OK	16
5.4.1.2	RUD3_ERROR	16
5.4.1.3	RUD3_DEVICE_NOT_FOUND	16
5.4.1.4	RUD3_DEVICE_CANNOT_OPEN	17
5.4.1.5	RUD3_INVALID_BUFFER_SIZE	17
5.4.1.6	RUD3_EPSO_ERROR	17
5.4.1.7	RUD3_USB_HID_ERROR	17
5.4.1.8	RUD3_TIMEOUT	17
5.4.1.9	RUD3_CARD_NOT_FOUND	17
5.4.1.10	RUD3_DEVICE_CANNOT_CLOSE	17
5.4.1.11	RUD3_WRITE_ERROR	17
5.4.1.12	RUD3_READ_ERROR	17
5.5	Flags compatible with the RogerRUD.dll	17
5.5.1	Detailed Description	17
5.5.2	Define Documentation	18
5.5.2.1	FDT_UNKNOWN	18
5.5.2.2	FDT_RUD3	18
5.6	General settings	18
5.6.1	Define Documentation	18
5.6.1.1	MAX_FRAME_LENGTH	18
5.6.1.2	ROGER_ALL_DEVICE_TYPE	18
5.6.1.3	ROGER_RUD3_DEVICE_TYPE	18

# Chapter 1

## Roger RUD-3 Library.

RogerRud3 library provides an API to communication with the Roger RUD-3 Mifare proxy readers and other Roger USB HID readers.

The library is written in C and provides supporting for searching reader connected to Windows system and provides communication with the selected reader

### 1.1 Requirements:

- operating system Windows XP or later
- RogerUsbHid.dll library in version 1.0.0.1 - API for communications with a USB HID (Human Interface Devices) devices
- use in technology .NET require import library functions into .NET - see example *RogerMfReaderExample*

### 1.2 Includes:

- RogerRUD3.dll - Dynamic-Link Library
- RogerRUD3.lib - Provides linking to dynamic library
- RogerRud3.h - Defines library function prototypes and constants



## Chapter 2

# Roger RUD-3 Library Version

- [Library Version](#)

[Function - Roger RUD-3 Library Version](#)

[Flags returned by library functions](#)





## Chapter 3

# Roger Find Readers

- [Device Count](#)
- [Serial Number](#)
- [Index by Serial Number](#)
- [Description by Serial Number](#)
- [Description by Index](#)
- [Port Name by Serial Number](#)
- [Device Type by Serial Number](#)

[Functions support readers search](#)

[Flags returned by library functions](#)



## Chapter 4

# Roger Communication

- [Open](#)
- [Close](#)
- [Write](#)
- [Read](#)

[Functions supporting communication](#)

[Flags returned by library functions](#)



## Chapter 5

# Module Documentation

### 5.1 Function - Roger RUD-3 Library Version

#### Functions

- DWORD [rud3LibVer](#) ( \_\_inout LPTSTR Version, \_\_in DWORD Length)

#### 5.1.1 Function Documentation

5.1.1.1 DWORD [rud3LibVer](#) ( \_\_inout LPTSTR *Version*, \_\_in DWORD *Length* )

#### Parameters

<i>Version</i>	Returns a pointer to a buffer with library version string
<i>Length</i>	The size of <i>Version</i> in bytes.

#### Returns

FLAGS:

- [RUD3\\_OK](#)
- [RUD3\\_ERROR](#)

### 5.2 Functions support readers search

#### Functions

- DWORD [rud3GetDeviceCount](#) ( \_\_out PDWORD DeviceCount, \_\_in USHORT DeviceType)
- DWORD [rud3GetSerialNumber](#) ( \_\_in DWORD DeviceCount, \_\_in USHORT DeviceType, \_\_out LPSTR SerialNumber)
- DWORD [rud3GetIndex](#) ( \_\_in LPSTR SerialNumber, \_\_out PDWORD DeviceCount)

- DWORD [rud3GetDescriptionBySerialNumber](#) (\_\_in LPSTR SerialNumber, \_\_out LPSTR Description)
- DWORD [rud3GetDescriptionByIndex](#) (\_\_in DWORD DeviceIndex, \_\_in USHORT DeviceType, \_\_out LPSTR Description)
- DWORD [rud3GetPort](#) (\_\_in LPSTR SerialNumber, \_\_out LPSTR Port)
- DWORD [rud3GetDeviceType](#) (\_\_in LPSTR SerialNumber, \_\_out PDWORD DeviceType)

## 5.2.1 Function Documentation

### 5.2.1.1 DWORD [rud3GetDeviceCount](#) ( \_\_out PDWORD *DeviceCount*, \_\_in USHORT *DeviceType* )

Function returns the number of readers connected to system.

#### Parameters

<i>DeviceCount</i>	Returns a pointer to the number of connected readers
<i>DeviceType</i>	Readers type connected to system

#### Returns

FLASG:

- [RUD3\\_OK](#)
- [RUD3\\_ERROR](#)

#### Note

The *DeviceType* is equivalent to Roger USB Product ID Number, value 0x0000 includes all supported readers.

### 5.2.1.2 DWORD [rud3GetSerialNumber](#) ( \_\_in DWORD *DeviceCount*, \_\_in USHORT *DeviceType*, \_\_out LPSTR *SerialNumber* )

Function returns a serial number of the reader indicated by *DeviceCount*.

#### Parameters

<i>DeviceCount</i>	Index of reader connected to system.
<i>DeviceType</i>	Readers type connected to system.
<i>SerialNumber</i>	Returns a pointer to the serial number.

#### Returns

FLAGS:

- [RUD3\\_INVALID\\_BUFFER\\_SIZE](#)
- [RUD3\\_DEVICE\\_NOT\\_FOUND](#)
- [RUD3\\_DEVICE\\_CANNOT\\_OPEN](#)

- [RUD3\\_OK](#)
- [RUD3\\_ERROR](#)

**Note**

The *DeviceType* is equivalent to Roger USB Product ID Number, value 0x0000 includes all supported readers.

The buffer *SerialNumber* must be large enough. For USB devices, the maximum string length is 126 chars.

**5.2.1.3** `DWORD rud3GetIndex ( _in LPSTR SerialNumber, _out PDWORD DeviceCount )`

Function returns an index of the reader indicated by *SerialNumber*.

**Parameters**

<i>SerialNumber</i>	A pointer to the serial number
<i>DeviceCount</i>	Returns a pointer to the index of reader connected to system

**Returns**

FLAGS:

- [RUD3\\_INVALID\\_BUFFER\\_SIZE](#)
- [RUD3\\_DEVICE\\_NOT\\_FOUND](#)
- [RUD3\\_DEVICE\\_CANNOT\\_OPEN](#)
- [RUD3\\_OK](#)
- [RUD3\\_ERROR](#)

**Note**

The buffer *SerialNumber* must be large enough. For USB devices, the maximum string length is 126 chars.

**5.2.1.4** `DWORD rud3GetDescriptionBySerialNumber ( _in LPSTR SerialNumber, _out LPSTR Description )`

Function returns a description of the reader indicated by *SerialNumber*.

**Parameters**

<i>SerialNumber</i>	A pointer to the serial number
<i>Description</i>	Returns a pointer to the description.

**Returns**

FLAGS:

- [RUD3\\_INVALID\\_BUFFER\\_SIZE](#)
- [RUD3\\_DEVICE\\_NOT\\_FOUND](#)
- [RUD3\\_DEVICE\\_CANNOT\\_OPEN](#)
- [RUD3\\_OK](#)
- [RUD3\\_ERROR](#)

#### Note

Buffers *SerialNumber* and *Description* must be large enough. For USB devices, the maximum string length is 126 chars.

5.2.1.5 **DWORD** rud3GetDescriptionByIndex ( *\_in* **DWORD** *DeviceIndex*, *\_in* **USHORT** *DeviceType*, *\_out* **LPSTR** *Description* )

Function returns a description of the reader indicated by *DeviceIndex*.

#### Parameters

<i>DeviceIndex</i>	Index of reader connected to system.
<i>DeviceType</i>	Readers type connected to system
<i>Description</i>	Returns a pointer to the description.

#### Returns

FLAGS:

- [RUD3\\_INVALID\\_BUFFER\\_SIZE](#)
- [RUD3\\_DEVICE\\_NOT\\_FOUND](#)
- [RUD3\\_DEVICE\\_CANNOT\\_OPEN](#)
- [RUD3\\_OK](#)
- [RUD3\\_ERROR](#)

#### Note

The *DeviceType* is equivalent to Roger USB Product ID Number, value 0x0000 includes all supported readers.

The buffer *Description* must be large enough. For USB devices, the maximum string length is 126 chars.

5.2.1.6 **DWORD** rud3GetPort ( *\_in* **LPSTR** *SerialNumber*, *\_out* **LPSTR** *Port* )

Function returns a port name of the reader - indicated by *SerialNumber*.

#### Parameters

<i>SerialNumber</i>	A pointer to the serial number
<i>Port</i>	Returns a pointer to the port name.



**Returns**

FLAGS:

- [RUD3\\_INVALID\\_BUFFER\\_SIZE](#)
- [RUD3\\_DEVICE\\_NOT\\_FOUND](#)
- [RUD3\\_DEVICE\\_CANNOT\\_OPEN](#)
- [RUD3\\_OK](#)
- [RUD3\\_ERROR](#)

**Note**

The function returns **USB\_HID** if the reader is connected or ? if not connected.  
The buffer *Port* must be large enough.

5.2.1.7 **DWORD** [rud3GetDeviceType](#) ( \_\_in LPSTR *SerialNumber*, \_\_out PDWORD *DeviceType* )

Function returns a type of the Roger RUD device - indicated by *SerialNumber*.

**Parameters**

<i>SerialNumber</i>	A pointer to the serial number
<i>DeviceType</i>	Returns a pointer to the device type

**Returns**

FLAGS:

- [RUD3\\_INVALID\\_BUFFER\\_SIZE](#)
- [RUD3\\_DEVICE\\_NOT\\_FOUND](#)
- [RUD3\\_DEVICE\\_CANNOT\\_OPEN](#)
- [RUD3\\_OK](#)
- [RUD3\\_ERROR](#)

**Note**

The function returns [FDT\\_RUD3](#) if the reader is *Roger USB HID readers* or [FDT\\_UNKNOWN](#) if isn't - compatible with the *RogerRUD.dll*.

## 5.3 Functions supporting communication

**Functions**

- **DWORD** [rud3Open](#) ( \_\_out PHANDLE *RudHandle*, \_\_in LPSTR *SerialNumber* )
- **DWORD** [rud3Close](#) ( \_\_in PHANDLE *RudHandle* )
- **DWORD** [rud3Write](#) ( \_\_in PHANDLE *RudHandle*, \_\_in PCHAR *Buff*, \_\_in **DWORD** *BuffLength*, \_\_out **LPDWORD** *NumberOfBytesWritten*, \_\_in **DWORD** *TimeOut* )

- DWORD [rud3Read](#) ( \_\_in PHANDLE RudHandle, \_\_out PCHAR Buff, \_\_in DWORD BuffLength, \_\_out LPDWORD NumberOfBytesRead, \_\_in DWORD TimeOut)

### 5.3.1 Function Documentation

#### 5.3.1.1 DWORD rud3Open ( \_\_out PHANDLE *RudHandle*, \_\_in LPSTR *SerialNumber* )

Function opens connection with readers - indicated by *SerialNumber* and returns a pointer to the handle of the device.

##### Parameters

<i>RudHandle</i>	Returns a pointer to the handle of the device. When function return error, <i>RudHandle</i> point to <i>INVALID_HANDLE_VALUE</i>
<i>SerialNumber</i>	A pointer to the serial number

##### Returns

FLAGS:

- [RUD3\\_DEVICE\\_NOT\\_FOUND](#)
- [RUD3\\_DEVICE\\_CANNOT\\_OPEN](#)
- [RUD3\\_OK](#)

##### Note

The handle of the device *RudHandle* is created by function **CreateFile()** - with **FILE\_FLAG\_OVERLAPPED** flag

#### 5.3.1.2 DWORD rud3Close ( \_\_in PHANDLE *RudHandle* )

Function close connection with readers.

##### Parameters

<i>RudHandle</i>	A pointer to the handle of the device returned by <a href="#">rud3Open</a> .
------------------	--

##### Returns

FLAGS:

- [RUD3\\_DEVICE\\_NOT\\_FOUND](#)
- [RUD3\\_DEVICE\\_CANNOT\\_CLOSE](#)
- [RUD3\\_OK](#)

### 5.3.1.3 `DWORD rud3Write ( _in PHANDLE RudHandle, _in PCHAR Buff, _in DWORD BuffLength, _out LPDWORD NumberOfBytesWritten, _in DWORD TimeOut )`

Functions write a data to the reader.

#### Parameters

<i>RudHandle</i>	A pointer to the handle of the device returned by <a href="#">rud3Open</a>
<i>Buff</i>	A byte array containing a data to write.
<i>BuffLength</i>	The size of <i>Buff</i> in bytes. Maximum <i>Buff</i> size is <a href="#">MAX_FRAME_LENGTH</a> .
<i>NumberOf-BytesWritten</i>	A pointer to the number of bytes written. This parameter can be NULL.
<i>TimeOut</i>	The time-out interval, in milliseconds. The function waits until ended send data or the interval elapses. If <i>TimeOut</i> is zero the function sets default value of 500 ms

#### Returns

FLAGS:

- [RUD3\\_DEVICE\\_NOT\\_FOUND](#)
- [RUD3\\_INVALID\\_BUFFER\\_SIZE](#)
- [RUD3\\_WRITE\\_ERROR](#)
- [RUD3\\_TIMEOUT](#)
- [RUD3\\_OK](#)
- [RUD3\\_ERROR](#)

#### Note

The Function deletes all pending input reports in a input queue - see **HidD\_FlushQueue()**

If *TimeOut* is set to a time less than *bInterval* - variable in the structure of the USB report, function may return error [RUD3\\_TIMEOUT](#).

### 5.3.1.4 `DWORD rud3Read ( _in PHANDLE RudHandle, _out PCHAR Buff, _in DWORD BuffLength, _out LPDWORD NumberOfBytesRead, _in DWORD TimeOut )`

Function read a data from the reader.

#### Parameters

<i>RudHandle</i>	A pointer to the handle of the device returned by <a href="#">rud3Open</a>
<i>Buff</i>	Returns a byte array containing a data read from the reader.
<i>BuffLength</i>	The number of bytes to be read. The number of bytes cannot exceed <a href="#">MAX_FRAME_LENGTH</a>
<i>NumberOf-BytesRead</i>	A pointer to the variable that receives the number of bytes read. This parameter can be NULL.
<i>TimeOut</i>	Time in ms when the data read must end, for value 0 function sets default value 500 ms

### Returns

FLAGS:

- [RUD3\\_DEVICE\\_NOT\\_FOUND](#)
- [RUD3\\_INVALID\\_BUFFER\\_SIZE](#)
- [RUD3\\_READ\\_ERROR](#)
- [RUD3\\_TIMEOUT](#)
- [RUD3\\_OK](#)
- [RUD3\\_ERROR](#)

### Note

If *TimeOut* is set to a time less than *bInterval* - variable in the structure of the USB report, function may return error [RUD3\\_TIMEOUT](#).

## 5.4 Flags returned by library functions

- `#define RUD3\_OK 0x00`
- `#define RUD3\_ERROR 0x01`
- `#define RUD3\_DEVICE\_NOT\_FOUND 0x02`
- `#define RUD3\_DEVICE\_CANNOT\_OPEN 0x04`
- `#define RUD3\_INVALID\_BUFFER\_SIZE 0x05`
- `#define RUD3\_EPSO\_ERROR 0x06`
- `#define RUD3\_USB\_HID\_ERROR 0x07`
- `#define RUD3\_TIMEOUT 0x08`
- `#define RUD3\_CARD\_NOT\_FOUND 0x09`
- `#define RUD3\_DEVICE\_CANNOT\_CLOSE 0x0A`
- `#define RUD3\_WRITE\_ERROR 0x0B`
- `#define RUD3\_READ\_ERROR 0x0C`

### 5.4.1 Define Documentation

#### 5.4.1.1 `#define RUD3_OK 0x00`

Return OK

#### 5.4.1.2 `#define RUD3_ERROR 0x01`

Error

#### 5.4.1.3 `#define RUD3_DEVICE_NOT_FOUND 0x02`

Error - device not found - USB HID

5.4.1.4 `#define RUD3_DEVICE_CANNOT_OPEN 0x04`

Error - cannot open connection with the reader

5.4.1.5 `#define RUD3_INVALID_BUFFER_SIZE 0x05`

Error - invalid buffer size

5.4.1.6 `#define RUD3_EPSO_ERROR 0x06`

5.4.1.7 `#define RUD3_USB_HID_ERROR 0x07`

USB HID protocol returned error

5.4.1.8 `#define RUD3_TIMEOUT 0x08`

Error - timeout elapsed - functions [rud3Write](#) or [rud3Read](#)

5.4.1.9 `#define RUD3_CARD_NOT_FOUND 0x09`

5.4.1.10 `#define RUD3_DEVICE_CANNOT_CLOSE 0x0A`

Error - cannot close connection with the reader

5.4.1.11 `#define RUD3_WRITE_ERROR 0x0B`

Write function [rud3Write](#) returned error

5.4.1.12 `#define RUD3_READ_ERROR 0x0C`

Read function [rud3Read](#) returned error

## 5.5 Flags compatible with the RogerRUD.dll

- `#define FDT_UNKNOWN 0x00`
- `#define FDT_RUD3 0x03`

### 5.5.1 Detailed Description

See function [rud3GetDeviceType](#)

## 5.5.2 Define Documentation

### 5.5.2.1 #define FDT\_UNKNOWN 0x00

Not found or unknown devices

### 5.5.2.2 #define FDT\_RUD3 0x03

RUD-3 Mifare proxy reader or other Roger USB HID readers

## 5.6 General settings

- #define [MAX\\_FRAME\\_LENGTH](#) 64
- #define [ROGER\\_ALL\\_DEVICE\\_TYPE](#) 0x0000
- #define [ROGER\\_RUD3\\_DEVICE\\_TYPE](#) 0x0001

### 5.6.1 Define Documentation

#### 5.6.1.1 #define MAX\_FRAME\_LENGTH 64

Maximum frame size in bytes

#### 5.6.1.2 #define ROGER\_ALL\_DEVICE\_TYPE 0x0000

Select all supported devices

#### 5.6.1.3 #define ROGER\_RUD3\_DEVICE\_TYPE 0x0001

Select only Roger RUD-3 readers

# Index

- FDT\_RUD3
  - Flags compatible with the RogerRUD.dll, [18](#)
- FDT\_UNKNOWN
  - Flags compatible with the RogerRUD.dll, [18](#)
- Flags compatible with the RogerRUD.dll, [17](#)
  - FDT\_RUD3, [18](#)
  - FDT\_UNKNOWN, [18](#)
- Flags returned by library functions, [16](#)
  - RUD3\_CARD\_NOT\_FOUND, [17](#)
  - RUD3\_DEVICE\_CANNOT\_CLOSE, [17](#)
  - RUD3\_DEVICE\_CANNOT\_OPEN, [16](#)
  - RUD3\_DEVICE\_NOT\_FOUND, [16](#)
  - RUD3\_EPSO\_ERROR, [17](#)
  - RUD3\_ERROR, [16](#)
  - RUD3\_INVALID\_BUFFER\_SIZE, [17](#)
  - RUD3\_OK, [16](#)
  - RUD3\_READ\_ERROR, [17](#)
  - RUD3\_TIMEOUT, [17](#)
  - RUD3\_USB\_HID\_ERROR, [17](#)
  - RUD3\_WRITE\_ERROR, [17](#)
- Function - Roger RUD-3 Library Version, [9](#)
  - rud3LibVer, [9](#)
- Functions support readers search, [9](#)
  - rud3GetDescriptionByIndex, [12](#)
  - rud3GetDescriptionBySerialNumber, [11](#)
  - rud3GetDeviceCount, [10](#)
  - rud3GetDeviceType, [13](#)
  - rud3GetIndex, [11](#)
  - rud3GetPort, [12](#)
  - rud3GetSerialNumber, [10](#)
- Functions supporting communication, [13](#)
  - rud3Close, [14](#)
  - rud3Open, [14](#)
  - rud3Read, [15](#)
  - rud3Write, [14](#)
- General settings, [18](#)
  - MAX\_FRAME\_LENGTH, [18](#)
- ROGER\_ALL\_DEVICE\_TYPE, [18](#)
- ROGER\_RUD3\_DEVICE\_TYPE, [18](#)
- MAX\_FRAME\_LENGTH
  - General settings, [18](#)
- ROGER\_ALL\_DEVICE\_TYPE
  - General settings, [18](#)
- ROGER\_RUD3\_DEVICE\_TYPE
  - General settings, [18](#)
- RUD3\_CARD\_NOT\_FOUND
  - Flags returned by library functions, [17](#)
- RUD3\_DEVICE\_CANNOT\_CLOSE
  - Flags returned by library functions, [17](#)
- RUD3\_DEVICE\_CANNOT\_OPEN
  - Flags returned by library functions, [16](#)
- RUD3\_DEVICE\_NOT\_FOUND
  - Flags returned by library functions, [16](#)
- RUD3\_EPSO\_ERROR
  - Flags returned by library functions, [17](#)
- RUD3\_ERROR
  - Flags returned by library functions, [16](#)
- RUD3\_INVALID\_BUFFER\_SIZE
  - Flags returned by library functions, [17](#)
- RUD3\_OK
  - Flags returned by library functions, [16](#)
- RUD3\_READ\_ERROR
  - Flags returned by library functions, [17](#)
- RUD3\_TIMEOUT
  - Flags returned by library functions, [17](#)
- RUD3\_USB\_HID\_ERROR
  - Flags returned by library functions, [17](#)
- RUD3\_WRITE\_ERROR
  - Flags returned by library functions, [17](#)
- rud3Close
  - Functions supporting communication, [14](#)
- rud3GetDescriptionByIndex
  - Functions support readers search, [12](#)
- rud3GetDescriptionBySerialNumber
  - Functions support readers search, [11](#)

rud3GetDeviceCount  
    Functions support readers search, [10](#)

rud3GetDeviceType  
    Functions support readers search, [13](#)

rud3GetIndex  
    Functions support readers search, [11](#)

rud3GetPort  
    Functions support readers search, [12](#)

rud3GetSerialNumber  
    Functions support readers search, [10](#)

rud3LibVer  
    Function - Roger RUD-3 Library Version, [9](#)

rud3Open  
    Functions supporting communication, [14](#)

rud3Read  
    Functions supporting communication, [15](#)

rud3Write  
    Functions supporting communication, [14](#)