****

**eLocker API**

Version Number : 1.3.1

Date : 3 May 2016

Table of Contents

[Introduction 3](#_Toc447035307)

[Octopus Reader 4](#_Toc447035308)

[Locker Control 8](#_Toc447035309)

[I/O Board Control 10](#_Toc447035310)

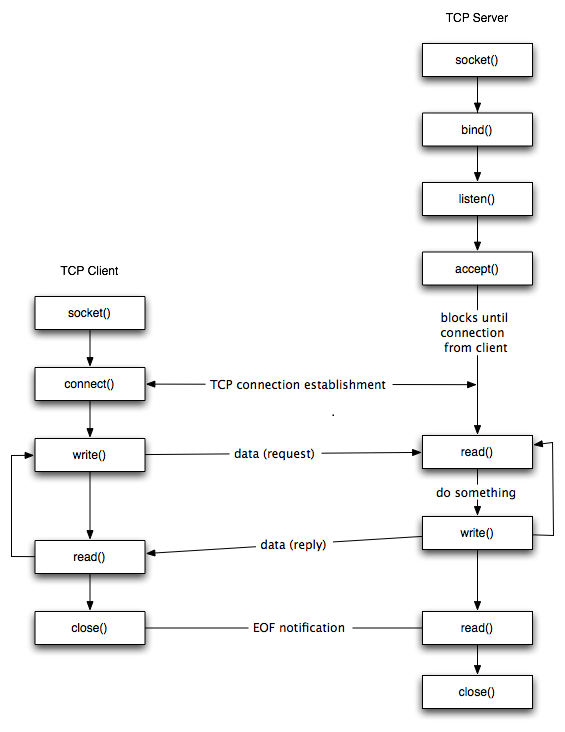
# Introduction

This document provides API showing how to interface with Octopus card reader, locker open control, locker box status and I/O board control. We use TCP client-server approach for data communication. Our server module abstract the low level API and library call. It eases of software development for multi-platform environment.

Protocols : TCP (as Server)

Address : 127.0.0.1

Port number : 9012

The sequence of function calls for the client and a server participating in a TCP connection is presented in the below diagram.

Command format : <command> [argument]\r\n

Response format : <command>: <return value>\r\n

where [argument] is optional, \r is RETURN character, \n is NEWLINE character

# Octopus Reader

1. Octopus Reader Initialization

Command : oct\_init

Response : oct\_init: <return value>

Purpose : Initialize and set the communication port

|  |  |
| --- | --- |
| Return value:  0 | Successful |
| 100001 | Failed to open communication port |
| 100005 | Invalid response from R/W |
| 100101 | Failed to create AR (SaveLog) |
| 100102 | Failed to create UD (SaveLog) |
| 100099 | Firmware upgrade has performed due to HouseKeeping() |

1. Octopus Reader Sound

Command : oct\_sound <argument>

Response : oct\_sound: <return value>

Purpose : Set sound tone based on argument.

argument=1 : Successful tone

argument=0 : Failed tone

|  |  |
| --- | --- |
| Return value:  0 | Successful |
| 100001 | The reader not connected |

1. Octopus Reader TimeVer

Command : oct\_timeVer

Response : oct\_timeVer: <return value>

Purpose : Setting the time and retrieve various version information from the reader

Where return value=<error code>, <sequence of the below data structure>

typedef struct {

unsigned int DevID; /\* Device ID \*/

unsigned int OperID; /\* Operator ID \*/

unsigned int DevTime; /\* Device Time \*/

unsigned int CompID; /\* Company ID \*/

unsigned int KeyVer; /\* Key Version \*/

unsigned int EODVer; /\* EOD Version \*/

unsigned int BLVer; /\* Blacklist Version \*/

unsigned int FIRMVer; /\* Firmware Version \*/

unsigned int CCHSVer; /\* CCHS Version \*/

unsigned int LocID; /\* Location ID \*/

unsigned int IntBLVer; /\* Interim Blacklist Version \*/

unsigned int FuncBLVer; /\* Functional Blacklist Version \*/

};

|  |  |
| --- | --- |
| First item of return value  0 | Successful |
| 100001 | The reader not connected |

e.g. successful

timeVer: 0, 5791862, 16909060, 511512220, 909, 0, 1001, 0, 8389048, 0, 0, 0, 0

e.g. failed

timeVer: 100001

1. Octopus Reader Poll

Command : oct\_poll

Response : oct\_poll: <card remaining value>, <Card ID>,<Customer Info >,<IDm>

Purpose : Detect the presence of an Octopus card/product and return relevant information if found

|  |  |
| --- | --- |
| **Return Value (first item):** | |
| Remaining Value on the card | If the function is successful, the return value is the remaining value on the Octopus card/product |
| 100001 | R/W not connected |
| 100005 | Invalid response from R/W |
| 100016 | Card read error |
| 100017 | Card write error |
| 100019 | Card is blocked |
| 100021 | The last add value date of card is greater than 1000 days |
| 100023 | Transaction Log full |
| 100024 | Card is blocked by this call. In this case, PollData contains the UD. |
| 100032 | No card present |
| 100034 | Card authentication error |
| 100035 | Card recover error |
| 100066 | System time error |
| **Note:** | If the function is successful, the return value is the remaining value on the Octopus card/product, otherwise the error code > 100000 will be returned. |

e.g. successful

poll: 948, 83812609, 15434-4199238-9-1-0, 060005018002572C

e.g. failed

poll: 100001

1. Octopus Reader Deduct

Command : oct\_deduct <value in 10 cents>

Response : oct\_deduct: <return value>

Purpose : Deduct value in 10 cents

|  |  |
| --- | --- |
| **Return Value:** | |
| Remaining Value on the card | If successful, the return value is the remaining value on the Octopus card/product; otherwise the error code > 100000 will be returned. |
| 100001 | R/W not connected |
| 100003 | Invalid parameters |
| 100005 | Invalid response from R/W |
| 100016 | Card read error |
| 100017 | The card polled before is not on the target or card communication has been interrupted since last poll |
| 100019 | Card is blocked. (A blocked card is a blacklisted card that should not be accepted for any transactions.) |
| 100020 | Card has not been polled before or no card present |
| 100022 | Incomplete transaction. Must retry  Note: A 100022(Retry please) occurs when the customer pull out the card too quickly. The R/W is not sure if the data is successfully written to the card or not. Hence, the R/W returns with this error code to advise the customer to present the card again. The R/W will correctly handle the different cases of retry; the same transaction will not be deducted twice. Refer to Volume A1 on Incomplete Transaction Handling. |
| 100023 | Transaction Log full |
| 100048 | Card has insufficient fund |
|  |  |

1. Octopus Reader Reset

Command : oct\_reset

Response : oct\_reset: <return value>

Purpose : This function performs software reset on the reader

|  |  |
| --- | --- |
| **Return Value:** | |
| 0 | If command is sent |

1. Octopus Reader Close

Command : oct\_close

Response : oct\_close: <return value>

Purpose : This function closes the communication port and opened files

|  |  |
| --- | --- |
| **Return Value:** | |
| 0 | If command is sent |

# Locker Control

Remark : One control board controls two bays. One bay contains 9 boxes (maximum).

1. Locker board initialization

Command : lock\_init <COM port>

Where <COM port> = 1,2,3,4,…..

Response : lock\_init: <return value>

Purpose : initialize the locker control board

|  |  |
| --- | --- |
| Return value:  0 | Successful |
| 100001 | Hardware failure |

1. Locker health check

Command : lock\_ready

Response : lock\_ready: <return value>

Purpose : check if the locker is ready

|  |  |
| --- | --- |
| Return value:  0 | Ready |
| 100001 | Hardware failure |

1. Locker open box

Command : lock\_open <bay number, box number>

Response : lock\_open: <return value>

Purpose : open specified box (1 – 9, from top to bottom) in specified bay (1 – 4, from left to right)

|  |  |
| --- | --- |
| Return value:  0 | Successful |
| 100002 | Not successful |

1. Locker read box status

Command : lock\_boxStatus <bay number, box number>

Response : lock\_boxStatus: <return value>

<return value> = S

S=1 indicates the box is opened while 0 indicates the box is closed

Purpose : read box status of specified bay (1 – 4, from left to right)

|  |  |
| --- | --- |
| Return value:  0 | Box status of corresponding box (0 or 1) |
| 100002 | Not successful |

1. Locker board close

Command : lock\_close

Response : lock\_close: <return value>

Purpose : release the locker control board resource

|  |  |
| --- | --- |
| Return value:  0 | Successful |
| 100001 | Hardware failure |

# I/O Board Control

1. I/O board initialization

Command : io\_init <COM port>

Where <COM port> = 1,2,3,4,….. or 0 = autoscan

Response : io\_init: <return value>

Purpose : initialize the I/O board

|  |  |
| --- | --- |
| Return value:  0 | Successful |
| 100001 | Hardware failure |

1. I/O board digital output

Command : io\_output <output n> <status>

where n=1 – 4, status= 0:off / 1:on

Response : io\_output: <return value>

Purpose : output on/off control (on = +12Vdc output with 500mA max, off = no voltage output)

|  |  |
| --- | --- |
| Return value:  0 | Successful |
| 100001 | Hardware failure |

1. I/O board relay output

Command : io\_relay <relay n> <status>

where n=1 - 3, status= 0:off, 1:on

Response : io\_relay: <return value>

Purpose : relay on/off control (on = relay close, off = relay open)

|  |  |
| --- | --- |
| Return value:  0 | Successful |
| 100001 | Hardware failure |

1. I/O board digital input

Command : io\_input <input n>

where n=1 - 5

Response : io\_input: <return value>

Purpose : input status

|  |  |
| --- | --- |
| Return value:  0 | Contact opened |
| 1 | Contact closed |
| 100001 | Hardware failure |

1. I/O board close

Command : io\_close

Response : io\_close: <return value>

Purpose : release the I/O board resource

|  |  |
| --- | --- |
| Return value:  0 | Successful |
| 100001 | Hardware failure |

