EREL Software Development Kit Users Manual

ABS Applied Biometric Systems GmbH
SDK Version 1.01.05
February 14, 2015

Table of Contents

Software Development Kit

Purpose

This document describes the EREL Software Development Kit (SDK) and the functions exported by the SDK for use by a calling application.

The SDK is a “thin” software interface between a calling application and EREL devices.

SDK supports the Windows 2000/XP/7/10 operating environments and serves the EREL devices with Ethernet interface.

SDK Files

The SDK consists of following files:

1.	erel11xc.h	SDK header file
2.	erel11xc.lib	SDK import library
3.	erel11xc.dll	SDK

SDK Testing

This SDK version was tested with EREL (Ethernet interface) scanners running firmware versions 3.x – 5.x, under Windows 2000/XP/7/10.

SDK Functions

This is list of SDK functions for EREL devices v5.x

erelOpenDevice

Syntax: erelOpenDevice(int typedevice, unsigned char * addr, int port)

Description: Opens and initializes the EREL device.

Parameters: typedevice – type of device (TYPEDEVICE_EREL112 - for all devices)
addr – IP address of device,
port – port of device (default port is 5000)

Return values: from 0 to 63 is handle of device
< 0 is error

erelCloseDevice

Syntax: erelCloseDevice(int handle)

Description: Closes of connection with EREL device.

Parameters: handle - handle of the device

Return values: rcOK

erelGetVersion

Syntax: erelGetVersion(int handle, unsigned char * version)

Description: Get version of EREL device.

Parameters: handle - handle of the device,
version - pointer to data from device (the size of array must be min. 8 bytes)

Return values: rcOK.

Parameters: handle - handle of the device

Return values: rcOK is OK
< 0 - error

erelRelayOn

Syntax: erelRelayOn(int handle, unsigned char nRelay)

Description: Activate the relay(s)

Parameters: handle - handle of the device,
nRelay – number of relay

```
#define  S_RELAY_1     0x01
#define  S_RELAY_2     0x02
#define  S_RELAY_3     0x04
#define  S_RELAY_4     0x08
#define  S_RELAY_5     0x10
#define  S_RELAY_6     0x20
#define  S_RELAY_7     0x40
#define  S_RELAY_8     0x80
```

Return values: current status of relays
< 0 - error

erelRelayOff

Syntax: erelRelayOff(int handle, unsigned char nRelay)

Description: Deactivate the relay(s)

Parameters: handle - handle of the device,
nRelay – number of relay

```
#define  S_RELAY_1     0x01
#define  S_RELAY_2     0x02
#define  S_RELAY_3     0x04
#define  S_RELAY_4     0x08
#define  S_RELAY_5     0x10
#define  S_RELAY_6     0x20
#define  S_RELAY_7     0x40
#define  S_RELAY_8     0x80
```

Return values: current status of relays
< 0 - error

erelRelayOnTimer

Syntax: erelRelayOnTimer(int handle, unsigned char nRelay, unsigned char nTime)

Description: Activate the relay for nTime seconds

Parameters: handle - handle of the device,
nRelay – number of relay
nTime –switching on time in seconds

Return values: current status of relays
< 0 - error

erelAllRelayOff

Syntax: erelAllRelayOff (int handle)

Description: Deactivate all relays

Parameters: handle - handle of the device,

Return values: current status of relays
< 0 - error

erelGetStatusRelay

Syntax: erelGetStatusRelay (int handle, unsigned char * out)

Description: Get status of relays

Parameters: handle - handle of the device,
out – pointer to status value

Return values: 0 is OK
!= 0 - error

erelGetStatusInput

Syntax: erelGetStatusInput (int handle, unsigned char * out)

Description: Get status of inputs (The normal status of input is 0xff). The active signal of input sets the bit to 0.

Parameters: handle - handle of the device
out – pointer to status value

Return values: 0 is OK
!= 0 - error

erelSetTimeM3

Syntax: erelSetTimeM3 (int handle, unsigned char day, unsigned char mon, unsigned short year, unsigned char hour, unsigned char min, unsigned char sec)

Description: Set date and time

Parameters: handle - handle of the device,
day - current day
mon - current month
year - current year
hour – current hour
min – current minute
sec – current second

Return values: 0 is OK
!= 0 - error

erelGetTimeM3

Syntax: erelGetTimeM3 (int handle, unsigned char *day, unsigned char *mon, unsigned short *year, unsigned char *hour, unsigned char *min, unsigned char *sec, unsigned char * dow)

Description: Activate the relay for nTime seconds

Parameters: handle - handle of the device,
day - pointer to current day value
mon - pointer to current month value
year - pointer to current year value
hour – pointer to current hour value
min – pointer to current minute value
sec – pointer to current second value
dow - pointer to current day of week value

Return values: rcOK - current status of relays
!= rcOK - error

erelSetInvertInpMask

Syntax: erelSetInvertInpMask (int handle, unsigned char maskl, unsigned char maskh)

Description: Set mask for input value (max 16 bits)

Parameters: handle - handle of the device,
maskl – low byte
maskh – high byte

Return values: rcOK is OK
-1 - error

erelGetInvertInpMask

Syntax: erelGetInvertInpMask(int handle, unsigned char * maskl, unsigned char * maskh)

Description: Get mask for input value (max 16 bits)

Parameters: handle - handle of the device,
maskl – pointer to low byte value
maskh – pointer to high byte value

Return values: rcOK is OK
-1 - error

erelSetInvertOutMask

Syntax: erelSetInvertOutMask (int handle, unsigned char maskl, unsigned char maskh)

Description:	Set mask for output value (max 16 bits)
Parameters:	handle - handle of the device, maskl – low byte maskh – high byte
Return values:	rcOK is OK -1 - error

erelGetInvertOutMask

Syntax:	erelGetInvertOutMask (int handle, unsigned char *maskl, unsigned char *maskh)
Description:	Get mask for output value (max 16 bits)
Parameters:	handle - handle of the device, maskl – pointer to low byte value maskh – pointer to high byte value
Return values:	rcOK is OK -1 - error