NDC MiCon Connector userguide v 1.4

name	NDC-MiConConnector-userguide.docx			
project	NDC MiCon/RS485			
content	Userguide for system integrators			
version	**state**	**Changes**	**author**	**Date**
1.0	Draft	start version	Rha	2021-07-28
1.1	Release	Fixed error in answer to GetIdentification	Rha	2021-09-07
1.2	Release	Clarification of CRC generation, examples included, details on error code, restrictions on documented fw version	Rha	2021-10-06
1.3	Release	Added section about reading SystemInfo	Rha	2022-04-27
1.4	Release	Added flow information in GetCurrentData	Rha	2022-08-18

References

No.	Description	filename	remarks

Contents

Inhalt

1 Preliminaries

This userguide document is targeted at “system integrators”, who are defined but not limited to partners who integrate ZENNER devices into their readout systems, mostly by implementing own software which accesses the devices either via IR and/or by parsing the data sent via radio.

The content of this document does not constitute a performance declaration and is by no means a complete description of the device. Instead, items are selected which seem to be relevant for typical system integrators. The document will be modified over time and is provided as-is without any warranties.

All information within this document is confidential and shall not be distributed further.

AUS DIESEM DOKUMENT ABGELEITETE PRODUKTZUSAGEN BEDÜRFEN DER SEPARATEN FREIGABE.
PRODUCT FEATURES DERIVED FROM THIS DOCUMENT REQUIRE SEPARATE CONFIRMATION AND RELEASE.

2 Overview

This document describes the usage of NDC MiCon Connector in connection with IUW (ZENNER's ultrasonic water meter) for the use case of reading the current meter value using software not provided by ZENNER.

The following picture shows the NDC MiCon Connector, SAP 158406. On the one side, it provides a NDC "antenna" for NFC communication with IUW. On the other side, it provides a SUB-D connector for serial communication via RS485.

The lower layers of NFC communication (physical layer and link layer) are handled by the MiCon Connector, while providing a transparent command interface between IUW and the communication partner behind the RS485 line.

3 Communication settings

3.1 SUB-D pin assignment

Pin assignment of SUB-D connector is as follows:

SUB D connector pin	Usage	Remarks
4	V_{CC}, 3 – 3.3 V	must be provided externally
5	GND	Common ground for power supply of MiCon connector and RS485
6	RS485_B	
7	RS485_A	

3.2 Communication settings

Communication settings for serial communication over RS485 are: 115200 baud, 8 data bits, 1 stop bit, even parity.

4 Communication protocol

For the use case described here (reading current meter value via NFC) there are only two commands which are necessary:

- GetIdentification (0x01)
- GetCurrentData (0x29)

The command GetIdentification must be issued first, before any other communication. It delivers the “seed” for CRC calculation needed for all later transmissions of further commands like GetCurrentData.

4.1 GetIdentification (0x01)

GetIdentification command is defined as follows:

Byte position (start)	Length [Byte]	Value	Meaning
0	1	0x03	length of command w/o CRC and w/o length field
1	1	0x01	Command ID
2	2	0x5A 0xA5	fixed value
4	2	0x2C17	fixed value (CRC)

Example

Example of a valid command byte sequence: 03015AA5172C

Response is defined as follows (only required fields documented here):

Byte position (start)	Length [Byte]	Value	Meaning
0	1	0x2D	length of response w/o CRC and w/o length field
1	1	0x01	command ID
2	1	0x01	response version, must be 1 or 2 here **
3	1	0x01	protocol version, must be 1 here **
4	5		not documented here
9	4		serial number of IUW*
13	If response version 1: 4		not documented here

	If response version 2: 6		
If response version 1: 17 If response version 2: 19	4		firmware version*, must be 1.5.1 or higher for this documentation to be applicable. Coding see table below
If response version 1: 21 If response version 2: 23	4		meter_ID value*
If response version 1: 25 If response version 2: 27	21		not documented here
If response version 1: 46 If response version 2: 48	2		CRC

*: note the byte order LSB first, **: other versions exist, but not documented here

Firmware version:

Mask	Meaning	Using
0xff000000	MajorVersion	Incremented with major functionality changes
0x00ff0000	MinorVersion	Incremented by small enhancements
0x0000f000	Revision	Incremented with troubleshooting function without change
0x00000fff	DeviceIdentity	Fix for each device type or series. See table below.

Example:

```
Fw version field 09300104
Hex interpreted:
0x04013009
  04____________ MajorVersion 4
    01__________ MinorVersion 1
      3_________ Revision 3
       009_____ DeviceIdentity (not documented here)
```

Example:

Response on command byte sequence 03015AA5172C:

2D0101010730496A9955332211260008010610050133 5F152C380500000E08842B22BA6B00B36B08540
0060B0000FBB6

Byte position (start)	Length [Byte]	Value	Meaning
0	1	0x2D	length of response w/o CRC and w/o length field
1	1	0x01	command ID
2	1	0x01	response version, example here for version 1
3	1	0x01	protocol version, must be 1
4	5	0730496A99	not documented here
9	4	55332211	serial number of IUW*
13	4	26000801	not documented here
17	4	06100501	fw version 1.5.1
21	4	335F152C	meter_ID value*
25	21	380500000E08842B22BA6B00B36B085400060B0000	not documented here
47	2	FBB6	CRC*

*: note the byte order LSB first

4.2 GetCurrentData (0x29)

GetCurrentData command is defined as follows:

Byte position (start)	Length [Byte]	Value	Meaning
0	1	0x03	length of command w/o CRC and w/o length field
1	1	0x29	Command ID
2	2	0x5A 0xA5	fixed value
4	2		CRC

Example:
Example of a valid command byte sequence (using meter_ID 0x00): 03295AA5B087

CRC calculation is to be done as follows:

- CRC including all bytes in the command
- CCITT CRC-16
- Polynom 0x1021
- Inital value: to be calculated based on the meter_ID retrieved by response to GetIdentification command, as follows:
  CRC initial value = (meter_ID>>16) XOR meter_ID

Response is defined as follows (only required fields documented here):

Byte position (start)	Length [Byte]	Value	Meaning
0	1	0x28	length of response w/o CRC and w/o length field
1	1	0x29	command ID
2	6		not documented here
8	1	0x00	Base meter unit: 0: m3 resp. m3/h 1:GAL resp. GAL/minute
9	8		accumulated volume, encoded in IEE754 double precision floating point format*
17	8		volume in flow direction, encoded in IEE754 double precision floating point format*
25	8		Volume in return direction, encoded in IEE754 double precision floating point format*
33	4		Flow, encoded in IEE754 floating point format*
37	4		not documented here
41	2		CRC

*: note the byte order LSB first

Example:

Calculation of CRC initial value from the response in the example above:

- meterID: 0x2C155F33
- meterID>>16 == 0x2C15
- meterID && 0xFFFF = 0x5F33

- (meterID >>16) XOR (meterID) = 0x7326
- CRC initial value := 0x7326
- CRC of command sequence 0x03 0x29 0x5A 0xA5: 0x6A7F
- Complete command sequence: 03 29 5A A5 7F 6A

Response on command sequence 03 29 5A A5 7F 6A:
2829150A060A1D0900827D4E86D91AA0401021A27C9CB5A04018ACD5A75E8E62C03BE0C03E713D
0042B4D9

Byte position (start)	Length [Byte]	Value	Meaning
0	1	28	length of response w/o CRC and w/o length field
1	1	29	command ID
2	6	150A060A1D09	not documented here
8	1	00	Base meter unit: 0: m3 resp. m3/h 1:GAL resp. GAL/minute
9	8	827D4E86D91AA040	accumulated volume, encoded in IEE754 double precision floating point format
17	24	1021A27C9CB5A04018ACD5A75E 8E62C03BE0C03E713D0042	not documented here
41	2	B4D9	CRC

- base meter unit: m3
- IEEE 754 value: 40A01AD9864E7D82
- converts to 2.061424... x10exp3 [m3]

4.3 Error handling

In case the CRC was wrong, device will answer with an error message:

Byte position (start)	Length [Byte]	Value	Meaning
0	1	0x05	length of response w/o CRC

1	1	0xFE	indicating an error
2	1		Source and type field: 0x00-0x7F: NFC communication error, source and type field contains copy of command ID of request 0x80-0xFE: error message of attached device, source and type field contains copy of command ID of request or'd with 0x80 0xFF: NDC connector error
3	3		response, depending on value of source and type field
6	2		CRC

Source and type field 0x00 – 0x7F:

Byte	Length [byte]	Value	meaning
1	1		actual transceiver state
2	2		coupler result code

Source and type field 0x80 – 0xFE:

Byte	Length [byte]	Value	meaning
1	1	0x01	NFC version (currently only: 0x01)
2	2		error code s. table below*

*: note the byte order LSB first

Error code	Meaning	remark
0x0000	No error, response ok	
0x0001	Illegal byte size	
0x0002	Received CRC error	
0x0003	Unknown, not supported command	
0x0004	Write permission error	
0x0005	Hardware access error	
0x0006	Read permission error	
0x0007	Not allowed on protected meter	
0x0008	Parameter value not allowed	
0x0009	busy	

Example:

Answer on request 03295AA57F6B with wrong CRC: 05 FE A9 01 0200 E1B5

Answer on correct request F103295AA57F6A but NDC not attached to device: 05 FE 29 03 8700 89A2

4.4 Additional commands

4.4.1 GetSystemState (0x0F)

Another interesting command is the command GetSystemState. It provides current status information of the device.

In short notation, request and response are as follows:

Request:

Length	Command	data	16 bit crc
0x03	0x0f	0x5A, 0xA5	

Response:

Length	Command	data	16 bit crc
0xnn	0x0f	variable bytes depend on device firmware version	

The data field of the response of IUW firmware 1.5.1 includes:

SystemState[UINT32] (for internal use)

NFC_PowerFailCounter [UINT16]

SystemInfo [UINT32]

SystemInfo is defined as follows:

Bit	*Symbol*	*Second line*	*Blinking*	*Danger sign*	*Description / reaction*
0		SFxx	1		Smart function event; xx = event number; see table below
1		IF01	1		NDC module is not able to use the selected communication scenario
2					
3					Device in sleep (delivery mode)
4				x	NTAG I2C fault
5					Display interpreter error
6	Batt				Calculated battery live time over
7	Batt			x	Battery live time over detected by under voltage (Device works)
8				x	Write protection not active
9				x	Ultrasonic channel 1 corrupt
10				x	Ultrasonic channel 2 corrupt
11				x	Temperature sensor corrupt
12		CHK	1		Test-View is active
13					Reverse flow
14		tOR	1	x	Temperature out of range
15		FOR	1	x	Flow out of range
16				x	CRC error firmware code
17				x	CRC error configuration
18					
19					
20					
21					
22		AIR	1		Bubbles in the water
23		dry			No water in the tube (or all ultrasonic channels corrupted)
24					
25					
28					
27					
28					
29		Err5		x	The meter lost its accumulated data
30		Err6		x	TDC-Error
31	Batt	Err7		x	Battery down (Device out of order)

Smart function event numbers are defined by the (loadable) smart function and not within the firmware itself.

Therefore new event numbers, which are not listed here, can exist by creating new functions. The following list shows the numbers used by the per factory default loaded smart functions.

Leakage	1	Burst	6
ReverseInstallation	2	Dry	7
BatteryWarning	3	Frost	8
Oversized	4	Backflow	9
Undersized	5	NoConsumption	10