

LF M3S SDK Specification

Preliminary

1st Draft



Version :

22.Mar.2013 10:14

Confidentiality Note

This document may only be circulated to those people involved in the project.
The document may not be passed on to third parties without permission of iDTRONIC.



Table of contents

1	Abstract	4
2	General product philosophy.....	5
3	Interoperability	5
4	Limitations.....	5
5	SDK library	5
5.1	OpenReader	6
5.2	GetReaderCaps	6
5.3	SetReaderCaps	6
5.4	ReadData.....	7
5.5	WriteData.....	7
5.6	CloseReader	8
6	Operation schedule.....	9
7	Notes	10
8	Document history.....	11



1 Abstract

This document is intended to keep all participating parties on the project at the same level of information and to summarize all kind of ideas, wishes, recommendations and must have features from customers, employees, support team and other kind of sources.

This preliminary specification describes the product's requirements for LF M3S SDK development.

When all preliminary specifications are clear and accepted by all participating parties, the next step should be to push this document into final specification documentation where all issues are described.



2 General product philosophy

LF readers/tags represent an old standard very well implemented in current application. Combination between reader and tag offers a very stable solution in time. In our days when mobility represents a key to success we are forced to migrate also the existing applications to current technology. As result we migrate our LF solution on M3 mobile device.

This reader is an approved solution and integration should be done through SDK. During time several feature was added to reader and no other SDK was developed. Our user request us a new version from SDK which can provide also new feature and future development. This new SDK provides the following features:

- No need of knowledge about low level protocol from LF-M3S reader
SDK library represent a transparent solution for communication with LF-M3S readers. Because LF-M3S communication protocol is not so easy to be implemented by each user the SDK library assists us to integrate LF-M3S reader in our end solution.
- Fast integration in end user solution
Communication library offers also high scalability for integration in end user product. Because this library contain only few functions, this will offer us possibility of fast integration in end user software avoid big problems created by low level protocols interchanging data.
- Compatibility with early version between SDK versions.
SDK offers high level of compatibility with early versions of library. So updating to a newer version should be complete transparent and should not generate any modification in developed software.

3 Interoperability

LF-M3S reader will be integrated in our SDK library to be able to offer high scalability and represents fast and common way to be accessed from end user developers from other programming languages. To demonstrate interoperability options we will deliver examples for C++ and C# programming languages.

4 Limitations

LF-M3S reader is able to detect and to exchange data only with specific LF cards. The following cards are detected and supported: EM4X00, Q5, Hitag, Titan, Atmel 5567.

5 SDK library

To help end user developers we decided to provide an SDK which emulate and optimize reader functionality for an easy integration. We decided that our library to export next functions:

1. OpenReader
2. GetReaderCaps
3. SetReaderCaps
4. ReadData
5. WriteData
6. CloseReader

This functions will be described in details in the following chapters.

5.1 OpenReader

Main proposal of this function is to initialize and check if the LF-M3S reader is already connected to serial host.

After calling this function the following error codes are returned:

- SDK handle to reader if successfully
- Null in case of error

5.2 GetReaderCaps

This function gets information about reader:

Input parameters are as follows:

- SDKHANDLE - Handle to reader opened previous
- EREADER_FEATURE - One of the following reader features:
 - ERD_SERIAL_NUMBER – return reader serial number
 - ERD_VERSION – return reader software version
- Pointer to byte array for returning data
- Reference to length of data

Data format request by each feature will be described in the following session:

- ERD_SERIAL_NUMBER
 - Buffer to byte array where reader serial number will be returned
 - Data length – size of data buffer
- ERD_VERSION
 - Buffer to byte array where reader software version will be returned
 - Data length – size of data buffer

As return we get the following errors:

- ER_OK – no error detected
- ER_FEATURE – invalid feature
- ER_MORE_DATA – more data available
- ER_INVALID_POINTER – data buffer is null

5.3 SetReaderCaps

This function sets some reader features and should be treated very carefully:

Input parameters are as follows:

- SDKHANDLE - Handle to reader open previous
- EREADER_FEATURE - One of next reader feature:
 - ERD_RESET – set reader attenuation
 - ERD_SET_LED – set reader sensitivity
 - ERD_TURNOFF_CONT_MODE – set reader frequency
 - ERD_START_BLOCK – perform reader reboot
 - ERD_LOGIN – restore factory settings
- Pointer to byte array with feature information
- Length of data byte array

Data format request by each feature will be described in the following session:

- ERD_RESET
 - Buffer empty
 - Data length – size of 0
- ERD_SET_LED
 - Buffer with 1 byte having next value
 - 0 – turn off all LEDs
 - 1 – turn on green LED
 - 2 – turn on red LED
 - Data length – size of byte
- ERD_TURNOFF_CONT_MODE
 - Buffer empty
 - Data length – size of 0
- ERD_START_BLOCK
 - Buffer contains 4 bytes – number blocks to read/write (as unsigned long)
 - Data length – size of data buffer – 4 bytes
- ERD_LOGIN
 - Buffer contains login password – Titan cards
 - Data length – size of data buffer – password length

As return we get the following errors:

- ER_OK – no error detected
- ER_FEATURE – invalid feature
- ER_MORE_DATA – more data available
- ER_INVALID_POINTER – data buffer is null

5.4 ReadData

As input parameters we have the following:

- SDKHANDLE - Handle to reader open previous
- Pointer to byte array for return data
- UInt32 - Length of data to be read

As return we get the following errors:

- ER_OK – no error detected
- ER_READ – invalid feature
- ER_INVALID_POINTER – data buffer is null

5.5 WriteData

Like input parameter we have next:

- SDKHANDLE - Handle to reader open previous
- Pointer to byte array with data
- UInt32 - Length of data to be written

As return we get the following errors:

- ER_OK – no error detected
- ER_WRITE – invalid feature

- ER_INVALID_POINTER – data buffer is null

5.6 CloseReader

This function ends communication with the RFID reader and turns the power off. As input parameter we have the following:

- SDKHANDLE - Handle to reader open previous

Like return we get next errors:

- ER_OK – no error detected
- ER_INVALID_POINTER – handle is null



6 Operation schedule

	Duration	Dead-line
1 st Implementing new SDK for LF-M3S reader		09.03.2012



[illegible]

8 Document history

Version / Date	Author	Changes
March, 2013		Initial Version

