NX2EXP-LF Reader Module

Interface description

handheld

Development executed by RadioForce GmbH Germany

1. Contents

2. The NX2EXP-LF

2.1. Introduction

The NX2EXP-LF is a Low Frequency RFID Module. It is designed to be mounted on a X2 Terminal from Handheld. The NX2EXP-LF can read the most used Low Frequency transponders in the market. This includes both Full Duplex ASK modulated tags and Half Duplex FSK modulated tags. It can be extended to read more types of transponders when the modulation type is one of the above.

2.2. Document Scope

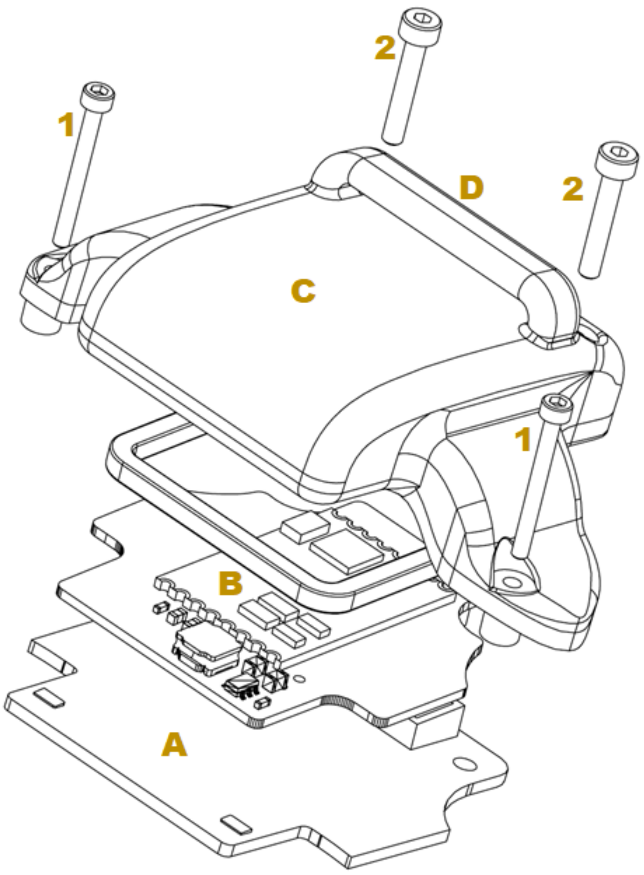
The goal of this document is to serve as an interface specification for the integration of the RFID Module. It describes both the electronical aspects of the interface and the software aspects of the interface.

2.3. Abbreviations and terminology

3. NX2EXP-LF Connections

3.1. Mechanical aspects

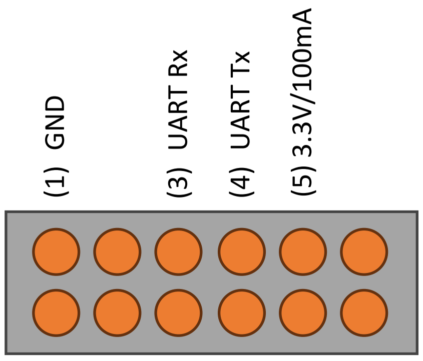
The NX2EXP-LF consists of the Reader Module PCB, the RFID Antenna and the housing. The housing consists of a top cap and a bottom closing plate. The antenna is glued inside the casing. The bottom closing plate is glued to the top cap. The mounting on the X2 Terminal is done by removing 2 existing screws from the X2 top. Then the casing fits on the backside of the terminal and can be fixed with 2 M1.6 screws and 2 M2 screws.



3.2. Electrical aspects

The NX2EXP-LF uses Spring loaded connection pins to make the electrical connection to the X2 Terminal. When the NX2EXP-LF module is screwed on the terminal the pins contact gold plated contacts on the X2 Terminal. The NX2EXP-LF uses 4 pins for power and communication.

Top of the Module



View from the bottom side

The interface has the following requirements:

Power Voltage	The NX2EXP-LF needs 3.3V
Power consumption	The NX2EXP-LF uses an average of 60mA with peak values to 100mA
Uart Voltage levels	The UART uses 3.3V signal levels
Uart Baudrate	115200
Uart Properties	8 Databits, 1 Stopbit, No parity

4. Using the NX2EXP-LF

The NX2EXP-LF is equipped with a host interface port. Using this port, the NX2EXP-LF can be controlled. There are several aspects of the NX2EXP-LF which are important for correctly and efficiently interfacing with the NX2EXP-LF.

4.1. Transponder Types

The NX2EXP-LF supports different transponder types. The currently support transponder types are:

1. FDX-B transponders (Type 02)
2. Tiris HDX transponders (Type 01)
3. EM4x02/Casi Rusco Transponders (Type 03)
4. Hitag 1/S Transponders (Type 04)

These transponder types work in different ways. This influences how the NX2EXP-LF behaves when trying to detect the transponder. The NX2EXP-LF is capable of autodetecting the transponder type by looping through the different transponder types. Because looping through all the different transponder types takes time it is always more efficient to set the transponder type which can to be detected.

4.1.1. FDX-B Transponder

The FDX-B transponder uses ASK modulation in Full Duplex mode. That means the RFID field is used to modulate the transponder number with load modulation. This tag is a Tag Talk First transponder which means it automatically sends out the transponder number.
This transponder normally operates on 134.2KHz, but is some cases are misused on 125KHz.
The FDX-B number contains 64bits of information but can use several encoding schemes. The standardized way of displaying the transponder number is using a BCD encoding. This is used often in animal identification. It is possible to also output the identification as a Raw Hex format code. Both the BCD and Hex encoding schemes are 16 digits.

4.1.2. Tiris HDX Transponder

The Tiris HDX transponder uses FSK modulation in Half Duplex mode. This means that the RFID Field has to be turned off for short moment to allow the transponder to transmit its code. Before it can do that, it needs to charge itself from the RF Field. This tag is a Tag Talk First transponder, it will send out its number shortly after the field is turned off. The transponder always operates at 134,2KHz. The ID number also is 64 bits like the FDX-B transponder. This transponder is also used often for animal identification. The number is mostly encoded as a Raw Hex format code but sometimes the BCD encoding is used.

4.1.3. EM4x02/Casi Rusco Transponder

The EM4x02 transponder also called unique transponder uses ASK modulation but with a slightly different data decoding. It is also a Tag Talk First transponder. The most used frequency for this transponder is 125KHz. The number consists of 5 Bytes of data, which are mostly encoded as a Hex string with 10 digits.

4.1.4. Hitag 1/S Transponder

The Hitag 1 and Hitag S transponders use the same protocol, so they are grouped as the same type. By reading the config block the different types can be distinguished. Both transponders are more advances transponders. They contain user memory and can be programmed to emulate other transponder types. When this transponder is programmed correctly it will be detected as a FDX-B transponder for example, or a EM4x02 transponder. To detect that the real transponder type is a Hitag transponder we need communicate in a more advanced way with the transponder. To start this communication, we need to force the transponder into direct mode. We do this by sending commands to the transponder. When we only want to detect the tag and read its Identification number, we need to perform an Inventory. The NX2EXP-LF performs this automatically when this transponder type is enabled. Because this transponder type has more memory including configuration memory there are more commands which should be used.

1. Selecting the transponder
2. Reading data from the transponder
3. Writing data to the transponder

Before we can read or write to the transponder it has to be selecting. This is done by sending a select command. When this is successful the transponder is stopping its emulated state and waits for other commands. The transponder remains in this state as long as an RF field is active.
The data can be read from the transponder or written to the transponder. Memory for this tag is arranged in 32-bit words, so 4 bytes or 8 hex digits, which we call blocks. There are certain blocks which are read-only, others can be read from and written to.

Note that when autodetection is active and both Hitag-1/S and FDX-B or EM4x02 are enabled it might be that it seems that the transponder type is not correctly identified, because it is detected as a Hitag-1/S transponder while it should be a FDX-B or EM4x02 transponder. But this is because it is common to emulate other transponder types and sell them as such.

4.2. Properties

Properties are used to store configuration parameters in the NX2EXP-LF. A property is identified by an ID number. A property has a certain type like Integer or String, but are mostly converted to and from strings when required. By changing a property, the behavior of the NX2EXP-LF will change. Below is a list of useful properties:

81001	Int32	Contains the enabled transponder types as a bitmask: Bit 0: Enable HDX transponder type Bit 1: Enable FDX-B transponder type Bit 2: Enable EM4x02 transponder type Bit 3: Enable Hitag-1/S transponder type Example: enable FDX-B and EM4x02 results in: 6
81002	Int32	Contains the field frequency. This is currently on 130000Hz. Because the antenna is tuned for this frequency this value should not be changed
81003	Int16	Contains the default startup mode: 0 = Field is turned off by default. 1 = Field is turned on by default
81004	Int16	Contains smart detection settings. To be defined
81005	Int16	Contains the encoding type for FDX-B transponder: 0 = Send out the number in BCD format 1 = Send out the number in raw Hex format 2 = Send out the number in raw Hex format with bytes swapped (Elatec format)

5. NX2EXP-LF Protocol

The NX2EXP-LF can be accessed with two different protocols:

1. ASCII based protocol
2. Binary based protocol (IDC protocol)

5.1. ASCII based protocol

The ASCII based protocol can be used to send commands to the NX2EXP-LF and to receive events from the NX2EXP-LF.

5.1.5. Message structure

The messages between the host and the NX2EXP-LF can be divided into 3 message types:

1. Commands (Send from the host to the NX2EXP-LF)
2. Responses (Send from the NX2EXP-LF to the host)
3. Events (Send from the NX2EXP-LF to the host)

The protocol uses the following data format:

Communication from Host to NX2EXP-LF

Message	<CR>

Response/Events from NX2EXP-LF to Host

Message	<CR>	<LF>

The command messages can be divided into 2 subtypes:

1. Requesting Information
2. Setting information

These messages follow the data format:

Request information

Command	[Parameters]	?	<CR>

Setting information

Command	[Parameters]	=	Data	<CR>

5.2. ASCII Messages definition

The following Command Messages are defined:

GetVersion	Retrieves the version of the NX2EXP-LF module hardware and firmware
GetProperty	Read the value of a property
SetProperty	Write the value of a property
GetFieldStatus	Get the current status of the RF field
SetFieldStatus	Set the status of the RF Field
GetInventory	Perform one inventory and get the response
SelectTransponder	Perform a select command to a transponder
ReadData	Perform a read command to a selected transponder
WriteData	Perform a write command to a selected transponder

On each command it is also possible to get a Error Response

5.2.1. GetVersion

The “GetVersion” command can be used to retrieve the version string. It contains information about the hardware version, the terminal interface and firmware version.

Command Code	V
Parameters	None
Type	Information Request
Response	Version string
Example Message	V?<CR>
Example Response	NX2EXP-LF|HW:V1|T:X2|FW:V0.5-b018<CR><LF>

The “Version string” is a combination of several fields. The fields are separated by the ‘|’ character. Each field except the first one is a key/value pair. The key/value uses the ‘:’ as a separator. The first field is always the name of the product. The other fields can contain the following information

HW	Hardware version
T	Terminal interface version
FW	Firmware version

5.2.2. GetProperty

The “GetProperty” command can be used to retrieve the value of a property. A property is a configuration parameter which is stored in the non-volatile memory of the NX2EXP-LF. A property is defined by certain characteristics like Identifier number, Type, Size and access details. The ASCII message only needs to deal with the identifier number as the value is always converted to a string.

Command Code	P
Parameters	PropertyID
Type	Information Request
Response	PropertyID and Value
Example Message	P81001?<CR>
Example Response	P81001=15<CR><LF>

5.2.3. SetProperty

The “SetProperty” command can be used to set the value of a property. A property is a configuration parameter which is stored in the non-volatile memory of the NX2EXP-LF. A property is defined by certain characteristics like Identifier number, Type, Size and access details. The ASCII message only needs to deal with the identifier number as the value is always converted from a string.

Command Code	P
Parameters	PropertyID=Value
Type	Information Set
Response	OK response or Error response
Example Message	P81001=7<CR>
Example Response	OK<CR><LF>

5.2.4. GetFieldStatus

The “GetFieldStatus” command can be used to retrieve the State of the RF Field. When the RF Field is activated, it means that autodetection is taking place. Note that is not always the represents the status of the field generation on that exact moment. Certain transponders need to have the field turned off for short moments to be able to send back data. When a Select command is issued the autodetection is automatically turned off but the field will still be active.

Command Code	F
Parameters	None
Type	Information Request
Response	Field status 0 = Off, 1 = On
Example Message	F?<CR>
Example Response	F=1<CR><LF>

5.2.5. SetFieldStatus

The “SetFieldStatus” command can be used to set the State of the RF Field. When the RF Field is activated, it means that autodetection is taking place.

Command Code	F
Parameters	=<Status> 0= Off, 1 = On
Type	Information Set
Response	OK or Error response
Example Message	F=1<CR>
Example Response	OK<CR><LF>

5.2.6. GetInventory

The “GetInventory” command can be used to start one inventory cycle and report back any detected transponders. One cycle means to try to detect all enabled transponder types. It is also possible to pass in the transponder type number to only try to detect one transponder type.

Command Code	I
Parameters	None or transponder type
Type	Information Request
Response	0 or more transponder detections. When done an OK will be send
Example Message	I01?<CR>
Example Response	D,02,0020000806A678B<CR><LF>OK<CR><LF>

5.2.7. SelectTransponder

The “SelectTransponder” command can be used to send a Select command to a transponder. This applies to transponders which need to be select to perform other actions with the transponder. An example is the Hitag-S transponder. After a succesfull select command the RF field will remain in the on state and any autodetection is stopped. Any further actions with the transponder should be performed without resetting the transponder. This means the transponder has to remain in the RF field.

Command Code	S
Parameters	<Transponder Type>=<Transponder UID>
Type	Information Set
Response	S<TransponderType>=<Config register contents>
Example Message	S03=055E8C10<CR>
Example Response	S03=A00000C1<CR><LF>

5.2.8. ReadData

The “ReadData” command can be used to send a read user data command to the transponder. This command should always be performed after selecting a transponder. Otherwise, the command will fail. This command can be repeated to read more than one block of data. The data amount is always 1 block. For most transponders this means 4 bytes of data. Note that the transponder type is determined by the one used for the select command.

Command Code	R
Parameters	<Block Offset>
Type	Information Request
Response	R<Block Offset>=<Data>
Example Message	R11?<CR>
Example Response	R11=FA59B107<CR><LF>

5.2.9. WriteData

The “WriteData” command can be used to send a write user data command to the transponder. This command should always be performed after selecting a transponder. Otherwise, the command will fail. This command can be repeated to write more than one block of data. The data amount is always 1 block. For most transponders this means 4 bytes of data. Note that the transponder type is determined by the one used for the select command.

Command Code	W
Parameters	<Block Offset>=<Data>
Type	Information Set
Response	OK
Example Message	W11=12345678<CR>
Example Response	OK<CR><LF>

5.2.10. Detection

The “Detection” Event is sent by the NX2EXP-LF when it detects a transponder. It contains the transponder number and type.

Command Code	D
Parameters	Not applicable
Type	Event
Response	D,<Type>,<Ttransponder Number>
Example Message	Not applicable
Example Response	D,02,0020000806A678B<CR><LF>

5.2.11. Error

The “Error” Response is sent by the NX2EXP-LF when it encounters an error as a response on a command from the host. The Error response contains an error code to identify the problem.

Command Code	Any
Parameters	
Type	Error Response
Response	ERR=<Error Number
Example Message	W11=12345678<CR>
Example Response	ERR=3<CR><LF>

5.3. Binary based protocol (IDC protocol)

The binary protocol can be used to update the firmware of the NX2EXP-LF. It also allows reading and changing the properties of the NX2EXP-LF.

The protocol will not be described in this document.