ND103-v12 AMS D180222

TECHNICAL DATA SHEET
MW-D7B / MW-D7G

MW-D7G and MW-D7B

Documentation version: ND103-v12
Valid from firmware version:: MW-D7x-v2.2.A1.16

TABLE OF CONTENTS:

1 INTRODUCTION

MW-D7x is a wall-mounted reader of RFID cards which works on 125kHz rated frequency

Main features:

- Support of Unique, Q5, Hitag or HID transponders,
- Interfaces:
  - RS232
  - RS485
  - 1-WIRE
  - WIEGAND
- Built-in buzzer,
- Built-in LED RGB of common purpose
- Button
- Available in colors:
  - black (MW-D7B)
  - grey (MW-D7G)

2 TECHNICAL DATA

Supported functionality depending on transponder / card type:			
Transponder type	ID number readout	Full writing and readout of memory blocks	Writing and readout using encryption
Unique	YES	-	-
Q5	YES	YES	-
HID	YES	-	-
HITAG	YES	YES	NO

MW-D7x reader parameters:	
Supply voltage	7-15V
Max. supply current	120 mA
Rated operation radio frequency of module	125 kHz
Read-out distance of transponders	Up to 8 cm
Dimensions(wid.*len.*heig)	44x83x14 mm
IP rating	IP54
Front button	YES
Interfaces	RS-232 RS-485 Wiegand 1-WIRE
Input/Output	Anticollision (In/Out) PinOUT (OC type Out) PinIN0 (In) PinIN1 (In)
Load capacity of the PinOUT	max. 200mA
Maximum voltage at PinOUT	40V
Input voltage state L (PinIN0, PinIN1)	0…0,2V
Input voltage state H (PinIN0, PinIN1)	3,75…25V Internal pull-up +5V (10kΩ)
Operating temperature	0-60˚C

3 TERMINALS

Wire	Name	Function
Red	VCC	VCC (+)
Blue	GND	GND (-)
White	Anticollision	Output for connecting readers with each other, that are operating closely together
Brown	PinOUT	Output for any purpose
Green	PinINTERFACE1	Serial interface line (RS232_TX, RS485_B, WIEGAND0, 1WIRE)
Yellow	PinINTERFACE2	Serial interface line (RS232_RX, RS485_A, WIEGAND1)
Grey	PinIN0	Input for any purpose
Pink	PinIN1	Input for any purpose

4 INPUTS / OUTPUTS

4.1 INPUTS

Reader MW-D7x has three inputs:

1. PinIN0
2. PinIN1
3. Button

4.2 OUTPUTS

Reader MW-D7x has six outputs:

1. Color0 (LED RGB)
2. Color1 (LED RGB)
3. Color2 (LED RGB)
4. Color3 (LED RGB)
5. Buzzer
6. PinOUT
7.

WARNING:
The active state of the output buzzer locks transponders readings.

4.3 4.3. RGB LED

Reader MW-D7x using LEDs can display 4 colors: white, green, red and blue. Colour codes are shown in the table below:

Table 4.1 Color codes table

Colour code	Colour
0	Red
1	Green
2	Blue
3	White

Assigning a specific color to the ColorX output can be done with the *C_ConfigLed* command.

4.4 SIGNAL SOURCES CONTROLLING OUTPUTS

MW-D7x reader has 18 sources of logical signals. These signals can be used to control outputs. The table below contains a list of all sources and values of signals generated by them.

Table 4.2 Signal sources

ID	Nazwa	Opis
0	„0”	Signal source with value of 0
1	„1”	Signal source with value of 1
2	Button	Source reflecting the status of the front button. It has value of 1 when the button is pressed and value of 0 otherwise
3	Every card	Source reflecting information about the presence of the card in the field. It has value of 1 when the card is the field and value of 0 otherwise
4 5 6 7	RS_0 RS_1 RS_2 RS_3	Sources controlled via RS232 serial interface. See *C_WriteOutput command.*
8 9	PinIN0 PinIN1	Sources controlled by a physical input pin using the INPUT block
10 11 12 13	SigA0 SigA1 SigA2 SigA3	Sources controlled by SIG_Ax block outputs
14 15 16	SigB0 SigB1 SigB2	Sources controlled by SIG_Bx block outputs

17	SigB3	
18 19 20 21	SigC0 SigC1 SigC2 SigC3	Sources controlled by SIG_Cx block outputs

4.4.1 SOURCE „0” AND SOURCE „1”

Signal source „0” has always value of 0, while the signal source “1” has the value of 1.

4.4.2 SOURCE BUTTON

Source reflecting the status of the front button. It has the value 1 when the button is pressed and the value 0 otherwise.
ATTENTION:
If a button is pressed for more than 3 minutes, it will be recalibrated and source value reset to zero

4.4.3 SOURCE ANY CARD

Source reflecting information about the presence of card in the field of the reader. It has the value 1 when the card is in the field and the value 0 in the opposite case.

4.4.4 SOURCE RS_X

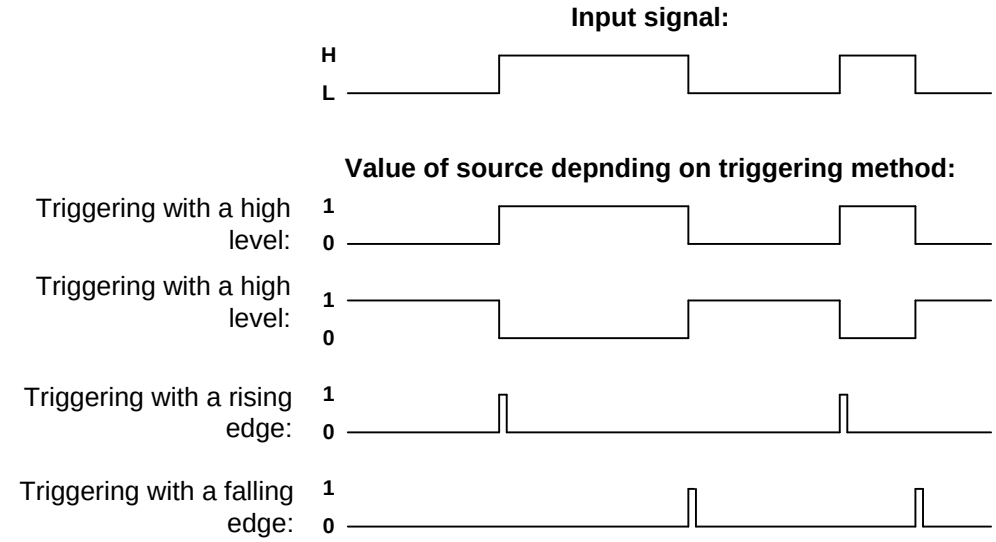
Sources controlled via RS232 serial interface. The source enables

- Setting the value 0
- Setting the value 1
- Setting the value 1 to a specified time, after which the source will automatically change the state to 0.

See RSx source state writing command

4.4.5 SOURCE PININX

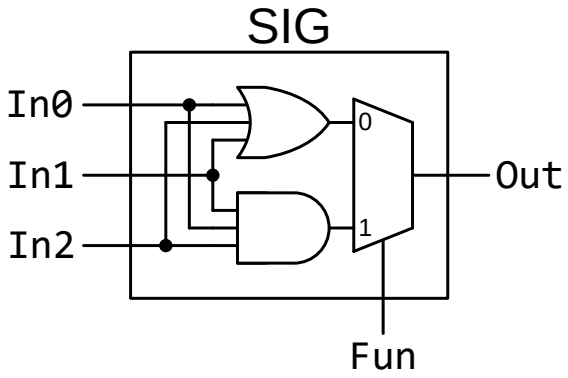
PinINx sources are controlled through physical inputs. Depending on the configuration, the source value has the value:



Configuration of the trigger method is done using Port configuration writing command.

4.4.6 SOURCE SIG_AX

MW-D7X reader has 4 virtual SIG_A blocks that allow you to perform logical operations on signals. Each block has 3 signal outputs, one function selection input and one output. Any signal source can be connected to the signal inputs of blocks. At the block output, depending on the *Fun* function selected, there will be a logical sum or logical product of input signals. The SIG blocks are configured using the *SIG_A block configuration* command.

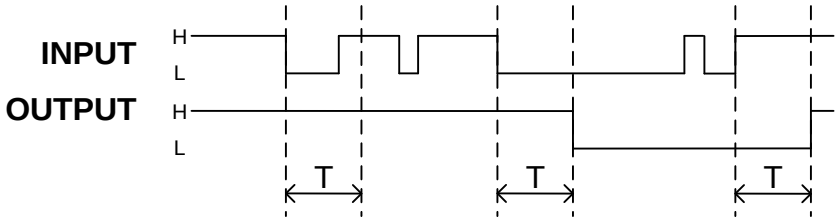


4.4.7 SOURCE SIG_BX

MW-D7X reader has 4 virtual SIG_B blocks that allow you to perform logical operations on signals. SIG_B block configuration is done using *SIG_B block configuration* command.

4.4.8 SOURCE SIG_CX

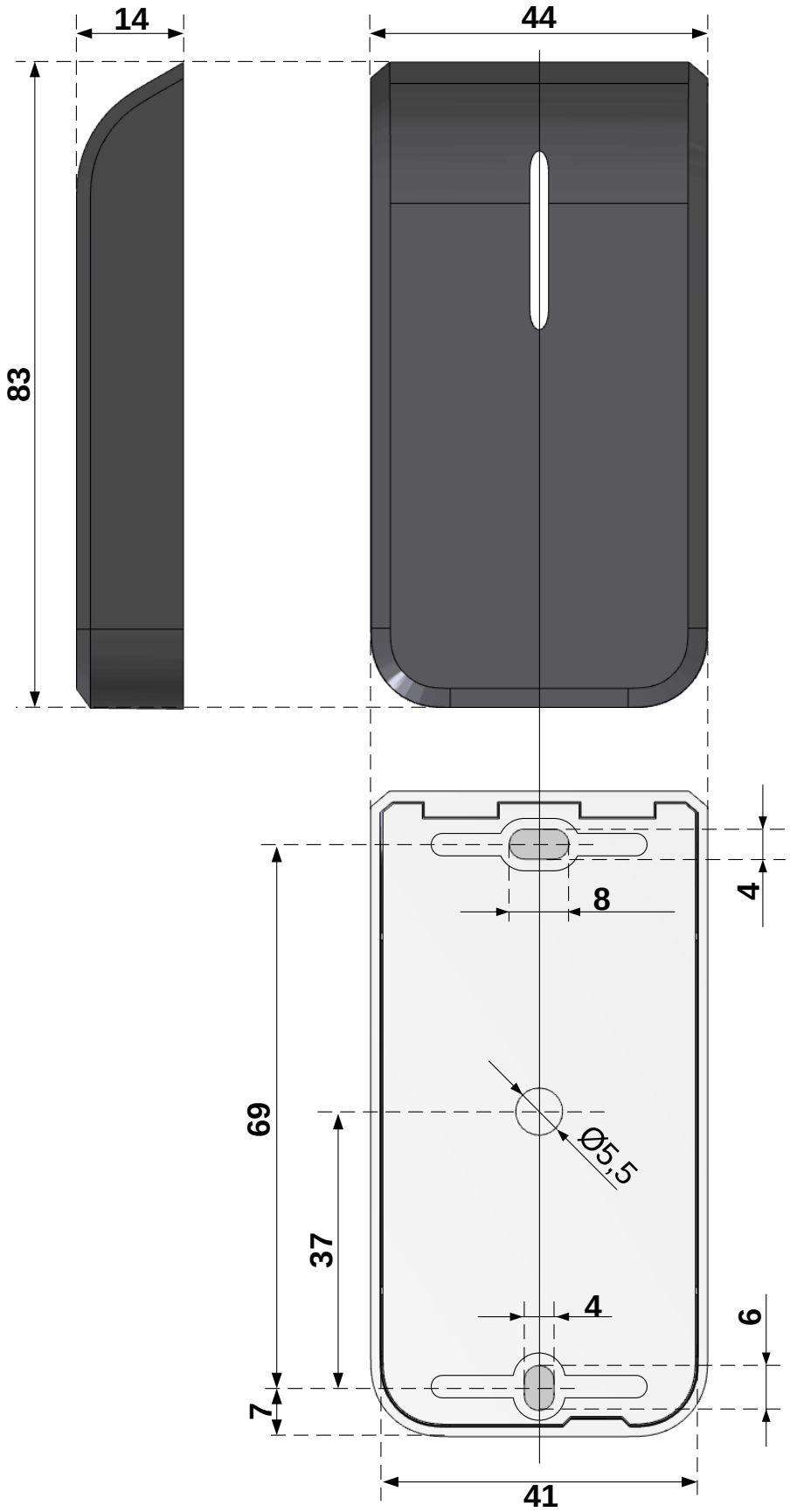
MW-D7X reader has 4 virtual SIG_C blocks that allow you to perform filtering of logical signals. State at the SIG_C output will change to the same as at input if the input state remains constant for the time defined by Time parameter. Configuration of SIG_C blocks is performed using SIG_B block configuration command.



4.1 Example of input and output course for SIG_C block

5 DIMENSIONS

Dimensions of the reader are shown in the figure below:



Cable length: 30cm

6 INTERFACE

MW-D7x reader supports 4 serial interfaces:

- RS232
- RS485
- 1-WIRE
- Wiegand

RS-232 and RS-485 interfaces are constantly listening in anticipation of the command. AutoReader sends the read ID via the interface selected in the AutoReader configuration.

6.1 6.1. INTERFACE 1-WIRE

After configuring the device to work in 1-WIRE mode, the reader emulates Dallas DS1990 series of "pills". As long as the card is in the field, reader will issue a unique number on the 1-WIRE bus. The format of the sent ID has the form:

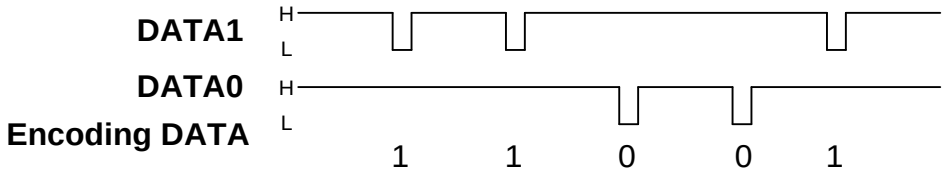
Family code	Transponder ID					Address	CRC
ConfFC	ID0	ID1	ID2	ID3	ID4	ConfAdr	xx

To change a parameter *Address* or *Family code*, please send a *C_SetInterfaceConfig* command to the reader.

6.2 6.2. INTERFACE WIEGAND

Reader, after being configured to operate in WIEGAND mode, sends a unique ID number of the read card in accordance with the Wiegand protocol with the following parameters:

Pulse duration (L level) 100us
Interval between impulses (H level) 1ms



MW-D7x reader allows you to change the WIEGAND frame length and to select the part of the ID card to be sent on the main line (bus).
Example:
Card ID = 0x123456789A = 0b000100100011010001010110011111000100110 10

WIEGAND parame-ters	Card ID / corresponding WIEGAND frame	
P1=26, P2=0	0b**000100100011010001010110**0111100010011010 P**000100100011010001010110**N	Card ID WIEGAND frame
P1=37, P2=0	0b**000100100011010001010110011111000100**11010 P**000100100011010001010110011111000100**N	Card ID WIEGAND frame
P1=26, P2=1	0b**000100100011010001010110011111000100**11010 P0101011001111000100110 10N	Card ID WIEGAND frame

P,N – parity bits

Another format e.g. WIEGAND, can be obtained by changing the configuration (Writing configuration of serial interface).

6.2.1 PARITY BITS

The MW-D7x reader enables configuration of the methods of counting parity bits. The configuration of Wiegand interface parameters can be done using the ***C_SetInterfaceConfig*** command.

Method 1 - example for WIGAND26

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
													E	E	E	E	E	E	E	E	E	E	E	E	P2
P1	O	O	O	O	O	O	O	O	O	O	O	O													

P1 – odd bit
P2 – even bit
O – denotes the bits from which the odd bit is calculated
E –denotes the bits from which the even bit is calculated

Method 2 - example for WIGAND26

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	E	E		E	E		E	E		E	E		E	E		E	E		E	E		E	E		P2
P1		O	O		O	O		O	O		O	O		O	O		O	O		O	O		O	O	

P1 – odd bit
P2 – even bit
O – denotes the bits from which the odd bit is calculated
E –denotes the bits from which the even bit is calculated

6.3 6.3. INTERFACE RS232 / RS485

MW-D7X reader, regardless of the configured interface, always listens to commands sent via RS232 or RS485 interface. In this documetation, the description of the protocol has been limited to the description of orders, responses and their parameters. The headline and CRC checksum are always present and are consistent with the full “Netronix Protocol” documentation.

Command frame:

Header	C_CommandName	Command_parameters 0…n	CRC

Response frame:

Header	C_CommandName +1	Response_parameters 0…m	OperationCode	CRC

ATTENTION:
Serial protocol operation can be tested by means of development tools including free of charge “FRAMER” software”.
(https://netronix.pl/en/software/framer#download)

7 COMMANDS AVAILABLE FOR RS232/RS485 INTERFACE

7.1 SERIAL INTERFACE CONFIGURATION

7.1.1 CONFIGURATION RECORD OF SERIAL INTERFACE

Command frame:

C_SetInterfaceConfig	Typ, P1, P2

Where:

Parameter name	Description	Value range
C_SetInterfaceConfig	Command for changing the serial interface settings	0x54
Typ	Interface type	0 – RS232 1 – RS485 2 – 1-WIRE 3 – WIEGAND
P1, P2	Parameters depending on value Type: **For Typ=0** P1 - logical address (RS232) P2 - transmission speed (RS232) **For Typ=1** P1 - logical address (RS485) P2 - transmission speed (RS485) **For Typ=2** P1 - ConfAdr (7th byte of Dallas frame) P2 - ConfFC (1st byte of Dallas frame) **For Typ=3** P1 - number of bits P2 MSB … LSB - - - - - - P L/M	 P1: 0x01 - 0xFE P2: - See Table 7.1 P1: 0x01 - 0xFE P2: - See Table 7.1 P1: 0x00 - 0xFF P2: 0x00 - 0xFF P1: 26 - 37 P2: **P** – This switch determines the method of counting the parity bits. P = 0 - method 1 P = 1 - method2 **L/M** – Switch determines which part of card ID will be sent in the WIEGAND frame.

Table 7.1 RS232 interface speed

ID	Speed
0	1200 bps
1	2400 bps
2	4800 bps
3	9600 bps
4	19200 bps
5	38400 bps
6	57600 bps
7	115200 bps

Response frame:

C_SetInterfaceConfig +1		Operation Code

7.1.2 CONFIGURATION READING OF SERIAL INTERFACE

Command frame:

C_GetInterfaceConfig	Typ

Where:

Parameter name	Description	Value range
C_GetInterfaceConfig	Command for changing the serial interface settings	0x56
Typ	Type of interface which configurations we want to read	0 – RS232 1 – RS485 2 – 1-WIRE 3 – WIEGAND

Response frame:

C_GetInterfaceConfig+1	Typ, P1, P2

Where:

Parameter name	Description	Value range
C_GetInterfaceConfig+1	Command for changing the serial interface settings	0x57
Typ	Interface type	0 – RS232 1 – RS485 2 – 1-WIRE 3 – WIEGAND
P1, P2	Parameters depending on value Type: For Typ=0 or Typ=1 P1 - device logical address P2 - transmission speed For Typ=1 P1 - Address (7th byte of Dallas frame) P2 - Family (1st byte of Dallas frame) For Typ=2 P1 - number of bits P2 – L/M	 P1: 0x01-0xFE P2: - See Table 7.1 P1: 0x00-0xFF P2: 0x00-0xFF P1: 26-48 P2: 0-1

7.2 COMMUNICATION COMMANDS FOR TRANSPONDERS

7.2.1 SELECTING THE TRANSPONDER TYPE

Command frame:

C_SetTransponderType	TransponderType, GAIN

Where:

Parameter name	Description	Value range
C_SetTransponderType	Command of transponder type changing	0x02
TransponderType	Transponder type we want to exchange data with	0x01 – Unique 0x02 – Q5 0x03 – HITAG 0x04 – HID

Response frame:

C_SetTransponderType +1		Operation Code

7.2.2 TRANSPONDER TYPE READOUT

Command frame:

C_GetTransponderType	TransponderType, GAIN

Where:

Parameter name	Description	Value range
C_GetTransponderType	Transponder type readout command	0x04

Response frame:

C_GetTransponderType +1	TransponderType	Operation Code

Where:

Parameter name	Description	Value range
C_SetTransponderType+1	Transponder type readout command	0x05
TransponderType	Identifier of transponder type with which we intend to exchange data	0x01 – Unique 0x02 – Q5 0x03 – HITAG 0x04 – HID

7.2.3 ON/OFF SWITCHING OF READER FIELD

Command frame:

C_TurnOnAntennaPower	State

Where:

Parameter name	Description	Value range
C_TurnOnAntennaPower	On/off switching of reader field	0x10
State	State	0x00 – switching the field off 0x01 – switching the field on

Response frame:

C_TurnOnAntennaPower +1		Operation Code

7.2.4 READING THE ID CARD UNIQUE NUMBER

Command frame:

C_Select	

Where:

Parameter name	Description	Value range
C_Select	ID readout	0x12

Response frame:

C_Select +1	Coll, TType, ID1…….IDn	Operation Code

Where:

Parameter name	Description	Value range
Coll	Collision information (HITAG transponders only)	0 – no collision 1 – collision of two or more transponders
TType	Information about the type of transponder from which the read ID number comes from	1 - Unique,Q5 3 - HITAG 4 - HID
ID1…IDn	Unique transponder number	ID1 – LSB, IDn – MSB

7.3 COMMANDS FOR COMMUNICATION WITH Q5 TRANSPONDERS

After selecting the Q5 type transponder, we have new commands at disposal, which will be used for two-way communication.

7.3.1 WRITING THE ID-UNIQUE NUMBER TO Q5 TRANSPONDER

Command frame:

C_UniqueWrite	Unique1..5, lock

Where:

Parameter name	Description	Value range
C_UniqueWrite	Writing command of id-unique	0x08
Unique1..5	5 bytes of ID number	0x00-0xff
lock	ID programming with rewrite lock	0 – without lock 1 – with lock

Response frame:

C_UniqueWrite +1		Operation Code

NOTE:
The Q5 type transponders do not have verification function of correct ID number writing. Getting proper code of operation does not guarantee correct assign of ID number. Make sure, that ID number has been assigned correctly reading the number with *C_Select* command.

7.3.2 SECTOR READOUT OF Q5 TRANSPONDER

Command frame:

C_ReadData	SectorNo,[Password1..4]

Gdzie:

Parameter name	Description	Value range
C_ReadData	Sector read command	0x1E
SectorNo	Read sector number	0x00-0x07
Password	Option - if sector which is being read is 4-byte password protected	0x00-0xff

Response frame:

C_ReadData +1		Operation Code

7.3.3 WRITING THE SECTOR OF Q5 TRANSPONDER

Command frame:

C_WriteData	SectorNo, Data1...4, Lock,[Password1..4]

Where:

Parameter name	Description	Value range
C_WriteData	Sector writing command	0x1C
SectorNo	Sector writing number	0x00-0xff
Data1..4	4 bytes of data	0x00-0x07
lock	Programming the sector with rewrite lock	0 – without lock 1 – with lock
Password1..4	Option – if we want to protect a sector with 4-byte password	0x00-0xff

Response frame:

C_WriteData+1		Operation Code

NOTE:
The Q5 type transponders do not have verification function of correct data write into sectors. Getting proper code of operation does not guarantee correct write. Make sure, that data has been written correctly reading it with *C_ReadBlock* command.

7.4 COMMANDS FOR COMMUNICATION WITH HITAG TRANSPONDERS

7.3.4 READING THE PAGE OF HITAG TRANSPONDER

Command frame:

C_ReadData	PageNo

Where:

Parameter name	Description	Value range
C_ReadData	Page readout command	0x1E
PageNo	Page readout number	0x00-0x3f

Response frame:

C_ReadData +1		KodOperacji

7.3.5 PAGE WRITING TO HITAG TRANSPONDER

Command frame:

C_WriteData	PageNo, Data1...4

Where:

Parameter name	Description	Value range
C_WriteData	Writing page command	0x1C
PageNo	Writing page number	0x00-0x3f
Data1..4	4 bytes of data	0x00-0xff

Response frame:

C_WriteData +1		OperationCode

7.3.6 SLEEP MODE OF HITAG TRANSPONDER

When operating with multiple HITAG transponders at the same time, it is necessary to enter unused transponders into sleep mode with *C_Halt* command.

Command frame:

headline	C_Halt		CRC

Where:

Parameter name	Description	Value range
C_Halt	Transponder sleep mode command	0x40

Response frame:

headline	C_Halt+1		Operation Code	CRC

7.4 SOURCES, ELECTRICAL INPUTS AND OUTPUTS

7.4.1 WRITING RSX SOURCE STATE

Command frame:

headline	C_WriteSourceRSx	IONo, State, [Time]	CRC

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_WriteSourceRSx	Record of Rsx source state	0x70
Source	Number of RSx source	0x04-0x07
State	Desired output status	0x00 lub 0x01
[Time]	Optional parameter. Time after which Rsx source will return to state 0 (x10ms)	0x00-0xFF

Response frame:

headline	C_WriteSourceRSx+1		Operation Code	CRC

7.4.2 SOURCE STATUS READOUT

Command frame:

headline	C_ReadSource	Source	CRC

Where:

Parameter name	Opis parametru	Zakres wartości
C_ReadSource	Source status readout	0x72
Source	Source value	See ID number from *Table 4.2 Signal sources*

Response frame:

headline	C_ReadSource +1	State	Operation Code	CRC

Parameter name	Opis parametru	Zakres wartości
C_ReadSource+1	Source status readout	0x73
State	Source value	0x04-0x07

7.4.3 WRITING PORT CONFIGURATION

Command frame:

C_SetIOConfig	IONo, Dir, P0

Where:

When configuring port as an output:

Parameter name	Description	Value range
C_SetIOConfig	Writing any port configuration	0x50
IONo	IO port number to be configured	0x00..0x05
Dir	Port direction	0x00 – output
P0	Control signal source	See ID number from Table 4.2 Signal sources

When configuring port as an input:

Parameter name	Description	Value range
C_SetIOConfig	Writing any port configuration	0x50
IONo	IO port number to be configured	0x06 – 0x07
Dir	Port direction	1 – input
P0	Byte specifying triggering method. See chapter: 4.4.5 PinINx source	0 – not negated 1 – negated 2 – reaction to rising slope 3 – reaction to falling slope

Not all MW-D7x ports have any direction. For proper configuration, correct direction should be given for the port.

Table 7.2 List of existing ports that can be controlled in MW-D7x

Port number	Direction	Description
0	output	PinOUT Physical output
1	output	KOLOR0
2	output	KOLOR1
3	output	KOLOR2
4	output	KOLOR3
5	output	BUZZER
6	input	PinIN0 Physical input
7	input	PinIN1 Physical input

Response frame:

headline	C_SetIOConfig +1		Operation Code	CRC

7.4.4 PORT CONFIGURATION READOUT

Command frame:

headline	C_GetIOConfig	IONo	CRC

Where:

Parameter name	Description	Value range
C_GetIOConfig	Port configuration readout	0x52
IONo	IO port number whose configuration is to be read	0x00…0x07

Response frame:

headline	C_GetIOConfig +1	Dir, P0	Operation Code	CRC

Where:

Parameter name	Description	Value range
Dir, P0	Parameters have the same form as when saving the configuration	

7.4.5 SIG_A BLOCKS CONFIGURATION

Command frame:

headline	C_ConfigSIG_A	SigNo, [Function, In0, In1, In2]	CRC

Where:

Parameter name	Description	Value range
C_ConfigSIG_A	Read/write configuration of SIG_A block	0x5C
SigNo	SIG_A block number , whose configuration is to be read / written	0x00…0x03
Function	Optional parameter – if present, the command saves new configuration. It specifies the type of function executed by SIG_A block.	0 – OR function 1 – AND function
In1, In2, In3	Optional parameter – if present, the command saves new configuration. Sources of input signals.	See ID number from *Table 4.2 Signal sources*

Response frame:

headline	C_ConfigSIG_A +1	Function, In0, In1, In2	Operation Code	CRC

Where:
Meaning of response parameters is identical to those described earlier.

7.4.6 SIG_B BLOCKS CONFIGURATION

Command frame:

C_ConfigSIG_B	No, [Source, Mode, Negation, Time, 0Time, 1Time]

Parameters: Source, Mode, Negation, Time, 0Time, 1Time are optional and if present, new configuration will be saved.

Parameter name	DescriptionRead/write configuration of SIG_A block	Value range
C_ConfigSIG_C	Read/write configuration of SIG_B block	0x60
No	SIG_B block number	0x00..0x03
Source	Source of control signal	See ID number from *Table 4.2 Signal sources*
Mode	Determines output behaviour	00 – square wave generator 01 – change in output status to the opposite of previous state 10 – directly
Negation	Output negation	0 – negated output 1 – direct output
Time	Runtime of switching state after activation stops. This time is expressed as: Runtime x 100ms. During "Runtime", you can configure the output that can generate a square wave. Time of '1' and '0' is set by the following parameters: 0Time and 1Time	0 – 255
0Time	Logical '0' time	0 – 255
1Time	Logical '1' time	0 – 255

Response frame:

C_ConfigSIG_B+1	No, Source, Mode, Negation, Time, 0Time, 1Time

Where:
Meaning of response parameters is identical to those described earlier.

7.4.7 SIG_C BLOCKS CONFIGURATION

Command frame:

C_ConfigSIG_C	No, [Source, Time]

Parameters: *Source, Time* are optional and if present, new configuration will be saved.

Parameter name	Description	Value range
C_ConfigSIG_C	Read/write configuration of SIG_C block	0x62
No	SIG_B block number	0x00..0x03
Source	Source of control signal	See ID number from *Table 4.2 Signal sources*
Time	Filtering time (x100ms)	0-255

Response frame:

C_ConfigSIG_C+1	No, Source, Time

Where:
Meaning of response parameters is identical to those described earlier.

7.4.8 COLOUR CONFIGURATION

Command frame:

headline	C_ConfigLed	[C0, C1, C2, C3]	CRC

Where:

Parameter name	Description	Value range
C_ConfigLed	Read/Write displayed colour configuration	0x5E
[C0,C1, C2, C3]	Optional parameters – if present, the command writes new configuration. C0 – kolor0 code, priority 1 (highest) C1 – kolor1 code, priority 2 C2 – kolor2 code, priority 3 C3 – kolor3 code, priority 4 (lowest)	See: *Table 4.1 Colour codes table*

Response frame:

headline	C_ConfigLed +1	C0, C1, C2, C3	Operation Code	CR

Where:

Meaning of response parameters is identical to those described earlier.

7.5 ACCESS PASSWORD

7.5.1 LOGIN TO READER

Command frame:

headline	C_LoginUser	Data1…n, 0x0	CRC

Where:

Parameter name	Description	Value range
C_LoginUser	Login to reader	0xb2
Data1…n	Any byte string	Any of ranges 0x01 ... 0xff. String length can be from 0 to 8 bytes
0x00	'0' ending the string	0x00

Response frame:

headline	C_LoginUser +1		Operation Code	CRC

7.5.2 PASSWORD CHANGE

Command frame:

headline	C_ChangeLoginUser	Data1…n, 0x0	CRC

Where:

Parameter name	Description	Value range
C_ChangeLoginUser	Password change	0xb4
Data1…n	Any byte string, that will be valid access password	Any of ranges 0x01 ... 0xff. Chain length can be from 0 to 8 bytes
0x00	'0' ending the string	0x00

If Data1 = 0x00, then the reader will not be password protected. You can set a new password at any time so that the reader will be protected by a password.

Responjse frame:

headline	C_ChangeLoginUser+1		Operation Code	CRC

7.5.3 LOGOUT FROM READER

The order will void last password you provided.

Ramka rozkazu:

headline	C_LogoutUser		CRC

Where:

Parameter name	Description	Value range
C_LogoutUser	Logout from reader	0xd6

Response frame:

headline	C_LogoutUser +1		Operation Code	CRC

7.6 *AUTOREADER* MECHANISM

7.6.1 SAVING MACHINE CONFIGURATION

This command configures the operating mode of machine reading unique UID transponder number.

Described reader gives you opportunity to temporarily suspend machine operation in case of proper transmission on the RS link.

If reader works in mixed mode, i.e.
UID reading machine is running, and:
the master device (computer, controller) communicates with a reader or with use of a transponder reader,
it is necessary to properly configure a reader so that in the case of transmissions with a reader or with a transponder, the reading machine suspends its work.

Ramka rozkazu:

headline	C_SetAutoReaderConfig	ATrig, AOfflineTime, Aserial, AMode, Abuzz, Amulti, AInterface	CRC

Where:

Parameter name	Description	Value range
C_SetAutoReaderConfig	Saving machine configuration	0x58
ATrig	Defines when a UID reading machine should operate	0 - machine switched off permanently 1 - machine switched on permanently 2 - is activated automatically when there is no transmission on RS / USB for a longer period than AOfflineTime 3 - turns on automatically when there are no commands for communication with a transponder for longer period than AOfflineTime
AOfflineTime	'No transmission' time on RS / USB T = AofflineTime * [100 ms] No transmission can refer to any commands (Atrig = 2), or commands to communicate with a transponder (Atrig = 3). Communication commands with a transponder are: C_TurnOnAntennaPower C_Select	0x00…0xff
ASerial	Automatic sending of a UID transponder number after automatic reading from a transponder	0 - never 1 – only first time transponder application 2 – sends everything
AMode	Configurational byte determining sent ID format	I=1 – number in reverse order E=1 - extended information about collision signaling and card type

	Format: MSB LSB I E F<1,0> C<1,0> D ID ATTENTION: 'E' bytes and F<1,0> only matters for Ainterface = 0 or Ainterface=1. C1, C0, D bytes only matters for ASCII format (F<1,0>=1)	F<1,0>=0 – ID in Netronix frame format F<1,0>=1 – ID in ASCII format F<1,0>=2 – ID in binary format C<1,0>=0 – without the end of line mark C<1,0>=1 – CR end mark C<1,0>=2 – LF end mark C<1,0>=3 – CRLF end mark ID - extended information about the reader's address set for the RS485 bus
ABuzz	Automatic readout signaling using buzzer after automatic UID readout from a transponder	0 - never 1 - only first time transponder application 2 – signals all
AMulti	A configuration byte that specifies the types of transponders read by autoreader. The bit set to 1 indicates that the transponder type will be read. Format: MSB LSB - - H2 HID H1 U/Q5 - -	U/Q5 – Unique / Q5 H1 – Hitag1 HID – HID H2 – Hitag2
AInterface	Choosing the interface after which the autoreader machine sends the read ID	0 – RS232 1 – RS485 2 – 1-WIRE 3 – WIEGAND

Response frame:

headline	C_ SetAutoReaderConfig +1		Operation Code	CRC

7.6.2 CONFIGURATION READOUT FROM THE MACHINE

Command frame:

headline	C_ GetAutoReaderConfig		CRC

Where:

Nazwa parametru	Opis parametru	Zakres wartości
C_GetAutoReaderConfig	Configuration readout from the machine	0x5a

Response frame:

headline	C_ GetAutoReaderConfig +1	ATrig, AOfflineTime, ASerial, AMode, Abuzz, AMulti	Operation Code	CRC

Where:
Meaning of response parameters is identical to those described earlier.

7.7 OTHER COMMANDS

7.7.1 7.8.1. REMOTE READER RESET

Command frame:

headline	C_Reset		CRC

Where:

Nazwa parametru	Opis parametru	Zakres wartości
C_Reset	Remote reset of reader	0xd0

Response frame:

headline	C_Reset +1		Operation Code	CRC

7.8.2. Readout of a reader firmware version

Command frame:

headline	C_FirmwareVersion		CRC

Where:

Nazwa parametru	Opis parametru	Zakres wartości
C_FirmwareVersion	Readout of reader firmware version	0xfe

Response frame:

headline	C_FirmwareVersion+1	Data1…..n	Operation Code	CRC

Where:

Data1…n is a string of characters stored in the form of ASCII codes.

7.8 MEANING OF OPERATION CODES IN RESPONSE FRAMES

Table 7.3 Operation codes

Nazwa kodu operacji	Opis	Wartość
OC_Error	error	0x00
OC_ParityError	Parity error	0x01
OC_RangeError	Range error	0x02
OC_LengthError	Amount of data error	0x03
OC_ParameterError	Parameter error	0x04
OC_Busy	Temporary occupation of internal modules	0x05
OC_NoACKFromSlave	Lack of internal communication	0x22
OC_CommandUnknown	Unknown command	0x07
OC_WrongPassword	Wrong password or last password has expired, i.e. an automatic Log-Out has taken place	0x09
OC_NoCard	No transponder	0x0a
OC_BadFormat	Bad data format	0x18
OC_FrameError	Transmission error. It can indicate existing faults	0x19
OC_NoAnswer	No response from transponder	0x1E
OC_TimeOut	Operation time exceeded. It can indicate a lack of transponder in the reader field	0x16
OC_Successful	Operation completed successfuly	0xFF

8 MODBUS RTU PROTOCOL

NOTE:
When using the MODBUS protocol, the reader should not send automatically the readed ID.
To disable automatic ID sending, enter the value 0 in the register with the address 1022 (ASerial).

8.1 SUPPORTED MODBUS PROTOCOL FUNCTIONS

Function	Description
0x01	Read Coils
0x02	Read Discrete Inputs
0x03	Read Holding Regs
0x04	Read Input Regs
0x05	Write Single Coil
0x06	Write Single Reg
0x10	Write Multiple Regs

8.2 MODBUS ADRESS

8.2.1 REGISTER ADDRESSES FOR READING TRANSPONDER ID

Address	Type	R/W	Description
996	Holding Reg	R/W	New card state Reading: 1-new transponder detect Writing: 0 – clear this flag
997	Holding Reg	R	B<15:8> - Transponder type B<7:0> - Number of collisions
998	Holding Reg	R	ID Length
999	Holding Reg	R	Time counter since last reading (x100ms)
1000	Holding Reg	R	Transponder ID [0]
1001	Holding Reg	R	Transponder ID [1]
1002	Holding Reg	R	Transponder ID [2]
1003	Holding Reg	R	Transponder ID [3]
1004	Holding Reg	R	Transponder ID [4]
1005	Holding Reg	R	Transponder ID [5]
1006	Holding Reg	R	Transponder ID [6]
1007	Holding Reg	R	Transponder ID [7]

8.2.2 REGISTER ADDRESSES FOR READING/SAVING AUTOREADER CONFIGURATION

Address	Type	R/W	Description
1020	Holding Reg	R/W	ATrig
1021	Holding Reg	R/W	AOfflineTimer
1022	Holding Reg	R/W	ASerial
1023	Holding Reg	R/W	B<15:8> - AModeParam, B<7:0> - AMode
1024	Holding Reg	R/W	ABuzz
1025	Holding Reg	R/W	AMulti
1026	Holding Reg	R/W	AInterface

The meaning of registers is the same as for the C_SetAutoReaderConfig command.

8.2.3 REGISTER ADDRESSES FOR READING/SAVING RS232 / RS48 CONFIGURATION

Address	Type	R/W	Description
1030	Holding Reg	R/W	Device address on the RS232 bus
1031	Holding Reg	R/W	Baudrate on the RS232 bus (See **Błąd! Nie można odnaleźć źródła odwołania.**)
1032	Holding Reg	R/W	Device address on the RS485 bus

1033	Holding Reg	R/W	Baudrate on the RS485 bus (See **Błąd! Nie można odnaleźć źródła odwołania.**)

8.2.4 REGISTER ADDRESSES FOR READING/SAVING BLOCK SIG_A CONFIGURATIONS

Address	Type	R/W	Description
1040	Holding Reg	R/W	SigA_0 – Function
1041	Holding Reg	R/W	SigA_0 – In0
1042	Holding Reg	R/W	SigA_0 – In1
1043	Holding Reg	R/W	SigA_0 – In2
1044	Holding Reg	R/W	SigA_1 – Function
1045	Holding Reg	R/W	SigA_1 – In0
1046	Holding Reg	R/W	SigA_1 – In1
1047	Holding Reg	R/W	SigA_1 – In2
1048	Holding Reg	R/W	SigA_2 – Function
1049	Holding Reg	R/W	SigA_2 – In0
1050	Holding Reg	R/W	SigA_2 – In1
1051	Holding Reg	R/W	SigA_2 – In2
1052	Holding Reg	R/W	SigA_3 – Function
1053	Holding Reg	R/W	SigA_3 – In0
1054	Holding Reg	R/W	SigA_3 – In1
1055	Holding Reg	R/W	SigA_3 – In2

8.2.5 REGISTER ADDRESSES FOR READING/SAVING BLOCK SIG_B CONFIGURATIONS

Address	Type	R/W	Description
1060	Holding Reg	R/W	SigB_0 – Source
1061	Holding Reg	R/W	SigB_0 – Mode
1062	Holding Reg	R/W	SigB_0 – Negation
1063	Holding Reg	R/W	SigB_0 – Time
1064	Holding Reg	R/W	SigB_0 – 0Time
1065	Holding Reg	R/W	SigB_0 – 1Time
1066	Holding Reg	R/W	SigB_1 – Source
1067	Holding Reg	R/W	SigB_1 – Mode
1068	Holding Reg	R/W	SigB_1 – Negation
1069	Holding Reg	R/W	SigB_1 – Time
1070	Holding Reg	R/W	SigB_1 – 0Time
1071	Holding Reg	R/W	SigB_1 – 1Time
1072	Holding Reg	R/W	SigB_2 – Source
1073	Holding Reg	R/W	SigB_2 – Mode
1074	Holding Reg	R/W	SigB_2 – Negation
1075	Holding Reg	R/W	SigB_2 – Time
1076	Holding Reg	R/W	SigB_2 – 0Time
1077	Holding Reg	R/W	SigB_2 – 1Time
1078	Holding Reg	R/W	SigB_3 – Source
1079	Holding Reg	R/W	SigB_3 – Mode
1080	Holding Reg	R/W	SigB_3 – Negation
1081	Holding Reg	R/W	SigB_3 – Time
1082	Holding Reg	R/W	SigB_3 – 0Time
1083	Holding Reg	R/W	SigB_3 – 1Time

8.2.6 REGISTER ADDRESSES FOR READING/SAVING BLOCK SIG_C CONFIGURATIONS

Address	Type	R/W	Description
1090	Holding Reg	R/W	SigC_0 – Source
1091	Holding Reg	R/W	SigC_0 – Time
1092	Holding Reg	R/W	SigC_1 – Source
1093	Holding Reg	R/W	SigC_1 – Time
1094	Holding Reg	R/W	SigC_2 – Source
1095	Holding Reg	R/W	SigC_2 – Time
1096	Holding Reg	R/W	SigC_3 – Source
1097	Holding Reg	R/W	SigC_3 – Time

8.2.7 REGISTER ADDRESSES FOR READING/SAVING LED CONFIGURATIONS

Address	Type	R/W	Description
1100	Holding Reg	R/W	C0 – color code 0 (See Table 4.1)
1101	Holding Reg	R/W	C1 – color code 1 (See Table 4.1)
1102	Holding Reg	R/W	C2 – color code 2 (See Table 4.1)
1103	Holding Reg	R/W	C3 – color code 3 (See Table 4.1)

8.2.8 REGISTER ADDRESSES FOR READING/SAVING I/O CONFIGURATIONS

Address	Type	R/W	Description
1104	Holding Reg	R/W	Source – PinOut
1105	Holding Reg	R/W	Source – Colour0
1106	Holding Reg	R/W	Source – Colour1
1107	Holding Reg	R/W	Source – Colour2
1108	Holding Reg	R/W	Source – Colour3
1109	Holding Reg	R/W	Source – Buzzer
1110	Holding Reg	R/W	Source – PinIn0
1111	Holding Reg	R/W	Source – PinIn1

8.2.9 REGISTER ADDRESSES FOR SAVING VALUES OF RSX SOURCES

Address	Type	R/W	Description
1200	Holding Reg	W	RS0
1201	Holding Reg	W	RS1
1202	Holding Reg	W	RS2
1203	Holding Reg	W	RS3

8.2.10 REGISTER ADDRESSES FOR READING SOURCE VALUES

Address	Type	R/W	Description
100	Coil / Discrete Inputs	R	„0”
101	Coil / Discrete Inputs	R	„1”
102	Coil / Discrete Inputs	R	Button
103	Coil / Discrete Inputs	R	AnyCard
104	Coil / Discrete Inputs	R	RS_0
105	Coil / Discrete Inputs	R	RS_1
106	Coil / Discrete Inputs	R	RS_2
107	Coil / Discrete Inputs	R	RS_3
108	Coil / Discrete Inputs	R	PinIn0
109	Coil / Discrete Inputs	R	PinIn1
110	Coil / Discrete Inputs	R	SigA0
111	Coil / Discrete Inputs	R	SigA1
112	Coil / Discrete Inputs	R	SigA2
113	Coil / Discrete Inputs	R	SigA3
114	Coil / Discrete Inputs	R	SigB0
115	Coil / Discrete Inputs	R	SigB1
116	Coil / Discrete Inputs	R	SigB2

117	Coil / Discrete Inputs	R	SigB3
118	Coil / Discrete Inputs	R	SigC0
119	Coil / Discrete Inputs	R	SigC1
120	Coil / Discrete Inputs	R	SigC2
121	Coil / Discrete Inputs	R	SigC3

8.3 ENCAPSULATION NETRONIX PROTOCOL INSIDE MODBUS RTU

Any command from the Netronix protocol can be executed using the appropriate registers from the MODBUS protocol.

Addr.	Type	R/W	Name	Description
2008	Holding Reg	R/W	Trigger/Status	This register is used to trigger command processing and to check the processing status. Allowed values: 0x0000 – Module in IDLE mode 0x0001 – Triggering processing 0x00EE – Error 0x00FF – Command completed. The answer is in the work registers.
2009	Holding Reg	R/W	Len	Rejestr ten zawiera długość zapisanej komendy / długość odpowiedzi (ilość rejestrów zapisanych/do odczytania)
2010-2073	Holding Reg	R/W	Working registers	Rejestry te służą do zapisania komendy / odczytania odpowiedzi. Jeden rejestr przechowuje wartość 1 bajta komendy/odpowiedzi z ramki Netronix

8.3.1 WORKFLOW

Write to the MODBUS registers the command from the Netronix protocol according to the following scheme:

Netronix Protocol Command Frame

Adr	Len	Cmd	P[0]	P[1]	CRC	CRC
--	07	A0	11	22	--	--

-4

MODBUS RTU Register

	T/S	Len	Working registers						
Addr.:	2008	2009	2010	2011	2012	2013	2014	2015	2016
Value:	00 00	00 03	00 A0	00 11	00 22	-- --	-- --	-- --	-- --

Write the value 0x0001 to the Trigger/Status registry

MODBUS RTU Register

	T/S	Len	Working registers						
Addr.:	2008	2009	2010	2011	2012	2013	2014	2015	2016
Value:	00 01	00 03	00 A0	00 11	00 22	-- --	-- --	-- --	-- --

Read the Trigger/Status register until the value 0x00FF appears in it. The value 0x00FF means that the answer is ready and can be read from the MODBUS registers.

MODBUS RTU Register

	T/S	Len	Working registers						
Addr.:	2008	2009	2010	2011	2012	2013	2014	2015	2016
Value:	00 FF	00 05	00 A1	00 AA	00 BB	00 CC	00 FF	--	--

+4

Netronix Protocol Response Frame

Adr	Len	Cmd	P[0]	P[1]	P[3]	OC	CRC	CRC
--	09	A1	AA	BB	CC	FF	-- --	-- --

8.3.2 EXAMPLE OF USE - READING THE FIRMWARE VERSION

Assumptions:
Reader logical address (Netronix / Modbus RTU protocol) - 0x01.

1. Specify what the frame should look like in the Netronix protocol. Only the Cmd field and parameters are relevant. The entire frame for the C_FirmwareVerison command looks like this:

Adr	Len	Cmd	CRC	
0x01	0x05	0xFE	0xC6	0x14

   Cmd = 0xFE
   Param – none

2. The amount of data should be entered in the ***Len*** register and the command code (and optional parameters) should be entered in the ***Working Registers***:
   RTU Tx > 01 10 07D8 0002 04 0001 00FE 0925
   RTU Rx > 01 10 07D8 0002 C087

   Len = 0001
   WorkingRegister[0] = 00FE

3. Write the value 0x0001 to the ***Trigger/Status*** register. This will trigger the execution of the command stored in the Working Registers.
   RTU Tx > 01 06 07D7 0001 F946
   RTU Rx > 01 06 07D7 0001 F946

   Trigger/Status=0001

4. Then read the ***Trigger/Status*** register until the value 0x00FF is read. The value 0x00FF means that the command has been carried out and the ***Working Registers*** have the answer.
   RTU Tx > 01 03 07D7 0001 3546
   RTU Rx > 01 03 02 00FF F804

   Trigger/Status=00FF

5. Then read the ***Len*** register. This register contains information about the number of registers in which the response is stored.
   RTU Tx > 01 03 07D8 0001 0545
   RTU Rx > 01 03 02 0011 7848

   Len=0011

6. In the last step, read the Len first working registers.

```
RTU Tx > 01 03 07D9 0011 5549
RTU Rx > 01 03 22
         00FF
         004D 0057 002D 0052 0037 002D 0056 0033
         002E 0032 002E 0041 0031 002E 0035
         00FF
         8E C6
```

00FF – Command code +1 (response to C_FirmwareVersion)
004D 0057 002D … 0031 002E 0035 – firmware version – „MW-R7-v3.2.A.1.5”
00FF – Operation code (success)

9 RETURN TO FACTORY SETTINGS

To return to factory settings, within 3 to 10 seconds after starting a device, press the the front button for approx. 3 seconds. When returning to factory settings, the following reader parameters are permanently set:

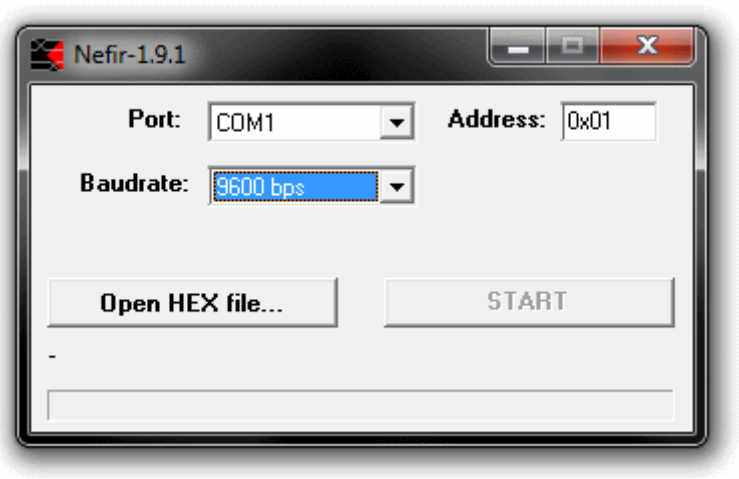
Table 8.1 Factory settings

Parameter name or functionality	Value or setting	
Interface		
RS232 interface	Address: 0x01 Speed: 0x03	9600bps
RS485 interface	Address: 0x01 Speed: 0x03	9600bps
1-WIRE interface	Family: 0x01 Address: 0x00	
Wiegand interface	Byte quantity: 37	
Transponder readout		
Transponder type	0x01	UNIQUE
Autoreader	Trigger: 0x02	
	Timeout: 0x14	2s
	Mode: 0xFF	All supported types
	Aserial: 0x01	on first application
	Amode: 0x40	Netronix format, extended information about collision signals and card type
	Abuzzer: 0x01	On first application
	AInterface: 0x00	RS232
Inputs/Outputs		
PinIN0 input	Trigger: low state	
PinIN1 input	Trigger: low state	
PinOUT output	Source control: Button	
Kolor0 output	Source control: PinIN1	
Kolor1 output	Source control: Button	
Kolor2 output	Source control: '0'	
Kolor3 output	Source control: '1'	
Buzzer output	Source control: PinIN0	
Colour setting		
LED configuration	C0: GREEN C1: BLUE C2: WHITE C3: RED	
SIGNAL blocks		
SigA0	In0: “0”; In1: “0”; In2: “0”; Function: OR	
SigA1	In0: “0”; In1: “0”; In2: “0”; Function: OR	
SigA2	In0: “0”; In1: “0”; In2: “0”; Function: OR	
SigA3	In0: “0”; In1: “0”; In2: “0”; Function: OR	
SigB0	Source: “0”, Mode: 2, Negation: 1 Time:0, Time0: 0, Time1: 0	
SigB1	Source: “0”, Mode: 2, Negation: 1 Time:0, Time0: 0, Time1: 0	
SigB2	Source: “0”, Mode: 2, Negation: 1 Time:0, Time0: 0, Time1: 0	
SigB3	Source: “0”, Mode: 2, Negation: 1 Time:0, Time0: 0, Time1: 0	
Password		
Password	“”, 0x3C	No password, 60s

10 BOOTLOADER – UPDATE DEVICE FIRMWARE

In order to upload new firmware to the device, follow the procedure:

1. Connect the device to the RS232 serial port on your computer
2. Run the program *NEFIR.exe*
3. Set the correct COM port and baud rate to 9600bps
4. Press *Open HEX File* button and load the file with the new firmware
5. Press the START button, which triggers the reloading firmware
6. Wait for the end of the process



Drawing 9.1 Program window while reloading firmware