User Manual E.010

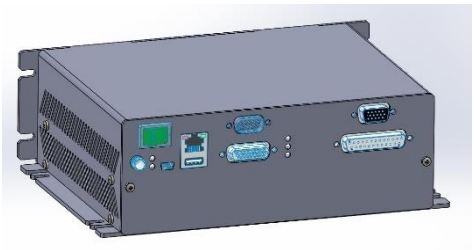
Digital Controllers Series EBC/EBD-120

Release: 4.0 Date: 2024-07-23

This document is valid for the following products:

- **EBC/EBD-120310**
  Digital Controller, 1 Channel
- **EBC/EBD-120320**
  Digital Controller, 2 Channels
- **EBC/EBD-120325**
  Digital Controller, 2.5 Channels
- **EBC/EBD-120330**
  Digital Controller, 3 Channels
- **EBC/EBD-120335**
  Digital Controller, 3.5 Channels

EBC/EBD-120310



EBC/EBD-120320 (2 Status-LEDs) and
EBC/EBD-120330 (3 Status-LEDs)

This manual is for information only and subject to change without notice. Images may not be up to date.

©nanoFaktur GmbH

Peterzeller Str. 8c
78048 Villingen-Schwenningen, Germany
Tel: +49 7721 9464700
Email: info@nanoFaktur.com

Contents

1. Declaration of Conformity

according to ISO / IEC Guide 22 and EN 45014

Manufacturer:	nanoFaktur GmbH	CE
Address of Manufacturer:	Peterzeller Str. 8c 78048 Villingen-Schwenningen Germany	

The manufacturer declares that the products

Short Description:	Digital Controller, Voltage-Amplifier
Model Numbers:	EBC/EBD-120310, EBC/EBD-120320, EBC/EBD-120325, EBC/EBD-120330, EBC/EBD-120335
Product Options:	Standard

is consistent with the following standards and directives:

2014/35/EU, Low Voltage Directive (LVD)
2014/30/EU, EMC Directive
2011/65/EU/ RoHS Directive

Safety (Low Voltage Directive):	IEC 61010-1:2010, IEC 61010-1:2010/AMD1:2016
EMC:	EN 61326-1:2013
RoHS:	EN IEC 63000:2018

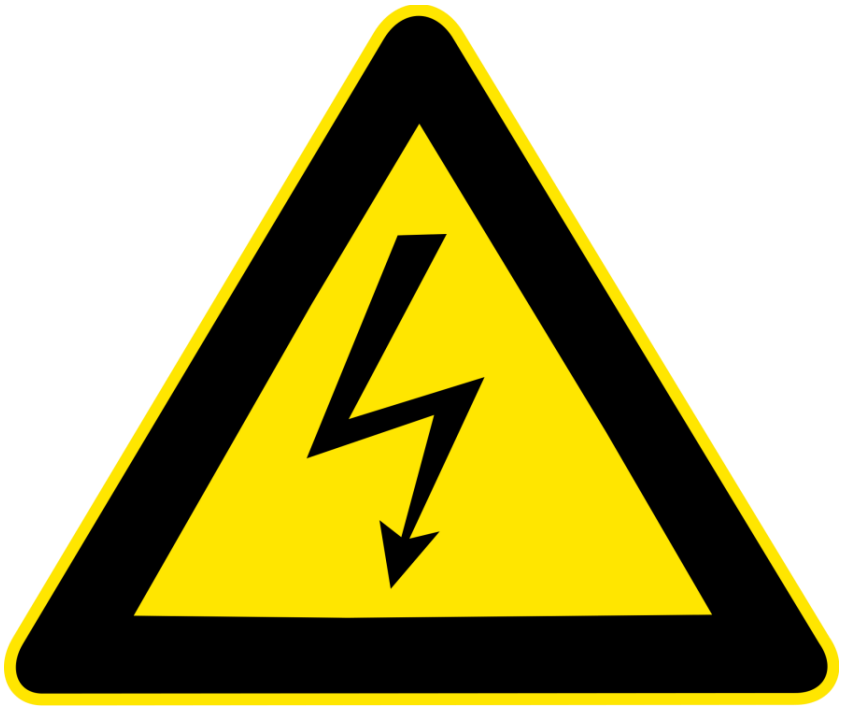
Electrical equipment which is intended to be integrated in other electrical equipment or enclosures, only conforms to the cited EMC Standards and normative documents, if the user ensures compliant enclosures and connections when integrating. Appropriate measures: installation of the components in qualified shielded enclosures and the usage of qualified connectors.

Villingen-Schwenningen, Germany, March, 2024

Dipl.-Ing. (FH) Klaus Pollak, Director

2. Caution! High Voltage!

All Chapter 2 must be read!



The devices described in this manual are high-voltage amplifiers. Improper handling may harm the health of the operator or even be lethal.

Operating personnel must have basic understanding of voltage-amplifying electronics and piezo-electricity. Persons handling open (OEM, bare boards) electronics must be entitled to handle high-voltage electronics according to the valid rules. Safety cannot be guaranteed if the devices are operated incorrectly.

Warranty and liability of nanoFaktur are void if housed devices are opened or in case of any other hardware-manipulations. Use the determined interfaces for connections only. Intentional or default improper operation void all warranty of liability.

2.1. Prescribed Usage

EBC-120 and EBD-120 controllers are intended to drive capacitive loads in particular piezo-electric actuators. Additionally, EBC drives and evaluates capacitive sensors, EBD does the same for strain-gage sensors (SGS). Actuators and sensors different from the latter described must not be connected. Other actuators and sensors than those of nanoFaktur may only be connected if particularly allowed by nanoFaktur. Observe the precautions written in this manual.

EBC-120 and EBD-120 controllers conform to Measurement Category I (CAT I) and may not be used for other CATs. Other use of the devices than instructed in this manual may affect safeguards provided.

2.2. Ambient Restrictions

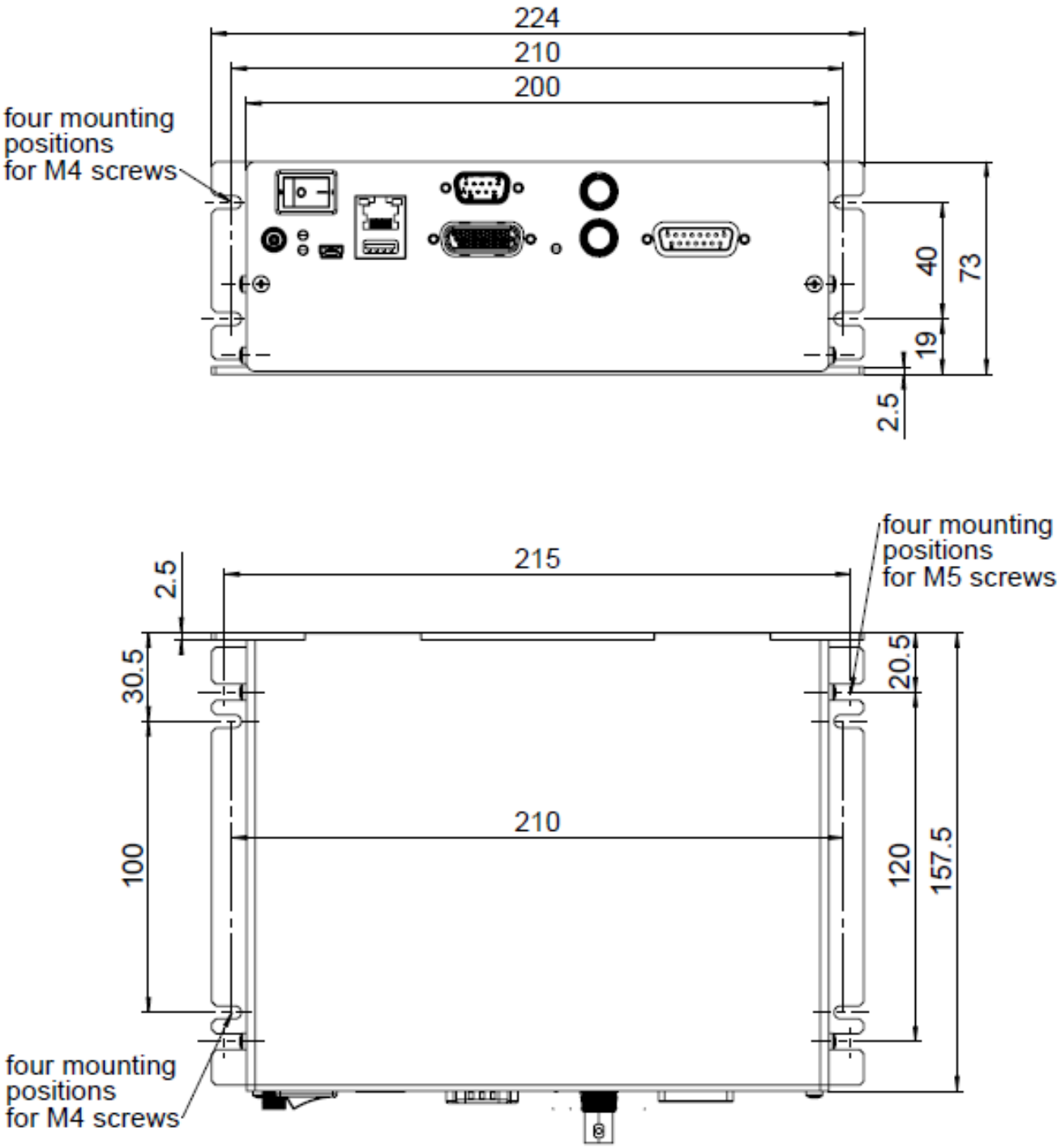
- Indoor use only
- Altitude less than 2000 m
- Temperature from 5 to 40 °C
- Relative humidity (RH) up to 80% for temperatures up to 31 °C, decreasing linearly to 50% RH at 40 °C
- Transient over-voltages as typical for public power supply.
  Note: The nominal level of the transient overvoltage is the standing surge voltage according to the overvoltage category II (IEC 60364-4-443).
- Degree of pollution: 2

3. EBC/EBD-1203nn Digital Controllers

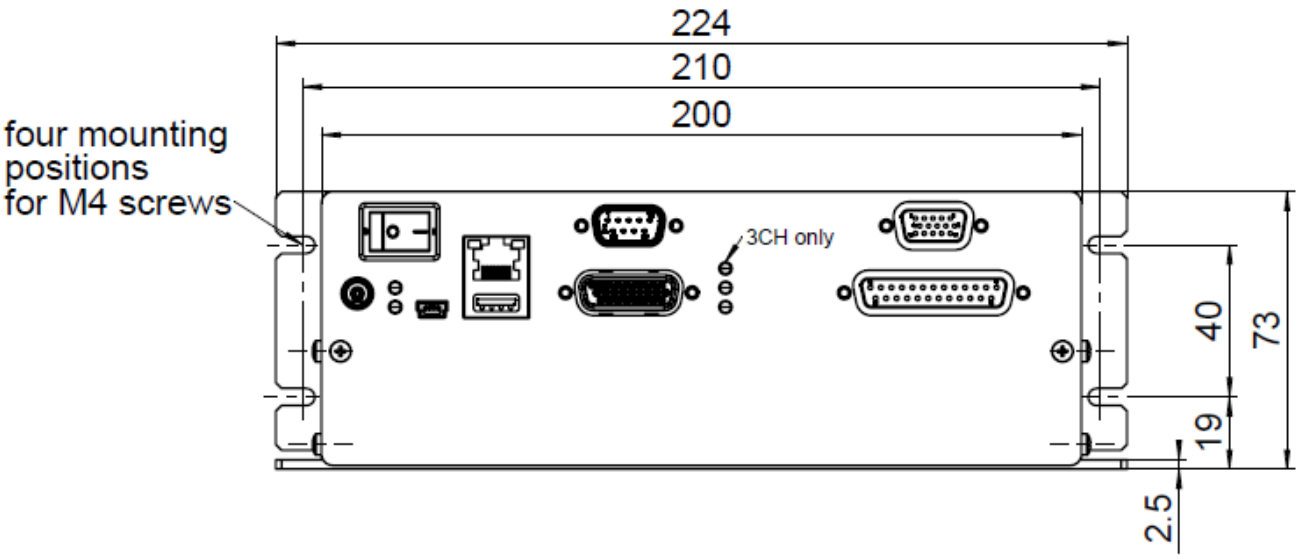
3.1. Specification

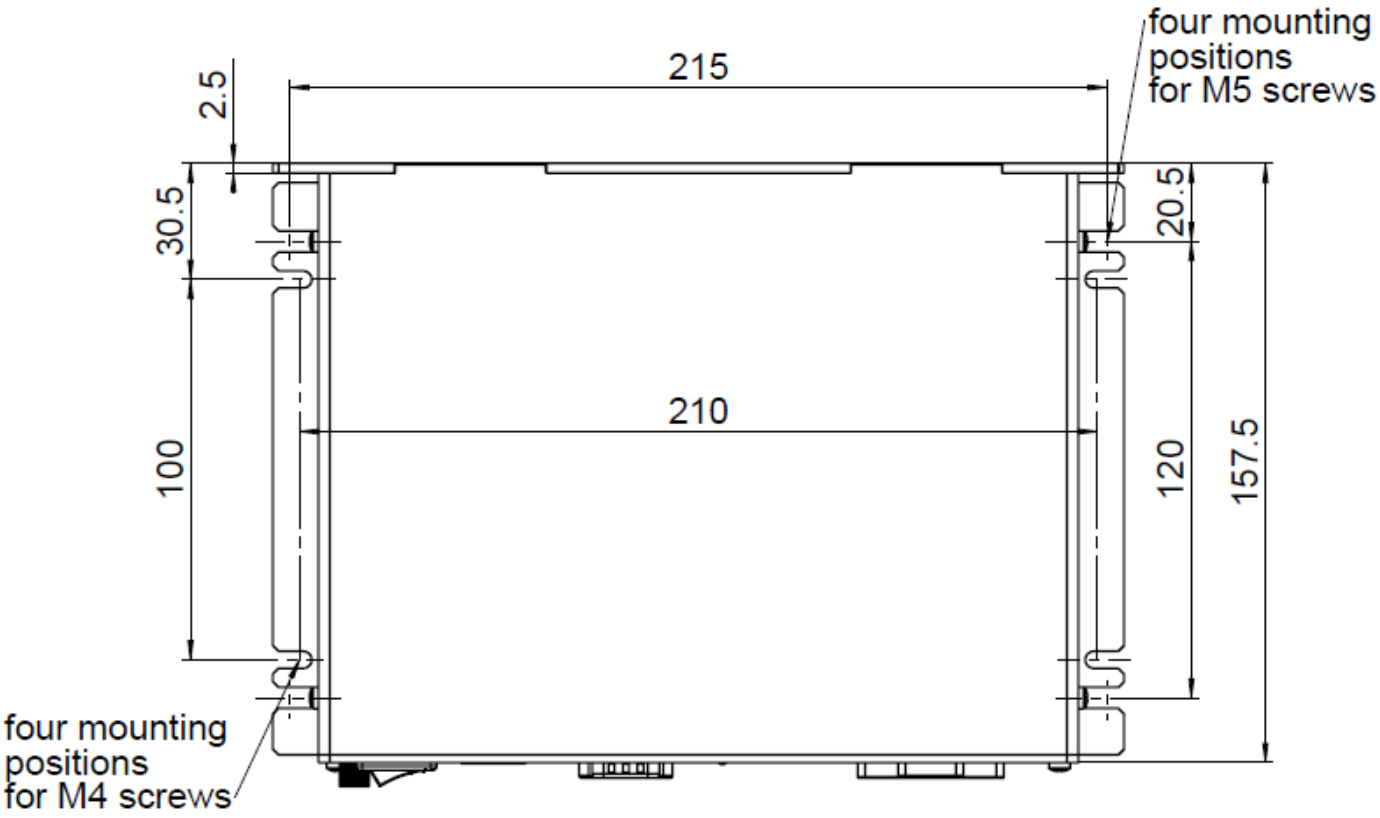
Products	Digital Controllers for Piezo-Actuators/Displacement-Sensors		
Product Numbers	**EBC/EBD-1203nn (nn = number of channels)**	**Unit**	**Tolerance**
Channels	1, 2, 2.5, 3, 3.5 (.5 stands for additional open-loop channel)		
Output Voltage for Piezo	-45 to +180	V	max.
Current	300	mA	peak max.
Average Current	120	mA	max.
Sensors	Capacitive (EBC), Strain-Gages (EBD)		
Resolution	20	Bit	
Control Loop Time	10 for EBC* and EBD-1202*	µs	
	20 for EBD-1203*	µs	
Control Parameters	PID, 2 Notch Filters per channel		
Software	nFControl Windows™ GUI		
Operating System Requirements	Microsoft ® Windows™ 7/8/10		
Wave-Generators	3 arbitrary wave-generators		
Data-Recorders	16 data recorders		
Memory	Total 4Mega data points for wave-/recorder-tables	DW	
Digital Interfaces	Ethernet, USB 2.0, RS232, Parallel (optional)		
Analog Input	settable, -10 to +10, -5 to +5, 0 to 10	V	18bit/±10V
Analog Output	-5 to +5 (Varies with controller type)	V	
Connection, Analog I/O	2x BNC (1 CH), DSub15f (2, 3 CH Versions)		
Connection, Piezo and Sensor	DSub15f (1 CH), DSub25f (2, 3 CH Versions)		
Connection, P/S	SwitchCraft RASPC10PS		
P/S, counter-socket	SwitchCraft PowerJack S10KS12		
Power Supply (Vin)	external, included		
Vin Voltage	24 (2A min.)	VDC	
Vin Protection	Power input reverse polarity protection		
Static Power Consumption	Ca. 10	W	
Ambient Temperature	+10 to +40	°C	
Design	aluminium black		
Dimensions, H x W x D	73 x 200 (215 incl. shackles) x 157.5 (plus interfaces)	mm	
Standards	2014/35/EU, Low Voltage Directive (LVD) 2014/30/EU, EMC Directive 2011/65/EU/ RoHS Directive Safety (Low Voltage Directive): IEC 61010-1:2010, IEC 61010-1:2010/AMD1:2016 EMC: EN 61326-1:2013 RoHS: EN IEC 63000:2018		

3.2. Dimensions EBC/EBD-120310



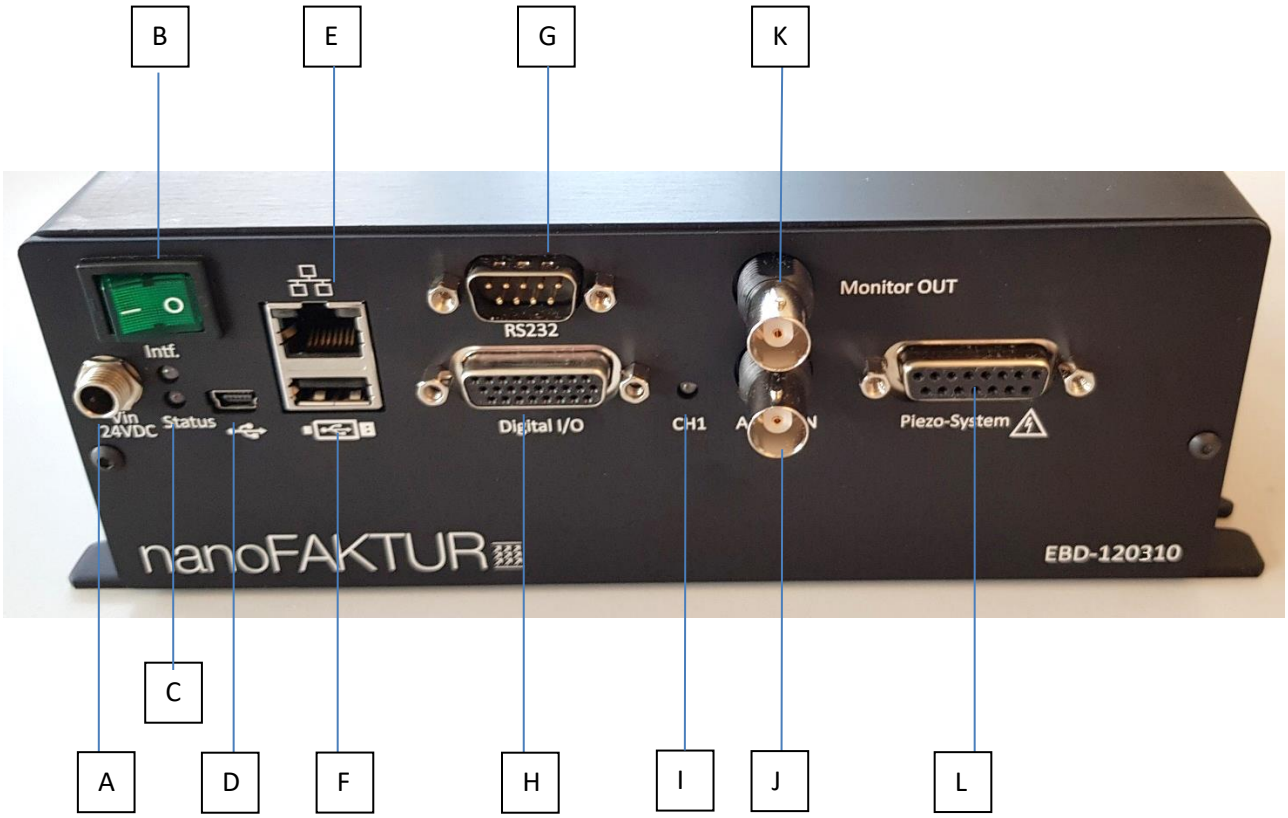
3.3. Dimensions EBC/EBD-120320 to 120335





4. Interfaces

4.1. EBC/EBD-120310, Front



A: SwitchCraft RASPC10PS, Power Supply, 24 VDC 2A

B: Power Switch: green LED indicates controller is powered-on

C: 2 LEDs, Interface/DSP Status, **green**: normal status, **red**: command error / no piezo voltage output

D: Mini USB-B, Communication

E: RJ45, Ethernet, Communication

F: USB-A, Reserved

G: DSub9m, RS232, Communication

H: DSub26f, Digital I/O-Port

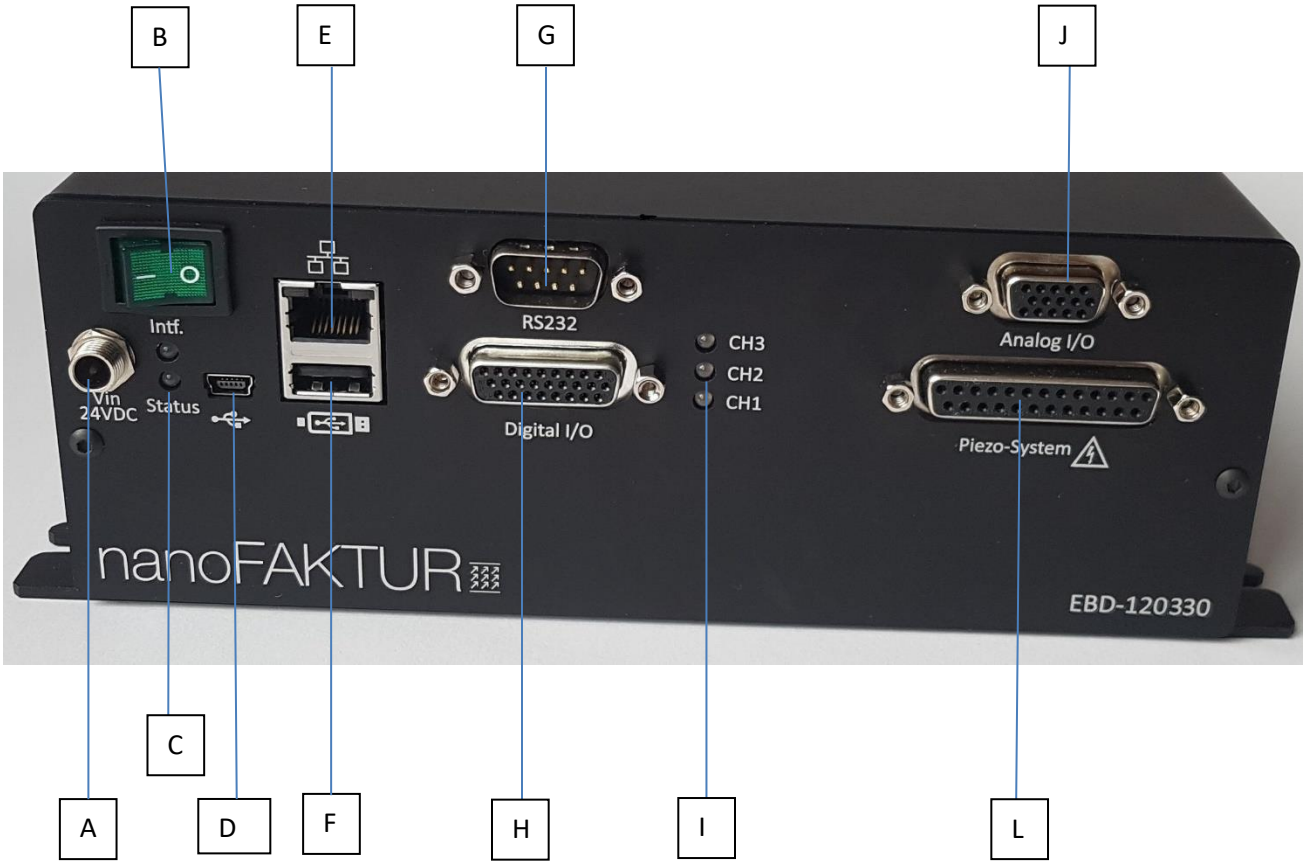
I: LED, Servo-Status, **green**: on-target, **yellow**: overflow, off: servo-off

J: BNC, Analog-Input, 0..+10 V (-10..+10 V, -5..+5 V settable at works), resolution 18bit @ -10..+10V

K: BNC, Monitor output, analogue -5..+5 V (may vary for different controllers)

L: DSub15f, Connection for assigned piezo-actuator/position-sensor

4.2. EBC/EBD-120320 to 120335, Front



A: SwitchCraft RASPC10PS, Power Supply, 24 VDC 2A

B: Power Switch: green LED indicates controller is powered-on

C: 2 LEDs, Interface/DSP Status, **green**: normal status, **red**: command error / no piezo voltage output

D: Mini USB-B, Communication

E: RJ45, Ethernet, Communication

F: USB-A, Reserved

G: DSub9m, RS232, Communication

H: DSub26f, Digital I/O-Port

I: 2/3 LEDS, Servo-Status, **green**: on-target, **yellow**: overflow, off: servo is off

J: DSub15f (VGA-kind, 9-pin housing), analog I/O), resolution 18bit @ -10..+10V

L: DSub25f, Connection for up to 3 assigned piezo-actuators/position-sensors

4.3. Protective Ground Connection

The protective Ground (PE) connection is given by a M4 threaded hole in all EBC-120 and EBD-120 models at the back of the left side seen from the front.

4.4. Pin-Assignments

4.4.1. Analog I/O of EBC/EBD-120320 to 120330

DSub15f (VGA-kind, 9-pin housing)

No.	Name	Direction	Function
1	AIN3	Input	3rd Analog input, for EBC/EBD-120330 only
2	GND	-	Ground
3	AIN2	Input	2nd Analog input
4	GND	-	Ground
5	AIN1	Input	1st Analog input
6	OUT3	Output	3rd Monitor output, for EBC/EBD-120330 only
7	GND	-	Ground
8	OUT2	Output	2nd Monitor output
9	GND	-	Ground
10	OUT1	Output	1st Monitor output
11	AIN4	Input	4th Analog input
12	R	-	Reserved
13	NC	-	Not internal connection
14	GND	-	Ground
15	GND	-	Ground

Note: for EBC/EBD-120310, analog-I/O connectors are BNC.

4.4.2. Digital I/O Connector

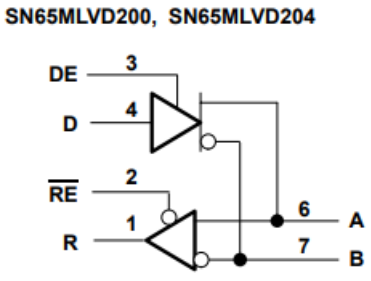
Connector Manufacture: Amphenol FCI

Manufacture part number: ICD26S13E4GX00LF

No.	Name	Direction	Function
1	OUT1	Output	1st digital output (3.3V or 5V)
2	OUT2	Output	2nd digital output (3.3V or 5V)
3	OUT3	Output	3rd digital output (3.3V or 5V)
4	VOL	Output	Voltage (3.3V or 5V, as reference)
5	GND	-	Ground
6	DIO1+	Input/output	Differential signal 1 positive[1]
7	DIO2+	Input/output	Differential signal 2 positive
8	DIO3+	Input/output	Differential signal 3 positive
9	DIO4+	Input/output	Differential signal 4 positive (option as Sync+)[2]
10	IN1	Input	1st digital input (TTL, 5V tolerant)
11	IN2	Input	2nd digital input (TTL, 5V tolerant)
12	IN3	Input	3rd digital input (TTL, 5V tolerant)
13	GND	-	Ground
14	GND	-	Ground
15	DIO1-	Input/output	Differential signal 1 negative
16	DIO2-	Input/output	Differential signal 2 negative
17	DIO3-	Input/output	Differential signal 3 negative
18	DIO4-	Input/output	Differential signal 4 negative (option as Sync-)[2]
19	RSTn	Input	System reset (low = active)
20	RSTn_St	Output	System reset output (low = active)
21	GND	-	Ground
22	VOL_SEL	Input	digital output voltage selection Float or High: 3.3V (default) Pulldown to GND: 5V
23	RSV		Reserved, do not connect
24	GND	-	Ground
25	NC	-	No internal connection
26	NC	-	No internal connection

Note:

1. the differential signal driver/receiver of
   - EBC-120* is SN65MLVD200ADR (Link: SN65MLVD200/2/4/5: Multipoint-LVDS Line Drivers and Receivers datasheet (Rev. E))
   - EBD-120* is LTC2854 (Link: LTC2854/LTC2855 – 3.3V 20Mbps RS485/RS422 Transceivers with Integrated Switchable Termination (analog.com))



2. from FPGA-firmware revision 3.04.

4.4.3. DSub15f and DSub25f, Stage-Connectors

EBC/EBD 1 channel versions are equipped with DSub15 stage-connectors, while 2- and 3-channel versions have DSub25.

DSUB15	DSUB25	Name	Function
8	13	RSV1	Reserved signal
15	25	ID-Chip	ID-Chip signal
7	12	RSV2	Reserved signal
14	24	RSV3	Reserved signal
6	11	Vp	Positive power supply
13	23	GND	Ground
5	10	Vn	Negative power supply
12	22	Vref+	Voltage reference positive
4	9	Vs1-	Ch1 Sensor signal negative
11	21	Vref-	Voltage reference negative
3	8	Vs1+	Ch1 Sensor signal positive
10	20	GND	Ground
2	7	NC	No internal connection
9	19	Hv1-	Ch1 Piezo driving signal negative
1	6	Hv1+	Ch1 Piezo driving signal positive
-	5	Vs2+	Ch2 Sensor signal positive
-	18	Vs2-	Ch2 Sensor signal negative
-	4	Vs3+	Ch3 Sensor signal positive
-	17	Vs3-	Ch3 Sensor signal negative
-	2	Hv2+	Ch2 Piezo driving signal positive
-	3	Hv2-	Ch2 Piezo driving signal negative
-	16	Hv3+	Ch3 Piezo driving signal positive
-	15	Hv3-	Ch3 Piezo driving signal negative
-	14	Hv4+	Ch4 Piezo driving signal positive
-	1	Hv4-	Ch4 Piezo driving signal negative

5.Software and Communication

5.1. Software Installation

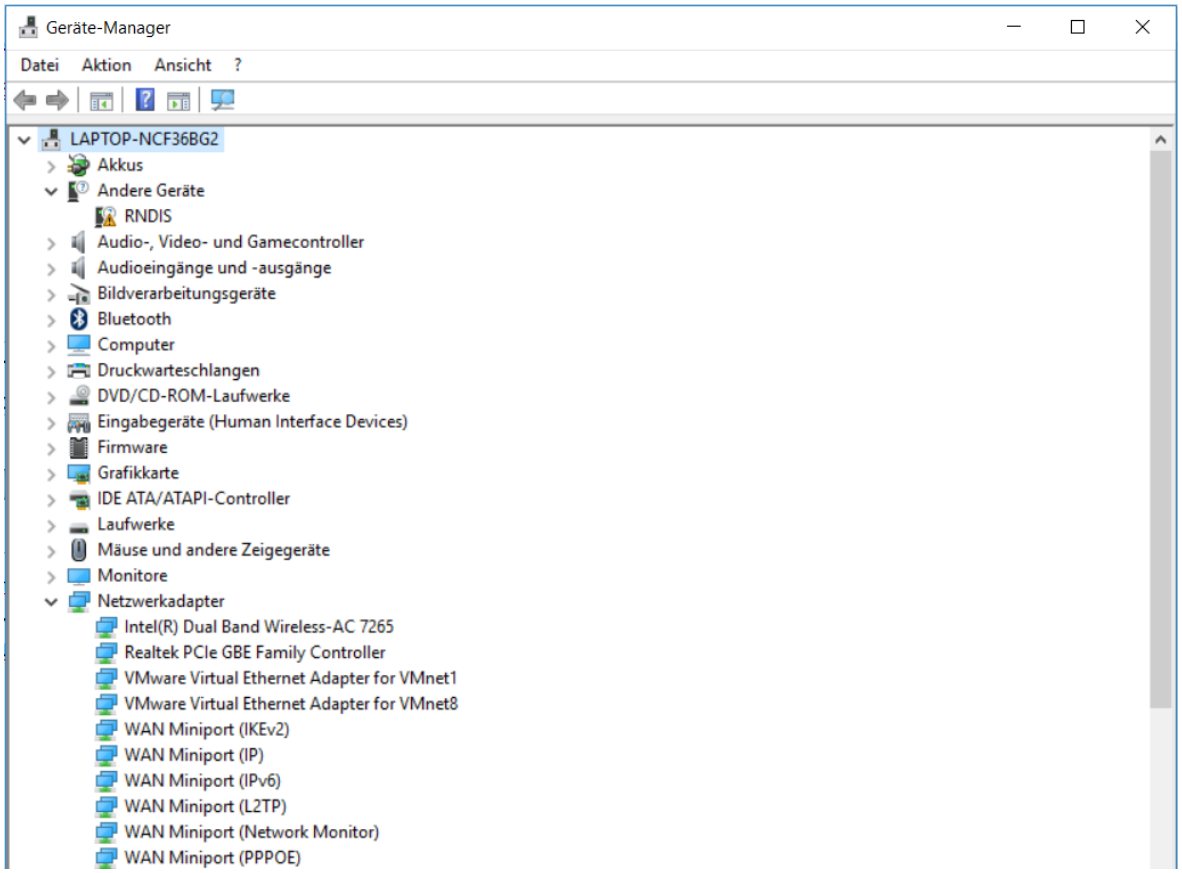
5.1.1. Download up-to-date software and documentation

https://www.nanofaktur.com/controllers/digital-controllers-closed-loop/support
EBx-1203nn Software Support.zip
Password: nFSoftw

The following chapter describes how a USB-port is configured as an RNDIS network port, that will be regarded as an TCP/IP interface by the system. The chapter after this is describing how to configure either the RNDIS or the Ethernet-interface as TCP/IP port.

5.1.2. Install Windows® driver (RNDIS) for USB Interface

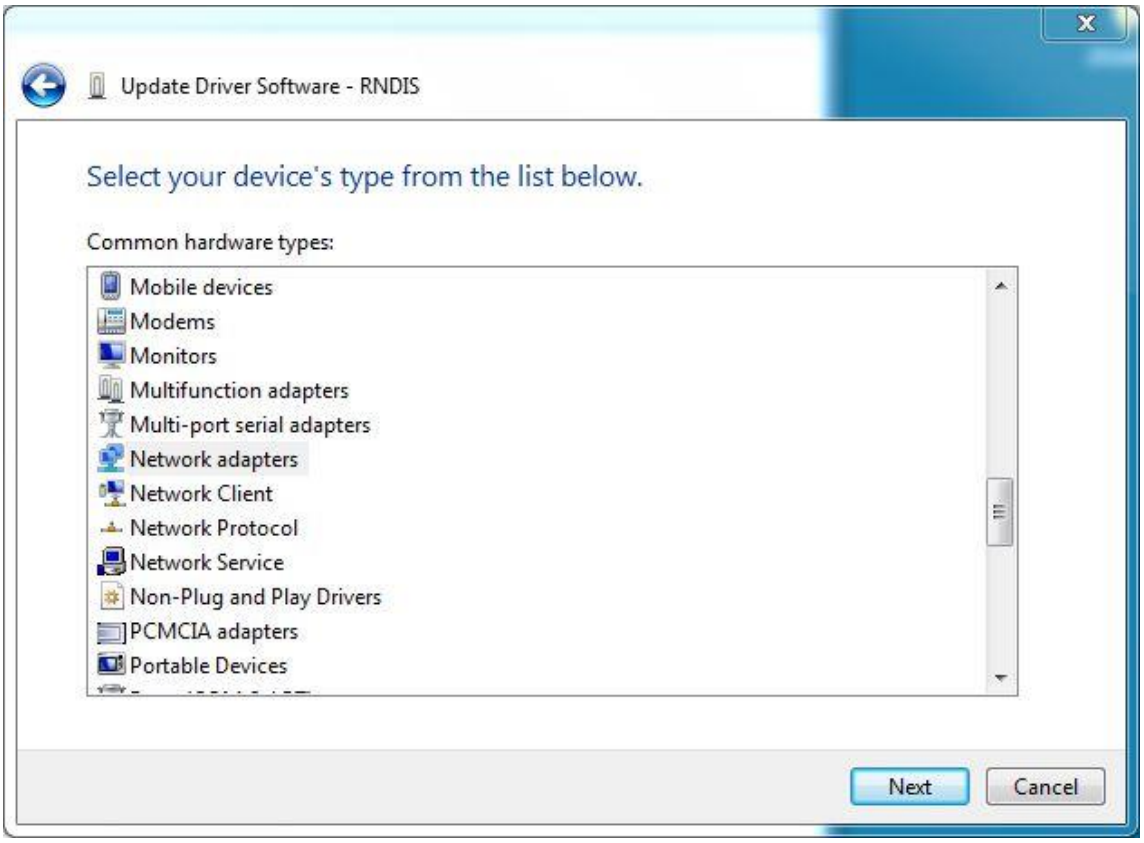
When the EBC/EBD-USB is connected to PC (ca. 10 seconds after controller has been switched on), Windows shows the new device RNDIS:



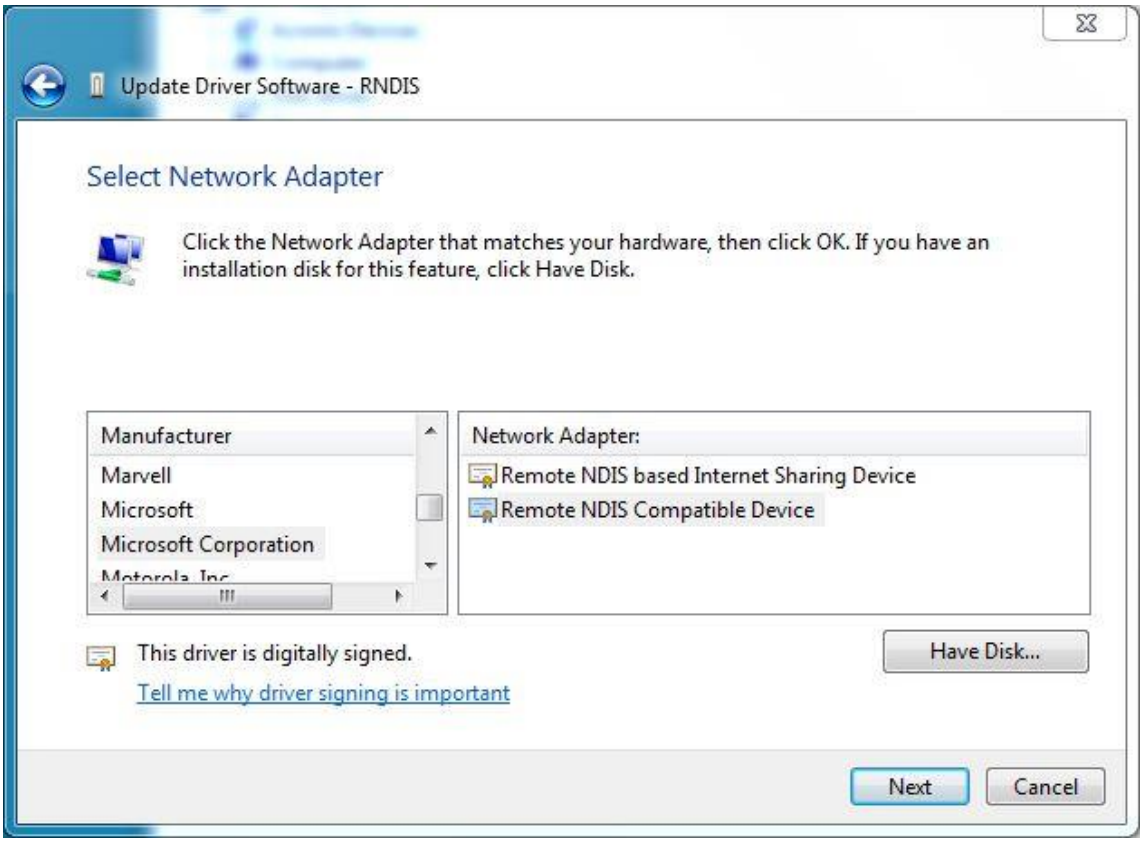
Right click on RNDIS and install driver:

Select “Browse my computer for driver software” and then “Let me pick from a list…”

Select “Network adapters”



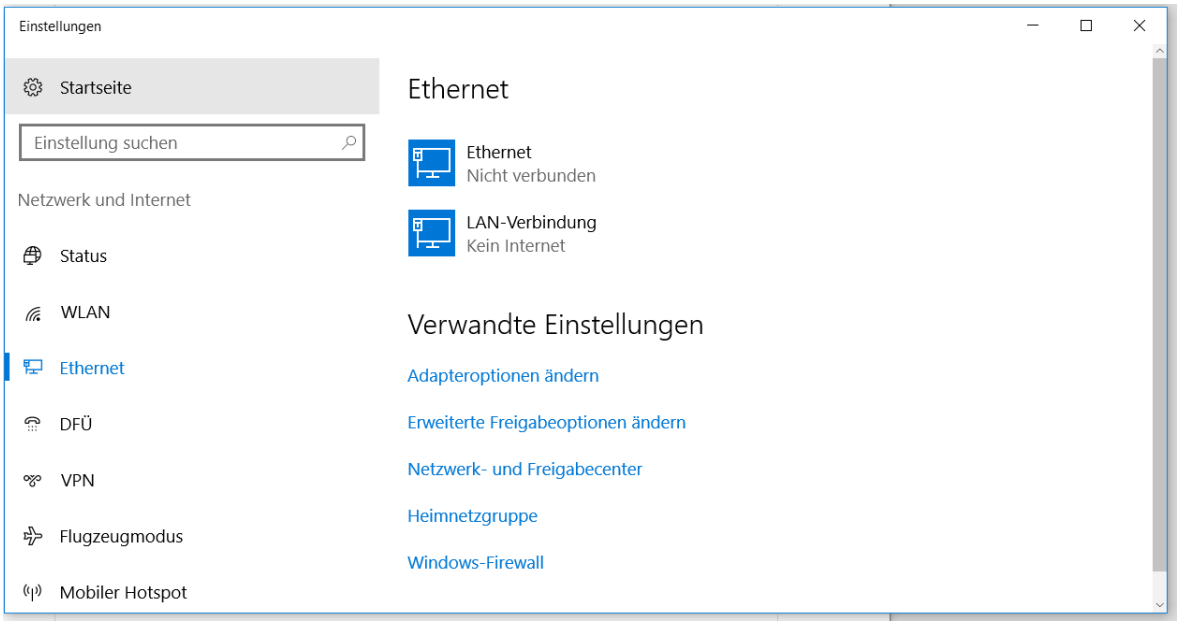
Use the default driver from Microsoft



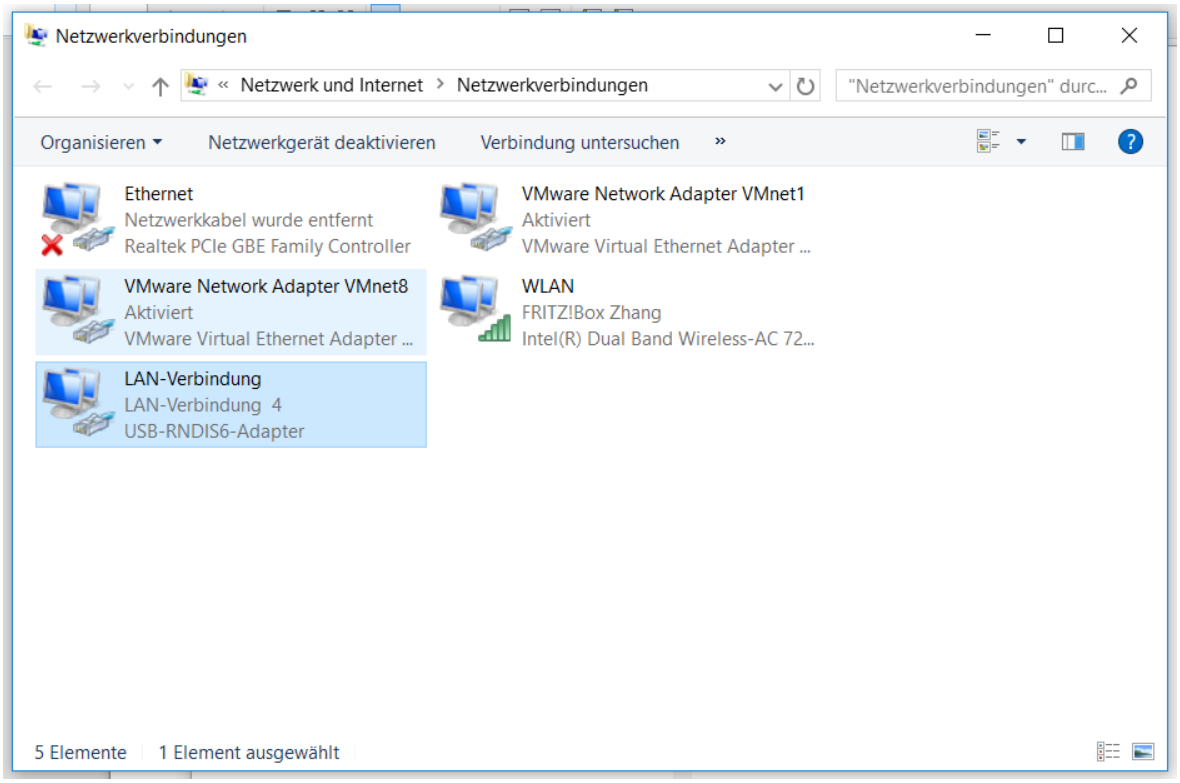
Now Windows shows NDIS Compatible Device

5.1.3. Setup TCP/IP for Ethernet and USB-RNDIS

From Windows Setup, click “Ethernet”

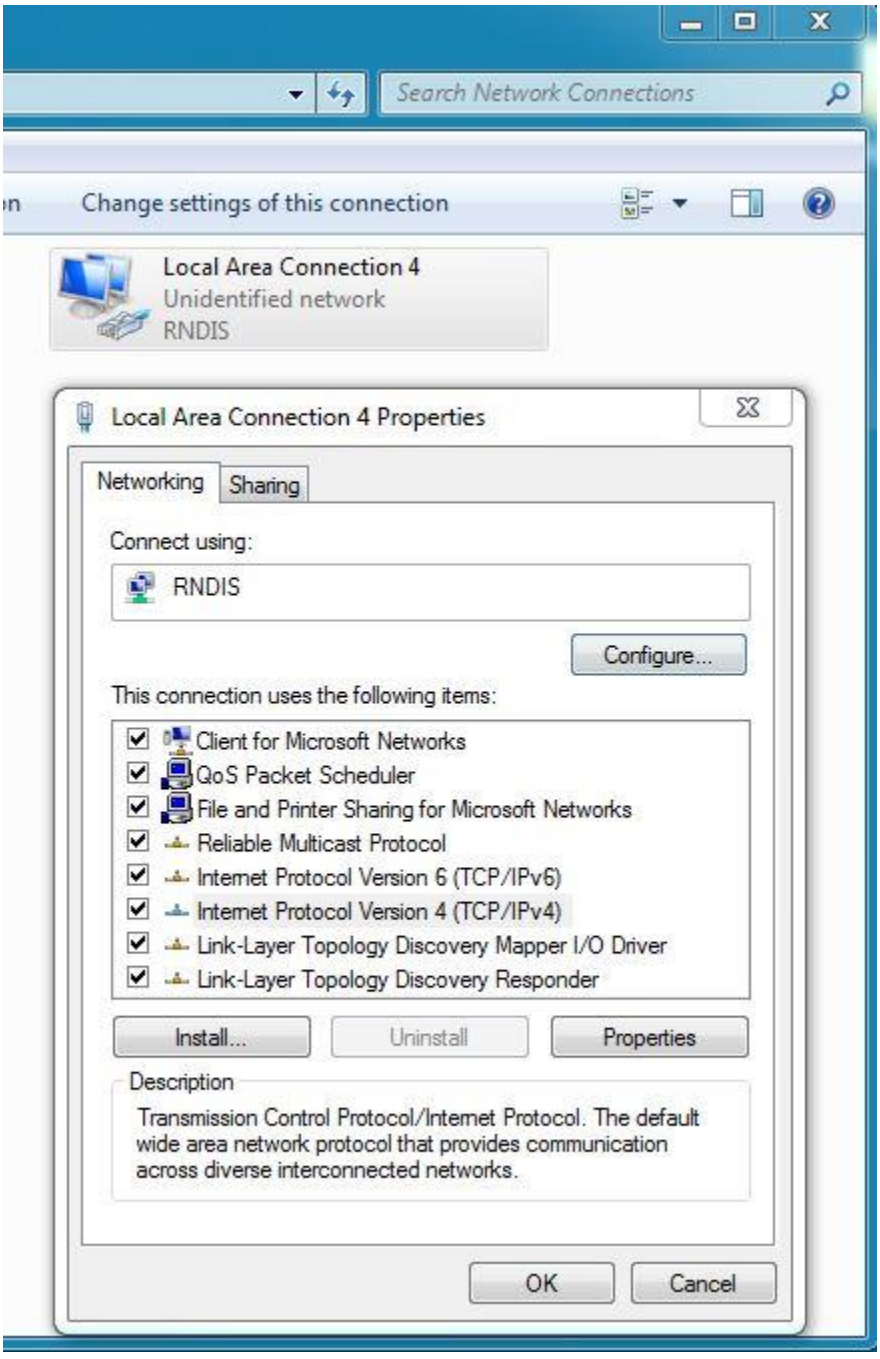


As shown below, there are one RNDIS-Adapter for USB connection and an Ethernet for normal LAN connection.

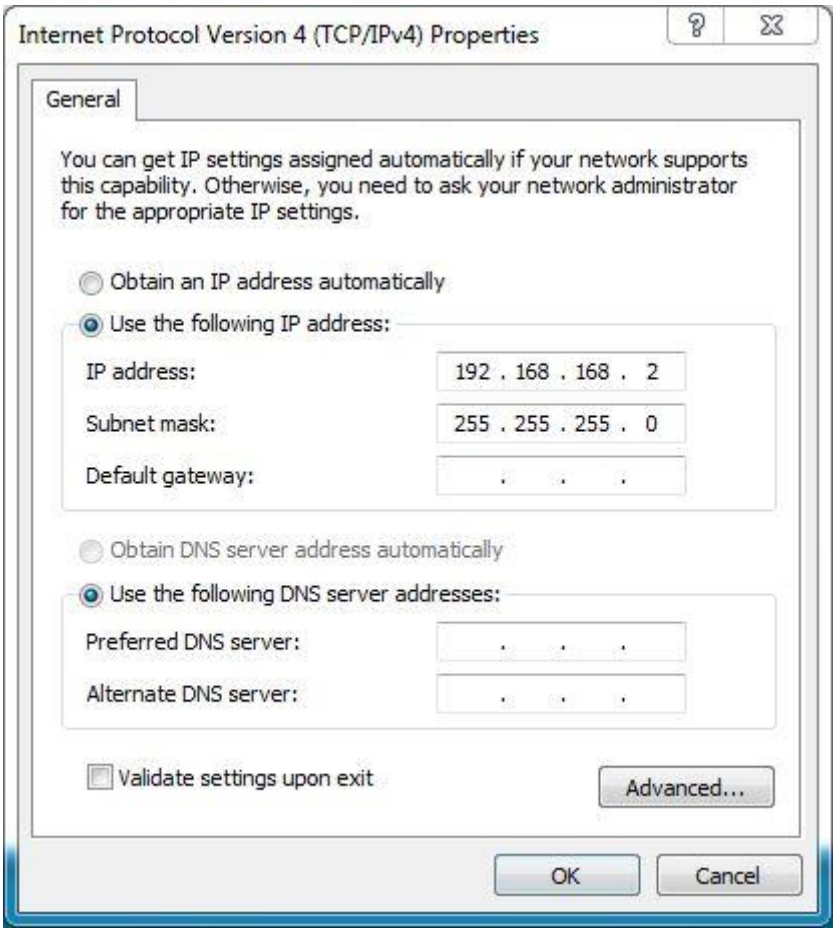


Setup the IP address: (Controller’s IP address for USB is fixed to 192.168.168.168)

The PC's RNDIS address must be in the same subnet class as the controller, i.e. 192.168.168.xxx.



Select “(TCP/IPv4)” and click the “Properties”



In this example, the PC’s IP is set to 192.168.168.2

For Ethernet connection, the controller’s IP address can be set to be static or DHCP.

A static IP-address is commonly used when connecting a controller to a PC directly. DHCP should be used when a controller is in a network.

Use parameter 0xFF010001 to select method to use (0=static, 1=DHCP)

When controller is configured to be static (default), setup controller’s IP address with parameter 0xFF010012. And the PC side should also be setup in the same subnet class (like RNDIS).

5.2. Operation and Tuning via the nFControl GUI

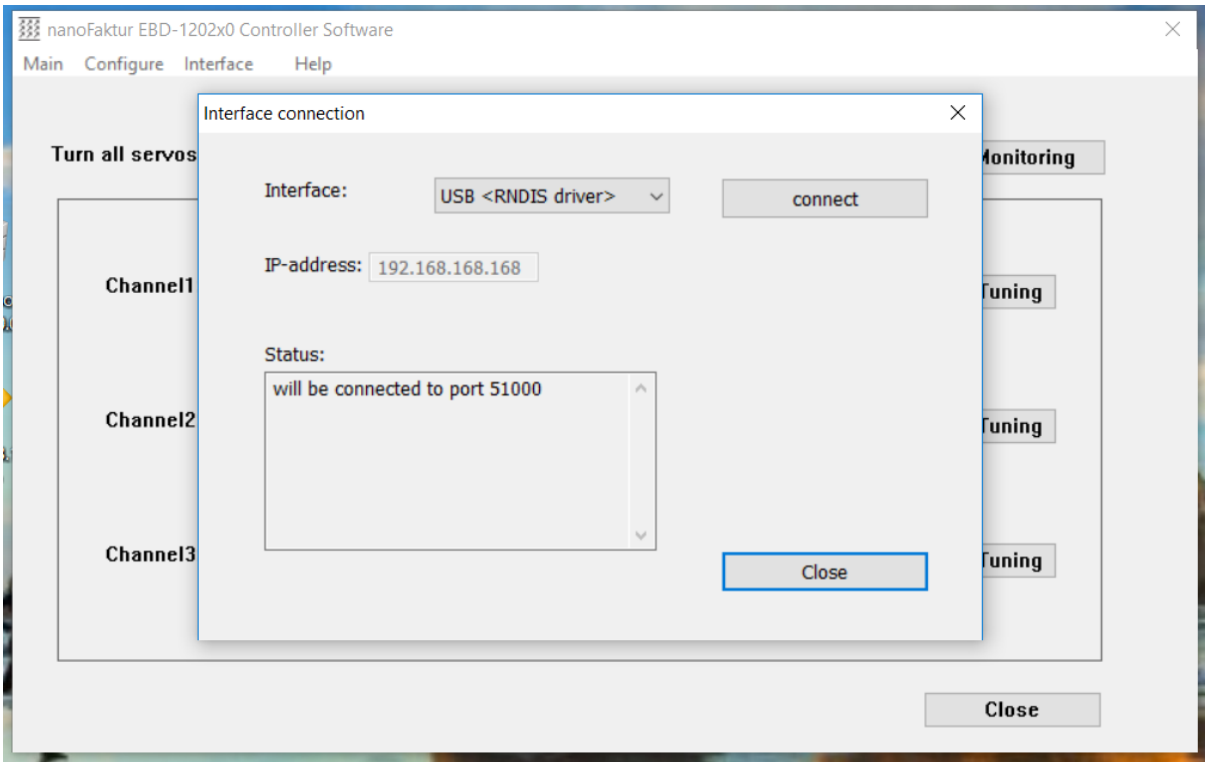
nanoFaktur GmbH provides a GUI for Windows®: nFControl.exe. This is convenient for the first usage and tuning of controllers. Interface functions are realized in the nF_interface.dll. It can be used for C, Labview or Python programming (32bits-mode example code is provided).
Demo source code (written in C which be compiled with MinGW-32bits) is also provided.

5.2.1. Install nFControl GUI

At first, there is no setup.exe for this demo software. Just un-zip all the files into a local folder.

5.2.2. Quick guide

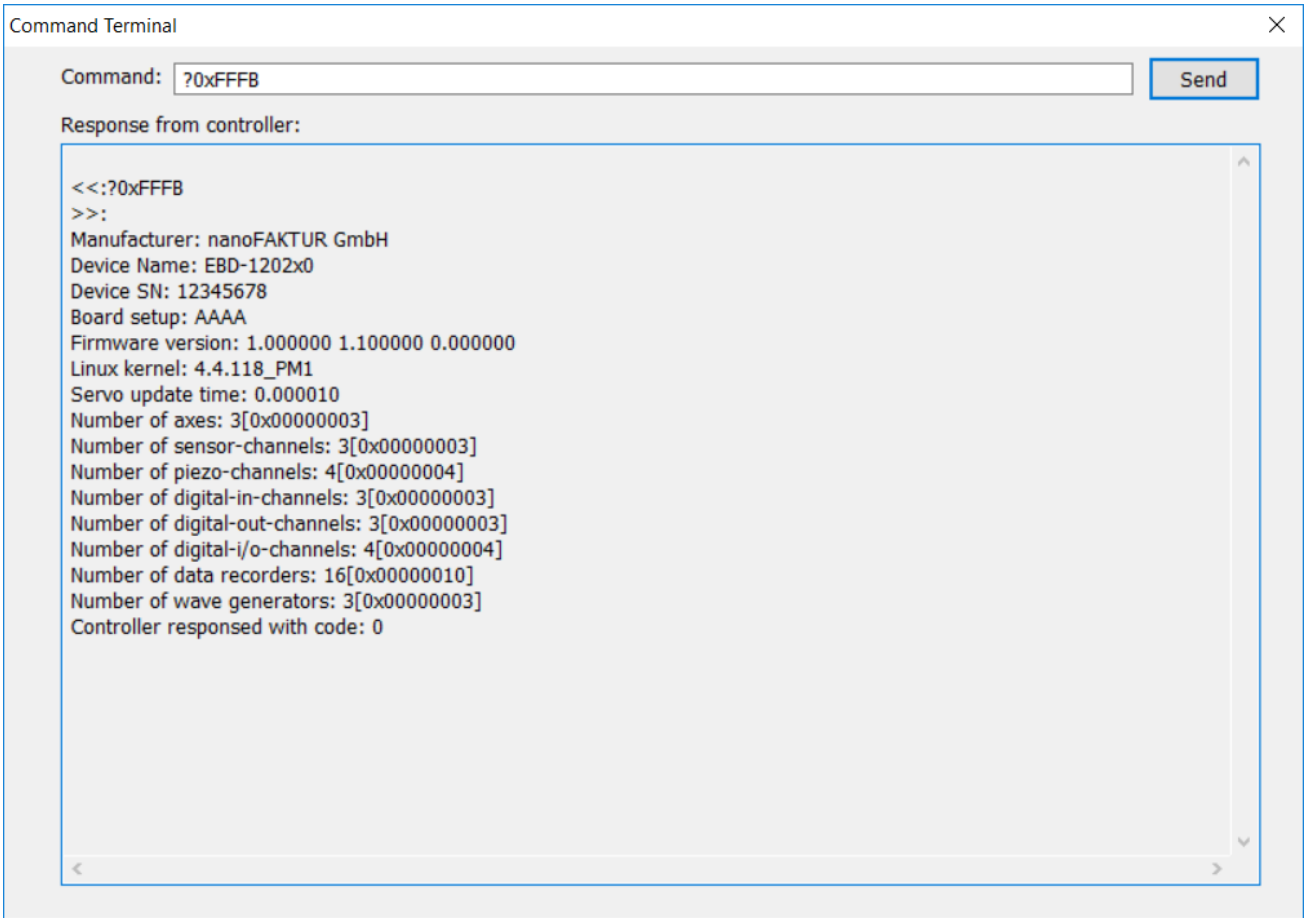
1) connect to controller (or simulation) with top-menu '***Interface***'



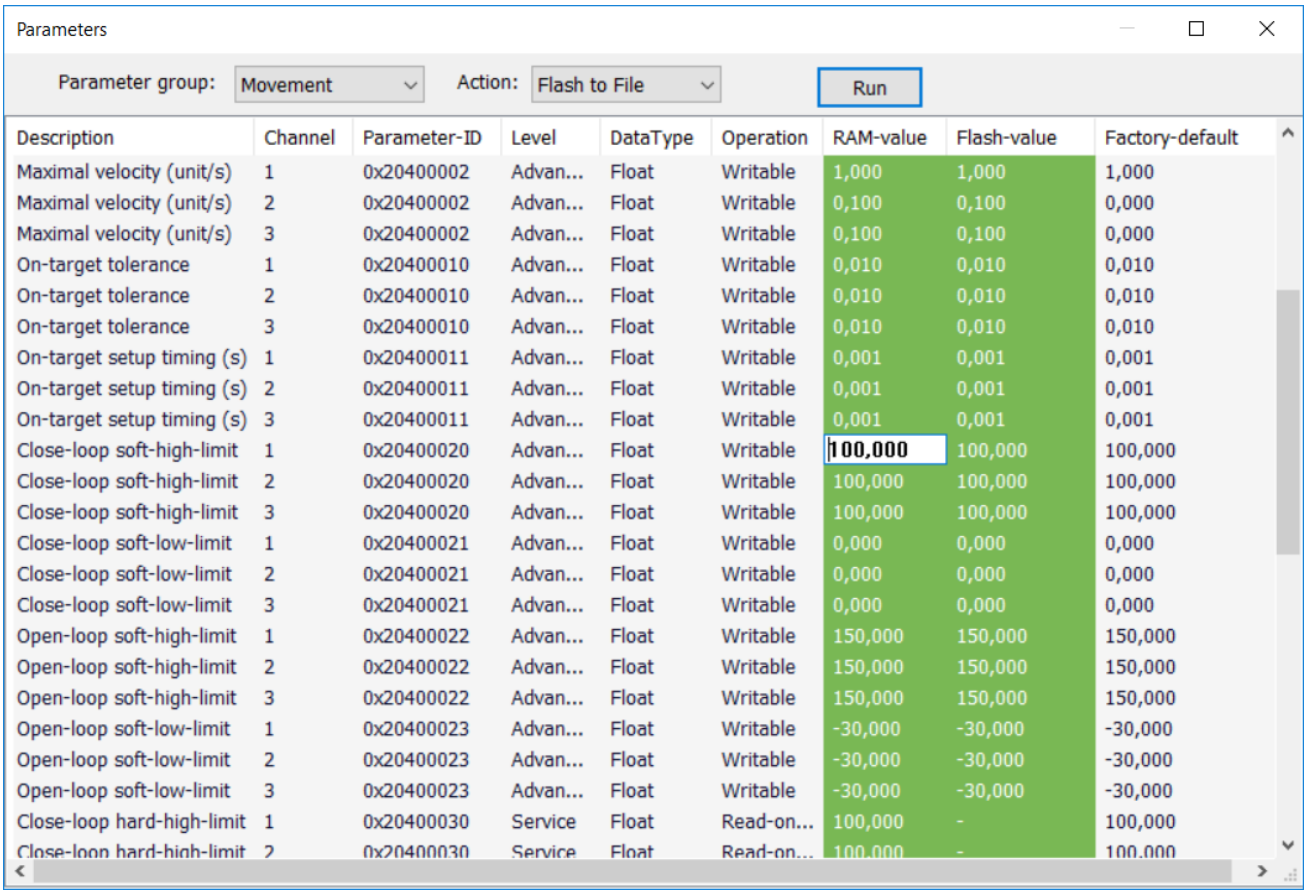
Note:

Some command needs "Command-level 1". nFControl.exe sends "0xFFF0 1" after interface 'connect'. It is recommended to take the same step by customer's Software or in controller's Macro.

2) test commands with '***Main -> Command Terminal***'
send "?0xfffb" to show controller information,
send "?0xffff" to show all valid commands,
etc.



3) view and/or modify parameter-values with '***Main -> Parameters***'



Description	Channel	Parameter-ID	Level	DataType	Operation	RAM-value	Flash-value	Factory-default
Maximal velocity (unit/s)	1	0x20400002	Advan...	Float	Writable	1,000	1,000	1,000
Maximal velocity (unit/s)	2	0x20400002	Advan...	Float	Writable	0,100	0,100	0,000
Maximal velocity (unit/s)	3	0x20400002	Advan...	Float	Writable	0,100	0,100	0,000
On-target tolerance	1	0x20400010	Advan...	Float	Writable	0,010	0,010	0,010
On-target tolerance	2	0x20400010	Advan...	Float	Writable	0,010	0,010	0,010
On-target tolerance	3	0x20400010	Advan...	Float	Writable	0,010	0,010	0,010
On-target setup timing (s)	1	0x20400011	Advan...	Float	Writable	0,001	0,001	0,001
On-target setup timing (s)	2	0x20400011	Advan...	Float	Writable	0,001	0,001	0,001
On-target setup timing (s)	3	0x20400011	Advan...	Float	Writable	0,001	0,001	0,001
Close-loop soft-high-limit	1	0x20400020	Advan...	Float	Writable	100,000	100,000	100,000
Close-loop soft-high-limit	2	0x20400020	Advan...	Float	Writable	100,000	100,000	100,000
Close-loop soft-high-limit	3	0x20400020	Advan...	Float	Writable	100,000	100,000	100,000
Close-loop soft-low-limit	1	0x20400021	Advan...	Float	Writable	0,000	0,000	0,000
Close-loop soft-low-limit	2	0x20400021	Advan...	Float	Writable	0,000	0,000	0,000
Close-loop soft-low-limit	3	0x20400021	Advan...	Float	Writable	0,000	0,000	0,000
Open-loop soft-high-limit	1	0x20400022	Advan...	Float	Writable	150,000	150,000	150,000
Open-loop soft-high-limit	2	0x20400022	Advan...	Float	Writable	150,000	150,000	150,000
Open-loop soft-high-limit	3	0x20400022	Advan...	Float	Writable	150,000	150,000	150,000
Open-loop soft-low-limit	1	0x20400023	Advan...	Float	Writable	-30,000	-30,000	-30,000
Open-loop soft-low-limit	2	0x20400023	Advan...	Float	Writable	-30,000	-30,000	-30,000
Open-loop soft-low-limit	3	0x20400023	Advan...	Float	Writable	-30,000	-30,000	-30,000
Close-loop hard-high-limit	1	0x20400030	Service	Float	Read-on...	100,000	-	100,000
Close-loop hard-high-limit	2	0x20400030	Service	Float	Read-on...	100,000	-	100,000

For parameters with 'Writable' property, double click to change the value.

RAM-value (volatile) is current being used value (will lost when controller is rebooted or powered off).

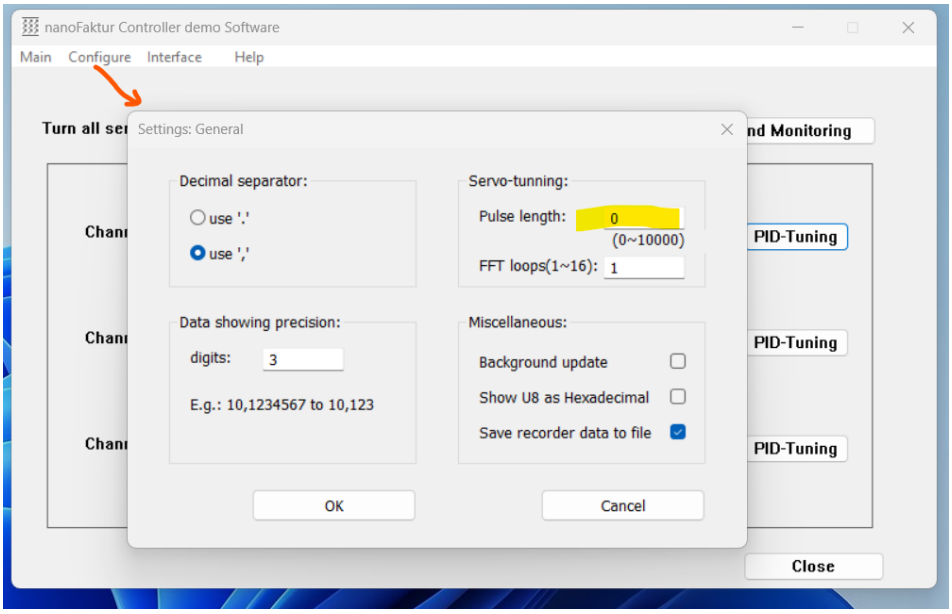
The flash-values (non-volatile) are used when the controller gets restarted.

Factory-default: the value being used before delivery (can be used as reference).

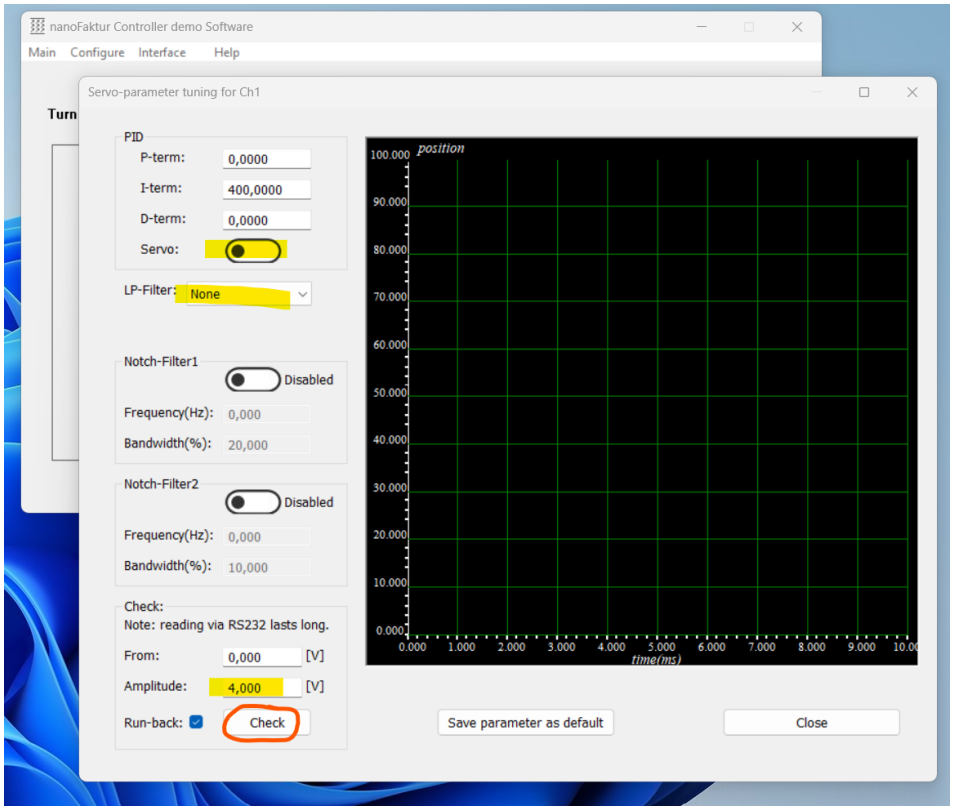
5.2.3. PID tuning

- Check Resonance frequency

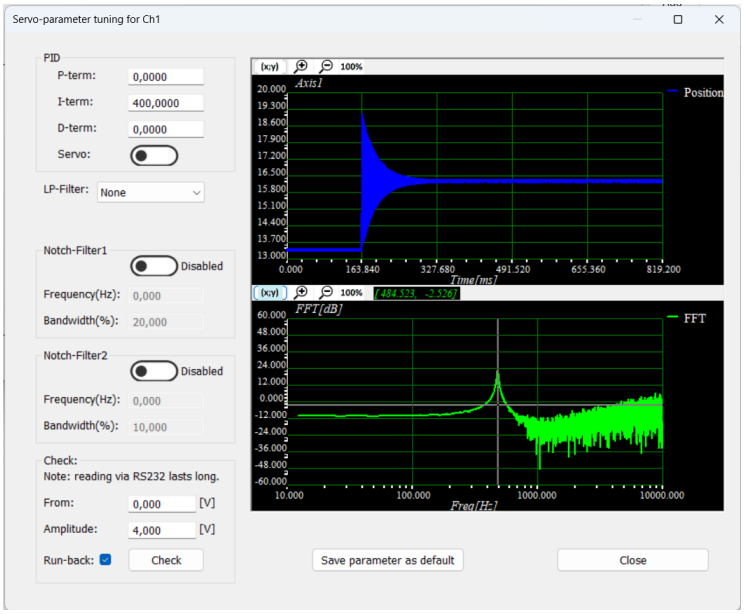
Setup the software: “Pulse length” to 0 for voltage step-mode (or value>=1 for ‘pulse-mode’)



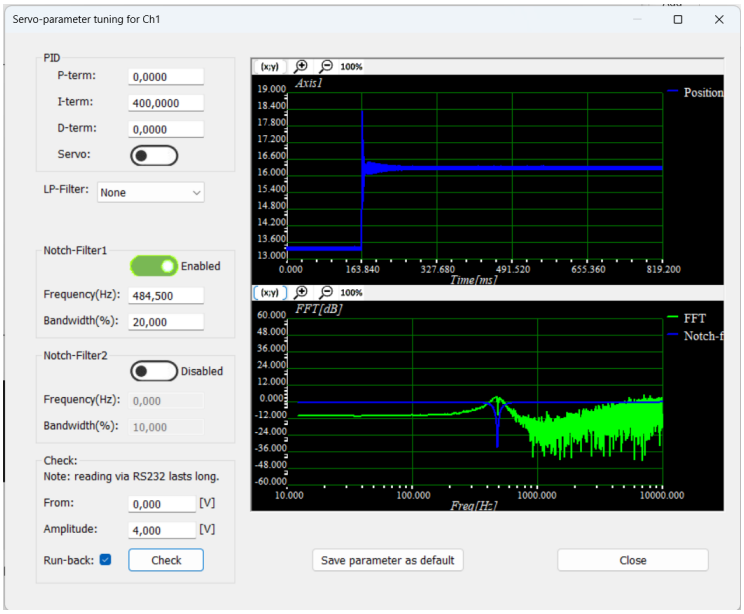
Set Low-pass-filter (LP-Filter) to ‘None’



Click “Check”

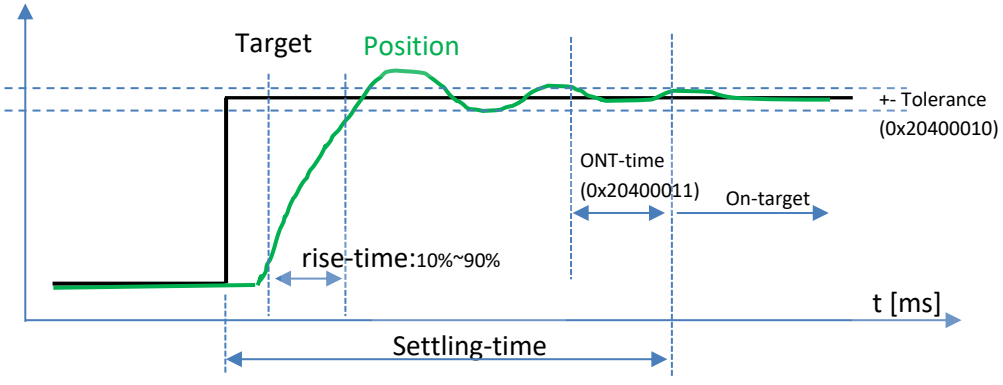


As shown in this example, set Notch-filter to 484.5Hz and ‘Check’ again

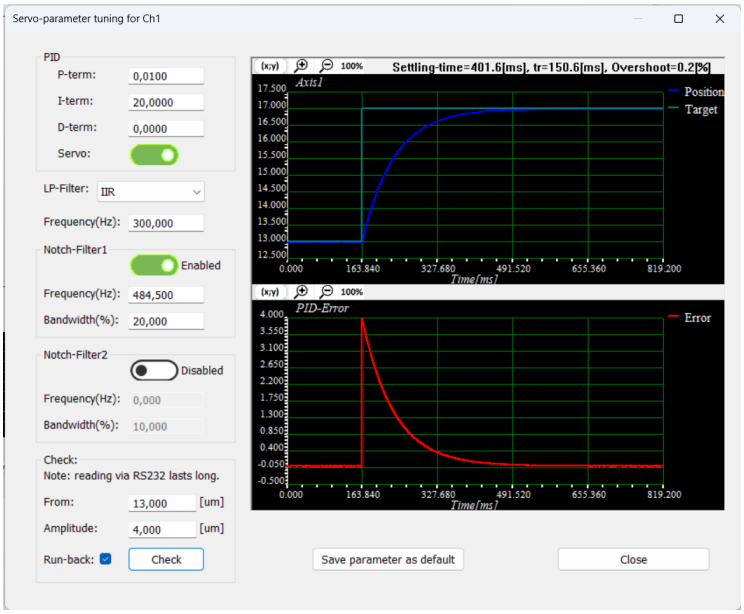


Note: the value varies with different mechanics and/or last.

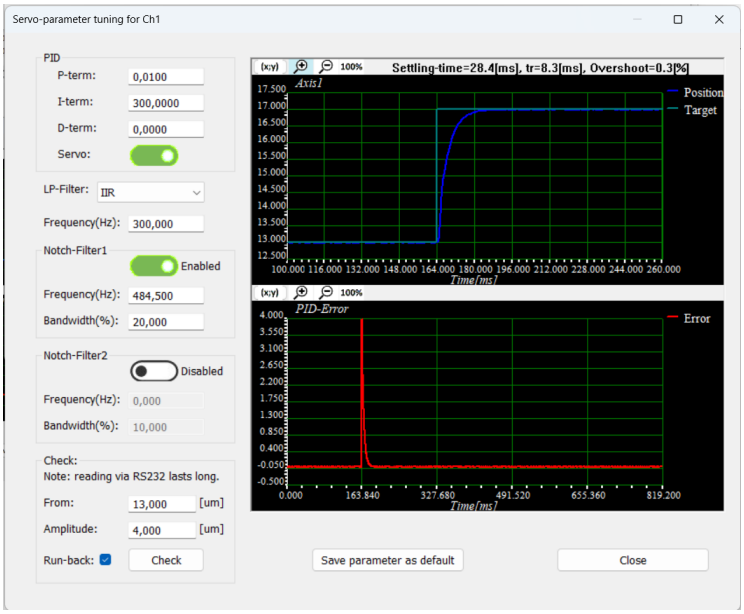
Item-definition for PID-related parameters



- Optimize the performance
  Turn servo-ON.
  Suppose the application requires a speed of 200um/s, then set the step to 4um.



Adjust the PID parameters until the rising-time (tr) reaches ca. 10ms (4um/10ms is enough for 200um/1s or 2um/10ms).



Note: in order to avoid oscillations during the PID parameter tuning, the recommended values are:

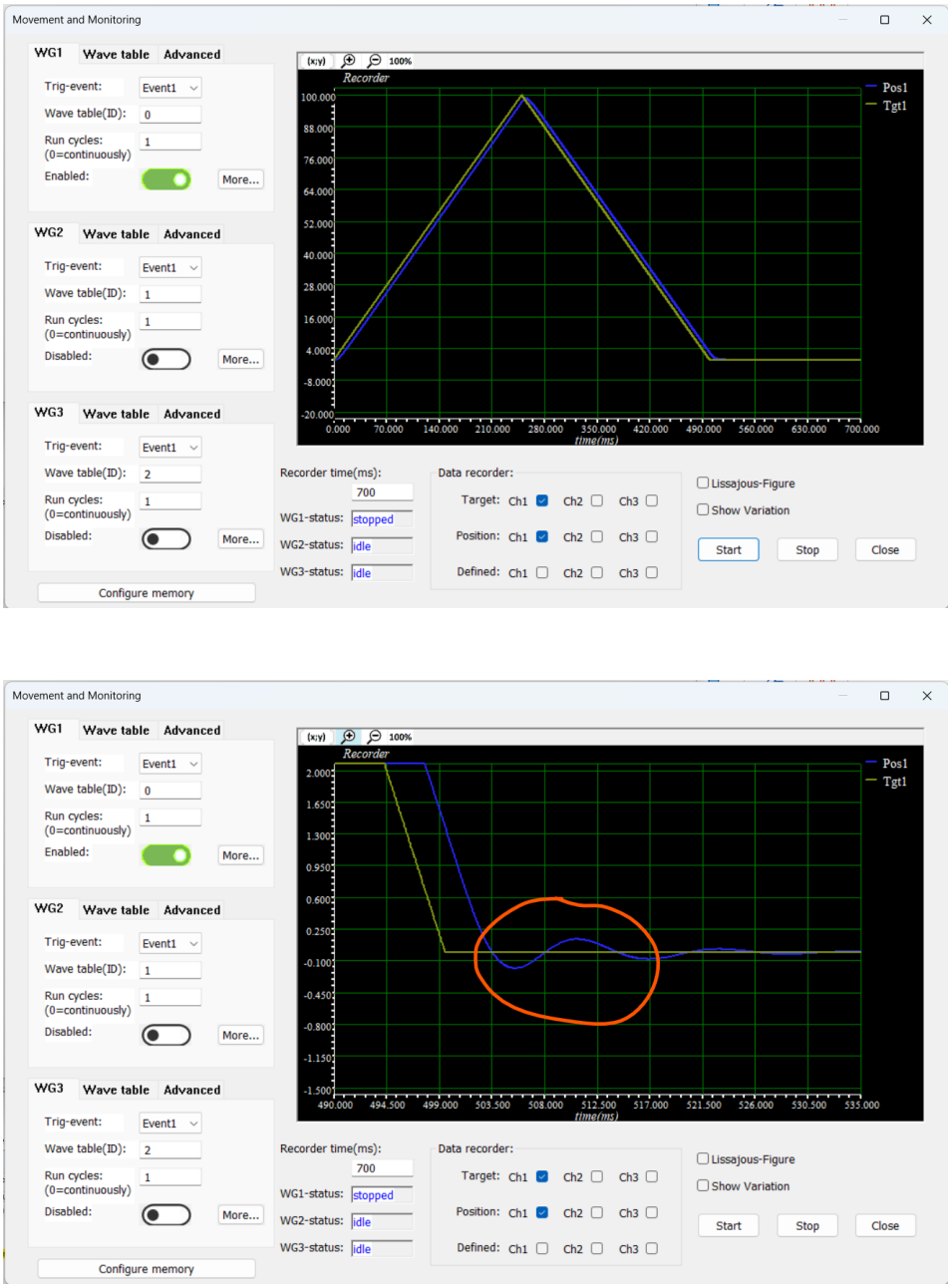
P-Term: starts from 0.01, step 0.01
I-Term: starts from 10.0, step 10.0
D-Term: not used normally

Note: when stage ossilates by tuning, Piezo-drving module may switches off (DSP-Led truns to red). Send 0x204F to reset the status.

Read the Wiki example https://en.wikipedia.org/wiki/PID_controller#/media/File:PID_Compensation_Animated.gif for deeper understanding.

- Overshoot issue

For a “Triangle” wave-form (yellow line: Tgt1), the real-position can have overshoot.

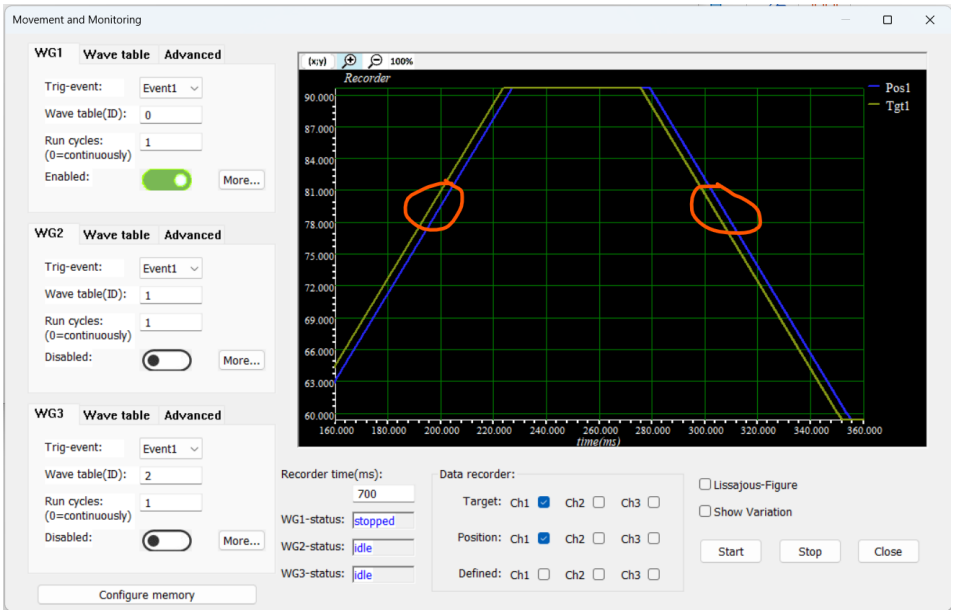


Enlarged figure for showing the overshoot

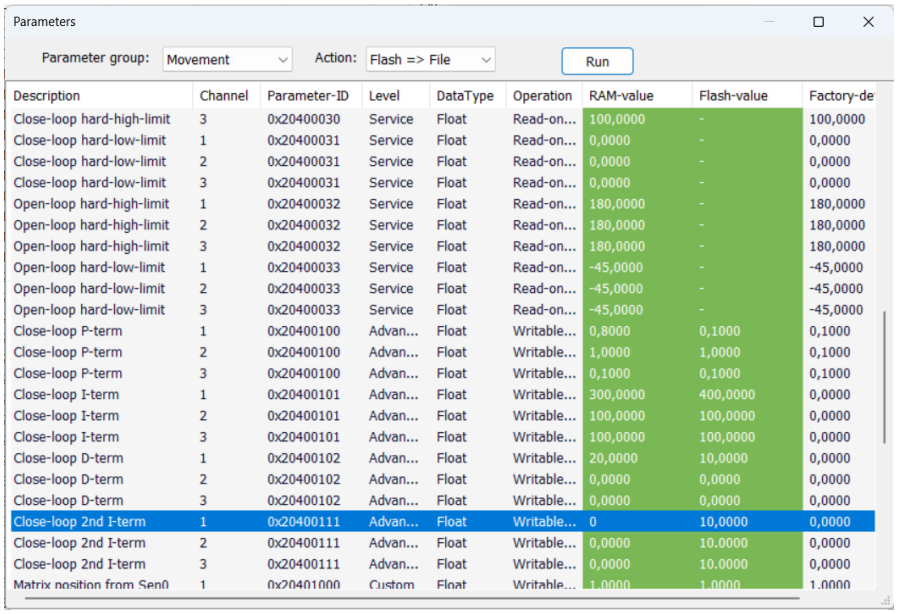
The error can be reduced by limiting the acceleration of ‘Target’, i.e. Target should be changed smoothly.

- “following error“ Issue

As the example above, real-position is not exactly the target value.



The difference between real-position and target can be reduced by the “Close-loop 2nd I-Term”. But in this case, the overshoot error could be increased.



Description	Channel	Parameter-ID	Level	DataType	Operation	RAM-value	Flash-value	Factory-de
Close-loop hard-high-limit	3	0x20400030	Service	Float	Read-on...	100,0000	-	100,0000
Close-loop hard-low-limit	1	0x20400031	Service	Float	Read-on...	0,0000	-	0,0000
Close-loop hard-low-limit	2	0x20400031	Service	Float	Read-on...	0,0000	-	0,0000
Close-loop hard-low-limit	3	0x20400031	Service	Float	Read-on...	0,0000	-	0,0000
Open-loop hard-high-limit	1	0x20400032	Service	Float	Read-on...	180,0000	-	180,0000
Open-loop hard-high-limit	2	0x20400032	Service	Float	Read-on...	180,0000	-	180,0000
Open-loop hard-high-limit	3	0x20400032	Service	Float	Read-on...	180,0000	-	180,0000
Open-loop hard-low-limit	1	0x20400033	Service	Float	Read-on...	-45,0000	-	-45,0000
Open-loop hard-low-limit	2	0x20400033	Service	Float	Read-on...	-45,0000	-	-45,0000
Open-loop hard-low-limit	3	0x20400033	Service	Float	Read-on...	-45,0000	-	-45,0000
Close-loop P-term	1	0x20400100	Advan...	Float	Writable...	0,8000	0,1000	0,1000
Close-loop P-term	2	0x20400100	Advan...	Float	Writable...	1,0000	1,0000	0,1000
Close-loop P-term	3	0x20400100	Advan...	Float	Writable...	0,1000	0,1000	0,1000
Close-loop I-term	1	0x20400101	Advan...	Float	Writable...	300,0000	400,0000	0,0000
Close-loop I-term	2	0x20400101	Advan...	Float	Writable...	100,0000	100,0000	0,0000
Close-loop I-term	3	0x20400101	Advan...	Float	Writable...	100,0000	100,0000	0,0000
Close-loop D-term	1	0x20400102	Advan...	Float	Writable...	20,0000	10,0000	0,0000
Close-loop D-term	2	0x20400102	Advan...	Float	Writable...	0,0000	0,0000	0,0000
Close-loop D-term	3	0x20400102	Advan...	Float	Writable...	0,0000	0,0000	0,0000
Close-loop 2nd I-term	1	0x20400111	Advan...	Float	Writable...	0	10,0000	0,0000
Close-loop 2nd I-term	2	0x20400111	Advan...	Float	Writable...	0,0000	10,0000	0,0000
Close-loop 2nd I-term	3	0x20400111	Advan...	Float	Writable...	0,0000	10,0000	0,0000
Matrix position from Sen0	1	0x20401000	Custom	Float	Writable...	1,0000	1,0000	1,0000

Note: before PID tunning, set this parameter to 0.0

5.3. Commanding

All EBC/EBD* controllers use binary command set. Data should be sent in Little-Endian format, i.e: Byte0 - Byte1 - Byte2 -...
For double-word, the lowest address (byte0) should be sent first.
Note: In case of interrupted communication, controllers wait 2-seconds for the rest package. After that an interface-timeout-error is set and the incomplete package will be removed.

5.3.1. Package format

Below shows the package definition, please see nF_common.h in the software package for details:

```
TYPEDEF STRUCT __ATTRIBUTE__((PACKED)) _CMD_PACKAGE_ {
  u16    len;          // total package length in bytes
  u16    CmdId;        // command ID
  u16    CustomId;     // value will be just returned (can be used to diff software routine)
  u8     opt;          // option: e.g. slave should acknowledge, CRC or check-sum, etc.
  u8     seq;          // sequence number (set by controller for response)
  u8     IntfId;       // interface ID: set by controller
  u8     ChkSumHeader; // check-sum for the header (when no command data, package ends here)
  u8     *pParam;      // command data, variable length (not included in package when length is 0)
  u8     ChkSumData; // check-sum for command data (not included in package when data length is 0)
}CMD_PACKAGE;
```

Package to controller:

Byte1/2	Byte3/4	Byte5/6	Byte7	Byte8	Byte9	Byte10	Byte11…	LastByte
Len	CmdId	CustomId	opt	0	0	CRC	Command-Data	CRC-Data

For Command-Data:

Byte1	Byte2~N	Byte N+1	ByteN+2…	…
Data format A	DataA	Data format B	Data B	…

As an example, the ‘pop controller’s error’ should be sent without data:

Len = 0x000A (=10Bytes, from item “Len” to “CRC”)
CmdId = 0x1000
CustomId = 0x0000 (any value for host software only)
Opt = 0x00 (for reading)
Seq = 0x00 (not used by controller)
InfId = 0x00 (not used by controller)
CRC(ChkSumHeader) = 0xE5 (Package check sum should be 0xFF)

=> Data to controller: 0x0A - 0x00 - 0x00 - 0x10 - 0x00 - 0x00 - 0x00 - 0x00 - 0x00 - 0xE5

For package with data, data format should also be sent. For example: ‘Set target’

len = 0x0012 (18 bytes)
CmdId = 0x2004 (Open-loop target)
CustomId = 0x0000
Opt = 0x21 (for writing and requires-acknowledge)

Seq = 0x00
InfId = 0x00
CRC (ChkSumHeader) = 0xA8
Format1 = 0x00 (U8)
Data1 = 0x00 (index of the first axis)
Format2 = 0x02 (Float)
Data2 = 10.55
CRC-Data (ChkSumData) = 0xFF - sum(from Format1 to Data2)

=> Data to controller: 0x12 0x00 0x04 0x20 ....

Package from controller:
The response package from controller has same definition. Below shows part of the controller's response (command 0xFFFB: show system information)

dd 01 fb ff 00 00 10 00 00 17 04 4d 61 6e 75 66
61 63 74 75 72 65 72 3a 00 04 6e 61 6e 6f 46 41
4b 54 55 52 20 47 6d 62 48 00 0a 04 44 65 76 69
63 65 20 4e 61 6d 65 3a 00 04 45 42 44 2d 31 32
30 32 78 30 00 0a 04 44 65 76 69 63 65 20 53 4e
3a 00 04 31 32 33 34 35 36 37 38 00 0a 04 42 6f
<...>

This first 10 Bytes is the header. Then follows <format><data><format><data>...
In the above package:

<format:04=string> <data: string-end with NULL> <... repeat>
<format: 0a=Linefeed, no data>
<format:04=string> <data: string-end with NULL>

So the response can be shown as:

Manufacturer: nanoFAKTUR GmbH ***<Linefeed>***
Device Name: <...>

Please check the software header file for details.
In the nF_interface.dll, there are functions like nF_intf_write_command() and nF_intf_read_command() for simplifying the procedure. There are also some 'top-level' functions such as nF_set_dev_axis_target().
C-source code is also provided for demo purposes.

In the nF_interface.dll, there is a function nF_set_log_state(option) for command logging. If option is set to 2, the command/response-package will also be saved.

5.3.2. Command-Set

Note:

"[ ]" means optional parameters

"{ }" means one or more parameters

"- " command operation invalid

For firmware from revision ARM=2.0 / DSP=2.0

Command - ID	Parameters for reading	Parameters for writing	Level	Description
System				
0xFFFF	[{<CommandID>}]	-	0	show information of commands
0xFFFE	[{<ParameterID>}]	-	0	show information of parameters
0xFFFD	[{<ErrorCode>}]	-	0	show information of errors
0xFFFC	[{<CommandSyntax>}]	-	0	show information of command syntax
0xFFFB		-	0	show information of system
0xFFFA		-	0	show information of interface
0xFFF9	[{<ID>}]	-	0	show command/parameter options
0xFFF1	-	<option> <Parameter>	1	See Firmware update
0xFFF0	[{<Level>}]	{<CommandLevel> [password]}	0	command level set/get (No password for level 0 and 1) **Note: some parameters/commands are only allowed in Command-Level=1. So beware to change the command level first before operation.**
0xFFEF		-	0	show system status
0xFF00	-		0	system reset
Macro				
0xE012	-	<miliSecond>	1	delay macro (ms)
0xE011	-		0	stop running macros
0xE010		<name>[<times>]	0	macro-running set/get
0xE006	-	<name>	1	delete macro
0xE005		<name>	1	default macro set/get
0xE004	<name>	-	0	show macro content
0xE003		-	0	list all macros
0xE002	-		1	end of recording commands to macro
0xE001	-	<name>	1	recording commands to macro (append)
0xE000	-	<name>	1	recording commands to new macro (new/replace)
Event				
0xD042	[{<index>}]	{<index><set/clear>}	0	event status set/get
0xD041	[{<index>}]	{<index><enable/disable>}	0	event enable set/get
0xD040	[{<index>}]	{<index><mode><source>}	0	event configuration set/get
ID-Chip				
0xD0FF		-	1	Get IDChip status, i.e. number of IDChip found and the type (I2C or 1-wire)
0xD000			1	IDChip parameters store / restore
Parameter R/W				
0x6005	[{<index><parameterID>}]	-	0	get factory-default parameters

Command - ID	Parameters for reading	Parameters for writing	Level	Description
0x6004	-	<option>	0	restore parameters from non-volatile (flash) to volatile (RAM)
0x6003	-	<option>	0	save RAM parameters to flash (option is 100)
0x6002	[{<index><parameterID>}]	{<index><parameterID><data>}	0	flash parameters set/get Index is channel or axis paramaterID: see chapter 5.3.3
0x6001	[{<index><parameterID>}]	{<index><parameterID><data>}	0	RAM parameters set/get Index is channel or axis paramaterID: see chapter 5.3.3
Digital I/O				
0x50FF		-	0	show information of digital-I/O
0x5053	[{<index>}]	{<index><set/clear>}	0	digital-out status set/get
0x5052	[{<index>}]	{<index><mode>}	0	digital-out mode set/get
0x5051	[{<index>}]	{<index><enable/disable>}	0	digital-out enable set/get
0x5050		<pulse-length>	0	digital-out pulse length (N) set/get
0x5042	[{<index>}]	-	0	get digital-in level status
0x5041	[{<index>}]	-	0	get digital-in enabled-status
0x5040	[{<index>}]	{<index><invert/normal>}	0	digital-in polarity set/get
Data Recorder				
0x4060	[{<recGroup>}]	-	0	get recorder status recGroup: recorder-group ID
0x4051	[{<recGroup>}]	{<recGroup><eventID>}	0	recorder event selection set/get
0x4050	[{<index>}]	{<index><mode><source>}	0	recorder source configuration set/get index: recorder ID
0x4042	[{<recGroup>}]	-	0	get recorded data-length
0x4041	[{<recGroup>}]	{<recGroup><rate>}	0	recorder rate set/get
0x4040	[{<recGroup>}]	{<recGroup><enable/disable>}	0	recorder enable set/get
0x4011	<index><from><length> [format]	-	0	get recorder table data
0x4010		{<recGroup><tabNum><tabSize>}	1	recorder table configuration set/get
0x4000	-	[{<index>}]	0	clear recorder table
Wave generator				
0x3160	[{<index>}]	-	0	get wave-generator status index: wave-generator-group ID
0x3151	[{<index>}]	{<index><eventID>}	0	wave-generator event selection set/get
0x3143	[{<index>}]	{<index><mode>}	0	Wave-generator trigger-output-mode set/get
0x3142	[{<index>}]	{<index><cycle>}	0	wave-generator cycles set/get
0x3141	[{<index>}]	{<index><rate>}	0	wave-generator rate set/get
0x3140	[{<index>}]	{<index><enable/disable>}	0	wave-generator enable set/get
0x3111	<index><from><length>	-	0	Get trigger table data
0x3110		{<index><tabNum><tabSize>}	1	Trigger table configuration set/get Index: trigger table ID
0x3101	{<index><position>}	{<index><position><value>}	0	Trigger table set/get
0x3100	-	[{<index>}]	0	clear trigger table
0x30FF		-	0	show information of wave command

Command - ID	Parameters for reading	Parameters for writing	Level	Description
0x3012	[{<index>}]	{<index><waveTableIndex>}	0	connect generator to wave table set/get
0x3011	{<uint>}	-	0	get wave table data
0x3010		{<index><tabNum><tabSize>}	1	wave table configuration set/get index: wave table ID
0x3001	<index><from><length>	{<uint>*5}+{<float>}	0	wave table set/get (see command 0x30FF)
0x3000	-	[{<index>}]	0	clear wave table
High-voltage (HV)				
0x22FE	[{<index>}]	{<index><enable/disable>}	1	HV-output-enable set/get
0x22FD	[{<index>}]	-	0	get HV interlock status
0x2214	[{<index>}]	-	0	get HV setting
0x2212	[{<index>}]	-	0	get HV temperature
0x2211	[{<index>}]	-	0	get HV value
0x2210	[{<index>}]	-	0	get DAC value
Sensor				
0x21FE	[{<index>}]	{<index><source>}	1	Monitor selection set/get
0x21FD	[{<index>}]	{<index><mode>}	1	Analog I/O configuration set/get
0x2150	-	{<index><voltage>}	1	Sensor auto zero position
0x2116	[{<index>}]	-	0	get monitor voltage
0x2115	[{<index>}]	-	0	get monitor ADC
0x2114	[{<index>}]	-	0	get analog-in voltage
0x2113	[{<index>}]	-	0	get analog-in ADC
0x2112	[{<index>}]	-	0	get filtered sensor value
0x2111	[{<index>}]	-	0	get sensor value
0x2110	[{<index>}]	-	0	get sensor ADC
Motion				
0x20FF	[{<index>}]	-	0	show information of axes
0x2052	[{<index>}]	{<index><acceleration>}	0	Acceleration set/get
0x2051	[{<index><option>}]	{<index><option><value>}	0	Position trigger Step-mode get/set
0x2050	[{<index>}]	{<index><velocity>}	0	Velocity set/get
0x204F			0	Stage status get/reset Check with "?0xFFF9 0x204F" for detail
0x2046	[{<index>}]	{<index>}	0	Home-position set/get
0x2045	[{<index>}]	{<index><PID-error-source>}	0	PID error source selection set/get
0x2044	[{<index>}]	{<index><enable/disable>}	0	Event controlled motion set/get
0x2043	-	[{<index>}]	0	Stop motion
0x2042	[{<index>}]	{<index><enable/disable>}	0	Trajectory controlling set/get
0x2041	[{<index>}]	{<index><source>}	0	target source selection set/get
0x2040	[{<index>}]	{<index><enable/disable>}	0	servo controlling set/get
0x2015	[{<index>}]	-	0	get currently controlling target, in case of trajectory controlled movement
0x2014	[{<index>}]	-	0	get currently controlling voltage, in case of trajectory controlled movement
0x2013	[{<index>}]	-	0	get currently position error

Command - ID	Parameters for reading	Parameters for writing	Level	Description
0x2012	[{<index>}]	-	0	get currently target
0x2011	[{<index>}]	-	0	get overflow status
0x2010	[{<index>}]	-	0	get on-target status
0x2005	-	{<index><target>}	0	set open-loop target relative
0x2004	[{<index>}]	{<index><target>}	0	open-loop target set/get
0x2003	-	{<index><target>}	0	set close-loop target relative
0x2002	[{<index>}]	{<index><target>}	0	close-loop target set/get
0x2001	[{<index>}]	-	0	get position
Controller status				
0x1001		-	0	Get delayed response (for lengthy operation)
0x1000		-	0	pop controller's error

Note: *Depending on the firmware version, the command set may vary. Please use command "? 0xFFFF" to see all valid commands*.

Note: *some commands are only allowed in Command-Level 1. So beware to first change the command level ("0xFFF0 1") before operation.*

For example, the description for command "*pop controller's error*" is:

"**4096[0x1000] 1[0x00000001] 255[0x000000ff] Pop controller's error** "

4096[0x1000]: command ID, Decimal and hexadecimal
1[0x00000001]: options for reading
255[0x000000ff]: options for writing

Byte	Value	Description of options
0	01	Syntax: (please see **Parameters for writing** in the table above) 0xFF = Operation not valid (i.e., command not for writing) 0x01 = No parameters
1	00	Command level: 0 = Normal 1 = Advanced Others = reserved
2	00	Reserved
3	00	Reserved

Please refer to "nF_common.h" for more details.

5.3.3. Parameters

Parameter-ID	Data format	Read-only	Level	Description
System				
0xFF010100	unsigned int	Y	0	RS232 Baudrate
0xFF010013	string		1	Network netmask to set
0xFF010012	string		1	IPv4 address to set
0xFF010005	string	Y	0	IPv6 address readback
0xFF010004	string	Y	0	Network gate-way readback
0xFF010003	string	Y	0	Network netmask readback
0xFF010002	string	Y	0	IPv4 address readback
0xFF010001	int		1	IPv4 address assignment (static=0/DHCP=1)
0xFF010000	string	Y	0	MAC address
0xFF000031	unsigned int	Y	0	Maximal number of wave tables
0xFF000030	unsigned int	Y	0	Memory(DW) for wave/recorder
0xFF000021	string	Y	0	Board name
0xFF000020	string	Y	0	Board setup
0xFF000013	unsigned int		1	Device ID
0xFF000012	string	Y	0	Device name
0xFF000011	string	Y	0	Device serial-number
0xFF000005	unsigned int	Y	0	Number of wave-generators
0xFF000004	unsigned int	Y	0	Number of recorders
0xFF000003	unsigned int	Y	0	Number of piezo-channels
0xFF000002	unsigned int	Y	0	Number of sensor-channels
0xFF000001	unsigned int	Y	0	Number of axes
Filter				
0xC0400904	float		1	User IIR-filter: A2
0xC0400903	float		1	User IIR-filter: A1
0xC0400902	float		1	User IIR-filter: B2
0xC0400901	float		1	User IIR-filter: B1
0xC0400900	float		1	User IIR-filter: B0
0xC0400834	float		1	Notch2 IIR-filter: A2
0xC0400833	float		1	Notch2 IIR-filter: A1
0xC0400832	float		1	Notch2 IIR-filter: B2
0xC0400831	float		1	Notch2 IIR-filter: B1
0xC0400830	float		1	Notch2 IIR-filter: B0
0xC0400821	float		0	Notch2 bandwidth (%)
0xC0400820	float		0	Notch2 -3dB frequency (0.0 = disabled)
0xC0400814	float		1	Notch1 IIR-filter: A2
0xC0400813	float		1	Notch1 IIR-filter: A1
0xC0400812	float		1	Notch1 IIR-filter: B2
0xC0400811	float		1	Notch1 IIR-filter: B1
0xC0400810	float		1	Notch1 IIR-filter: B0

Parameter-ID	Data format	Read-only	Level	Description
0xC0400801	float		0	Notch1 bandwidth (%)
0xC0400800	float		0	Notch1 -3dB frequency (0.0 = disabled)
0xC0400301	float		0	Sensor Notch bandwidth (%)
0xC0400300	float		0	Sensor Notch -3dB frequency (0.0 = disabled)
0xC0400200	unsigned int		0	Length of average-filter
0xC0400114	float		1	IIR-filter: A2
0xC0400113	float		1	IIR-filter: A1
0xC0400112	float		1	IIR-filter: B2
0xC0400111	float		1	IIR-filter: B1
0xC0400110	float		1	IIR-filter: B0
0xC0400101	float		0	IIR-LPF Quality factor
0xC0400100	float		0	IIR-LPF -3dB frequency
0xC0400000	unsigned int		0	low-pass-filter (LPF) types use "***?0xFFF9 0xC0400000***" to see valid settings
Data recorder				
0x40400001	unsigned int		0	recorder data-per-trigger
0x40400000	unsigned int		0	recorder rate
0x40000002	unsigned int	Y	0	Recorder-table-size of group (use 0x4010 to modify)
0x40000001	unsigned int	Y	0	Recorder-table-number in group (use 0x4010 to modify)
Wave generator				
0x31400001			0	wave generator offset
0x31400000			0	wave generator rate
0x30000002		Y	0	Wave-table-size of group (use 0x3010 to modify)
0x30000001		Y	0	Wave-table-number in group (use 0x3010 to modify)
Motion				
0x21400411	float		1	Monitor position-output gain (EBC/D-1203n0 only)
0x21400410	float		1	Monitor position-output offset (EBC/D-1203n0 only)
0x21400201	float		1	Analog input gain
0x21400200	float		1	Analog input offset
0x20400102	float		1	close-loop D-term
0x20400101	float		1	close-loop I-term
0x20400100	float		1	close-loop P-term
0x20400033	float	Y	0	open-loop hard-low-limit
0x20400032	float	Y	0	open-loop hard-high-limit
0x20400031	float	Y	0	close-loop hard-low-limit
0x20400030	float	Y	0	close-loop hard-high-limit
0x20400023	float		1	open-loop soft-low-limit
0x20400022	float		1	open-loop soft-high-limit
0x20400021	float		1	close-loop soft-low-limit
0x20400020	float		1	close-loop soft-high-limit
0x20400011	float		1	on-target setup timing (seconds)
0x20400010	float		1	on-target tolerance

Parameter-ID	Data format	Read-only	Level	Description
0x20400002	float		1	maximal velocity (unit/s)
0x20400001	float		1	maximal acceleration (unit/s^2)
0x20400000	int		1	Trajectory controlling (OFF=0 / ON=1)
0x20000002	string		1	axis unit
0x20000001	string		1	axis name
IDChip parameters				
0x21000005	string	Y	0	Stage serial number
0x21000004	string	Y	0	Stage production date
0x21000003	string	Y	0	Stage type
0x21000002	int	Y	0	Stage configuration
0x21000001	string	Y	0	Stage manufacture

Note: *Depends on firmware version, the parameter-list may vary. Please use command "? 0xFFFE" to see a detailed description of valid parameters.*

Note: *some parameters are only allowed in Command-Level 1. So beware to first change the command level ("0xFFF0 1") before operation.*

Note: *some parameters are also accessible with command, example for velocity: parameter-ID 0x20400002 and command-ID 0x2050 are identical.*

For example, the description for parameter "*Stage serial number*" is:
"**553648132[0x21000004] 2097153[0x00200001] 4[0x00000004] 260[0x00000104] Stage serial number** "

553648132[0x21000004]: parameter ID, Decimal and hexadecimal
2097153[0x00200001]:
0x0020 (the high 16bits) is reserved.
0x0001(the low 16bits) is number of channels/axes (i.e. 1 channel, valid index is 0)

4[0x00000004]: options for reading
260[0x00000104]: options for writing

Byte	Value	Description of options
0	04	Syntax: (please see **Parameters for writing** in the table above) 0x00 = Char 0x01 = Unsigned int32 0x02 = Float 0x04 = String please see nF_common.h for detail
1	00	Command level: 0 = Normal 1 = Advanced Others = reserved
2	00	Reserved
3	00	Reserved

Example to read the current active parameter (from RAM),
? 0x6001 0 0x21000004

Example to write to flash (non-volatile, takes into active by next reboot)

0x6002 0 0x21000004 '0000'

Example to backup all parameters to flash:

0x6003 100

5.4. Operation

5.4.1. Position Commanding and Reading

The controller is ready in about 10~15 seconds after powered-on. The default state is Servo-Off (open-loop) and 0V output (this behavior can be changed by default macro).

In order to move the stage to a desired position: (e.g. 1st Axes to position 1.0)

1) Set Servo-On

   Command-ID: 0x2040

   1st Parameter: axis index = 0 (index from 0, for 1st axis it should be 0, 2nd be 1, and so on)

   2nd Parameter: value = 1 (1=Servo-On, 0=Servo-Off)

   For simplification, it is listed as:

   0x2040 0 1

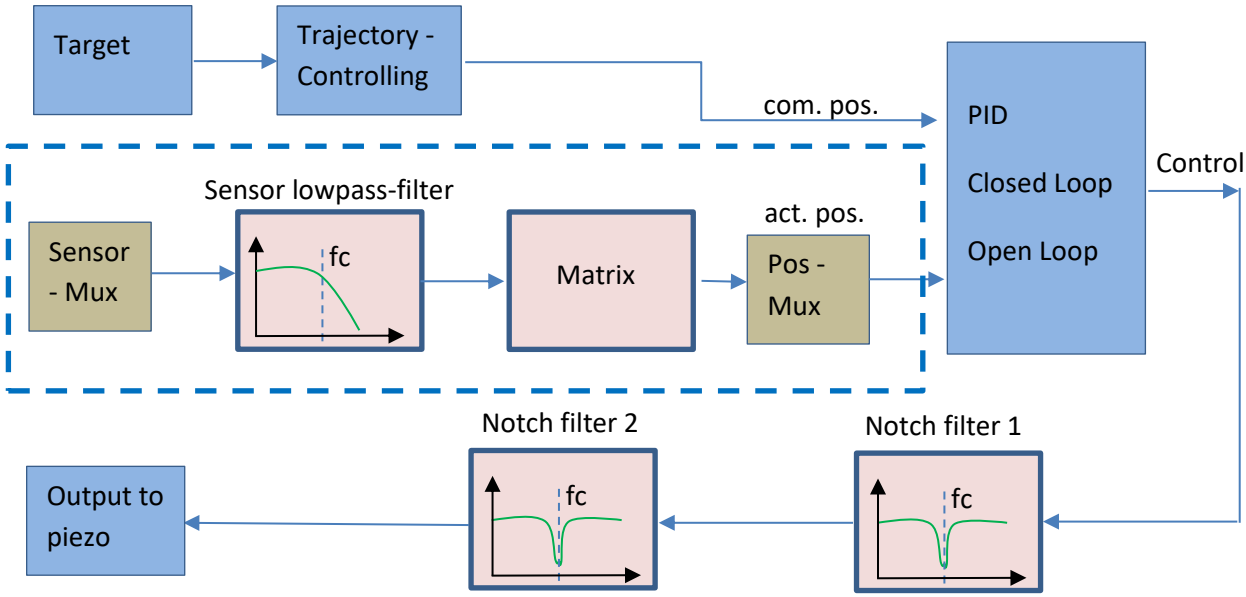
2) Set target to 1.0:

   0x2002 0 1.0

3) Check position:

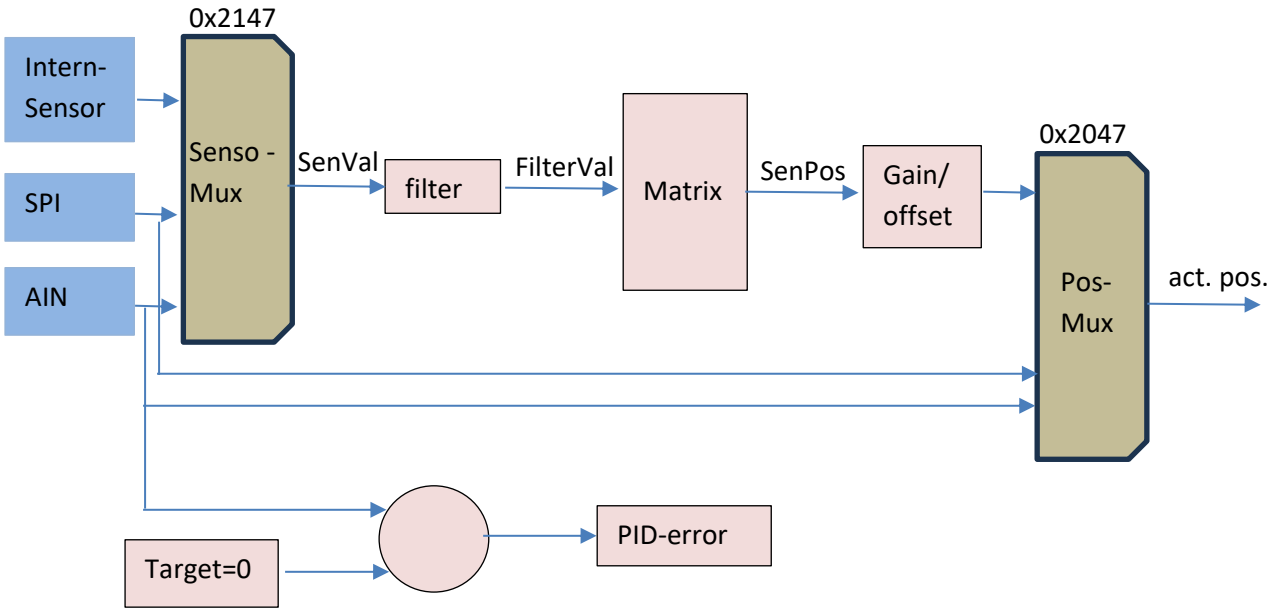
   ?0x2001 0 // read-back flag "?" is for command-terminal only

5.4.2. Function block diagram



Note: The sensor has one low-pass-filter (and value is converted to position with Matrix). The controlling voltage has 2 notch-filters.
Set *fc* to 0.0 to disable filter function.

Detail about Sensor/Position-Mux:



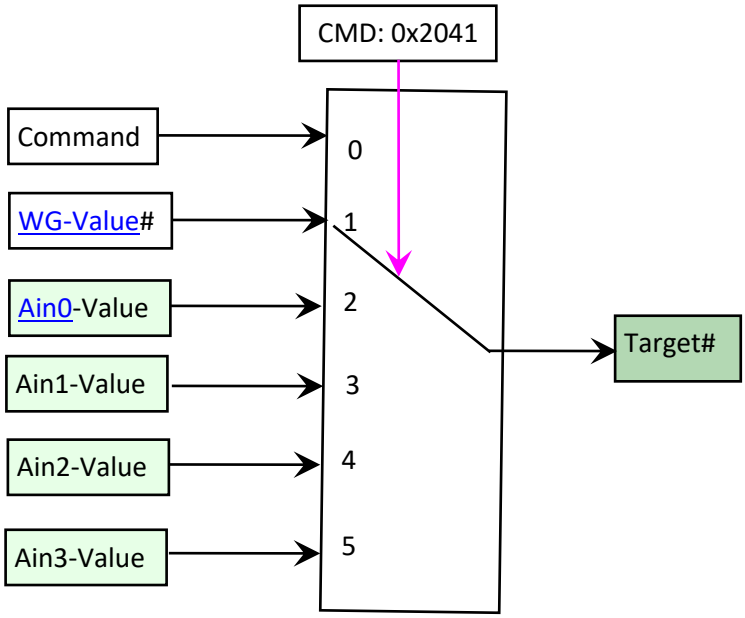
5.4.3. Target selection

As shown in the figure, targets can be selected from different sources for each axis.

For closed-loop (Servo On): The target is a position value.

For open-loop (Servo Off): The target is a voltage value.

Target sources include: command, wave-generator(WGO), or analog input. At start-up, the target is default to 'command'. (ID: 0x2002/0x2003 for closed-loop; 0x2004/0x2005 for open-loop).



For example, to connect the 2nd axis target to the 2nd analog-input (Ain1):

```
0x2041 1 3        // 3 stands for Ain1-Value
```

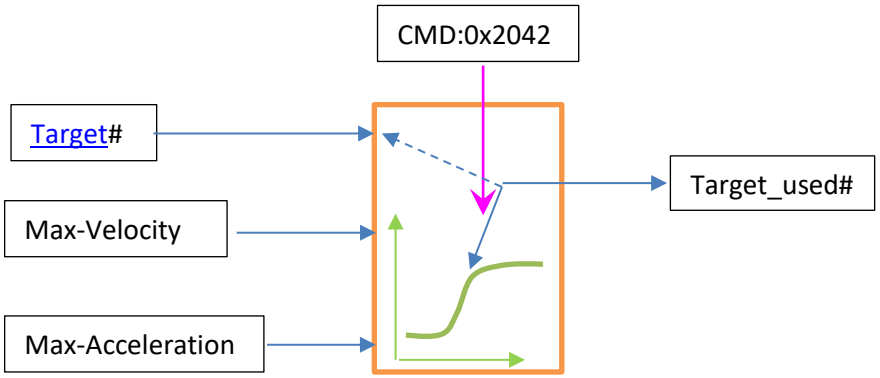
To select the wave-generator for the 1st axis:

```
0x2041 0  1       // 1=option for wave generator
```

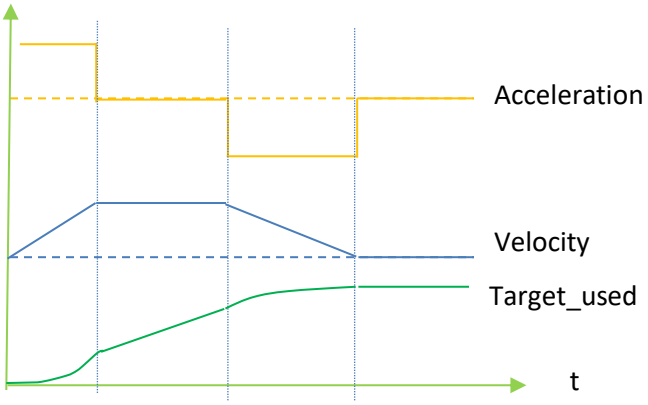
Note: please use "***?0xFFF9 0x2041***" to see all valid settings

5.4.4. Trajectory controlling

The selected target will be passed to trajectory module, which calculates the effective value used as target (target_used#) according to the maximum velocity and acceleration parameters.



The relation between target used and velocity / acceleration is shown below.



Example for using the trajectory module for the 1st axis

Set max-acceleration to e.g. 10.0um/(s*s). Optional, since parameter can be saved in flash.

0x6001 0 0x20400001 10.0 or with command

0x2052 0 10.0

Set max-velocity to 1um/s. Optional, since parameter can be saved in flash.

0x6001 0 0x20400002 1.0 or with command

0x2050 0 1.0

Enable trajectory controlling. Optional, since parameter can be saved in flash.

0x6001 0 0x20400000 1 or with command

0x2042 0 1

With the settings above, the speed-up(velocity from 0.0 to 1.0) and speed-down(velocity from 1.0 to 0.0) time is 0.1s.

Note:

When wave-generator is enabled, trajectory controlling should be disabled (otherwise the waveform may be deformed).

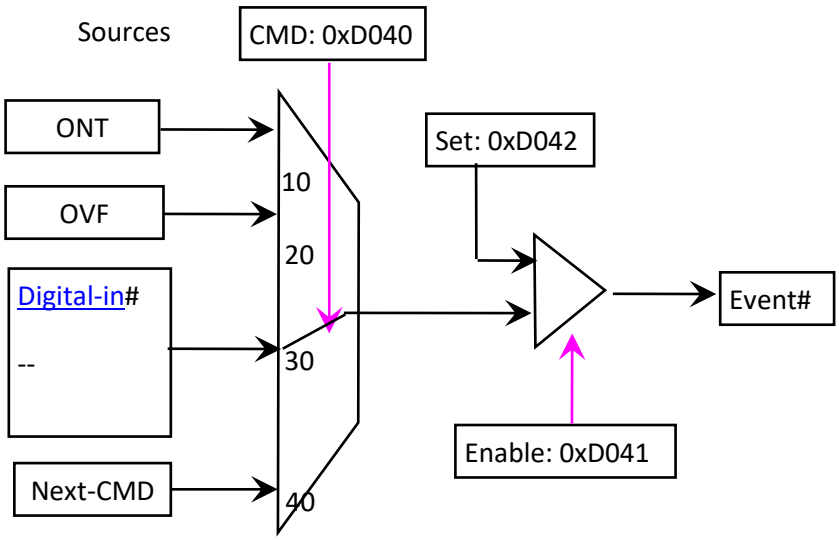
0x6001 <axis-ID> 0x20400000 0 or with command

0x2042 <axis-ID> 0

5.4.5. Event

Event is used for triggering data-recorders and/or wave-generators (synchronizing the operation) and movement. Up to 4 event signals (index 0~3) are available. To use an event, it should be first configured and then connected to data-recorders and/or wave-generators.

Event is set from different resources, such as internal status, digital-input or set directly.



Example: Event0 from the 2nd digital input. Event1 from "on-target" of both axis1 and axis2.

```
0xD040 0 30 1              // 30: digital in
0xD040 1 10 0  1 10 1      // 10: ONT
0xD041  0  1   1   1       // 1: enable
```

Note: please use "***?0xFFF9 0xD040***" to see all valid settings

5.4.6. Memory

Data-recorder- / wave-tables share the system memory. For example, if the total available memory size is 1 Mega (please check parameter 0xFF000030) and data recorder takes 512k, then wave table can use the rest 512k.
For maximal flexibility, both recorder-/wave-tables have two groups. Each group may have different numbers of tables. Tables inside one group have the same size.
For example:

Recorder-group0: 3 tables, each size 32k
Recorder-group1: 4 tables, each size 8k
Command: 0x4010 3 32768 4 8192

Wave-group0: 4 tables, each size 32k
Wave-group1: 0 tables
Command: 0x3010 4 32768 0 0

Note: The size means number of 32bits double-word (DW)

Total memory size can be read by parameter: 0xFF000030
Parameter ID for maximum number of wave tables: 0xFF000031
Parameter ID for maximum number of recorder tables: 0xFF000004

5.4.7. Wave-Generator

Wave table: Filled with waveform
0x3001 <WaveTableId> <mode> <waveform> <x0> <segLength> <waveform-parameters>

WaveTableId:

Wave table ID, unsigned int, index begins from 0, i.e. 2 means the 3rd wave-table.

mode:

0=new data,
1=add to old data

Waveform:

0: data point, software sets wave-table directly
1: sine
2: step
3: square
4: ramp
5: triangle
(Note: use "***?0xFFF9 0x3001***" to see all valid settings)

x0:

wave table data index, starts from 0

segLength:

total length of current segment

Waveform-parameters:

parameters for different waveforms may differ.

for data point (option: 0): float data to wave table.

E.g. set table2 with 5 data (0.0, 1.0, 2.0, 1.0, 0.0) , the command should be:

0x3001 2 0 0 0 5 0.0 1.0 2.0 1.0 0.0

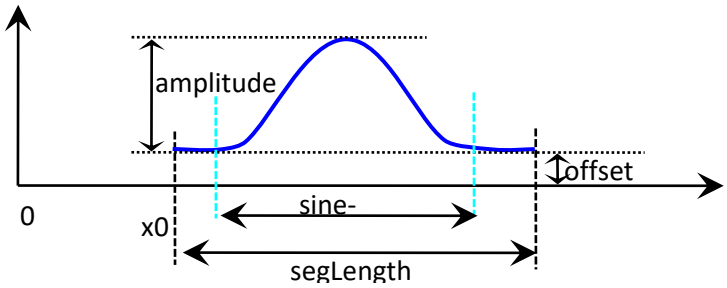
Or send 3 more data to table position 5~7

0x3001 2 0 0 5 3 1.0 1.5 1.0

Readback to verify: (table2, from 0, number-of-data is 8)

?0x3011 2 0 8

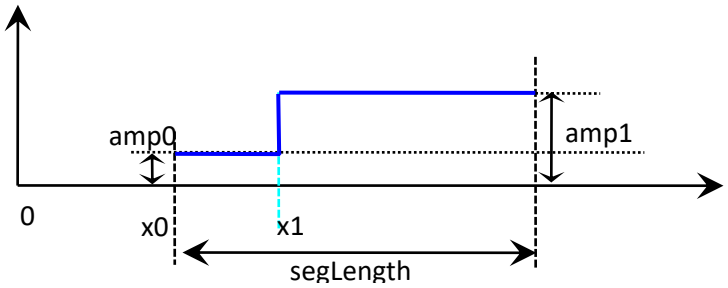
for sine: <amplitude> [<sine-length> <offset><phase>]



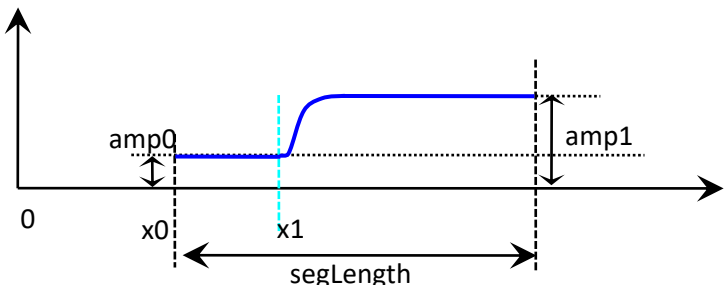
Example: set table2 with a new sine waveform, length = 5000 points, amplitude = 20.0, offset=0.0, phase = 90.0°,

0x3001 2 0 1 0 5000 20.0 5000 0.0 90.0

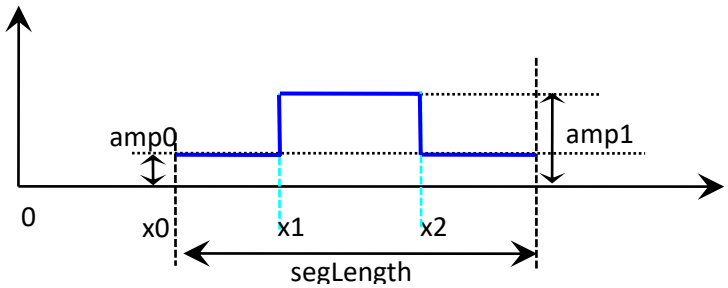
for step: <amplitude0> <x1> <amplitude1> [<max-acceleration><max-velocity>]



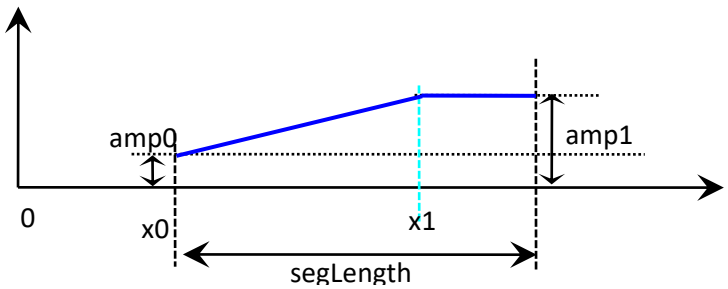
with max-acceleration/velocity controlling



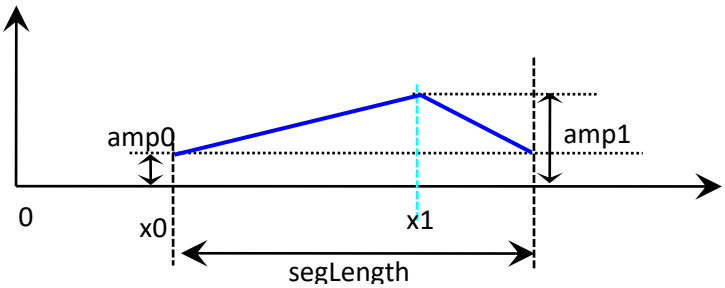
for square: <amplitude0> <x1> <amplitude1> <x2> [<max-acceleration><max-velocity>]



for ramp: <amplitude0> <x1> <amplitude1> [<max-acceleration>]



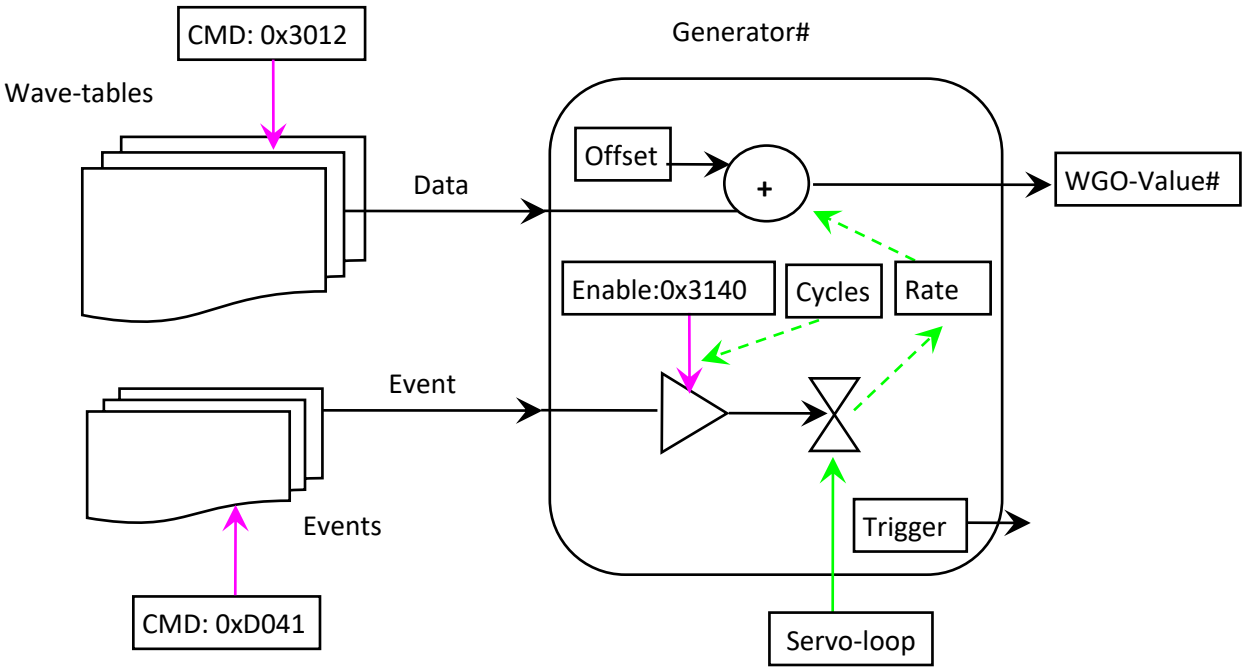
for triangle: <amplitude0> <x1> <amplitude1> [<max-acceleration>]



Note:

When the acceleration value is too small to create the waveform, command returns with error.

Wave-generator output(WGO): controller has up to 3 wave-generators, index 0~2.



As shown in 5.3.2, output of wave-generator# can be used as the target for axis-#, i.e. wave-generator0 to axis0, wave-generator1 to axis1, and so on.

To set the rate:

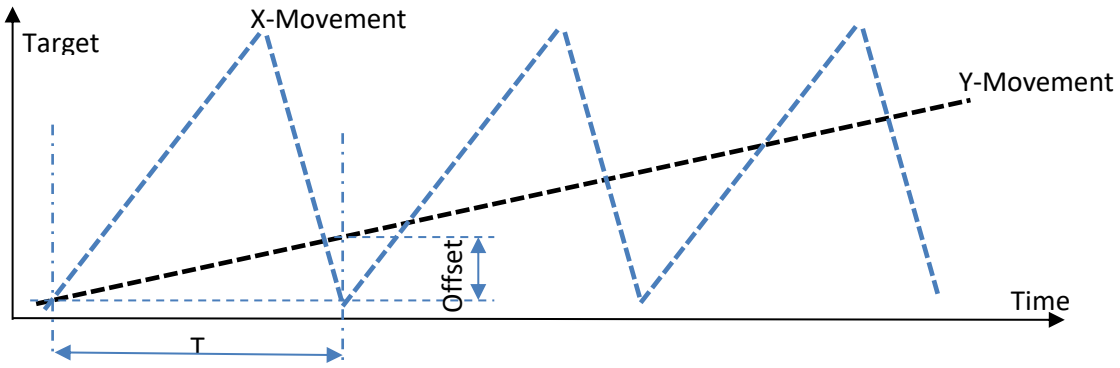
0x6001 <WGO-ID> 0x31400000 <rate> (e.g. 0x6001 0 0x31400000 10) or
0x3141 <WGO-ID> <rate> (e.g. 0x3141 0 10)

To set the Offset: 0x6001 <WGO-ID> 0x31400001 <0ffset> (e.g.: 0x6001 0 0x31400001 10.0)

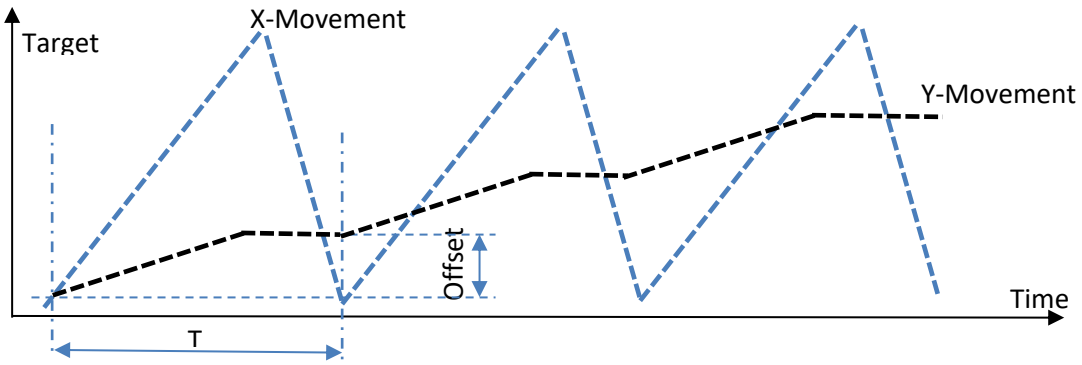
To set the cycles: 0x3142 <WGO-ID> <cycles> (e.g. 0x3142 0 3)

Note:
Parameter “Rate” means “takes the next wave-table data after N servo-loops”.
Parameter “Offset” (parameter-ID=0x31400001) is useful for scanning movement. As shown in the following examples, X/Y can be defined with same period T, and Y has offset after each cycle.

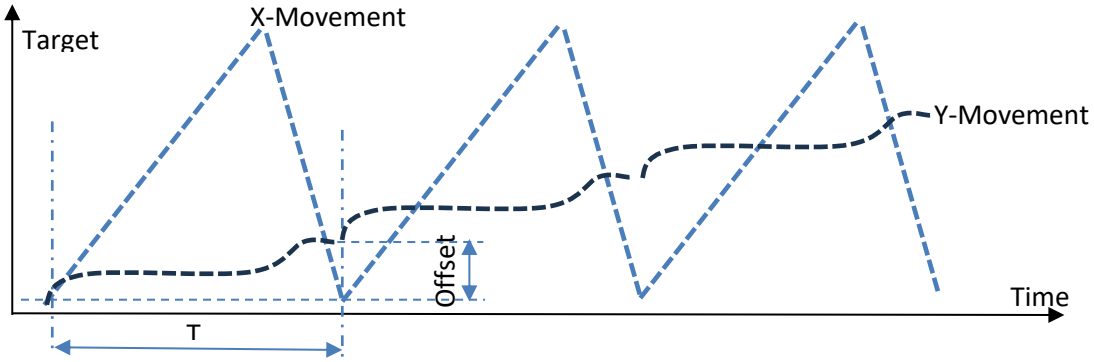
Example 1:



Example 2:



Example 3:



To use wave-generator: (for programming, please see the C/C++ demo code)

1) fill wave table:

   0x3001 2 0 1 0 5000 10
   (Table 2, 1=sine-waveform, 5000 points, amplitude 10)

*Note: Time-duration = segLength * Rate * Servo-time (10us or 20us, see parameter 0xFF00000F)*

2) axis target selection:

   0x2041 0 1

3) connect generator to wave-table:

   0x3012 0 2 // Table 2 to generator 0

4) select event for starting generator.

   0xd041 1 0 // optional: disable event when its current states not known
   0x3151 0 1 // generator 0 to event 1

5) enable generator 0.

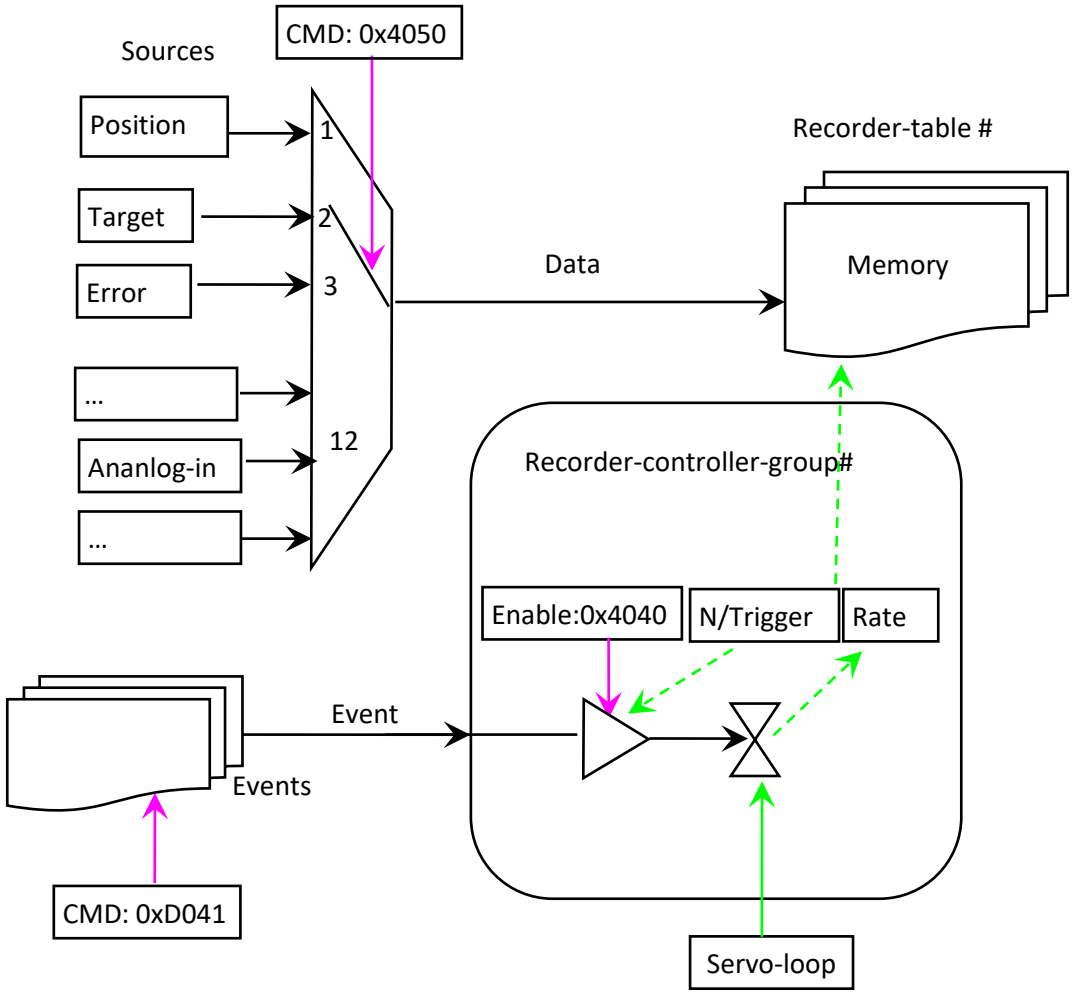
   0x3140 0 1

6) enable and set event 1.

   0xd041 1 1 // enable event 1
   0xd042 1 1 // set event 1, or from another source

Wave generator is now started.

5.4.8. Data-Recorder



There are 16 data recorders to store data synchronously in memory which can then be read-back by software. Recorders can be controlled by one of the two Recorder-controller-groups. All recorders in one group have the same table-rate and memory size.

To use data recorders:

1) configure the recorders to controller-group (optional). As an example: recorders with index 0~3 (each has 100000 data points) to group 0, and 4~5 to group 1 (8192 data points):
   0x4010 4 100000 2 8192

2) select data resource with command 0x4050. Command parameters are:
   0x4050 <recorder-index> <resource-option> <resource-channel>
   For example: To select recorder 0 with 1st axis position, and recorder 1 with 3rd analog input
   0x4050 0 1 0 1 12 2
   Note: please use "***?0xFFF9 0x4050***" to see all valid settings

3) select event for starting recorder-group.
   0xd041 1 0 // optional: disable event when its current states not known
   0x4051 0 1 // group 0 to event 1

4) enable recorder group 0.

```
0x4040 0 1
```

5) enable and set event 1.

```
0xd041 1 1          // enable event 1
0xd042 1 1          // set event 1
```

Now data will be saved at each servo-loop (for rate = 1).
In case of trigging by “next-command”:

```
0xd042 1 0          // clear last trigger status
0xd041 1 1          // enable event 1
0xd040 1 40 0       // triggered by next command
```

6) check recorded data length.

```
?0x4042 0           // Read-back
```

*Note: Time-duration = Table_size * Rate * Servo-time (10us or 20us, see parameter 0xFF00000F)*
In case of Servo-time=10us and rate=1, 8192 data needs 81.92ms.

7) when finished, read data.

```
?0x4011 0 0 1024        // Read 1024 data out of table 0, from position 0
?0x4011 0 1024 1024     // Read 1024 data out of table 0, from position 1024
…                       // Read more data out of table 0
?0x4011 1 8100 92       // Read 92 data out of table 1, from position 8100
```

5.4.9. Analog I/O

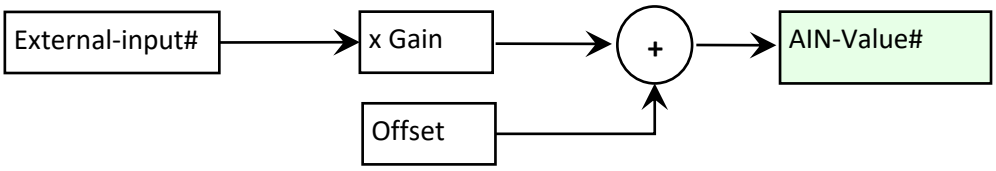
- Analog-input (AIN)

AIN can be used as target signal so that no software controlling is necessary (a default macro is required for configuration).

Table 5.4.5 : Description of analog inputs

Controller	Number of inputs	Index
EBC/EBD-120310	2	0, 3
EBC/EBD-120320	3	0, 1, 3
EBC/EBD-120330	4	0, 1, 2, 3

Calculating method:



For **EBC/EBD-120310** the upper BNC can be either configured as 4th input (default at power-on), or output.

To configure it as output:

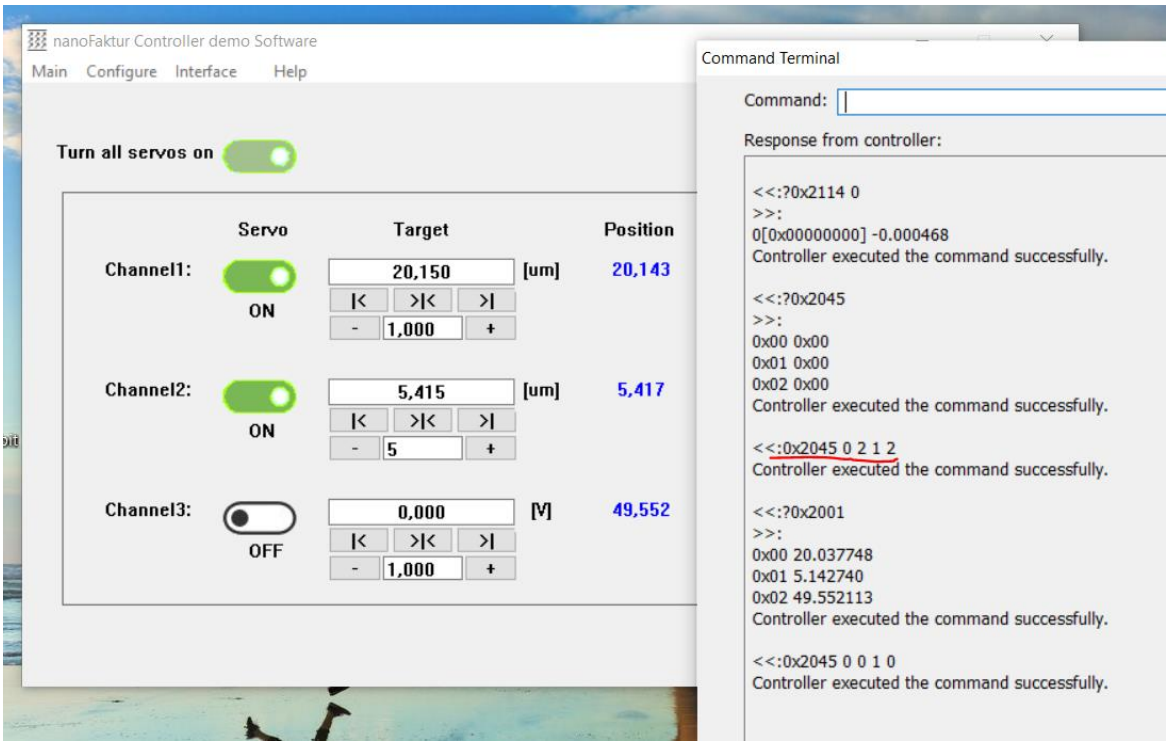
```
0x21FD  0  1          // 1=output, 0=input
```

AIN can also be used as PID-error signal: in case of using an external feedback signal for keeping at one special position (target is set to 0.0).

Command: 0x2045 <axis> <Ain-ID>
Note: check all options with ?0xFFF9 0x2045

Example:

```
0x2040 0 1            // Servo-on
0x2002 0  20.0        // Set target to 20um
?0x2114  0            // Optional: check input value of analog#1
0x2045 0  2           // PID-error signal from the 1st AIN (3=2nd AIN, 4=3rd AIN, etc.)
...
0x2045 0  0           // PID-error signal is set again to normal sensor signal
```



Note:

Use the Analog-input-gain to set the signal sensitivity and direction (-1.0 to invert the error signal).

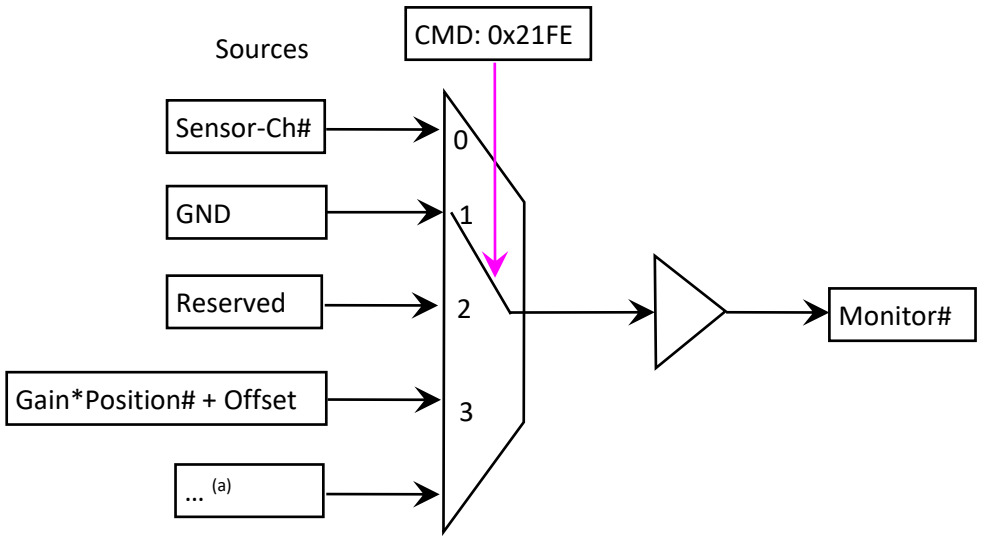
Parameters

Parameter group: Sensor Action: Flash => File Run

Description	Channel	Parameter-ID	Level	DataType	Operation	RAM-value	Flash-value
Stage travel range high	2	0x21000120	Service	Float	Read-on...	100,0000	-
Stage travel range high	3	0x21000120	Service	Float	Read-on...	1000,0000	-
Stage travel range low	1	0x21000121	Service	Float	Read-on...	0,0000	-
Stage travel range low	2	0x21000121	Service	Float	Read-on...	0,0000	-
Stage travel range low	3	0x21000121	Service	Float	Read-on...	0,0000	-
Analog input offset	1	0x21400200	Advan...	Float	Writable	0,0000	0,0000
Analog input offset	2	0x21400200	Advan...	Float	Writable	0,0000	0,0000
Analog input offset	3	0x21400200	Advan...	Float	Writable	0,0000	0,0000
Analog input offset	4	0x21400200	Advan...	Float	Writable	0,0000	0,0000
Analog input gain	1	0x21400201	Advan...	Float	Writable	1,0000	1,0000
Analog input gain	2	0x21400201	Advan...	Float	Writable	1,0000	1,0000
Analog input gain	3	0x21400201	Advan...	Float	Writable	1,0000	1,0000
Analog input gain	4	0x21400201	Advan...	Float	Writable	1,0000	1,0000
Monitor input offset	1	0x21400300	Advan...	Float	Writable	0,0000	0,0000
Monitor input offset	2	0x21400300	Advan...	Float	Writable	0,0000	0,0000
Monitor input offset	3	0x21400300	Advan...	Float	Writable	0,0000	0,0000

➢ Analog-Output

The analog output can be used as a monitor-signal.



Note (a): please use "***?0xFFF9 0x21FE***" to see all valid options

Option <3> for the EBC-120*/EBD-1203* controllers only:

Output-range: -5V ~ +10V

Output-Voltage = Position * Gain + Offset

For example of axis0:

Position: 0~25um

Output voltage: -5.0 ~ 5.0V

⇨ Gain = 0.4, Offset = -5.0

Command (or with Parameter-GUI):

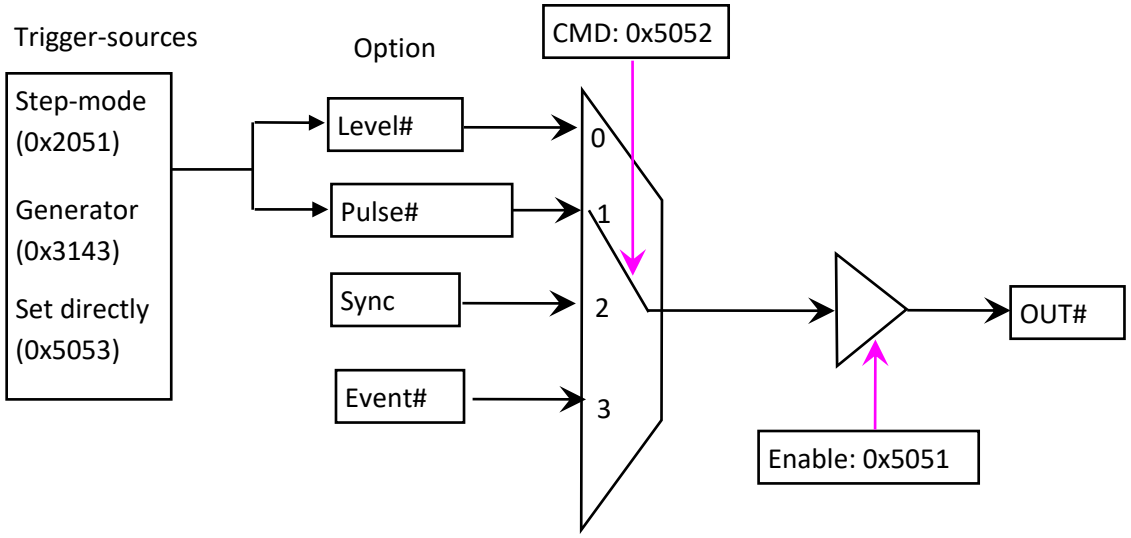
0x6001 0 0x21400410 -5.0 0 0x21400411 0.4

5.4.10. Digital I/O

As shown in Chapter 4.3.2, there are 3 input-only TTL signals, 3 output-only TTL signals and 4 differential signals which can be configured as inputs or outputs.

The input-only signals are 5V tolerant. And the level of output signals can be set to 3.3V or 5V with a configuration pin.

Output signals can be selected from trigger (including step-mode, wave-generator or set directly), system synchronization signal or event. The output should be enabled with command 0x5051.



Example: to show Event on OUT1

```
0x5051 0  1      // 0: index of the first output; 1: output enable, 0=disable
0x5052 0  3      // 0: index of the first output; 3: select Event0
```

Note:

- use "***?0xFFF9 0x5052***" to see valid settings
- Level: signal from Trigger sources directly (Priority: first Step-mode, then Generator and the last "Set" mode)
- Pulse: when Trigger sources set the signal, output will be changed to a pulse signal, the pulse length can be set with command 0x5050 <n*10us>, e.g.

```
0x5050 4         // for 40us (4*10us) pulse length,
```

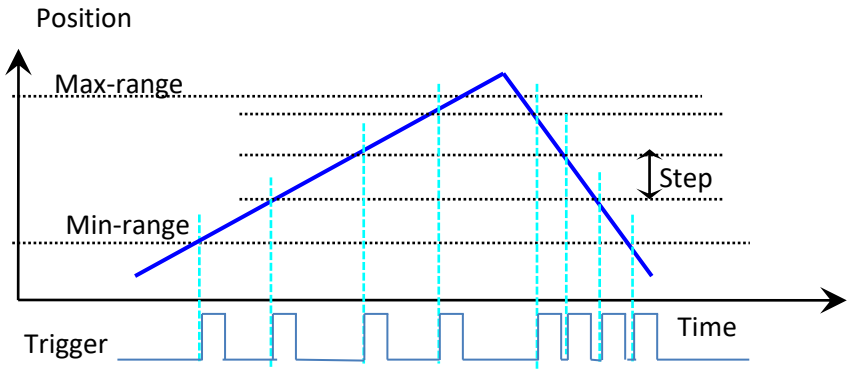
Table 5.4.10 : Description of digital output signals

Pin	Description	Type	Index
1 - OUT1	Digital-output only #1	TTL	0
2 – OUT2	Digital-output only #2	TTL	1
3 – OUT3	Digital-output only #3	TTL	2
6/15 – DIO1	Digital-in/out #1	LVDS	3(1)
7/16 – DIO2	Digital-in/out #2	LVDS	4(1)
8/17 – DIO3	Digital-in/out #3	LVDS	5(1)
9/18 – DIO4	Digital-in/out #4	LVDS	6(1)

Note: (1) FPGA firmware up from 2.00

About Trigger sources

(1) Step-mode:



Output trigger signal for each movement-step,

Example: to configure the trigger signal in range of 1~10um and step size of 0.5um

```
0x2051  0  0   0.5        // Step size
0x2051  0  1   1.0        // range from 1.0um
0x2051  0  2   10.0       // range to 10.0 um
0x2051  0  3   0          // mode = 0: both-side
```

Note:

- Command syntax: 0x2051 <axis> <option> <value>
- Check all valid options with ?0xFFF9 0x2051
- unit depends on stage, can be um or mrad and etc.
- The movement can be realized by any mode, like wave-generator or speed controlling method and so on.
- Use 0x5051 to enable/disable trigger-output
- Comparing source can be selected between real-position or target with command 0x2051 <axis> 4 <source>. For real-position (source=0), PID-tuning is required for a fast moving speed.
- For reliable trigger output, the axis motion should extended to (Min_Range – 0.5*Step_Size) ~ (Max_Range + 0.5*Step_Size).

Below is an example for trigger output in speed-controlled movement (step:0.5, range:1.0~10.0)

```
0x2041  0   0              // axis0 target from command
0x2040  0   1              // axis0 servo on
0x2002  0   0.0            // move to 0.0
0x2042  0   1              // axis0 speed controlling enabled
0x2050  0   1.0            // axis0 speed: 1.0 um/s
0x5051  0   1              // enable trigger0 output
0x5052  0   0              // trigger mode: level
0x2051  0   0  0.5 0  1  1.0  0  2 10.0  // configure the trigger signal
0x2003  0   10.5           // relative movement: to 10.5um
```

Check: Trigger signal on PIN1 of DIO-connector

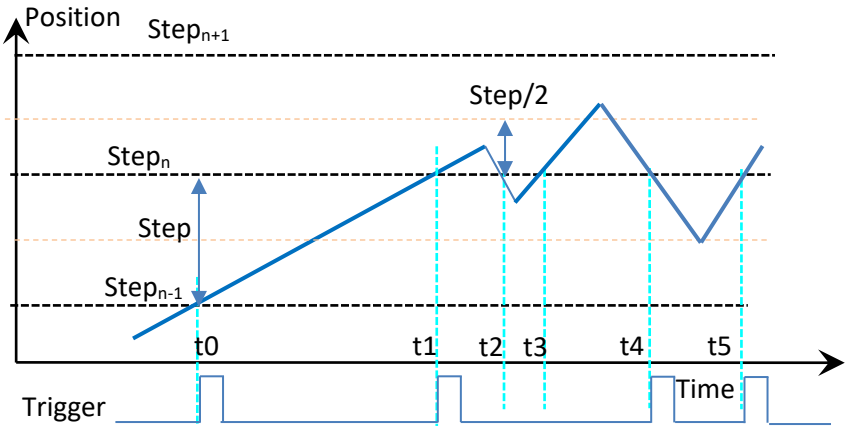
```
0x2003 0  -10.5            // move back
```

Note: the moving range requirement (Min_Range – 0.5*Step_Size) ~ (Max_Range + 0.5*Step_Size)

Note: to use DIO#1 as trigger output

```
0x5051 3 1                 // 3: ID of DIO#1
0x5052 3 1                 // pulse mode
0x2051 0 5 3               // trigger output to port (3=DIO#1)
```

About multi-trigger at one position:



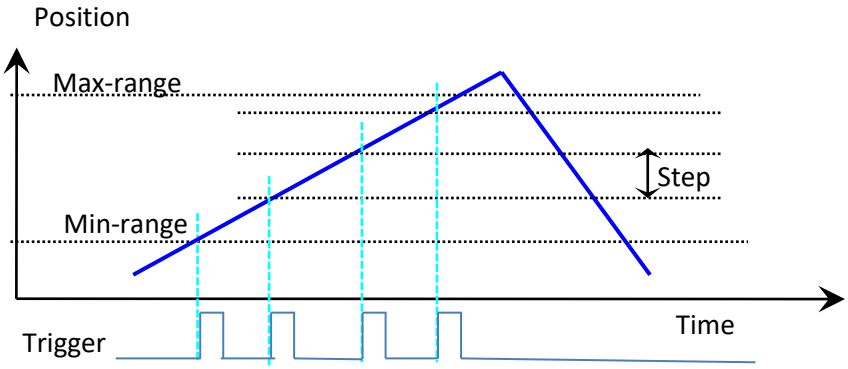
As shown above, there are pulses at time t0 and t1, but no pulse at t2 and t3 since position difference to the last step (Step$_n$) is less than Step/2. At position t4 and t5 there are output.

For one-side output:

0x2051 <axis> 3 1 // option: 3 = mode; value: 1 = one-side

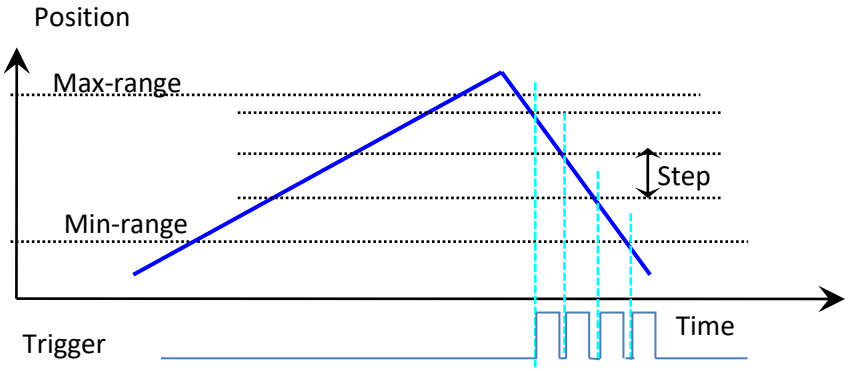
Example, output trigger only when moving from 50 to 60 um

0x2051 0 0 1.0 0 1 50.0 0 2 60.0 0 3 1



Example, output trigger only when moving from 60 to 50 um

0x2051 0 0 1.0 0 1 60.0 0 2 50.0 0 3 1

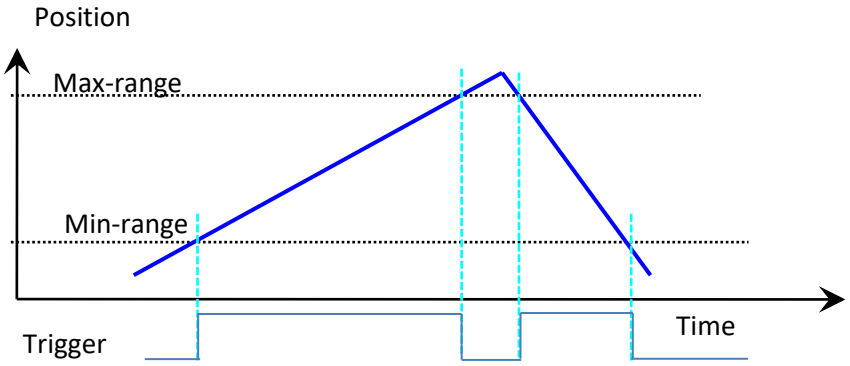


For threshold output:

0x2051 <axis> 3 2 // option: 3 = mode, value: 2 = threshold

Example, output trigger when moving from 50 to 60 um

0x2051 0 1 50.0 0 2 60.0 0 3 2

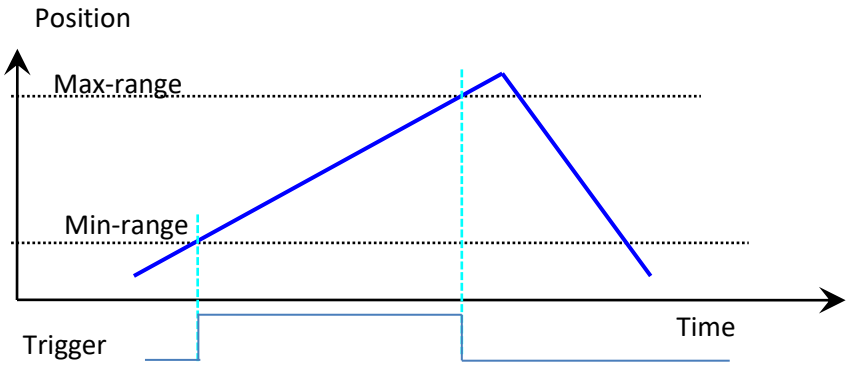


For one-side threshold output:

0x2051 <axis> 3 3 // option: 3 = mode, value: 3 = one-side threshold

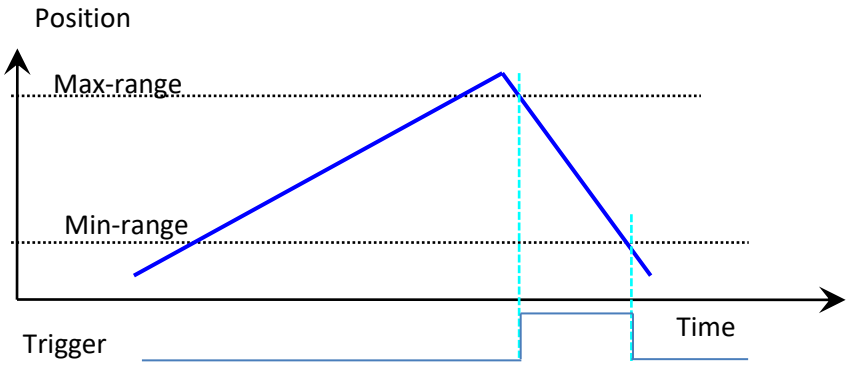
Example, output trigger when moving from 50 to 60 um

0x2051 0 1 50.0 0 2 60.0 0 3 3



Example, output trigger when moving from 60 to 50 um

0x2051 0 1 60.0 0 2 50.0 0 3 3



(2) Generator-mode:

Wave generator has 3 options for trigger signal, include “each point”, “end of frame” and “trigger table”. Use command 0x3143 to select which method, e.g.

```
0x3143  0   3          // generator 0 output trigger according to “trigger table”
0x3143  2   1          // generator 2 output trigger at each point
0x3143  1   2          // generator 1 output trigger at end of frame
```

Note:

Trigger signal lasts for one-servo-update time (normally 10us). In “each point” mode, if wave-generator-rate is 1 (i.e. data updated at each servo-loop), the trigger signal will be set all the time.

For using the trigger table:

```
0x3100  0              // clear trigger table 0
0x3101  0   1   1   0   100   1     // Output trigger at point 1 and point 100
?0x3111  0  0  128     // Read back to verify

0x2051  0  0   0.0     // set step size to 0.0 to disable the step-mode
0x3143  0   3          // select trigger mode for generator 0

0x5051  0   1          // enable trigger0 output
0x5052  0   0          // trigger mode: level

<Start the first generator with nFControl.exe>

// Check: Trigger signal on PIN1 of DIO-connector
```

Note:

Use 0x3145 to select the digital-output port. Example, to select DIO#1 as trigger output

```
0x5051 3  1            // enable DIO#1 output
0x5052 3 0             // mode: level
0x3145 0 3             // trigger of wave-generator#1 to DIO#1
```

(3) Set directly

This mode has the lowest priority, i.e. only when wave generator is not running and Step-mode disabled.

To set trigger:

```
0x5053  0   1          // output ‘1’ on PIN1
0x5053  0   0          // output ‘0’ on PIN1
```

5.4.11. Macro function

In case that the controller must be used without interface/software, the initialization should be performed automatically with saved commands. This comes to the macro function.

Table 5.4.12 Command for Macro function

Command ID	Syntax (read/write)	Function description	Comment
0xE000	-/ <string: name> write only	Start recording a new macro	The old macro will be replaced when a same name is given. Only saves command from one interface.
0xE001	-/ <string: name> write only	Append command to an existing macro	
0xE002	-/ <none> write only	End of recording	
0xE003	<none> / - read only	List all macros	
0xE004	-/ <string> read only	Show contents of macros	Command package in binary format
0xE005	<none> / <string>	Set/get default macro	Controller starts default macro each time it re-started
0xE006	-/ <string> write only	Delete macro	
0xE010	<none> /<string>[<times>]	Start macro / get running status	<times> is optional, 1 when not set.
0xE011	-/ <none> Write only	Stop running macro	
0xE012	-/ <ms> Write only	Delay millisecond	Only used in macro function

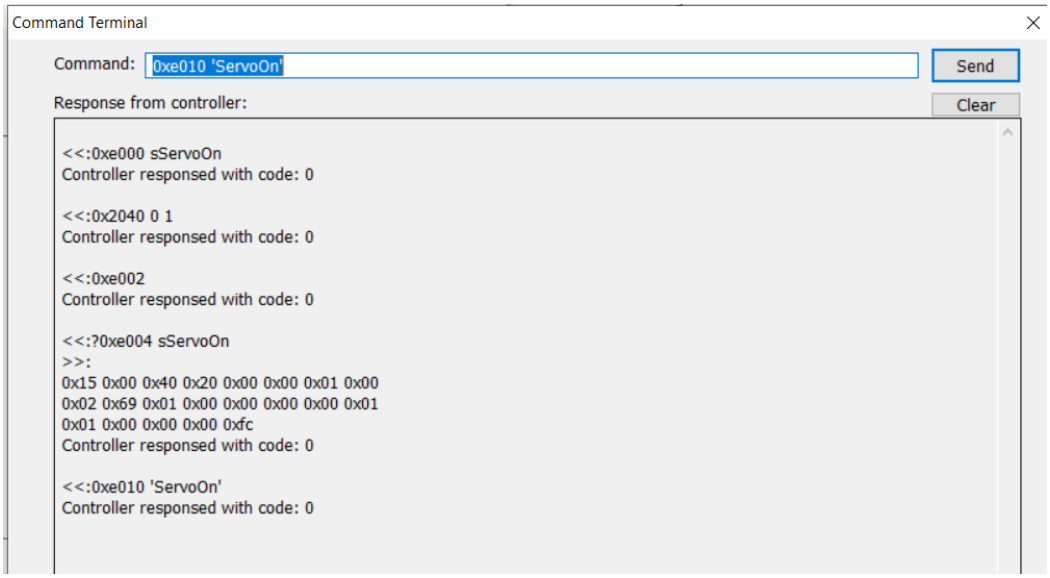
Note: Name are case sensitive, maximal 8 chars.

Example 1: define a macro which turns servo-on

```
0xE000  sServoOn        // macro name is 'ServoOn'
                        //  's' is only for software which means parameter is a string
                        // or just  0xE000  'ServoOn'
0x2040  0   1           // servo on
0xE002                  // end of recording macro

?0xE003                 // show all macros
?0xE004   'ServoOn'     // show all commands in macro

0xE010   'ServoOn'      // start macro : just to test
```

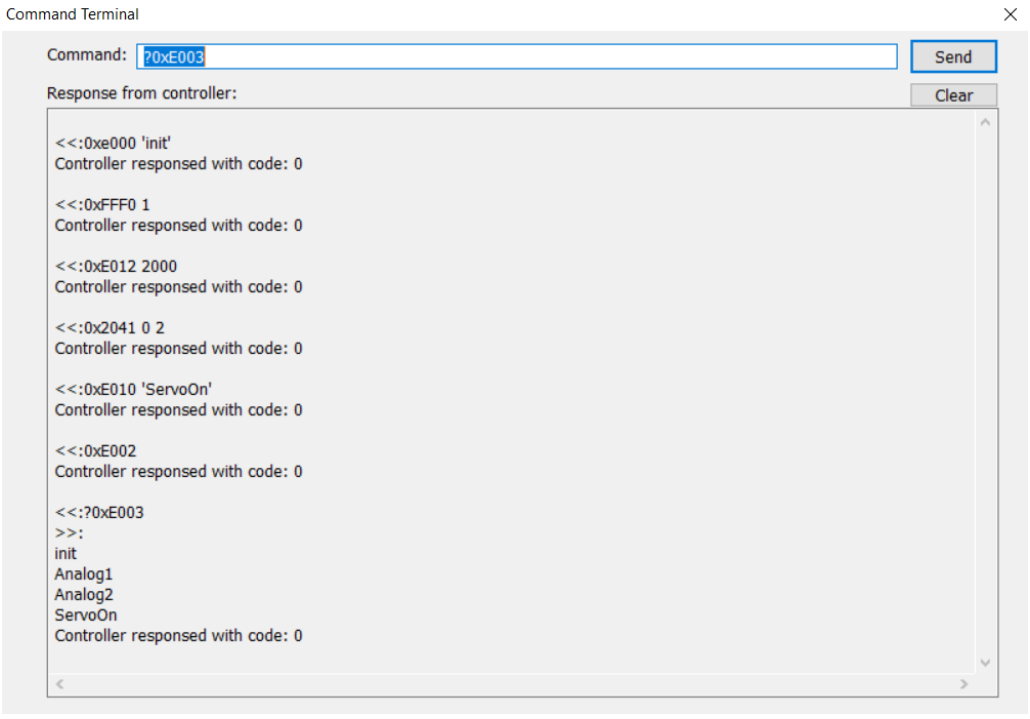


Example 2: enable analog controlling

```
0xE000  'init'          // macro name is 'init'
0xFFF0  1               // Command level to 1
0xE012  2000            // Delay 2s
0x2041  0  2            // select the 1st analog input as target of axis0
0xE010  'ServoOn'       // run macro, or just "0x2040 0  1"
0xE002                  // end of recording macro

?0xE003                 // show all macros
0xE010  'init'          // start macro : just to test

0xE005  'init'          // set as default macro
```




```
0xE005                  // without macro name, the default macro will be disabled
```

Example 3: Step movement

Move from 0 to 10um in step of 0.1, and then back to 0 in step of 0.2um

```
0xE000  'stepUp'          // macro for step-up
0x2003  0   0.1           // move axis0 relative (0.1um)
0xE012  10                // Delay 10ms
0xE002                    // end of recording macro

0xE000  'stepDown'        // macro for step-down
0x2003  0   -0.2          // move axis0 relative (0.2um)
0xE012  10                // Delay 10ms
0xE002                    // end of recording macro

0xE000  'Main'            // top-level macro
0x2040 0  1               // axis0 servo ON
0x2002 0  0.0             // move to position 0.0
0xE012  10                // Delay 10ms
0xE010 'stepUp'  100      // step up 100 times
0xE012  10                // Delay 10ms
0xE010 'stepDown'  50     // step down 50 times
0xE002
```

Run the macro:

```
0xE010 'Main'
```

5.4.12. Firmware update

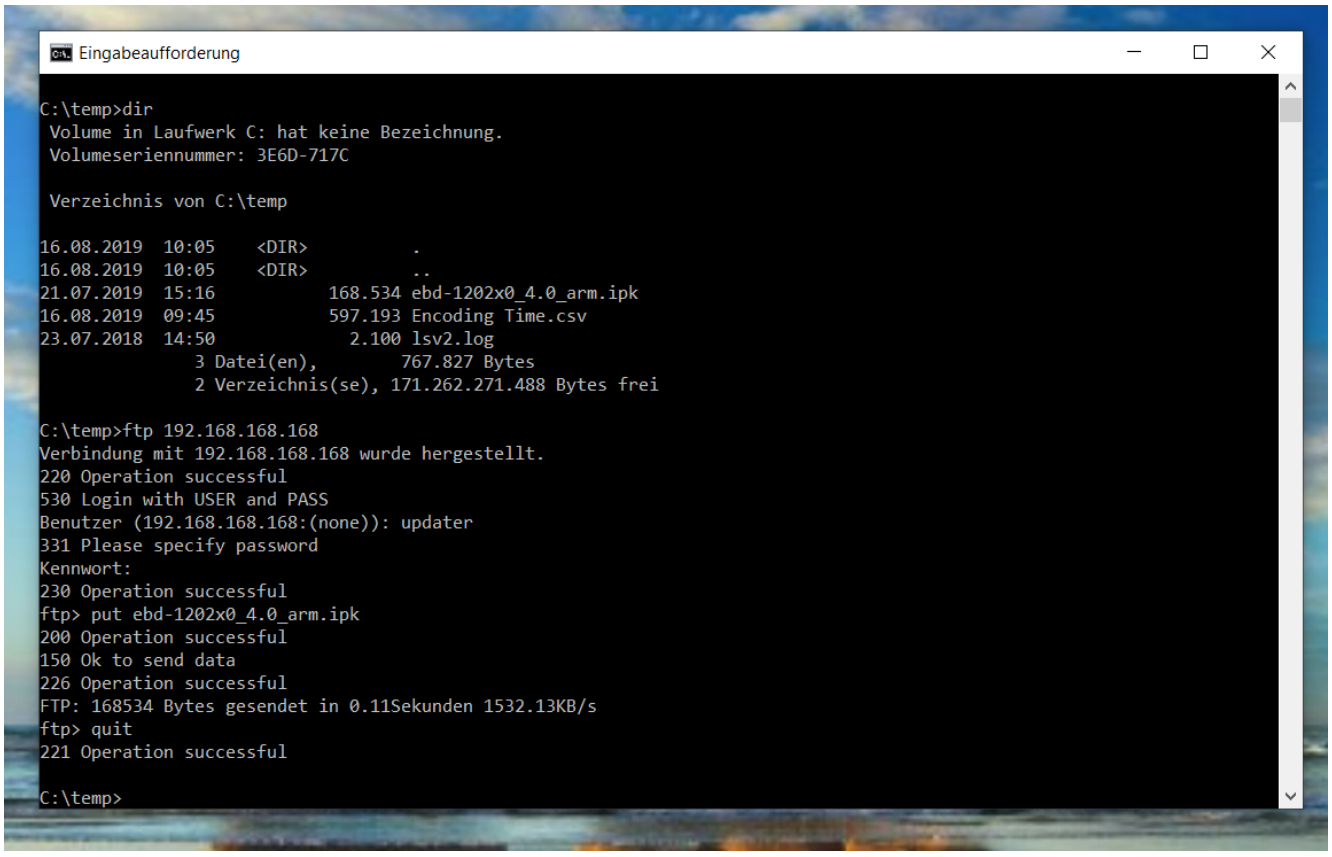
EBC/EBD-120* controllers use opkg (Open PacKaGe management) freeware which is intended for use on embedded GNU/Linux. A firmware collection file with 'ipk' extension should be used.

It is recommended to use nFUpdater.exe to update the firmware.
Note: valid interface is USB or Ethernet (RS232 is for EBC/EBD-060* controllers only).

To update firmware manually,
Steps 1: transfer file (e.g. ebd-1202x0_4.0_arm.ipk) to controller.
Software for Windows/linux is ftp. For Windows, use command console ('cmd' and send the following commands, ***case sensitive***)

```
>> ftp
>> open 192.168.178.2  // replace with the controller's IP address, or RNDIS/192.168.168.168
>> user
>> updater              // this is the user name
>> ebd-1202x0           // this is the password
>> put  ebd-1202x0_4.0_arm.ipk        // replace with the right firmware name
>> quit
```

Note: check the firewall setting to allow ftp protocol

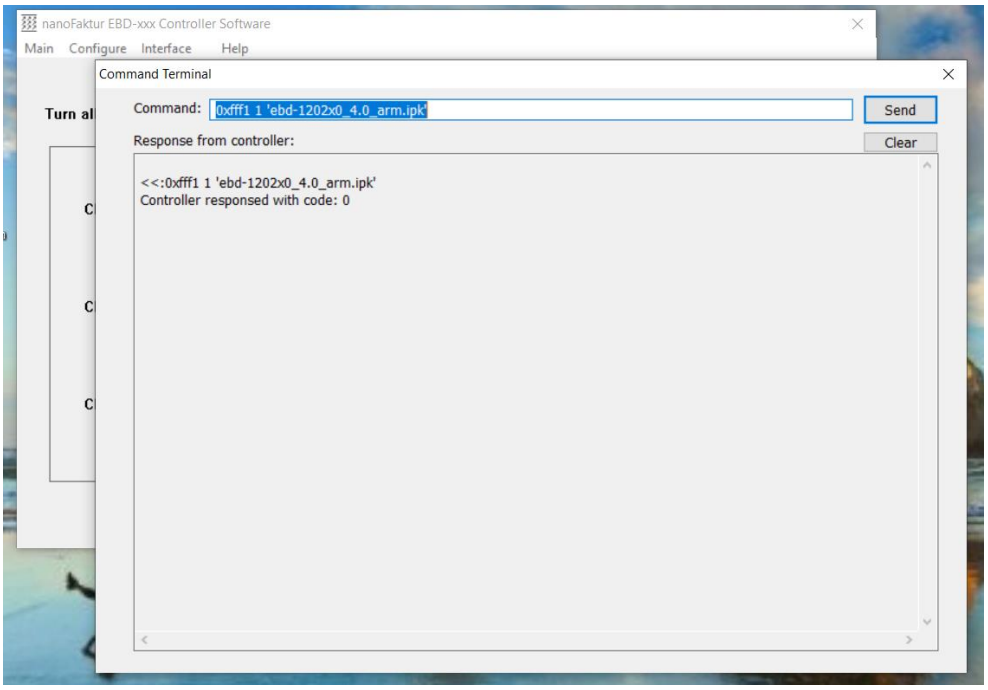


Windows example

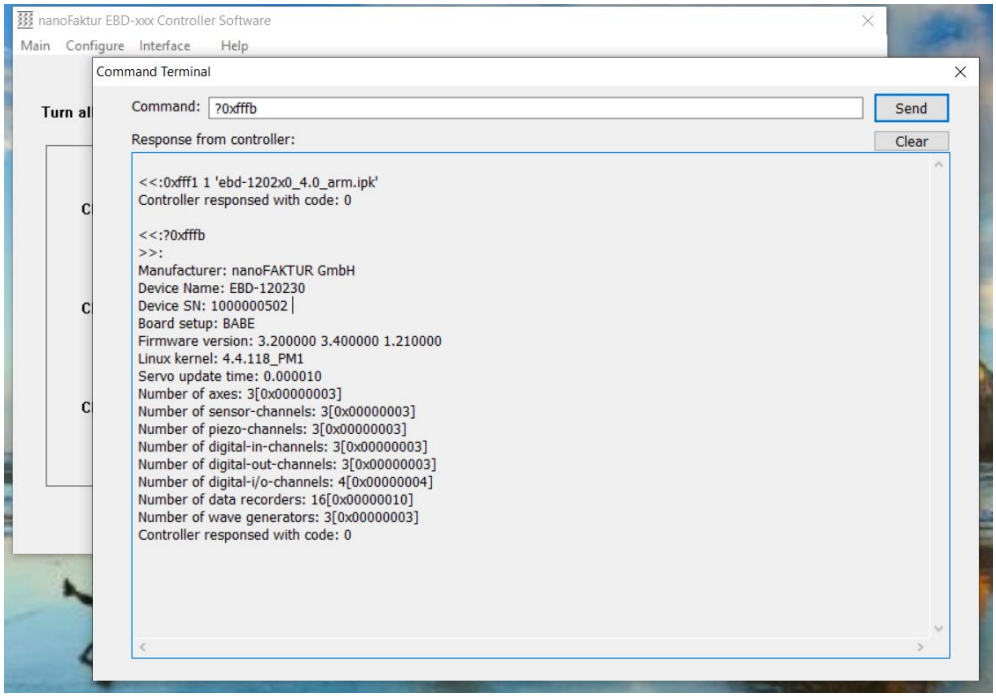
Step2: install the new firmware

Use command 0xFFF1 in nFControl -> Command-Terminal.

0xFFF1 1 'ebd-1202x0_4.0_arm.ipk' *// replace with the right firmware name*



After a successful execution, controller will be restarted and the new firmware will be installed.
Note: Interface of nFControl.exe should be re-connected.
Check the new firmware version with command ?0xFFFB



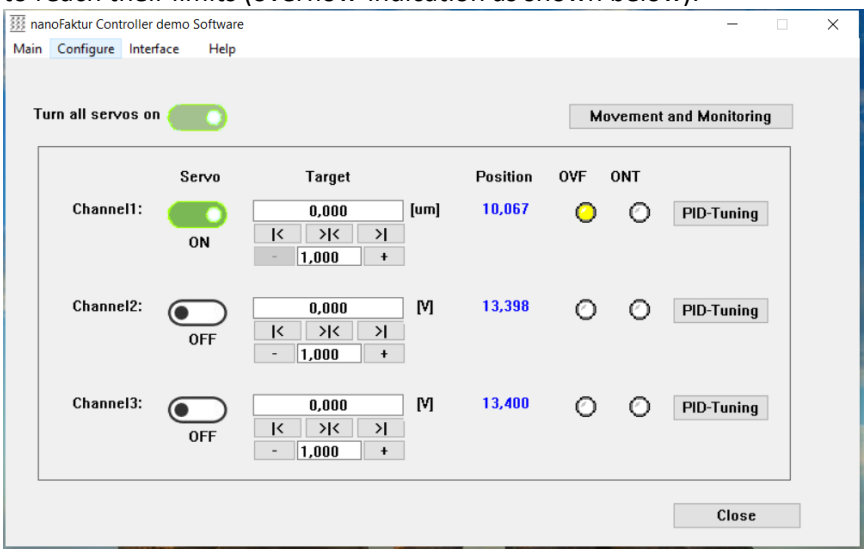
Note:

Since the updated firmware may have new parameters, which confirms controller's states and then be active, so please check parameters with nFControl.exe and save them again with

0x6003 100

5.4.13. Auto-zero function

Zero-positions may be shifted by load, transport or mounting. This can lead to mechanics not being able to reach their limits (overflow-indication as shown below).



In such case, the sensor zero-point can be shifted to another voltage with command 0x2150.

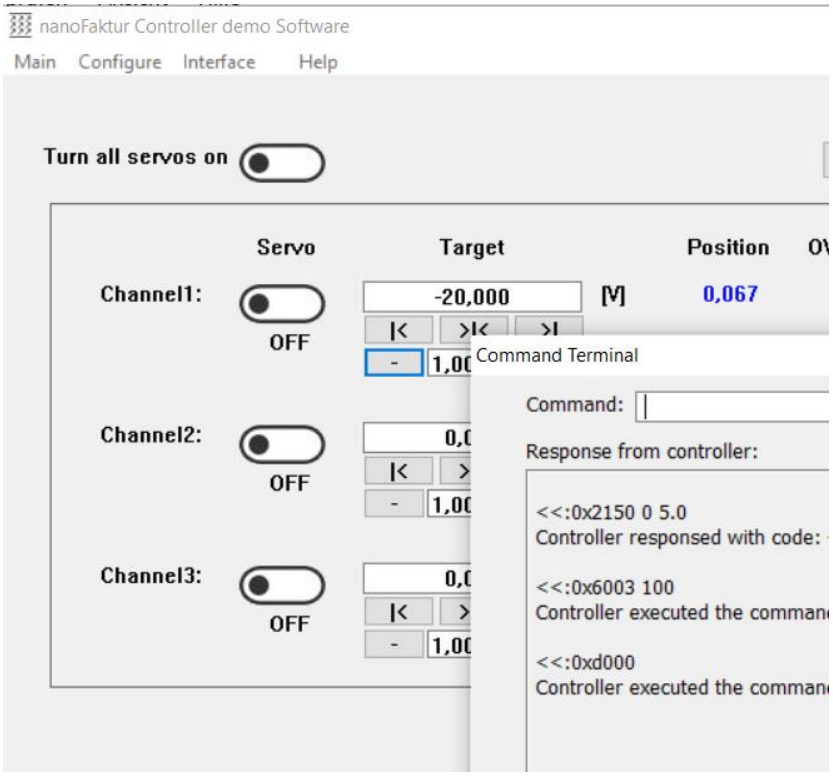
Step 1: Turn all Servo-mode to OFF

Step 2: Set voltage to example -20V

This means that the sensor value should show about 0.0 at -20V, depending on how much voltage is reserved for close-loop compensations. For example:

Set to -20V, when -20V ~ 150V corresponding to whole stage range or

Set to -30V, when -30V ~ 165V => whole stage range.



Step 3: Send the auto-zero-command

0x2150 0 2.0

Command ID: 0x2150

Syntax: <command-ID> <sensor-channel-index> <voltage-step>

Note: this can last for some seconds.

Step 4: Verify the stage can move the whole range:

Turn Servo-ON, and check the limit position.

Step 5: When OK, save the new parameters to controller and stage

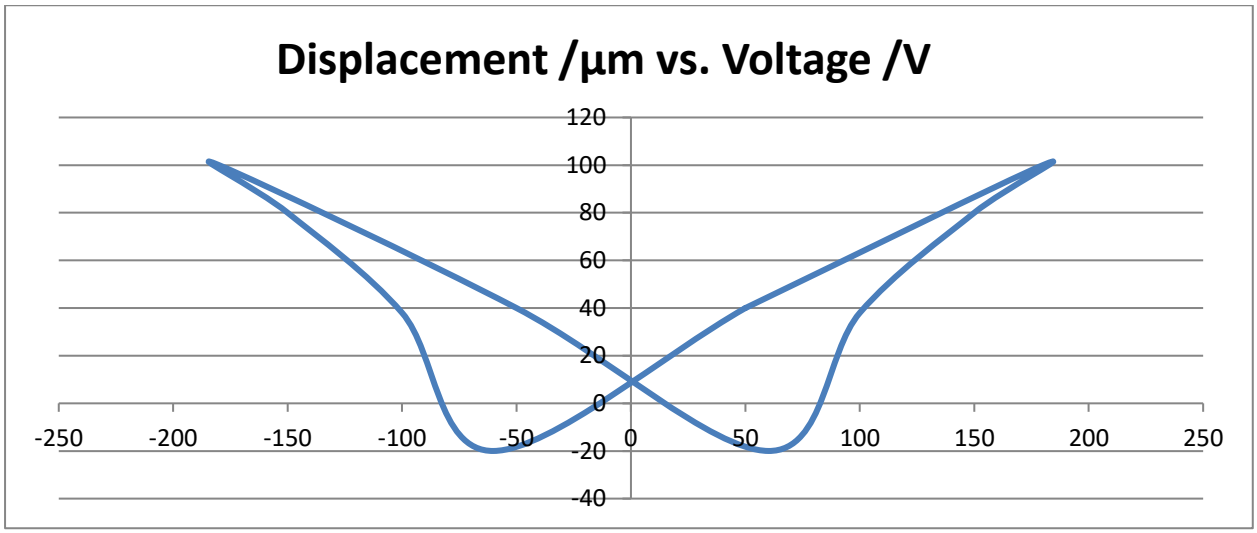
```
0x6003 100          // Save parameters to controller
0xD000              // Save parameters to ID-Chip of the stage
```

6. Explanation: Open- and Closed-Loop

6.1. Open Loop Behavior

Open loop operation means that simply voltages are applied to a PZT actuators, which then perform expansion or contraction. The latter is depending on which coupling constant is relevant and in which direction the voltages are applied.

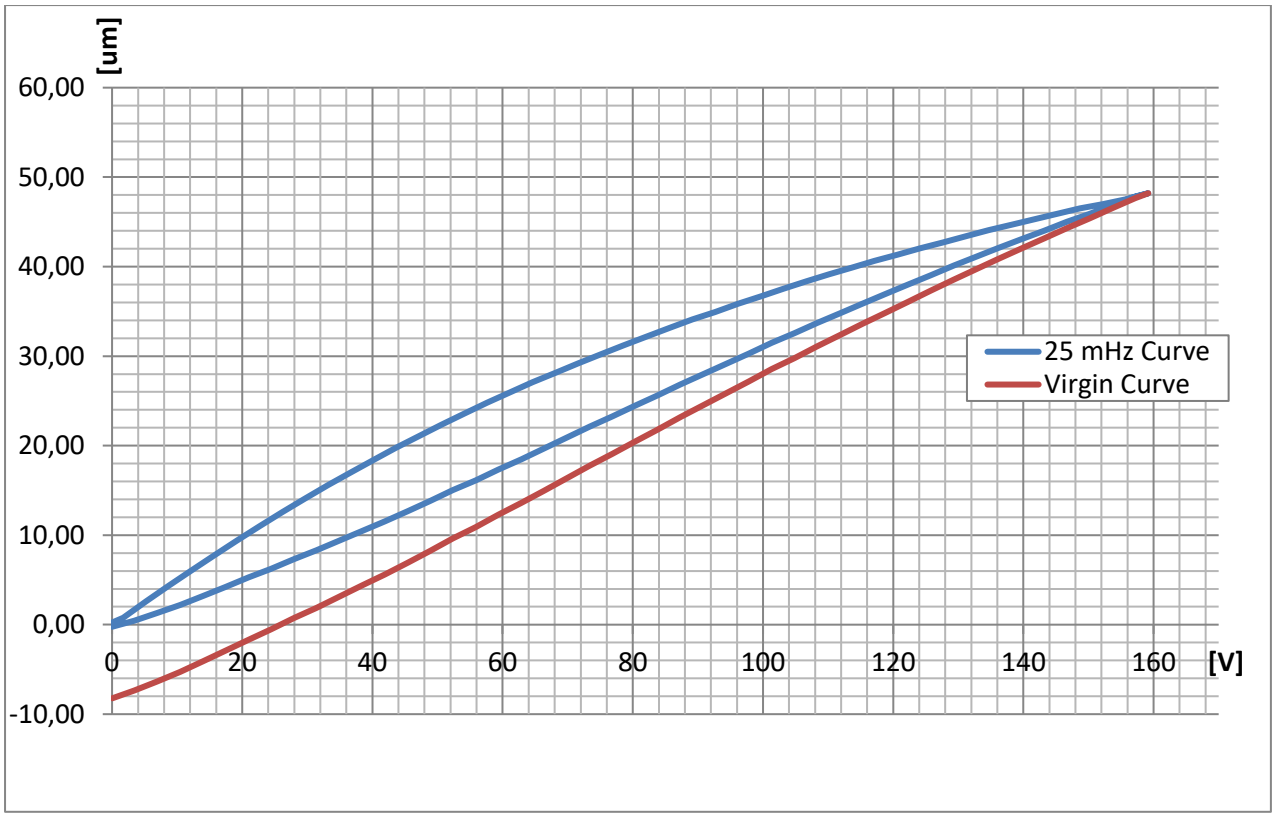
Typical behaviour of a d_{33} PZT-actuator at an electrical field of ± 3 KV/mm (counter-polarization occurs at the low turning points):



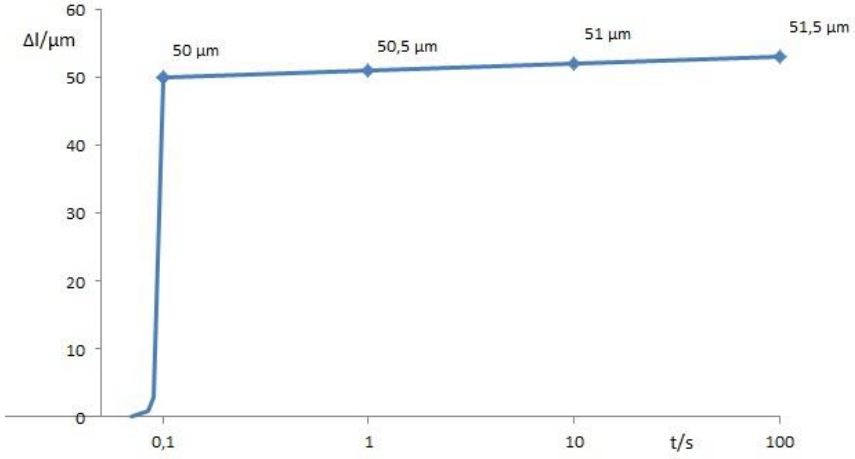
The symmetry of this butterfly-curve for plus and minus voltages and losses at repolarization lead to only unipolar usage making sense. Contraction at counter-fields can be added to 20%-30% of the nominal voltages. Maximum voltages and maximum counter-voltages depend on the kind of actuator. Follow manufacturer specifications!

nanoFaktur controllers in standard-configuration amplify by a factor of 15. The offset is at 0 V. A voltage of 0 to +10 V at the input-BNC will drive the output from 0 to +150 V. An extended output-range of about -45 to +180 V is serving as a reserve for closed-loop operation only.

6.2. Voltage-Behavior of a d_{33}- PZT

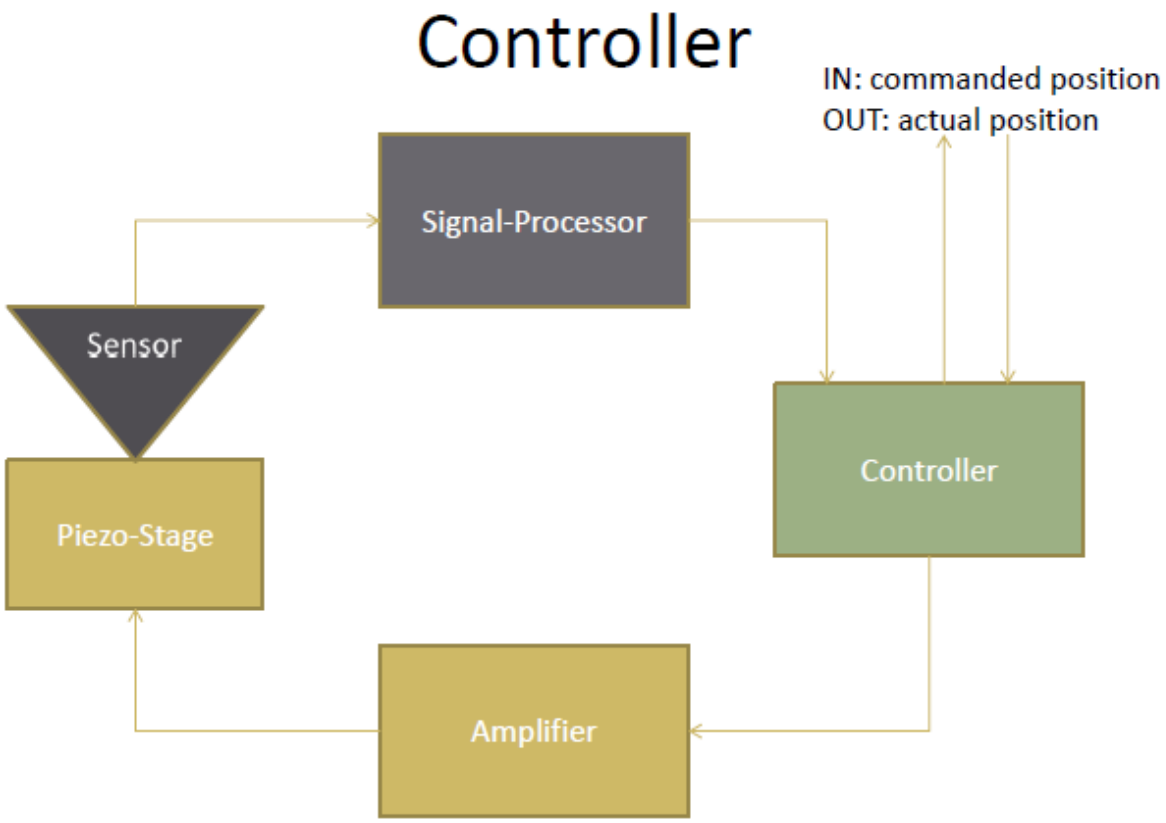


The curve above shows the typical behaviour of a nanoFaktur MPO-050050 PZT-multilayer stack. The stack was driven at voltages from 0 to 160 V, expansion monitored with a probe. When a PZT actuator is reactivated after a long rest, it performs a virgin curve. The Y-axis departure of the virgin-curve depends on the time of the rest. The existence of a gap between the zero-points of the virgin-curve and the later curve is indicating the phenomenon drift. A drift does always exist after the PZT has been used. The drift has a magnitude of 1% per time-decade.



6.3. Closed Loop Operation

Most applications require precise positioning and/or stable resting at positions. Hysteresis and drift described above shall be overcome. This can be achieved by using a linear position sensor measuring what the piezo-actuator is doing. A controller can compare the commanded position (input) with the actual position and generate a corrected voltage for the piezo. This is optimally done repeatedly (in loops). The frequency (band-width) of the loop is decisive for the dynamics of closed-loop operation. The inverse value of the band-width is the control-loop time. This is as low as 20 µs for 2nd generation EBC/EBD-controllers.



7. Troubleshooting

Please first update firmware to the newest stand.

Problem	Reason	Solution
USB connection		Install the right RNDIS driver
TCP/IP connection: no interface found	IP address not configured Network router not support 10Mbps.	Enable DHCP when used in a sub-net (First configure controller via another interface, and then save parameters in flash)
TCP/IP connection: already connected	TCP/IP allows only one connection	Software should disconnect the interface when closed
Can't send firmware for updating		Check the firewall setting of PC
Interface status LED red	Command error	Read error code with "? 0x1000"
DSP status LED red	Stage connecter not connected Unknown stage type Piezo-driving-module switched off System temperature problem DSP is busy	Read stage status with "? 0x204F" To reset status: "0x204F" To reset status: "0x22FE <ch> 0" "0x204F" Information: "?0xFFF9 0x204F" When busy: check "? 0x1001"
Channel LED status	Off: Open loop Blinking: Stage is Moving Green: On-target	
Channel LED yellow	Overflow	Check soft-limit Check PID parameters
No movement	Voltage limit reached Position limit reached Small velocity value and Trajectory enabled Target selection	Check open-loop soft-limit Check close-loop soft-limit Disable trajectory controlling Check "?0x2041"
Monitor output		Check the pin-definition For EBC/EBD-120310: output must be enabled