Technical Data Sheet

RFID reader

PAC-DUG
PAC-DUB

PAC-DUG

Documentation version:	PAC-DU-MAN-V4
Date:	11.09.2018
Valid from firmware version:	PAC-DU-v8.8.A1.1
Author:	Patryk Burczyński

Introduction

PAC-DUx device series is OEM miniature RFID card reader operating at frequency of 125 kHz.

Main features:

- Support of Unique, Q5, Hitag-1, Hitag-S or HID transponders,
- Built-in buzzer
- Built-in push-button for reset to default settings
- Built-in two LED's of common purpose and diode as a supply indicator
- changeable format of sending ID
- Data password protected
- Powered from USB
- Two colors of case:
  - PAC-DUB (black)
  - PAC-DUG (grey)

1 General Specifications

Supported functionality depending on transponder / card type:		
Transponder type	ID number read-out	Full write and read-out of memory blocks
Unique	YES	-
Q5	YES	YES
HID	YES	-
HITAG	YES	YES

PAC-DUx module parameters	
Supply voltage	5 V(USB)
Max. supply current	120 mA
Rated operation radio frequency of module	125 KHz
Read-out distance between transponders	Up to 10 cm
Dimension	92x146x29
USB communication	CDC Class, serial port emulation, compliant with „Netronix Protocol" Keyboard emulation
Temperature	0-60st.c

2 Serial transmission format

After drivers installation (www.netronix.pl) , PAC-DUx reader is seen by PC port as a virtual serial port.

In this data sheet USB protocol has been confined to descriptions of commands, responses and their parameters. Header and CRC control sum exist always and are compliant with full “Netronix Protocol” document.

Command frame:

Header	C_CommandName	Response_parameters1…n	CRC

Response frame:

Header	C_CommandName+1	Response_parametrers…m	OperationCode	CRC

Serial protocol operation can be tested by means of development tools including free of charge “FRAMER” software”.

2.1 Commands for communication with transponders

2.1.1 Selecting the type of transponder

Command frame:

C_SetTransponderType	TransponderType, GAIN

Where:

Parameter name	Parameter description	Value range
C_SetTransponderType	Command of transponder type changing	0x02
TransponderType	Transponder type we want exchange data with	0x01 – Unique 0x02 – Q5 0x03 – HITAG 0x04 – HID
GAIN	Gain of RFID receive circuit (recommended values 0x1 or 0x2)	0x0-0x3

Response frame:

C_SetTransponderType +1		Operation Code

2.1.2 Reading the type of transponder

Command frame:

C_GetTransponderType	

Where:

Parameter name	Parameter description	Value range
C_GetTransponderType		0x06

Response frame:

C_GetTransponderType +1	TransponderType, GAIN	Operation Code

Where:

Parameter name	Parameter description	Value range
C_GetTransponderType+1		0x07
TransponderType	Transponder type	0x01 – Unique

		0x02 – Q5 0x03 – HITAG 0x04 – HID
GAIN	Gain of RFID receive circuit	0x0-0x3

2.1.3 On/off switching of reader field

Command frame:

Header	C_TurnOnAntennaPower	State	CRC

Where:

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

Response frame:

Header	C_TurnOnAntennaPower +1		Operation Code	CRC

2.1.4 Reading the ID card unique number

Command frame:

C_Select	

Where:

Parameter name	Parameter description	Value range
C_Select	Odczyt ID	0x12

Response frame:

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

Where:

Parameter name	Parameter description	Meaning
Coll	Information on collision (HITAG transponders only)	0 – no collision 1 – collision of two or more transponders
TType	Information on transponder type, to whom the red ID number concerns	1 - Unique,Q5 3 - HITAG 4 - HID
ID1…IDn	Unique number of transponder	ID1 – LSB, IDn – MSB

2.2 Commands for communication with Q5 transponders

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

2.2.1 Writing the ID-Unique number to Q5 transponder

Command frame:

C_UniqueWrite	Unique1..5, lock

Where:

Parameter name	Parameter description	Value range
C_UniqueWrite	Command of id-unique write	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 write. 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.

2.2.2 Reading the sector of Q5 transponder

Command frame:

C_ReadBlock	SectorNo,[Password1..4]

Where:

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

Response frame:

C_ReadBlock +1		Operation Code

2.2.3 Writing the sector of Q5 transponder

Command frame:

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

Where:

Parameter name	Parameter description	Value range
C_WriteBlock	Sector write command	0x1C
SectorNo	Write sector 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_WriteBlock +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.

2.3 Commands for communication with HITAG transponders

2.3.1 Reading the page of HITAG transponder

Command frame:

C_ReadBlock	PageNo

Where:

Parameter name	Parameter description	Value range
C_ReadBlock	Page read command	0x1E
PageNo	Read page number	0x00-0x3f

Response frame:

C_ReadBlock +1		Operation Code

2.3.2 Writing the page to HITAG transponder

Command frame:

C_WriteBlock	PageNo, Data1...4

Where:

Parameter name	Parameter description	Value range
C_WriteBlock	Sector read command	0x1C
PageNo	Read page number	0x00-0x3f
Data1..4	4 bytes of data which is being red	0x00-0xff

Response frame:

C_WriteBlock +1		Operation Code

2.4 Reader inputs and outputs

Reader has a three outputs which are configurable.

2.4.1 Writing the output state

Command frame:

C_WriteOutputs	IONo, State

Where:

Parameter name	Parameter description	Value range
C_WriteOutputs	Output state write	0x70
IONo	I/O port number. The port should be configured as an output	0x1..0x7 for UW-U4R 0x1..0xC for UW-U4G
State	Requested output state	0x00 or 0x01

Response frame:

C_WriteOutputs +1		Operation Code

2.4.2 Reading the input state

Command frame:

C_ReadInputs	IONo

Where:

Parameter name	Parameter description	Value range
C_ReadInputs	Input state reed-out	0x72
IONo	I/O port number. Should be configured as an input.	0x0..0x7 dla UW-U4R 0x0..0xC dla UW-U4G

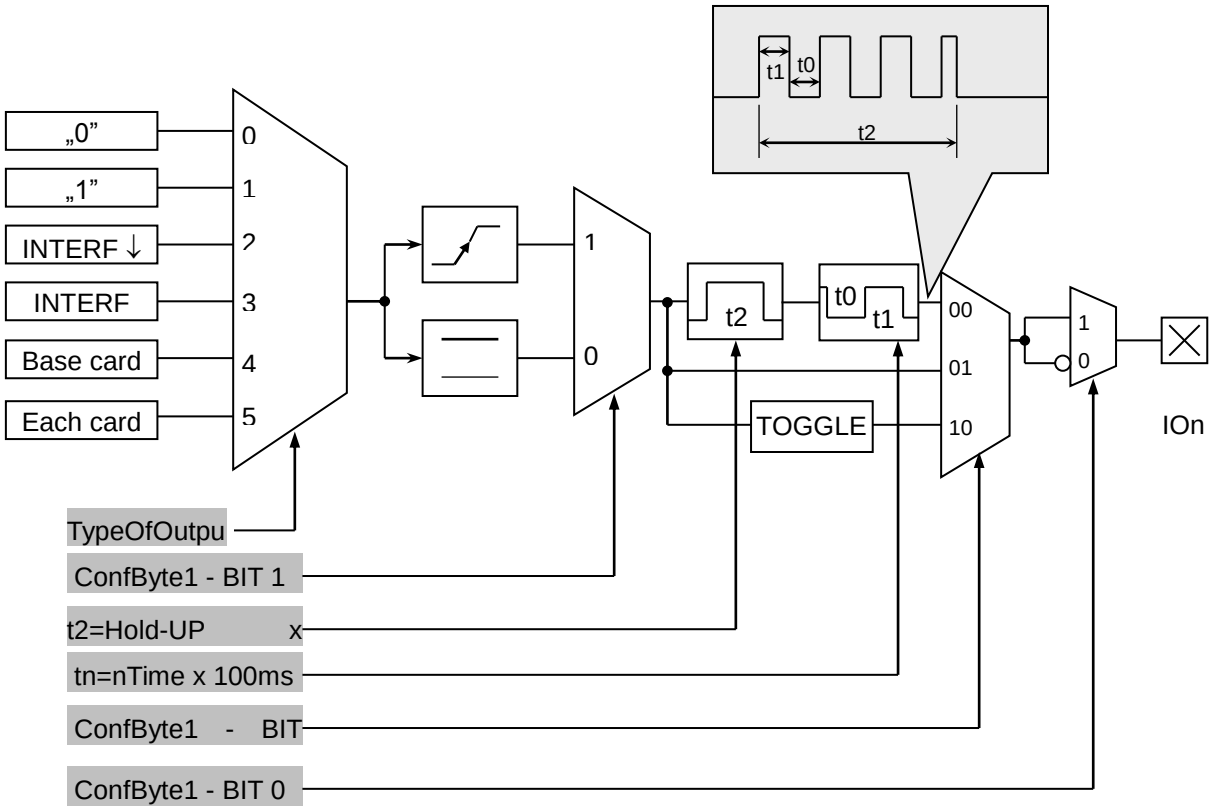
Response frame:

C_ReadInputs +1	State,[COUNTER]	Operation Code

Where:

Parameter name	Parameter description	Value range
State	Input state which has been red	
Counter	Counter state for counter type input.	

2.4.3 Writing the settings to any port



Command frame:

Header	C_SetIOConfig	IONo, IOConfigData1…n	CRC

If we set a port as output, IOConfigData1…n parameters are as below:

Dir, ConfByte1, TypeOfOutput, Hold-up, 0Time, 1Time

Where:

Parameter name	Parameter description	Value range
C_SetIOConfig	Writing the configuration of every port	0x50
IONo	I/O port number, which is to be configured	0x0..0x4
Dir	Port direction	0x00 – output
ConfByte1	One byte in which: BIT0 assigns output type as normally open or normally closed. BIT 1 determines reaction method of each output as sensitive for simulation changing (slope sensitive) or as sensitive for simulation state (state sensitive). BIT3:2 determines operation method of output referring to trigger signal state.	ConfByte1 Bit 0 0-Normally closed 1-Normally open ConfByte1 Bit 1 0-level sensitive 1-slope sensitive ConfByte1 Bit 3:2 00 – rectangular wave generator 01- directly 10 – output state change
TypeOfOutput	Source of driving signal	0x00 – permanently off 0x01 – permanently on 0x02 – driven via serial interface 0x03 – driven via serial with automatic reset(edge emulation) 0x04 – driven by internal access control mechanism ACM. This output is driven in case of applying the card to reader, which is written into internal card base. 0x05 – set in case of applying freely selected card to reader.
Hold-up	Time of maintaining the on state after actuation stopped. This time is specified as: Hold-up x 10 ms During “hold-up” time, it is possible to configure the output, whish is able to generate rectangular wave. By means of following parameters are configured “Logic 1” time and “Logic 0” time:	
0Time	Logic 0 time	
1Time	Logic 1 time	

If we set a port as a input, IOConfigData1…n parameters would be as below:
Dir, Triger, TypeOfInput, Delay,

Where:

Parameter name	Parameter description	Value range
C_SetIOConfig	Writing the configuration of freely selected port.	0x50
IONo	I/O port number, which is to be configured.	0x00,0x01,0x07
Dir	Port direction	0x01 – input
TypeOfInput	Input type	0x03
Delay	Delay	0x00

2.4.4 Reading-out the configuration of freely selected port

Command frame:

Header	C_GetIOConfig	IONo	CRC

Where:

Parameter name	Parameter description	Value range
C_GetIOConfig	Reading-out the configuration of freely selected port.	0x52
IONo	I/O port number, which configuration is to be red-out.	0x00…0x05

Response frame:

Header	C_GetIOConfig +1	IOConfigData1…n	OperationCode	CRC

Where:

Parameter name	Parameter description	Value range
IOConfigData1…n	This is the same, as in case of configuration write.	

Some I/O of CTU-D reader has no possibility to toggle port direction.
To accomplish proper configuration, input proper direction option to given port.

LIST OF EXISTING PORTS, WHICH CAN BE DRIVEN IN UW-M4R		
Port number	Direction	Description
0	input/output	GPIO1
1	input/output	GPIO 2
2	output	RELAY
3	output	BUZZER

Response frame:

Header	C_SetIOConfig +1		OperationCode	CRC

2.5 Access password

2.5.1 Logging to reader

Command frame:

Header	C_LoginUser	Data1…n, 0x0	CRC

Where:

Parameter name	Parameter description	Value range
C_LoginUser	Logging to reader	0xb2
Data1…n	This is any byte string	Any from range: 0x01…0xff. String length, which can be 0 to 8 bytes
0x00	Logic Zero, which terminates a string.	0x00

Response frame:

Header	C_LoginUser+1		OperationCode	CRC

2.5.2 Changing the password

Command frame:

Header	C_ChangeLoginUser	Data1…n, 0x0	CRC

Where:

Parameter name	Parameter description	Value range
C_ChangeLoginUser	Password change	0xb4
Data1…n	This is any byte string, which will form valid access password.	Any from range: 0x01…0xff. String length, which can be 0 to 8 bytes
0x00	Logic Zero, which terminates a string.	0x00

If =0x00, a reader will not be protected by password. At any moment, there is possible to set new password later on, to protect the reader by it.

Response frame:

Header	C_ChangeLoginUser+1		OperationCode	CRC

2.5.3 Logging out of the reader

This command sets latest password as an invalid.

Command frame:

Header	C_LogoutUser		CRC

Parameter name	Parameter description	Value range
C_LogoutUser	Logging out of the reader.	0xd6

Response frame:

Header	C_LogoutUser +1		OperationCode	CRC

2.6 ‘Autoreader’ configuration

2.6.1 Writing the “automatic read” configuration

This command sets operation method of automatic device, reading the unique transponder number UID.

The reader described below makes possible to hold-on operation of automatic device for a while, in case of suitable transmission via serial interface.

If the reader will operate in mixed mode i.e.:
- automatic reading device UID is enabled and:
- master device (computer, controller) communicates with reader or with transponders via reader,
it is required, to configure the reader correctly, so in case of communication with a reader or transponder, automatic reading device would hold-on its operation.

Command frame:

C_SetAutoReaderConfig	ATrig, AMode, AOfflineTime, ASerial, Abuzz,Amulti

Where:

Parameter name	Parameter description	Value range	
C_SetAutoReader Config	Writing the automatic device configuration.	0x58	
ATrig	Defines, when automatic reading device UID will operate.	0-automatic device disabled permanently 1-automatic device enabled permanently 2=enabled automatically in case of transmission lack on interface for a time longer than AOfflineTime 3=enabled automatically, in case of no recall of communication commands with transponder for a time longer than AOfflineTime	
AOfflineTime	Lack of transmission time on interface bus T= AofflineTime * [100ms] Lack of transmission can concern to any commands (Atrig=2), or commands for communication with transponder (Atrig=3). Commands for communication with transponder: C_TurnOnAntennaPower C_Select	0x00…0xff	
ASerial	Automatic sending the UID transponder number, after reading it automatically from transponder.	0-never 1-for the first applying the transponder only 2-sends all	
AMode	Selection the format of sending number 8 bits: LS MSB LSB D CR B C M E I A	C=1,CR=0 C=1,CR=1 C=0	End line sign CR+LF End line sign CR No end line sign
		M=1	Start with ”M” sign
		I=1	ID in invert order
		E=1	Extended data (collision and transponder type)
		A=0,B=0 A=1,B=0 A=0,B=1	ID in Nertonix frame ID in ASCII mode ID in binary mode
		D=1	ID converted to decimal mode, only in ASCII mode
ABuzz	Automatic indication of reading by means of buzzer, after automatic UID read-out from transponder.	0-never 1-for the first applying the transponder only 2-indicates all	
AMulti	Multi type of transponders read mode	0 – read a only selected by CSetTransponderType command	

		trnasponder type 0xff – read all known transponder types

Response frame:

Header	C_ SetAutoReaderConfig +1		OperationCode	CRC

2.6.2 Reading-out the configuration of automatic device

Command frame:

Header	C_ GetAutoReaderConfig		CRC

Where:

Parameter name	Parameter description	Value range
C_GetAutoReaderConfig	Read-out of automatic device configuration.	0x5a

Response frame:

Header	C_ GetAutoReader Config +1	ATrig, AOfflineTime, ASerial, Abuzz,Amulti	OperationCode	CRC

Where:
The meaning of response parameters is the same as described before.

2.7 Other commands

2.7.1 Change buzzer volume

Use this command to set buzzer volume and store setting in EEPROM memory.

Command frame:

Header	C_SetBuzzerConfig	Volume	CRC

Where:

Parameter name	Parameter description	Value range
C_SetBuzzerConfig		0xD8
Volume	Buzzer volume value	0x00-0x0F

Response frame:

Header	C_SetBuzzerConfig+1		OperationCode	CRC

2.7.2 Read buzzer volume

Use this command to read buzzer volume.

Command frame:

Header	C_GetBuzzerConfig	Volume	CRC

Where:

Parameter name	Parameter description	Value range
C_GetBuzzerConfig		0xDC

Response frame:

Header	C_GetBuzzerConfig+1	Volume	OperationCode	CRC

Where:

Parameter name	Parameter description	Value range
C_GetBuzzerConfig+1		0xDC
Volume	Buzzer volume value	0x00-0x0F

2.7.3 Remote reset of reader

Command frame:

Header	C_Reset		CRC

Where:

Parameter name	Parameter description	Value range
C_Reset	Remote reader reset	0xd0

Response frame:

Header	C_Reset +1		OperationCode	CRC

2.7.4 Setting the date and time

Following setting has no influence for reader operation today.

Command frame:

Header	C_SetRtc	Year, Month, Day, Hour, Minute, Second	CRC

Where:

Parameter name	Parameter description	Value range
C_SetRtc	Date and time set-up	0xb8
Year	year	0…99
Month	month	1…12
Day	day	1…31
Hour	hour	0…23
Minute	minute	0…59
Second	second	0…59

Response frame:

Header	C_SetRtc +1		OperationCode	CRC

2.7.5 Reading-out the date and time

Command frame:

Header	C_GetRtc		CRC

Where:

Parameter name	Parameter description	Value range
C_GetRtc	Read-out of date and time	0xb6

Response frame:

Header	C_GetRtc+1	Year, Month, Day, Hour, Minute, Second	OperationCode	CRC

Where:
The meaning of response parameters is the same as described before.

2.7.6 Keyboard

Command Frome:

Header	C_Keyboard	[Param]	CRC

Where:

Parameter name	Parameter description	Value range
C_Keyboard	Enable / disable keyboard	0x04
[Param]	One byte characters: MSB … LSB 0 0 0 F0 F1 INV ENTER EN	**F0=1 i F1=1** – Frame format 1WIRE. ID justified to left **F0=1 i F1=0** – Frame format 1WIRE. ID justified to right **F0=0 i F1=1** – hexadecimal format **F0=0 i F1=0** – decimal format **INV=1** – inverted order **ENTER = 1** – simulation of pressing the ENTER key after entering the ID **EN = 1** – keyboard enabled

Response frame:

Header	C_Keyboard +1	Param	KodOperacji	CRC

Where:

Parameter name	Parameter description	Value range
C_Keyboard+1	Enable / disable keyboard	0x05
Param	The meaning of response parameters is the same as described before.	

2.7.7 Reading-out the reader software

Command frame:

Header	C_FirmwareVersion		CRC

Where:

Parameter name	Parameter description	Value range

C_FirmwareVersion	Read-out of reader software version	0xfe

Response frame:

Header	C_FirmwareVersion+1	Data1…..n	OperationCode	CRC

Where:
Data1…n is sequence of dots, which are written as an ASCII codes.

2.8 Code meanings in response frames

Operation code name	Description	Value
OC_Error	Error	0x00
OC_ParityError	Parity error	0x01
OC_RangeError	Parameter range error	0x02
OC_LengthError	Data quantity error	0x03
OC_ParameterError	Parameter error	0x04
OC_Busy	Internal modules are busy at the moment.	0x05
OC_NoACKFromSlave	No internal communication	0x22
OC_CommandUnknown	Unknown command	0x07
OC_WrongPassword	Wrong password or last password terminated i.e. automatic LogOut occurred.	0x09
OC_NoCard	No transponder	0x0a
OC_BadFormat	Wrong data format.	0x18
OC_FrameError	Transmission error. Noise occurrence possible.	0x19
OC_NoAnswer	No response from transponder.	0x1E
OC_TimeOut	Operation time out. No transponder in reader field possible.	0x16
OC_Successful	Operation completed successfully.	0xff

3 Keyboard emulation

The device PAC-DUX can emulate USB keyboard (HID). During emulate the keyboard, each reading of the ID transponder through the mechanism AutoReader'a will simulate entering his ID. An example of the format ID sent depending on configuration is shown below:

ID:				0x1C34AB1F55
Configuration				Format
H/D	INV	ENTER	Caps Lock	
0	0	0	X	121142714197
0	0	1	X	121142714197 <ENTER>
0	1	0	X	369126749212
0	1	1	X	369126749212 <ENTER>
1	0	0	Off	1c34ab1f55
1	0	0	On	1C34AB1F55
1	0	1	Off	1c34ab1f55 <ENTER>
1	0	1	On	1C34AB1F55 <ENTER>

1	1	0	Off	551fab341c
1	1	0	On	551FAB341C
1	1	1	Off	551fab341c <ENTER>
1	1	1	On	551FAB341C <ENTER>

4 Reset to default settings

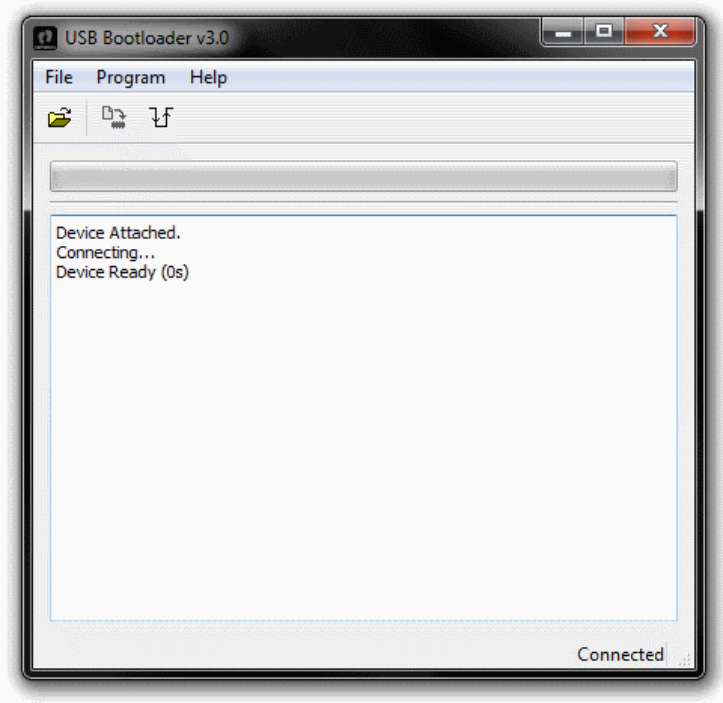
To restore default settings, connect reset terminal with ground for 2 s or longer. During restoring the defaults following reader parameters are fixed:

Access password	0x00 - no password
Port 0 – LED1	Card in field indicator
Port 1 -LED2	Read card indicator
Port 2 - BUZZER	Read card indicator
“Autoreader” configuration	0x02,0x14,0x01,0x04,0x01,0xff
Transponder type	Unique
Buzzer Volume	8

5 Bootloader - update device firmware

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

1. Disconnect the device from the USB port
2. While holding down the reset button connect the device to the USB port. The device should be detected as a HID device, and the LEDs on the unit should blink alternately.
3. Open the NX_HID_Bootloader.exe
4. Press Import Firmware Image (Ctrl + O) and then select a file with the firmware
5. Press Erase / Program / Verifi Device
6. Press the Reset DeviceX



Drawing 5.1 Program window

Latest news concerning to NETRONIX products
http://www.netronix.pl/

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Netronix ***manufacturer:***

Other Similar products are found below :

ACTS-2 AGEDI-B ANTENNA L112 50 MM COTER-E4I COTER-ECI CTU-D2R CTU-D2RM CTU-D4R CTU-D5N CTU-D5R CTU-M5N CTU-M5RM CTU-R5RM I2M-005 MM-R5 MW-D7G MW-R7B MW-R7G NANO-MS PAC-AUB PAC-DUB PAC-DUG PAC-PUB PLA-D6L PLA-MDK PLA-MDL PLA-R6L UW-D4G UW-DAL UW-R4G