nanoFAKTUR
piezo · positioning · systems

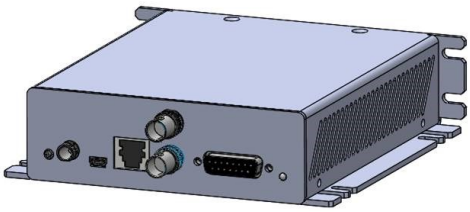
# User Manual E.006

# Exx-0603n Series Digital Controllers

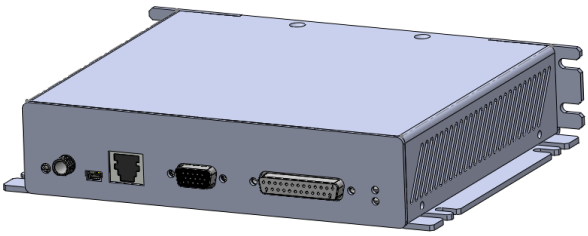
Release: 2.4 Date: 2024-08-30

**This document is valid for the following products:**

- **EOD-060310**
  Digital Controller, 1 Channel, OEM-Board, SGS
- **EOC-060310**
  Digital Controller, 1 Channel, OEM-Board, Cap. Sensor
- **EBD-060310**
  Digital Controller, 1 Channel, SGS
- **EBC-060310**
  Digital Controller, 1 Channel, Cap. Sensor
- **EBD-060320**
  Digital Controller, 2 Channels
- **EBD-060325**
  Digital Controller, 2 Channels plus 1 constant voltage
- **EBD-060330**
  Digital Controller, 3 Channels



EBD-060310/EBC-060310



EBD-060320, EBD-060325 (2 Status-LEDs) and EBD-060330 (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<br>78048 Villingen-Schwenningen<br>Germany | |

The manufacturer declares that the product

| | |
|---|---|
| Short Description: | Digital Controller, Voltage-Amplifier |
| Model Numbers: | EBC-060310, EBD-060310, EBD-060320, EBD-060325, EBD-060330, EOC-060310, EOD-060310 |
| Product Options: | EBD/EBC, Standard<br>EOD/EOC, with appropriate additional enclosure only, not in the scope of delivery |

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 |

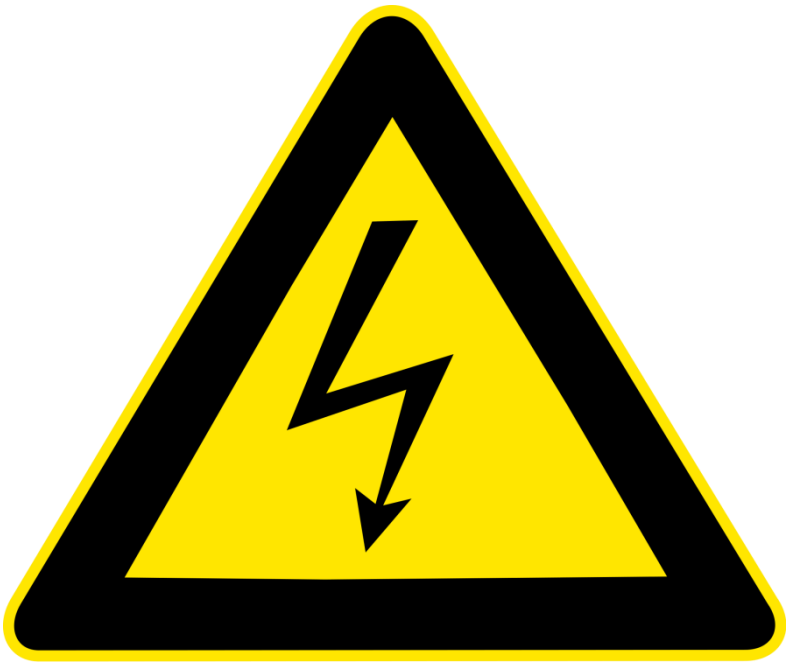
Electrical equipment which is intended to be integrated in other electrical equipment or enclosures, only conforms to the cited EMC Standards and normative documents, if the user ensures compliant enclosures and connections when integrating. Appropriate measures: installation of the components in qualified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

Villingen-Schwenningen, Germany, February, 2024

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

Caution! High Voltage!

## 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.

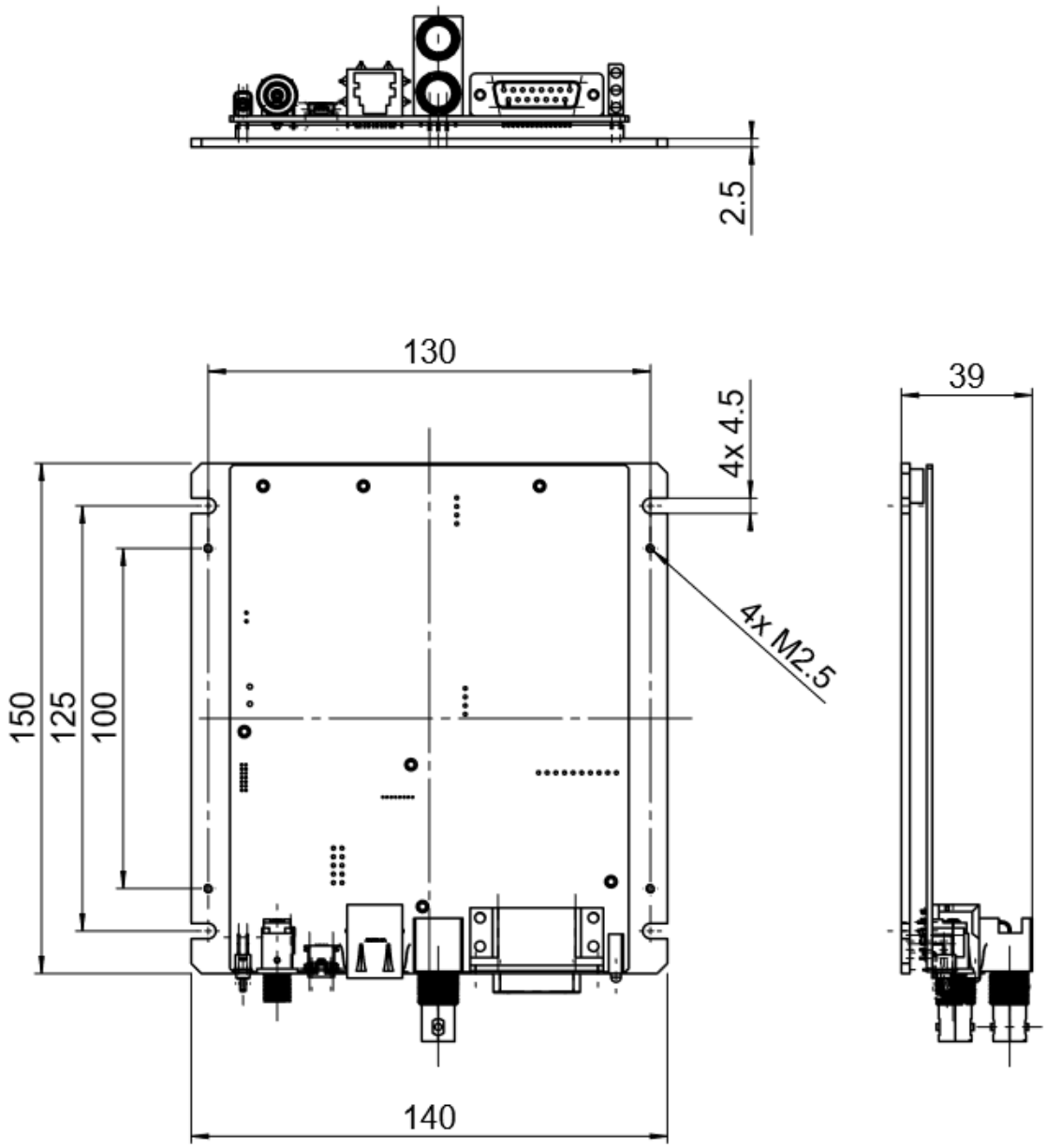
Warranty and liability of nanoFaktur are void if housed devices are opened or in case of any other hardware-manipulations. Use the determined interfaces for connections only. Intentional or default improper operation void all warranty of liability.

# 2. ExD-0603n Digital Controllers

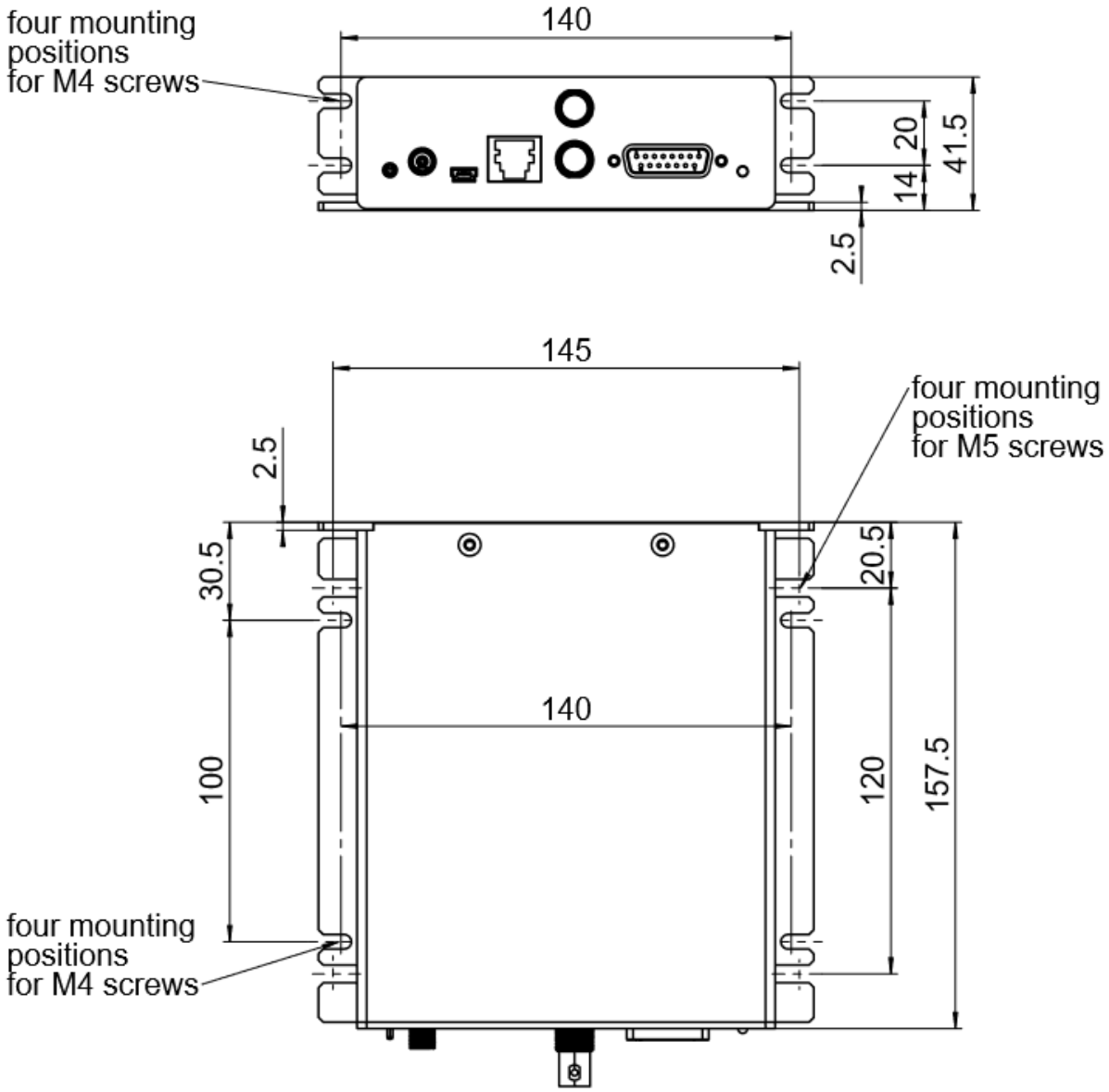
## 2.1. Specification

| Products | Digital Controllers for Piezo-Actuators/Displacement-Sensors | | |
|---|---|---|---|
| **Product Numbers** | **Exx-0603n (n=number of channels times 10)** | **Unit** | **Tolerance** |
| | | | |
| Channels | 1-3 | | |
| Output Voltage for Piezo | -45 to +180 | V | max. |
| Current | 150 | mA | peak max. |
| Average Current | 60 | mA | max. |
| | | | |
| Sensors | ExD-versions, Strain-Gages (SGS)<br>ExC-versions, capacitive sensors | | |
| Resolution | 18 | Bit | |
| | | | |
| Control Loop Time | 20 for ExD-060310/ExC-060310 | µs | |
| | 40 for 2-3 channel controllers | µs | |
| Control Parameters | PID, 2 Notch Filters per channel | | |
| Software | nFControl Windows™ GUI | | |
| Operating System Requirements | Microsoft ® Windows™ 7/8/10 | | |
| Wave-Generators | Sine-wave only | | |
| Data-Recorders | 2 Recorders (each has 512 data) | | |
| | | | |
| Digital Interfaces | USB 2.0 as COM-port, Ethernet,<br>RS232 (optional) | | |
| Analog Input | settable, -10 to +10, -5 to +5, 0 to 10<br>(Internal 1k Low pass filter) | V | |
| Analog Output | -5 to + 5 (for EBD-0602xx)<br>-5 to +10 (for EBD-0603xx) | 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 | | |
| Vin Voltage | 24 (1.5A min.) | VDC | |
| Vin Protection | Power input reverse polarity protection | | |
| Static Power Consumption | Ca. 7.5 | W | |
| | | | |
| Ambient Temperature | +10 to +40 | °C | |
| Standards (EOD/EOC with appropriate additional enclosure only, not in the scope of delivery) | 2014/35/EU, Low Voltage Directive (LVD)<br>2014/30/EU, EMC Directive<br>2011/65/EU/ RoHS Directive<br>Safety (Low Voltage Directive): IEC 61010-1:2010, IEC 61010-1:2010/AMD1:2016<br>EMC: EN 61326-1:2013<br>RoHS: EN IEC 63000:2018 | | |

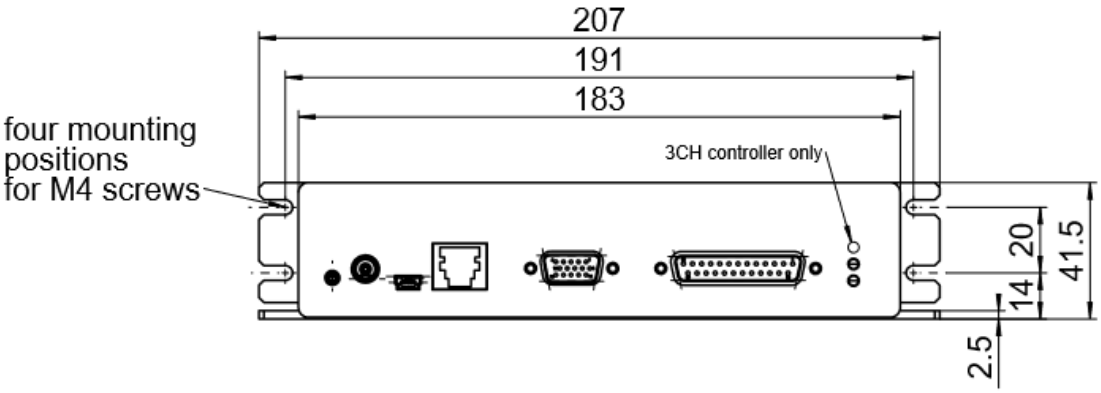
## 2.2. Dimensions EOx-060310

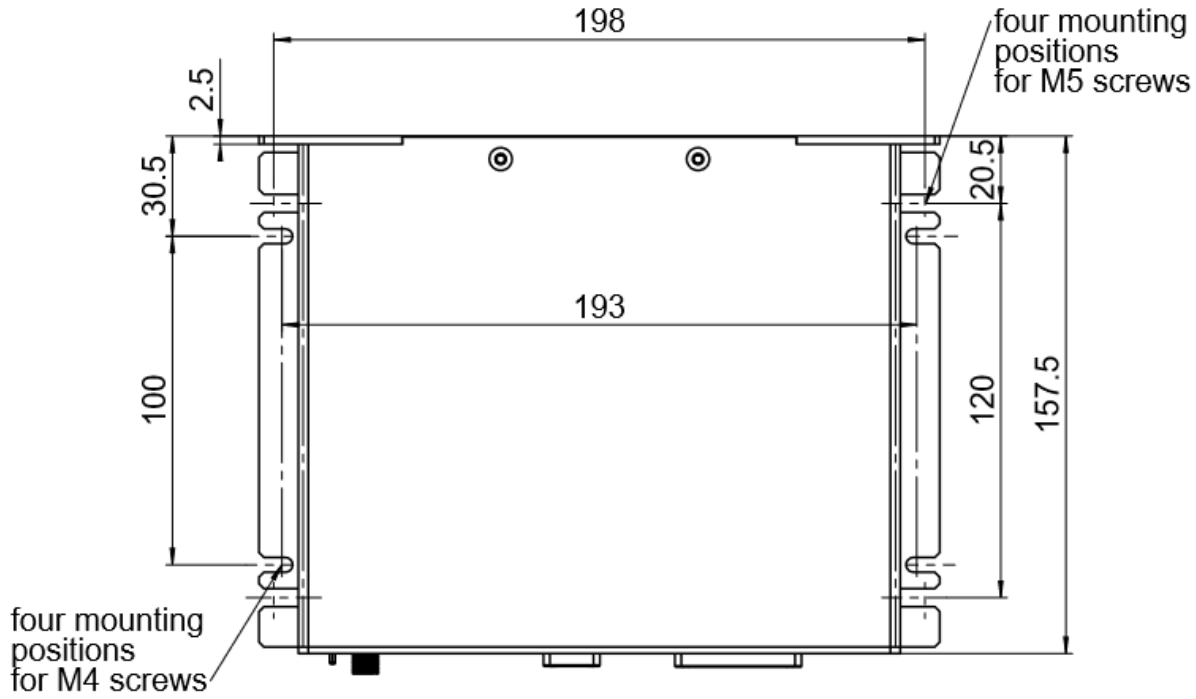


## 2.3. Dimensions EBx-060310/310



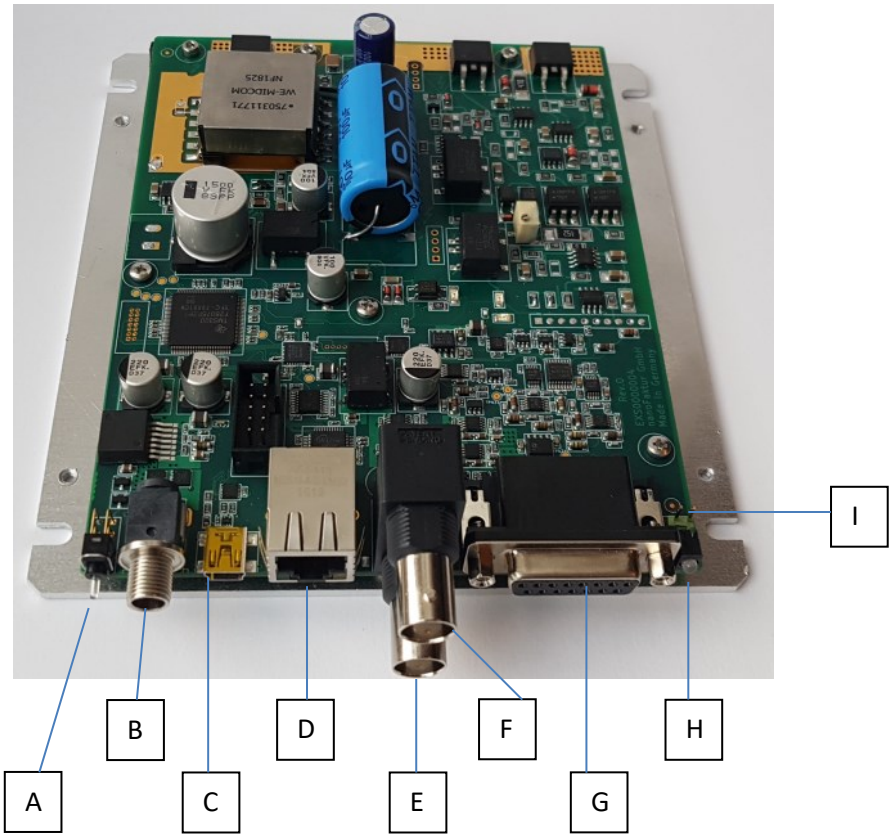
## 2.4. Dimensions EBD-060320, EBD-060325, EBD060330



# 3. Interfaces

## 3.1. EBx-060310, EOx-060310, Front




A: Switch, **green**: normal status, **red**: no stage connected, etc.

B: SwitchCraft RASPC10PS, Power Supply, 24 VDC 1.5A

C: Mini USB-B, Communication (as COM-port)

D: RJ45, Ethernet, Communication, optional

E: BNC (bottom), Analog-Input, 0..+10 V (-10..+10 V, -5..+5 V settable at works)

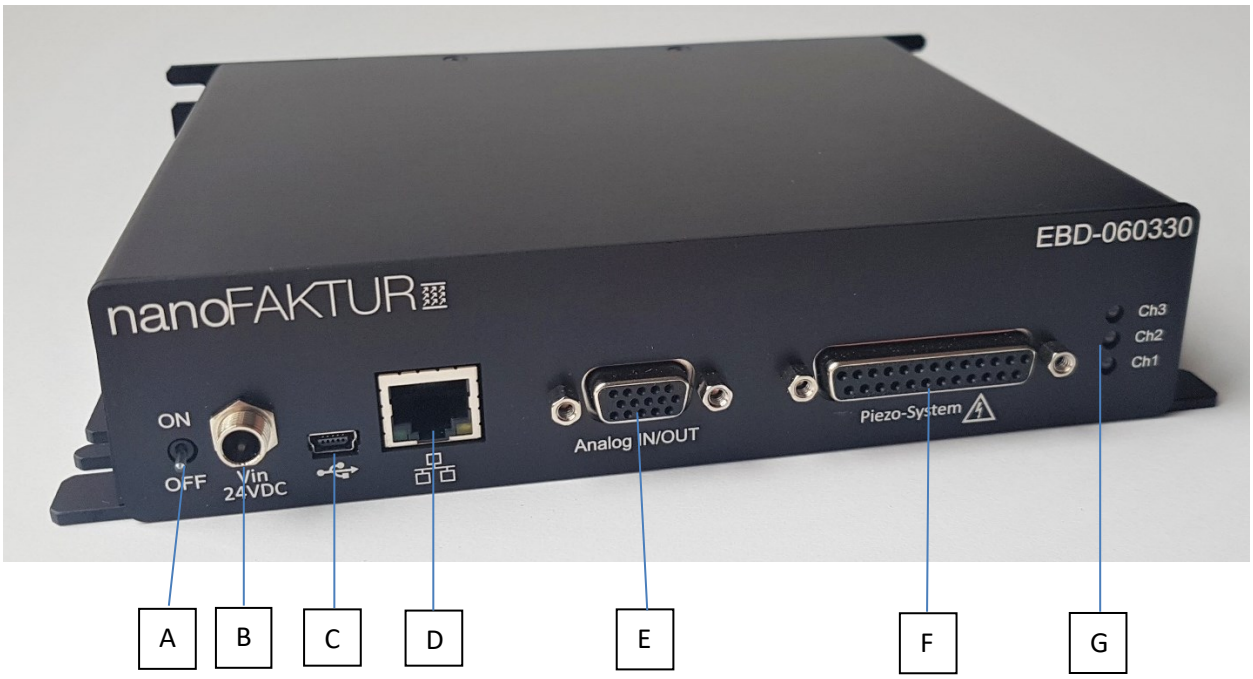
F: BNC (top), Analog-Output, analogue -5..+5 V

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

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

I: Trigger input/output (optional for EOD)

## 3.2. EBD-060320, EBD-060325, EBD-060330, Front




A: Switch, **green**: normal status, **red**: no stage connected, etc.

B: SwitchCraft RASPC10PS, Power Supply, 24 VDC 1.5A

C: Mini USB-B, Communication (as COM-port)

D: RJ45, Ethernet, Communication

E: DSub15, Analogue-I/O

F: DSub25f, Connection for assigned piezo-actuators/position-sensors

G: 2x(3x) LED, Servo-Status, **green**: on-target, **yellow**: overflow, off: servo-off

## 3.3. Pin-Assignments

### 3.3.1. Analog I/O of EBD-060320, EBD-060325 and EBD-060330

DSub15f (VGA-kind, 9-pin housing)

| No. | Name | Direction | Function |
|---|---|---|---|
| 1 | AIN3 | Input | 3rd Analog input, for EBD-060330 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 EBD-060330 only |
| 7 | GND | - | Ground |
| 8 | OUT2 | Output | 2nd Monitor output |
| 9 | GND | - | Ground |
| 10 | OUT1 | Output | 1st Monitor output |
| 11 | R | - | Reserved |
| 12 | R | - | Reserved |
| 13 | NC | - | Not connected |
| 14 | GND | - | Ground |
| 15 | GND | - | Ground |

### 3.3.2. DSub15f and DSub25f, Stage-Connectors

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 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 | NC | No connection |
| - | 1 | NC | No connection |

# 4.Software and Communication

## 4.1. Software Installation

### 4.1.1. Download up-to-date software and documentation

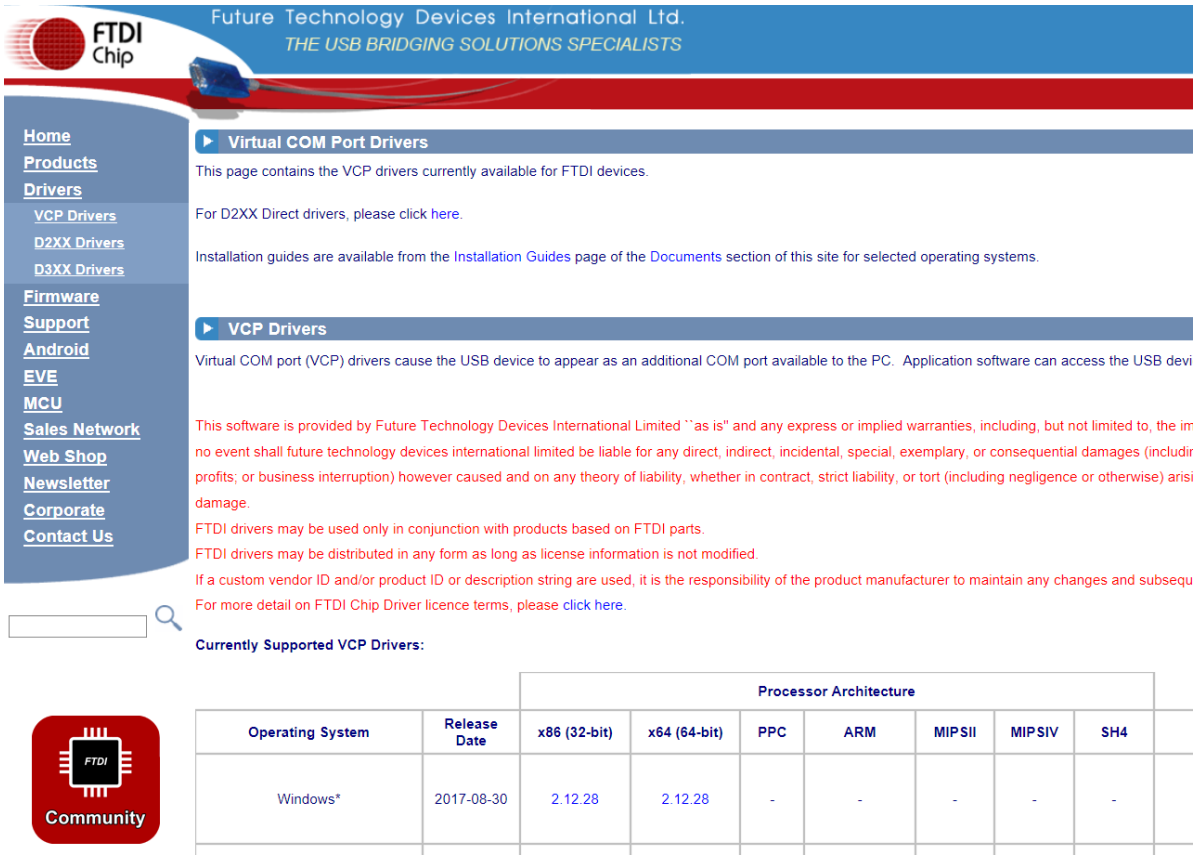
https://www.nanofaktur.com/ support
EBD-xxx2xx Software Support.zip
Password: nFSoftw

The following chapter describes how a USB-port is configured as an COM port. The chapter after this is describing how to configure the Ethernet-interface as TCP/IP port.

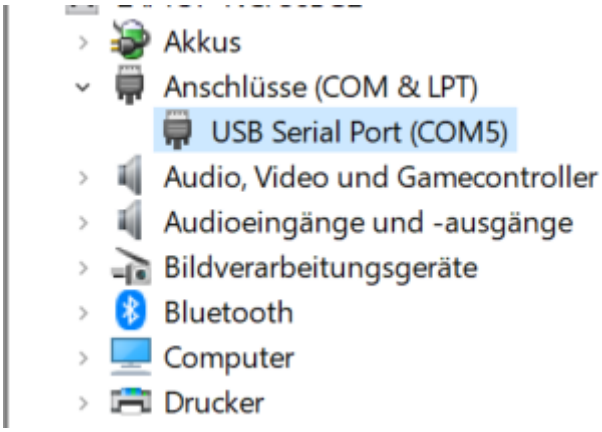
### 4.1.2. Install Windows® driver for USB (as COM-port)

When the EBD-0603x0/3x0 USB is connected to PC (ca. 2 seconds after controller has been switched on), Windows shows a new device: USB to COM port:

Please download the windows driver from FTDI (as shown below)




After installing the driver, the device-manager shows a new COM-port device:




The controller can be now connected via the nFControl.exe (as RS232).

### 4.1.3. Setup TCP/IP for Ethernet

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.

Note: It is recommended to use USB/RS232 interface to modify the Ethernet settings.

## 4.2. Operation and Tuning via the nFControl GUI

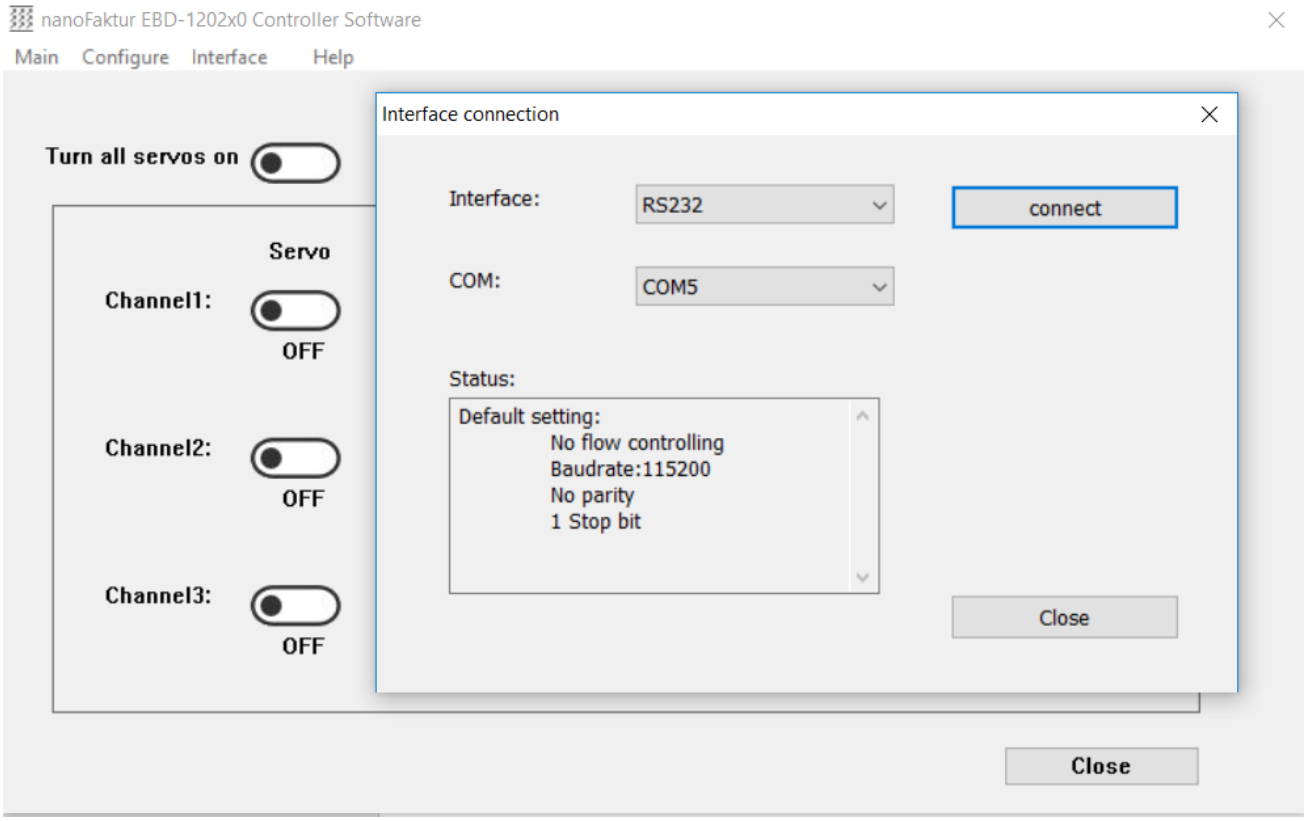
nanoFaktur GmbH provides a GUI for Windows®: nFControl.exe. This is convenient for the first usage and tuning of controllers. Interface functions are realized in the nF_interface.dll. It can be used for Labview or Python(32bits) programming.
Demo source code (written in C and can be compiled with MinGW-32bits)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

### 4.2.1. Install nFControl GUI

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

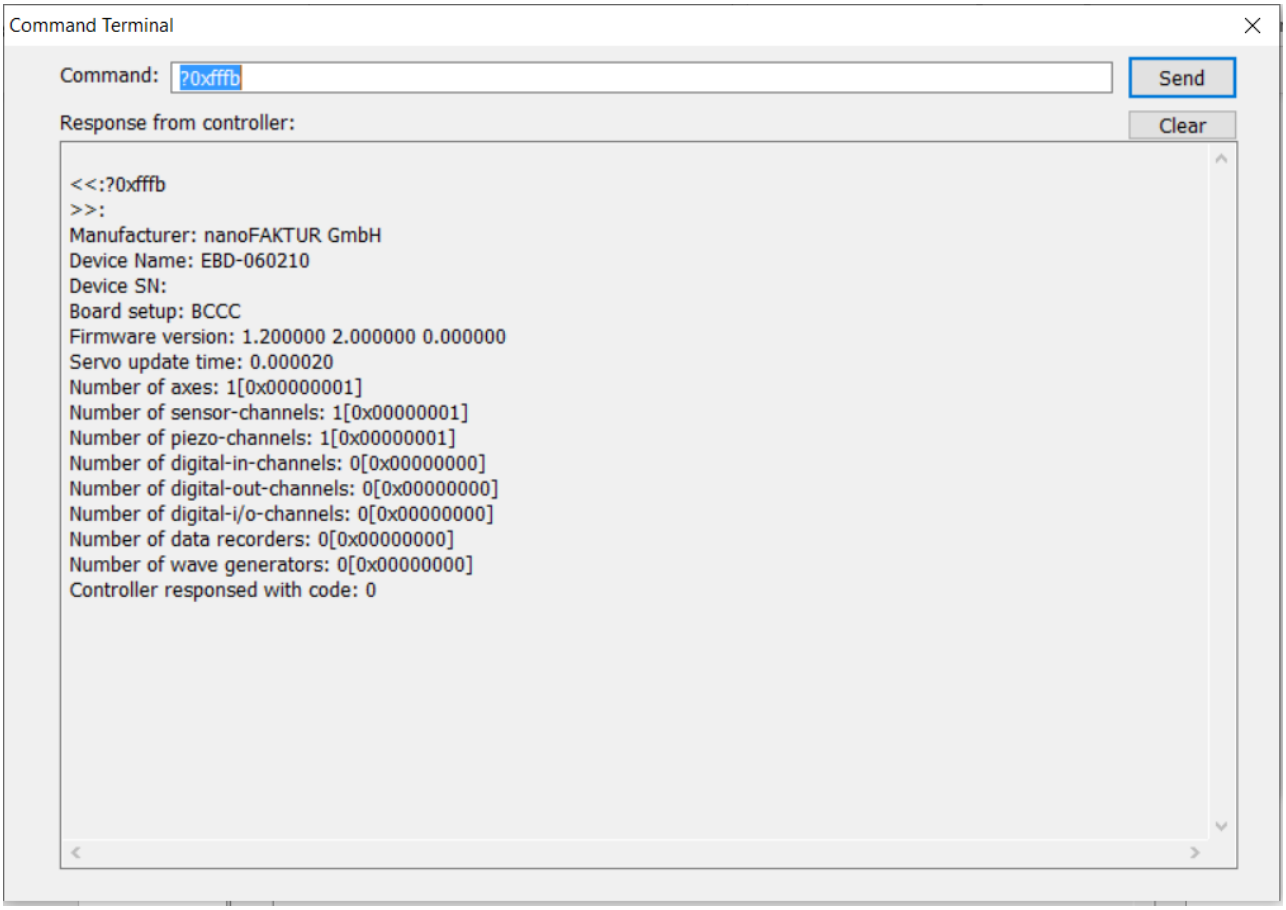
### 4.2.2. Quick guide

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

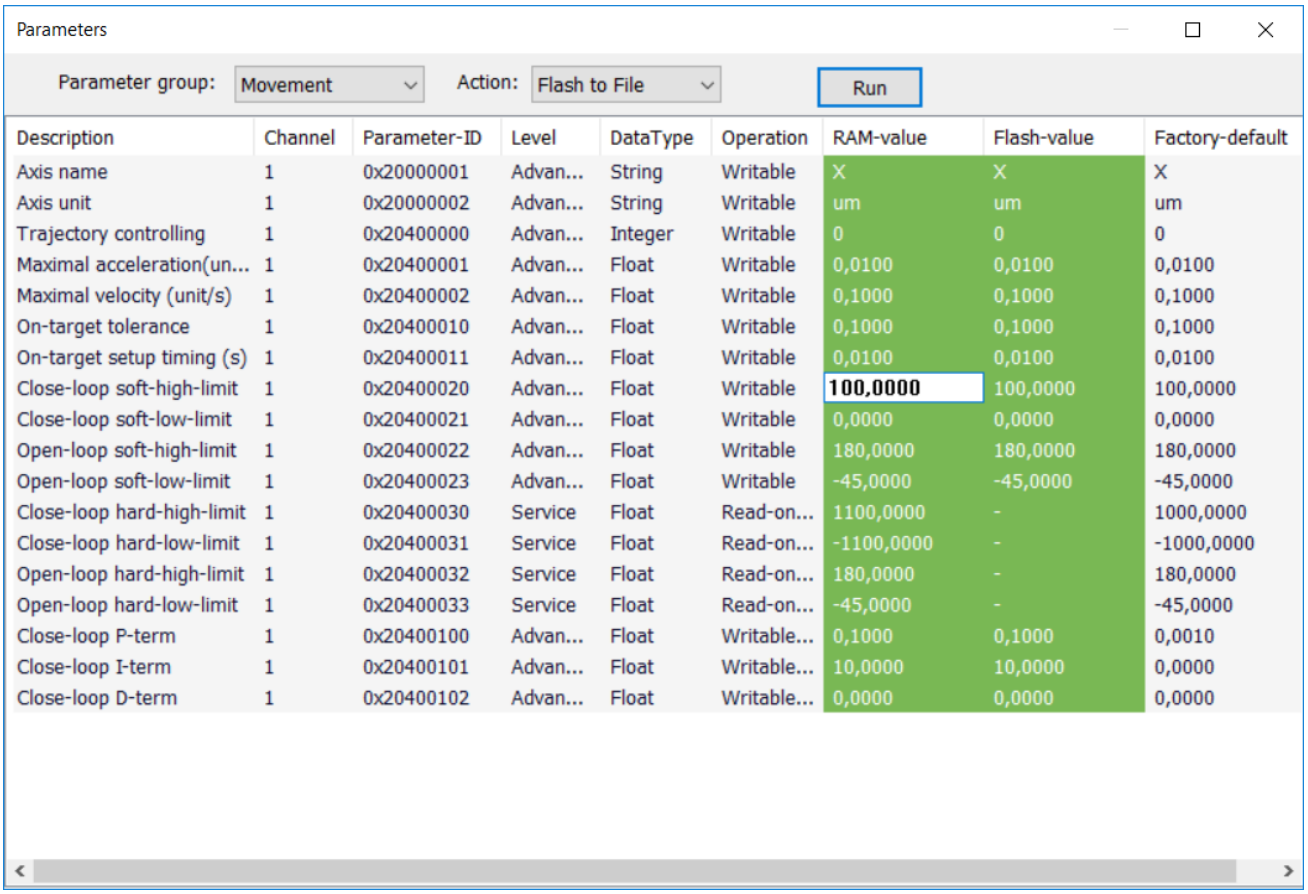



Note1: The USB interface is considered as an COM port.
Note2:
Some command needs "Command-level 1". nFControl.exe sends "0xFFF0 1" at interface 'connect'. It is recommended to take the same step by customer's Software or controller-Macro.

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 |
|---|---|---|---|---|---|---|---|---|
| Axis name | 1 | 0x20000001 | Advan... | String | Writable | X | X | X |
| Axis unit | 1 | 0x20000002 | Advan... | String | Writable | um | um | um |
| Trajectory controlling | 1 | 0x20400000 | Advan... | Integer | Writable | 0 | 0 | 0 |
| Maximal acceleration(un... | 1 | 0x20400001 | Advan... | Float | Writable | 0,0100 | 0,0100 | 0,0100 |
| Maximal velocity (unit/s) | 1 | 0x20400002 | Advan... | Float | Writable | 0,1000 | 0,1000 | 0,1000 |
| On-target tolerance | 1 | 0x20400010 | Advan... | Float | Writable | 0,1000 | 0,1000 | 0,1000 |
| On-target setup timing (s) | 1 | 0x20400011 | Advan... | Float | Writable | 0,0100 | 0,0100 | 0,0100 |
| Close-loop soft-high-limit | 1 | 0x20400020 | Advan... | Float | Writable | 100,0000 | 100,0000 | 100,0000 |
| Close-loop soft-low-limit | 1 | 0x20400021 | Advan... | Float | Writable | 0,0000 | 0,0000 | 0,0000 |
| Open-loop soft-high-limit | 1 | 0x20400022 | Advan... | Float | Writable | 180,0000 | 180,0000 | 180,0000 |
| Open-loop soft-low-limit | 1 | 0x20400023 | Advan... | Float | Writable | -45,0000 | -45,0000 | -45,0000 |
| Close-loop hard-high-limit | 1 | 0x20400030 | Service | Float | Read-on... | 1100,0000 | - | 1000,0000 |
| Close-loop hard-low-limit | 1 | 0x20400031 | Service | Float | Read-on... | -1100,0000 | - | -1000,0000 |
| Open-loop hard-high-limit | 1 | 0x20400032 | Service | Float | Read-on... | 180,0000 | - | 180,0000 |
| Open-loop hard-low-limit | 1 | 0x20400033 | Service | Float | Read-on... | -45,0000 | - | -45,0000 |
| Close-loop P-term | 1 | 0x20400100 | Advan... | Float | Writable... | 0,1000 | 0,1000 | 0,0010 |
| Close-loop I-term | 1 | 0x20400101 | Advan... | Float | Writable... | 10,0000 | 10,0000 | 0,0000 |
| Close-loop D-term | 1 | 0x20400102 | Advan... | Float | Writable... | 0,0000 | 0,0000 | 0,0000 |

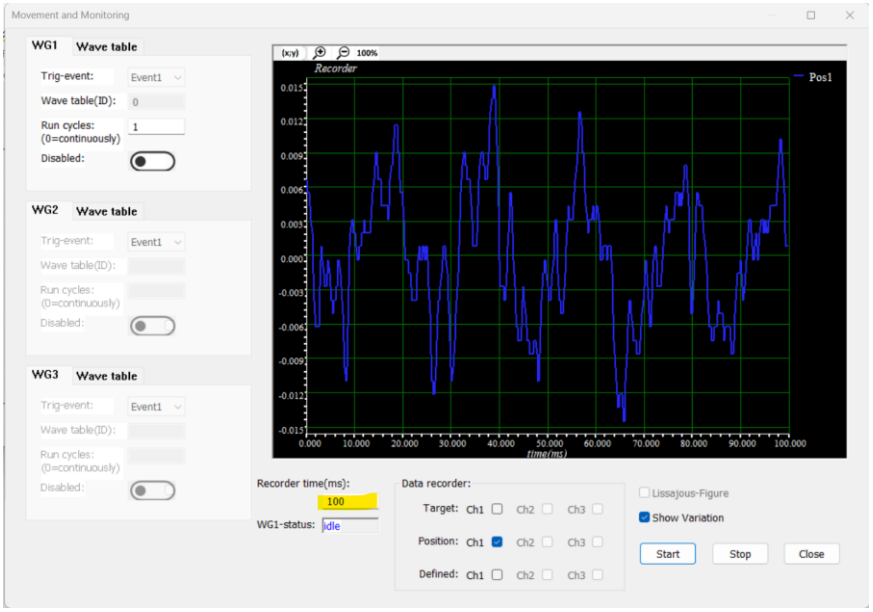
For parameters with 'Writable' property, double click to change the value.

RAM-value is lost when controller is rebooted or powered off.

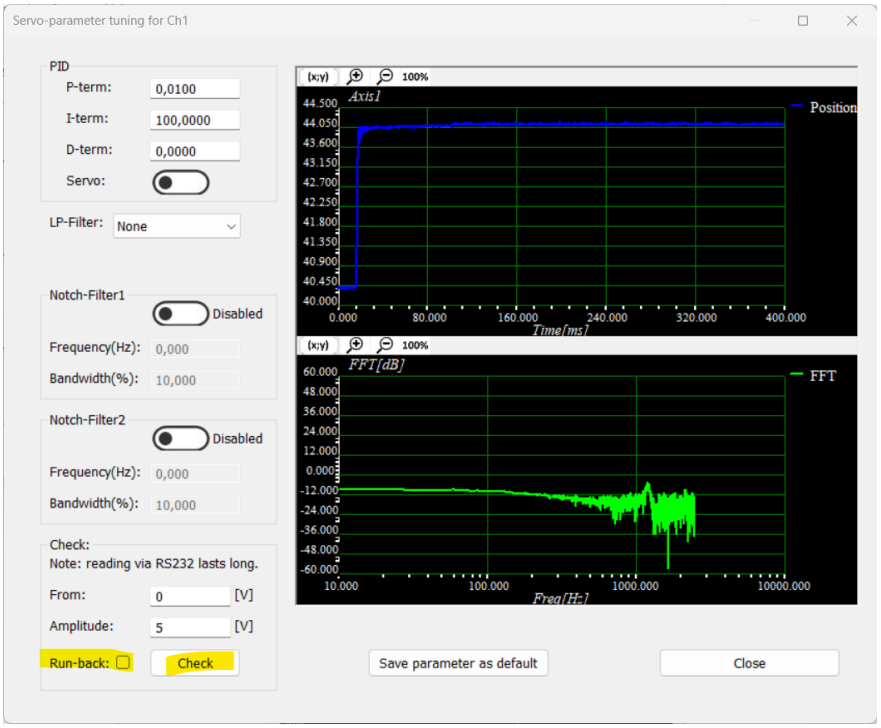
The flash-values are used when the controller gets restarted.

4) PID tuning

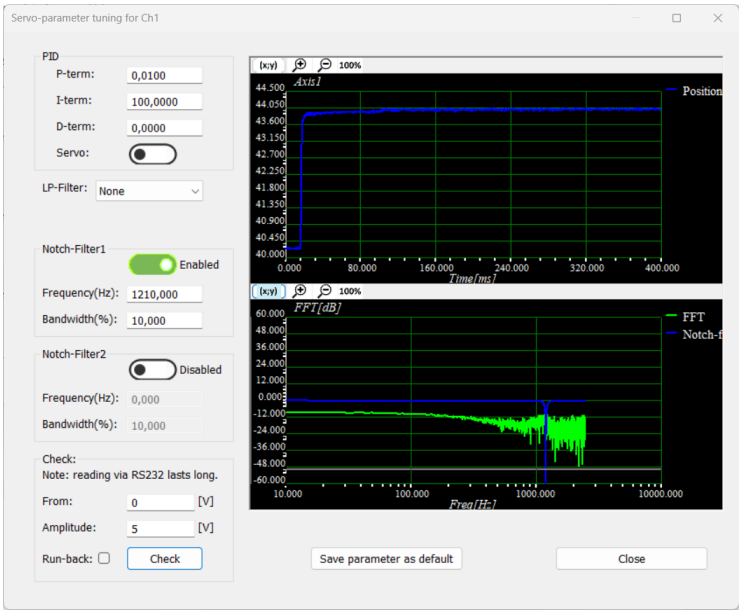
Configure the recording length by the “Movement and Monitoring” -> “Recorder time (ms)” . This value can be set to 100 ~ 500ms.



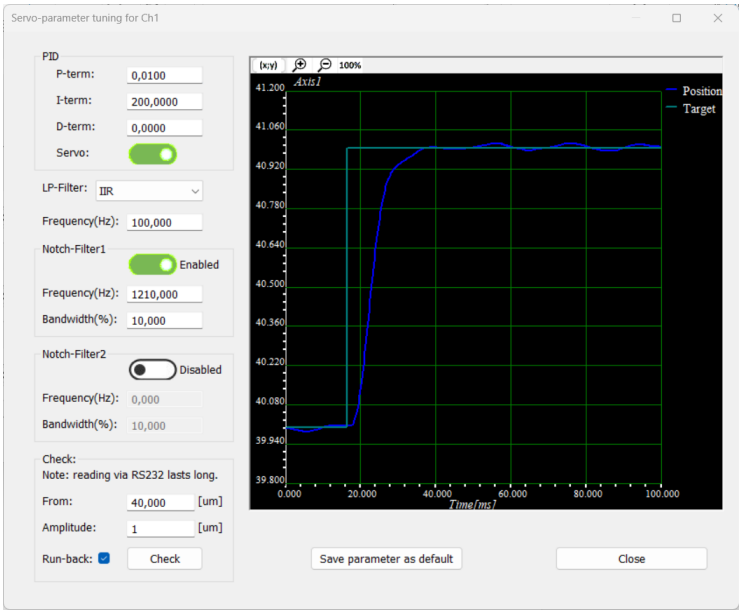

In open loop mode, click ‘***Check’*** to see the step response (depends on stage, the FFT varies).




Enable Notch-filter (as shown 1210Hz) and check again




Then turn '***Servo***' to on.




In order to avoid oscillations during the PID parameter tuning, first set P-Term to a small value e.g. 0.01,

Click the '***Check***' button to see the step response during parameter tuning.

Recommended values:

P-Term: starts from 0.01, step 0.01

I-Term: starts from 10.0, step 1.0 ~ 100.0

D-Term: 0.0 (valid from firmware rev.4.20)

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

## 4.3. Commanding

ExD-060xxx uses a 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.

### 4.3.1. 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 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 | Sum | Command-Data | SumData |

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 ChkSumHeader)
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)
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
ChkSumHeader = 0xA8
Format1 = 0x00 (U8)
Data1 = 0x00 (index of the first axis)
Format2 = 0x02 (Float)
Data2 = 10.55
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 see 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.

### 4.3.2. Command-Set

Note:

"[ ]" means optional parameters

"{ }" means one or more parameters

"- " command operation invalid

| 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 |
| 0xFFFB | | - | 0 | show information of system |
| 0xFFFA | | - | 0 | show information of interface |
| 0xFFF9 | [{<ID>}] | - | 0 | show list of command/parameter options |
| 0xFFF1 | - | <Parameter> | 1 | See Firmware update |
| 0xFFF0 | [{<Level>}] | {<CommandLevel> [password]} | 0 | command level set/get (No password for level 0 and 1) |
| 0xFF00 | - | | 0 | system reset |
| **Macro** | | | | |
| 0xE013 | - | <index> | 0 | Macro wait event (index: event-id) |
| 0xE012 | - | <millisecond> | 0 | Delay inside macro function |
| 0xE011 | - | | 0 | Stop running macros |
| 0xE010 | | <string>[<times>] | 0 | Show running status / Start macro |
| 0xE007 | | | 0 | Show macro memory status / memory defragmentation |
| 0xE006 | - | <string> | 0 | Delete macro |
| 0xE005 | | <string> | 0 | Default macro set/get |
| 0xE004 | <string> | - | 0 | Show macro content |
| 0xE003 | | - | 0 | List all macros |
| 0xE002 | - | | 0 | End of recording macro |
| 0xE000 | - | <string> | 0 | start recording macro |
| **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 | IDChip status |
| 0xD000 | | | 1 | IDChip parameters store / restore |
| **Parameter R/W** | | | | |
| 0x6005 | [{<index><parameterID>}] | - | 0 | get factory-default parameters |
| 0x6004 | - | <option> | 0 | restore flash to volatile parameters |
| 0x6003 | - | <option> | 0 | save volatile parameters to flash |
| 0x6002 | [{<index><parameterID>}] | {<index><parameterID><data>} | 0 | non-volatile parameters set/get |
| 0x6001 | [{<index><parameterID>}] | {<index><parameterID><data>} | 0 | volatile parameters set/get |

| Command - ID | Parameters for reading | Parameters for writing | Level | Description |
|---|---|---|---|---|
| **Data Recorder** | | | | |
| 0x4051 | [{<index>}] | {<index><eventID>} | 0 | recorder event selection set/get |
| 0x4050 | [{<uint>}] | {<index><mode><source>} | 0 | recorder source configuration set/get |
| 0x4042 | [{<index>}] | - | 0 | get recorded data-length |
| 0x4041 | [{<index>}] | {<index><rate>} | 0 | recorder rate set/get |
| 0x4040 | [{<index>}] | {<index><enable/disable>} | 0 | recorder enable set/get |
| 0x4011 | <index><from><length> [format] | - | 0 | get recorder table data |
| 0x4000 | - | [{<index>}] | 0 | clear recorder table |
| **Piezo-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 |
| 0x2210 | [{<index>}] | - | 0 | get DAC value |
| **Sensor** | | | | |
| 0x21FE | [{<index>}] | {<index><source>} | 1 | Monitor ADC selection set/get |
| 0x21FD | [{<index>}] | {<index><mode>} | 1 | Analog I/O configuration set/get |
| 0x2150 | - | [{<index>}] | 1 | Sensor offset auto-adjustment |
| 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 |
| 0x2050 | [{<index>}] | {<index><velocity>} | 0 | Velocity set/get |
| 0x204F | | | 0 | Stage status |
| 0x2044 | [{<index>}] | {<index><mode>} | 0 | Event controlled motion |
| 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 |
| 0x2014 | [{<index>}] | - | 0 | get currently controlling voltage |
| 0x2013 | [{<index>}] | - | 0 | get currently position error |
| 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 |

| Command - ID | Parameters for reading | Parameters for writing | Level | Description |
|---|---|---|---|---|
| 0x2002 | [{<index>}] | {<index><target>} | 0 | close-loop target set/get |
| 0x2001 | [{<index>}] | - | 0 | get position |
| **Controller status** | | | | |
| 0x1000 | | - | 0 | Get (and reset) error-code of controller |

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

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)<br>0xFF = Operation not valid (i.e., command not for writing)<br>0x01 = No parameters |
| 1 | 00 | Command level:<br>0 = Normal<br>1 = Advanced<br>Others = reserved |
| 2 | 00 | Reserved |
| 3 | 00 | Reserved |

### 4.3.3. Parameters

| Parameter-ID | Data format | Read-only | Level | Description |
|---|---|---|---|---|
| **System** | | | | |
| 0xFF010100 | unsigned int | Y | 0 | RS232 Baud-rate |
| 0xFF010014 | string | | 1 | Network gate-way to set |
| 0xFF010013 | string | | 1 | Network netmask to set |
| 0xFF010012 | string | | 1 | IPv4 address to set |
| 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 |
| | | | | |
| 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 |
| 0xFF00000F | float | Y | 0 | Control-loop time(s) |
| 0xFF000005 | unsigned int | Y | 0 | Number of wave-generators |
| 0xFF000004 | unsigned int | Y | 0 | Number of recorders |
| 0xFF000003 | unsigned int | Y | 0 | Number of pizeo-channels |
| 0xFF000002 | unsigned int | Y | 0 | Number of sesnor-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 |
| 0xC0400821 | float | | 0 | Notch2 bandwidth (%) |
| 0xC0400820 | float | | 0 | Notch2 -3dB frequency (0.0 = disabled) |
| 0xC0400801 | float | | 0 | Notch1 bandwidth (%) |
| 0xC0400800 | float | | 0 | Notch1 -3dB frequency (0.0 = disabled) |
| 0xC0400101 | float | | 0 | IIR-LPF Quality factor |
| 0xC0400100 | float | | 0 | IIR-LPF -3dB frequency |
| 0xC0400000 | unsigned int | | 0 | low-pass-filter (LPF) types<br>use "***?0xFFF9 0xC0400000***" to see valid settings |
| **Data recorder** | | | | |
| 0x40400000 | unsigned int | | 0 | recorder rate |
| **Motion** | | | | |
| 0x21400201 | float | | 1 | Analog input gain |
| 0x21400200 | float | | 1 | Analog input offset |
| 0x20400102 | float | | 1 | close-loop D-term |

| Parameter-ID | Data format | Read-only | Level | Description |
|---|---|---|---|---|
| 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 (seconds) | | 1 | on-target setup timing |
| 0x20400010 | float | | 1 | on-target tolerance |
| 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.*

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)<br>0x00 = Char<br>0x01 = Unsigned int32<br>0x02 = Float<br>0x04 = String<br>please see nF_common.h for detail |
| 1 | 00 | Command level:<br>0 = Normal<br>1 = Advanced<br>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

0x6002 0 0x21000004 '0000'

Example to store all parameters to flash:

0x6003 100

## 4.4. Operation

### 4.4.1. Position Commanding and Reading

The controller is ready in about 2 seconds after powered-on. The default state is Servo-Off and 0V output.

In order to move the stage to a desired position: (e.g. axis 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

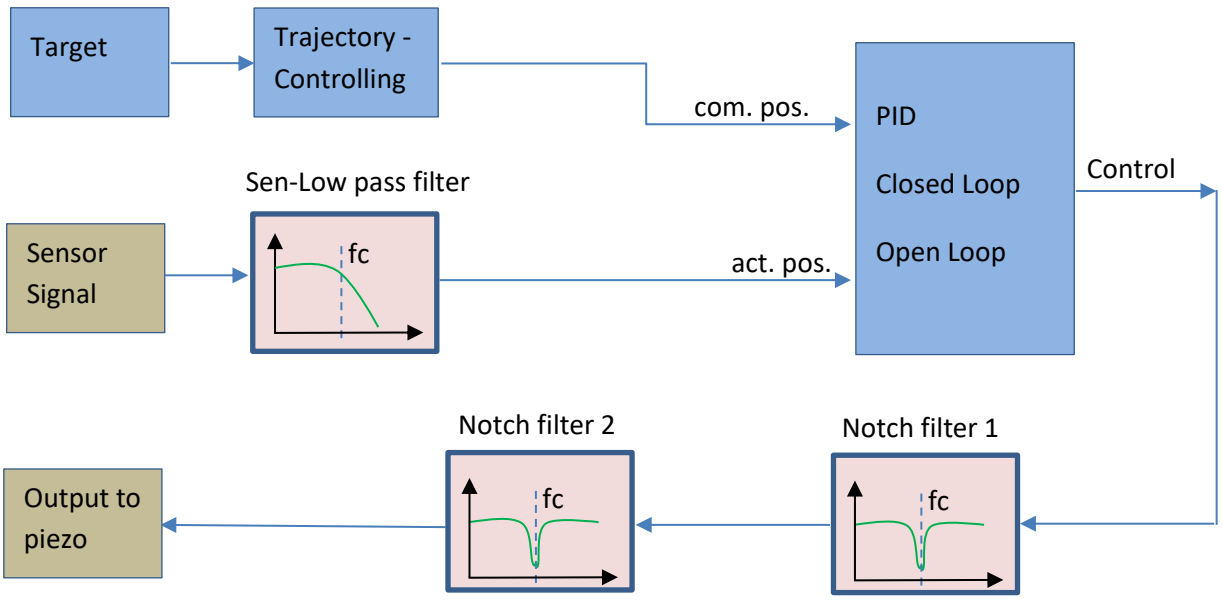
2) Set target to 1.0:

0x2002 0 1.0

3) Check position:

?0x2001 0 (command with read-back flag)

### 4.4.2. Function block diagram




Note: The sensor has one low-pass-filter. The controlling voltage has 2 notch-filters.
Set fc to 0.0 to disable filter function.

### 4.4.3. Target selection

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

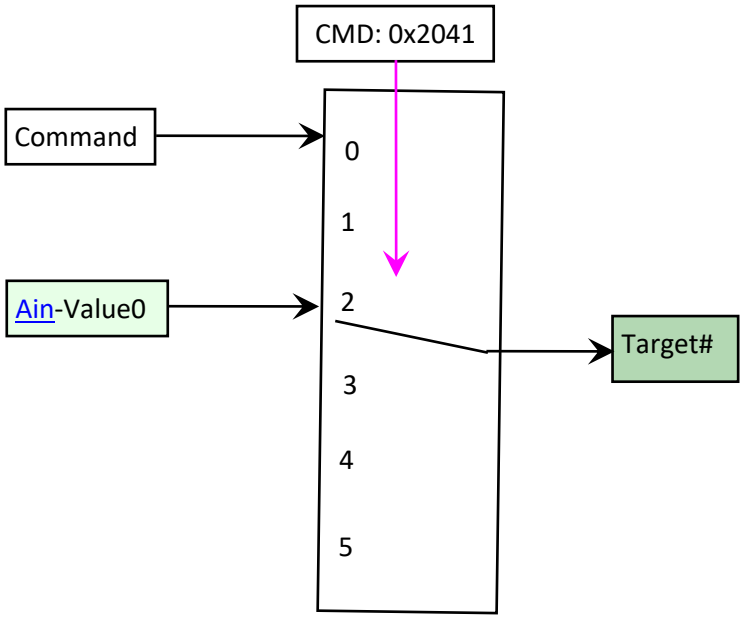
For closed loop (Servo-On): The target is a position value.

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

Target sources include command and 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 axis target to Ain-value:

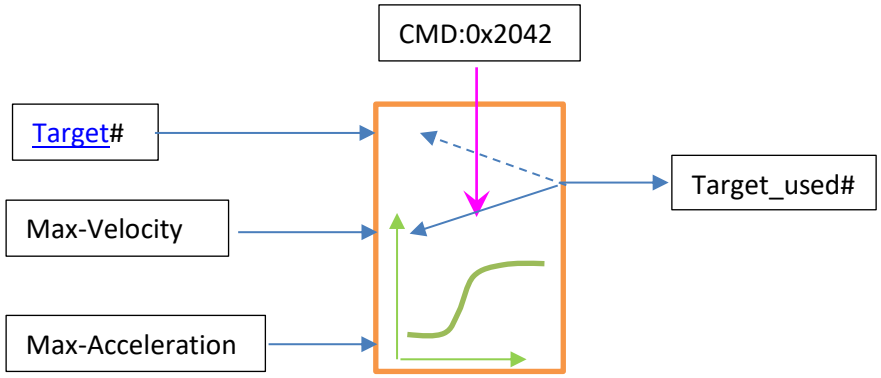
0x2041 0 2 (2 stands for Ain-Value0)



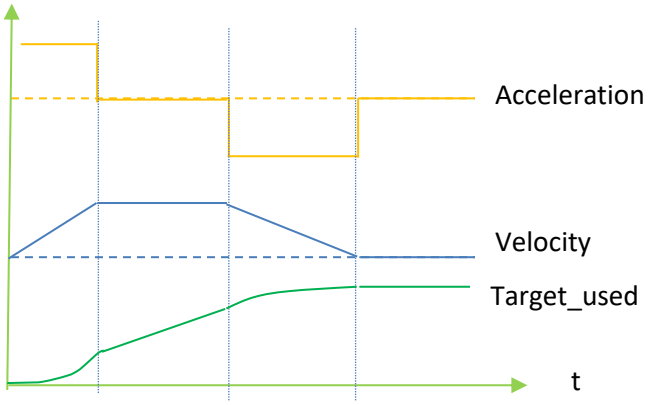

Note: use "***?0xFFF9 0x2041***" to see valid settings

### 4.4.4. Trajectory controlling

The selected target will be passed to the 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. 10um/(s*s). Optional, since parameter can be saved in flash.

0x6001 0 0x20400001 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

***Note:***

for scanning applications, trajectory controlling should be disabled.

0x6001 <axis-ID> 0x20400000 0 or 0x2042 <axis-ID> 0

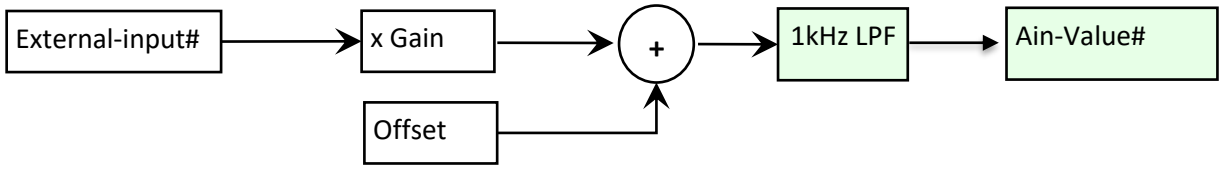
### 4.4.5. Analog I/O

Analog input can be used for target signals so that no software controlling is necessary (a default macro is required for configuration). Analog input has an internal 1kHz low pass filter.

Table 5.4.5 : Description of analog inputs

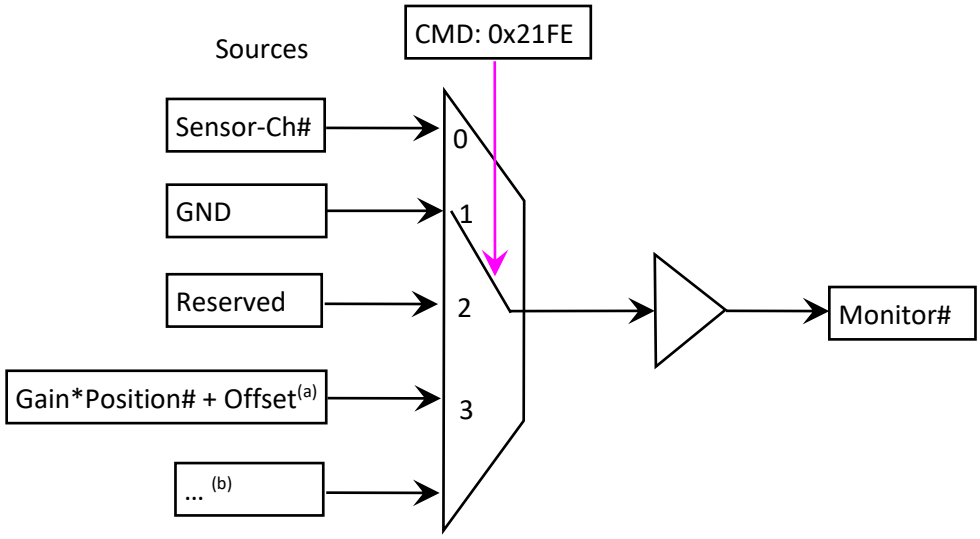
| Controller | Number of inputs | Index |
|---|---|---|
| **EBD-060310 / EOD-060310/ EBD-060310 / EOD-060310** | 1 | 0 |
| **EBD-060320** | 2 | 0, 1 |
| **EBD-060330** | 3 | 0, 1, 2 |

Calculating method:




Analog-Output:

The analog output can used as a monitor-signal.




Note (a): valid only for controller EBD-0603**

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

Option <3> for the EBD-0603** controller:

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

### 4.4.6. Event

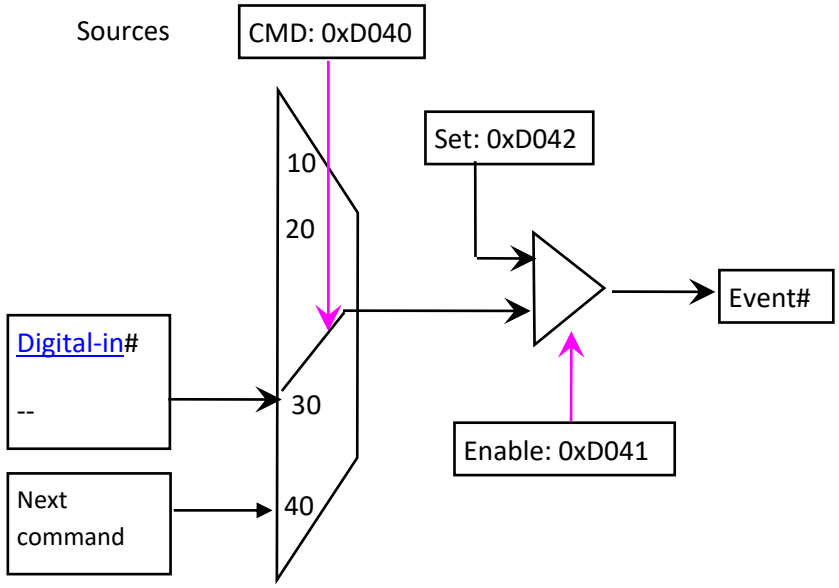
Event is used for triggering data-recorders and/or motion (synchronizing the operation). One event signals (index 0) is available. To use an event, it should be first configured and then connected to data-recorders.

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

Example: Event0 to the digital input.

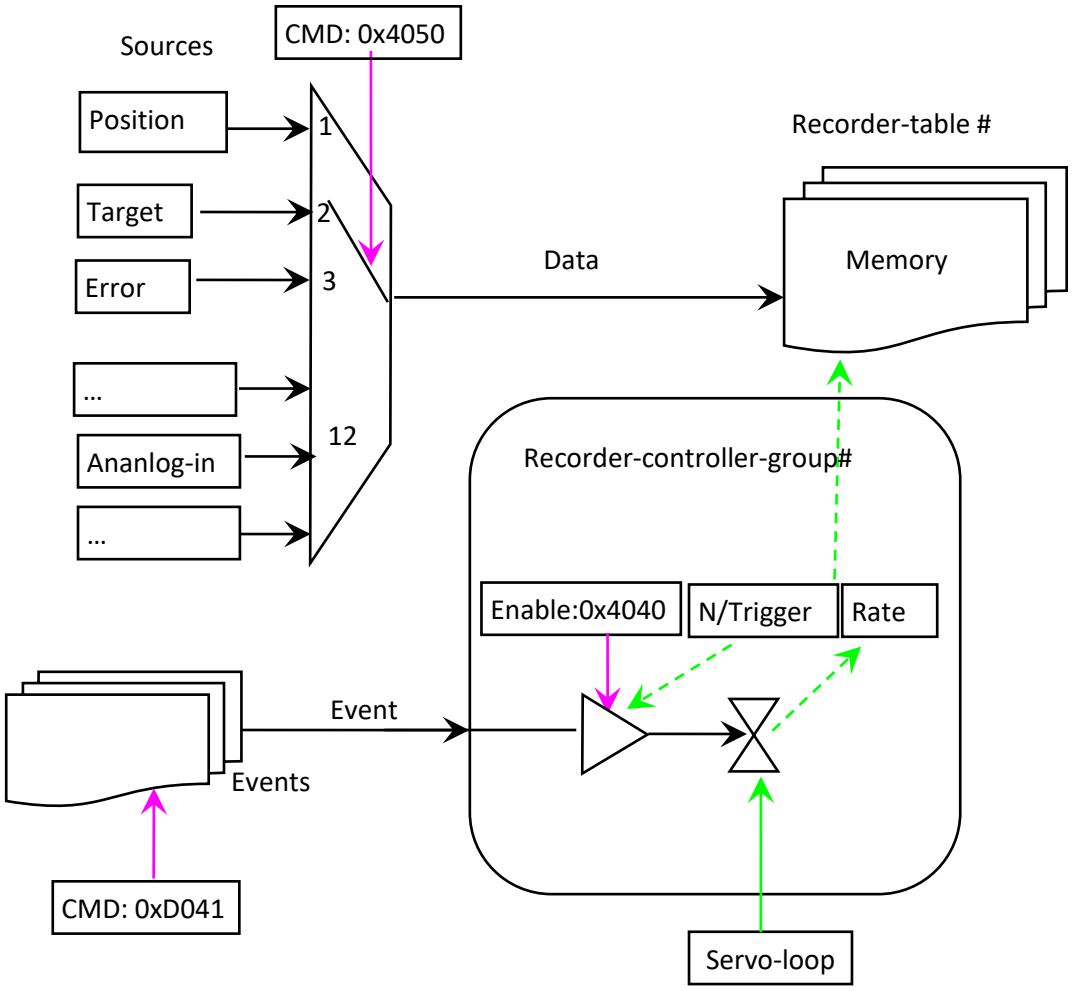
0xD040 0 30 1 (30: digital in)
0xD041 0 1 (1: enable)

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




Note: Digital input is only valid in EBD-060310/EBD-060310. Signal detection/delay timing is 20us.

### 4.4.7. Data-Recorder




There are 2 data recorders to store data synchronously in memory which can then be read-back by software. The memory size of each recorder is 512 DW (double-word).

To use data recorders:

1) select data sources for recorder with command 0x4050. Command parameters are:
   <recorder-index> <source-ID> <source-channel>
   For example: To select recorder 0 with axis position, and recorder 1 with analog input
   0x4050 0 1 0 1 12 0
   Note: use "***?0xFFF9 0x4050***" to see valid settings

2) enable recorder.
   0x4040 0 1

3) enable and set event.
   0xd041 0 1 (enable event)
   0xd042 0 1 (set event )
   Now data will be saved at each servo-loop (for rate = 1).

4) check recorded data length.
   ?0x4042 0 (Read-back command)
   In case of 50k servo-loop and rate=50, to fill up 512 data-buffer, it takes 512ms.

5) when finished, read data.
   ?0x4011 0 0 512 (read 512 data out of table 0, from position 0)
   ?0x4011 1 0 512 (read 512 data out of table 1, from position 0)

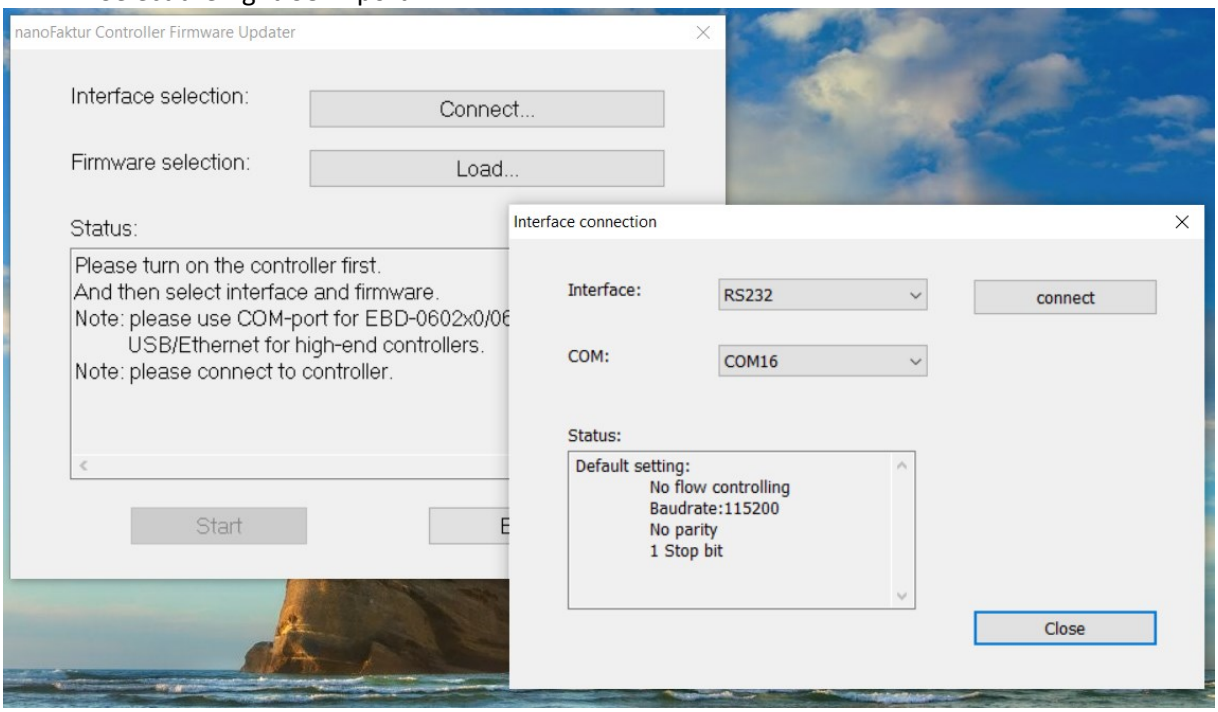
Example 2: recorder PID-Target and position triggered by command:

| | |
|---|---|
| 0xd041 0 0 | (Disable event) |
| 0x4040 0 0 | (Disable recorder) |
| 0x4050 0 7 0 1 1 0 | (Config recorder source) |
| 0x4040 0 1 | (Enable recorder) |
| 0xd041 0 1 | (Enable event) |
| 0xd040 0 40 0 | (Event option: triggered by next command) |
| 0x2003 0 0.1 | (Relative move and start-recorder) |
| ?0x4042 | |
| ?0x4011 0 0 512 | |

### 4.4.8. Firmware update

Power on the controller and start software: nFUpdater.exe
Note: the current controller can't use the Ethernet interface for firmware update.

1. Select the right COM-port




2. Load the firmware and then "Start"

After a successful execution, controller will be restarted, and the new firmware will be working.

Note: after update, please check parameters with nFControl.exe and save them with "0x6003 100" (since the updated firmware may have new parameters, which confirms controller's states and then be active).

### 4.4.9. 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.

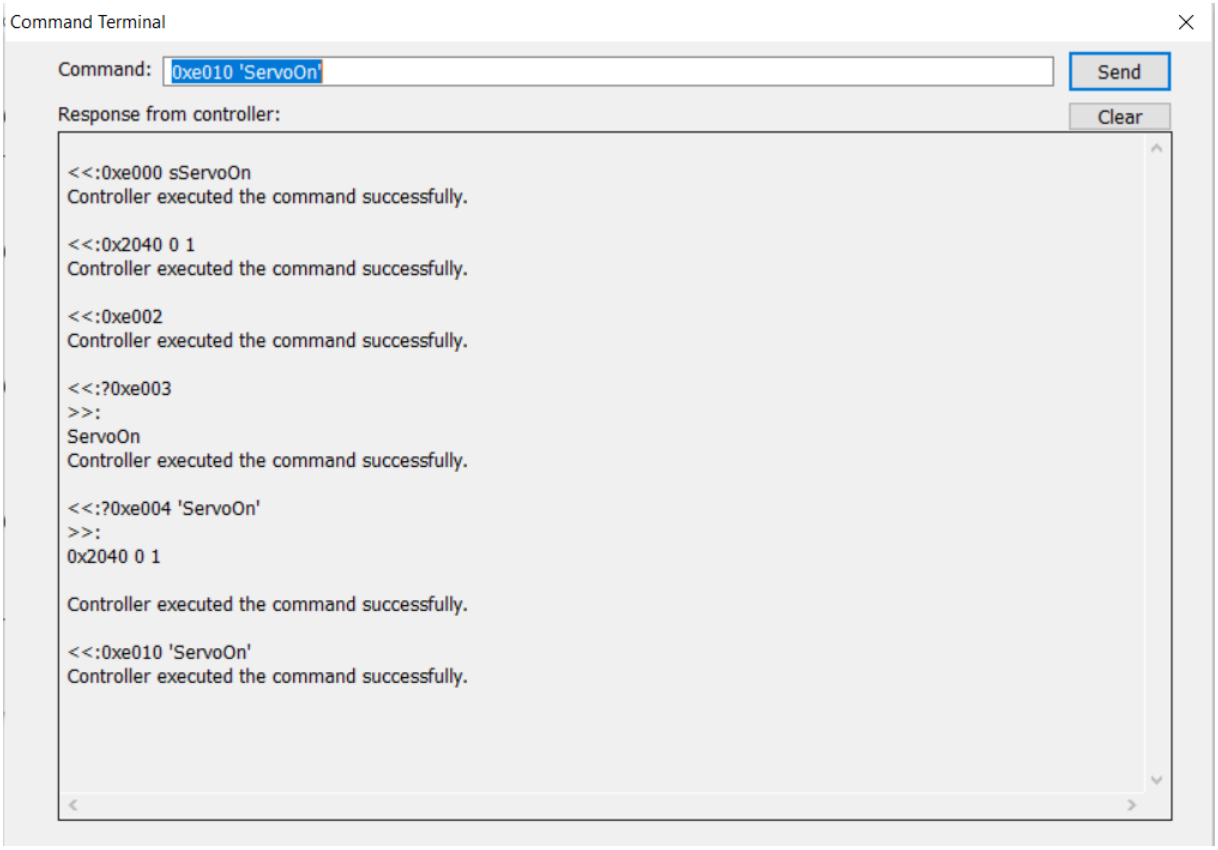
Table 5.4.12 Command for Macro function

| Command ID | Syntax (read/write) | Function description | Comment |
|---|---|---|---|
| 0xE000 | -/ <string: name><br>write only | Start recording a new macro | The old macro will be replaced when a same name is given.<br>Only saves command from one interface. |
| 0xE002 | -/ <none><br>write only | End of recording | |
| 0xE003 | <none> / -<br>read only | List all macros | |
| 0xE004 | -/ <string><br>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><br>write only | Delete macro | |
| 0xE010 | <none> /<string>[<times>] | Start macro / get running status | <times> is optional, 1 when not set. |
| 0xE011 | -/ <none><br>Write only | Stop running macro | |
| 0xE012 | -/ <ms><br>Write only | Delay millisecond | Only used in macro function |

Note: Name is case sensitive, maximal 8 chars.

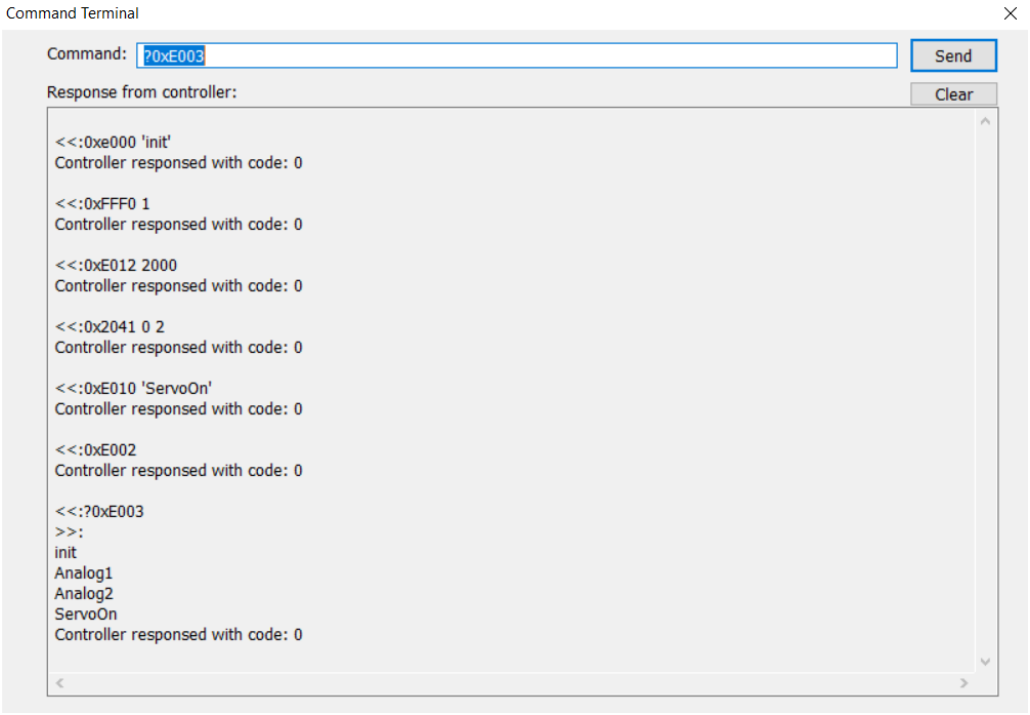
Example 1: define a macro which turns servo-on

```
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 |
| 0x2041 0 2 | --- select the 1st analog input as target of axis1 |
| 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:

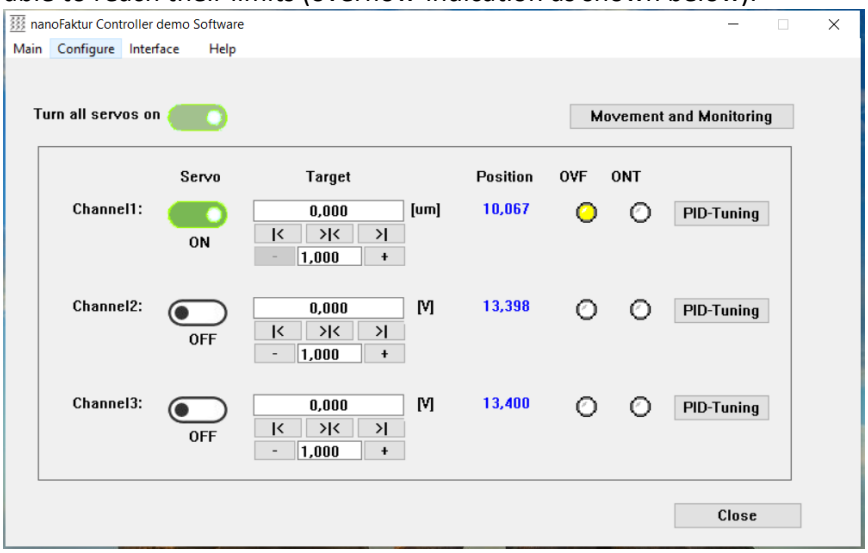
```
0xE010 'Main'
```

Note:

- Macro name is a string, please pay attention for the special character ' '
- Macro name is case sensitive.

### 4.4.10. 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 case of strain-gauge sensors, the offset can be shifted to another voltage with command 0x2150.

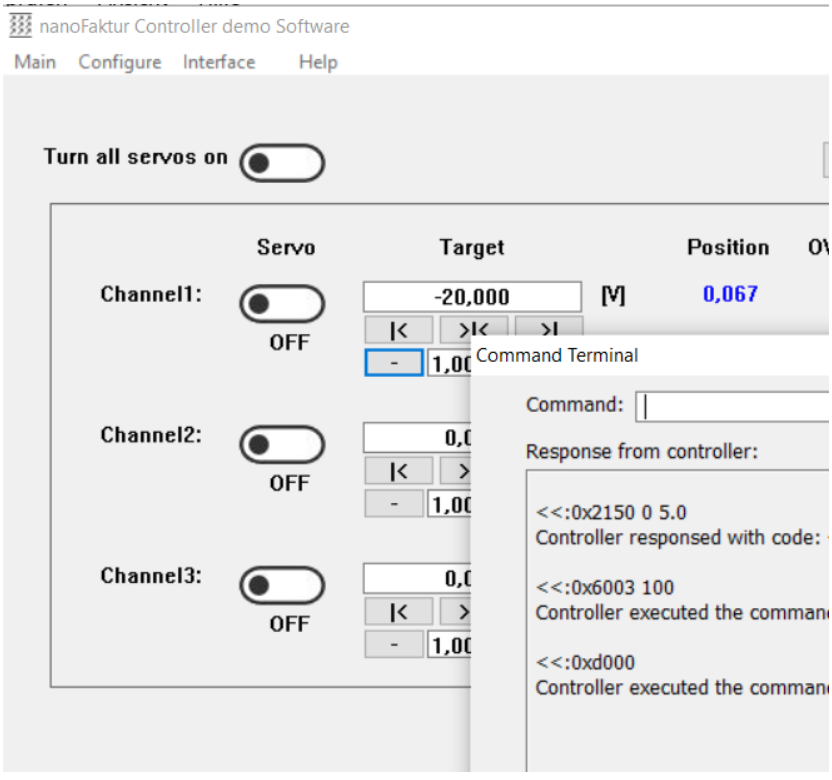
Step 1: Turn all Servo-mode to OFF

Step 2: Set voltage to example -20V

This means that the sensor value should be about 0.0 at -20V, depending on how much voltage is reserved for other 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 5.0

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

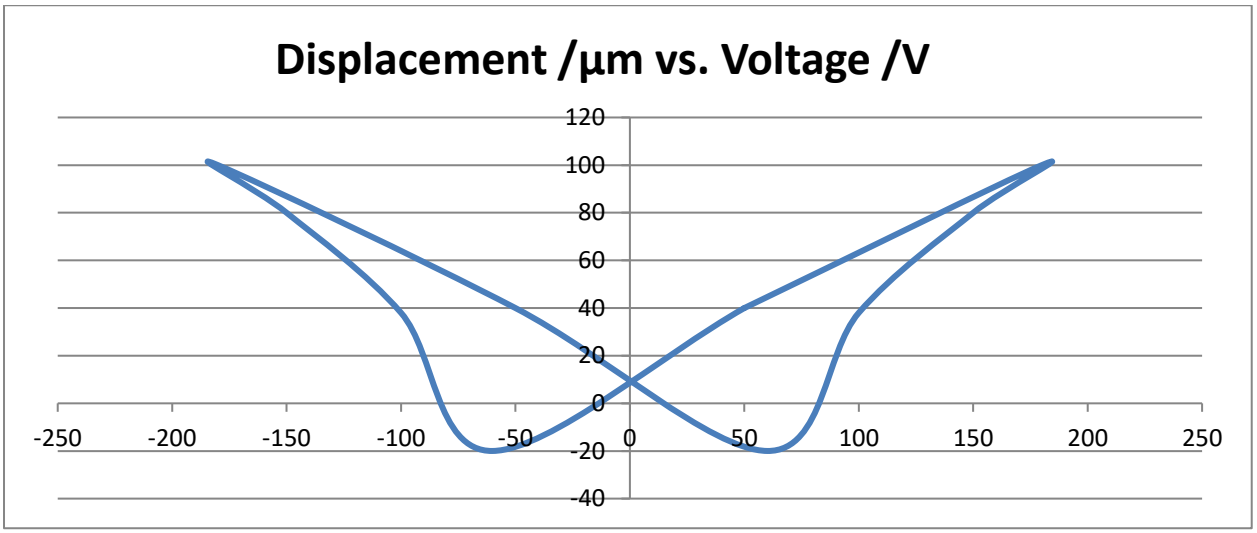
| | |
|---|---|
| 0x6003 100 | (Save parameters to controller) |
| 0xD000 | (Save parameters to IDChip of the stage) |

# 5. Explanation: Open- and Closed-Loop

## 5.1. 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.

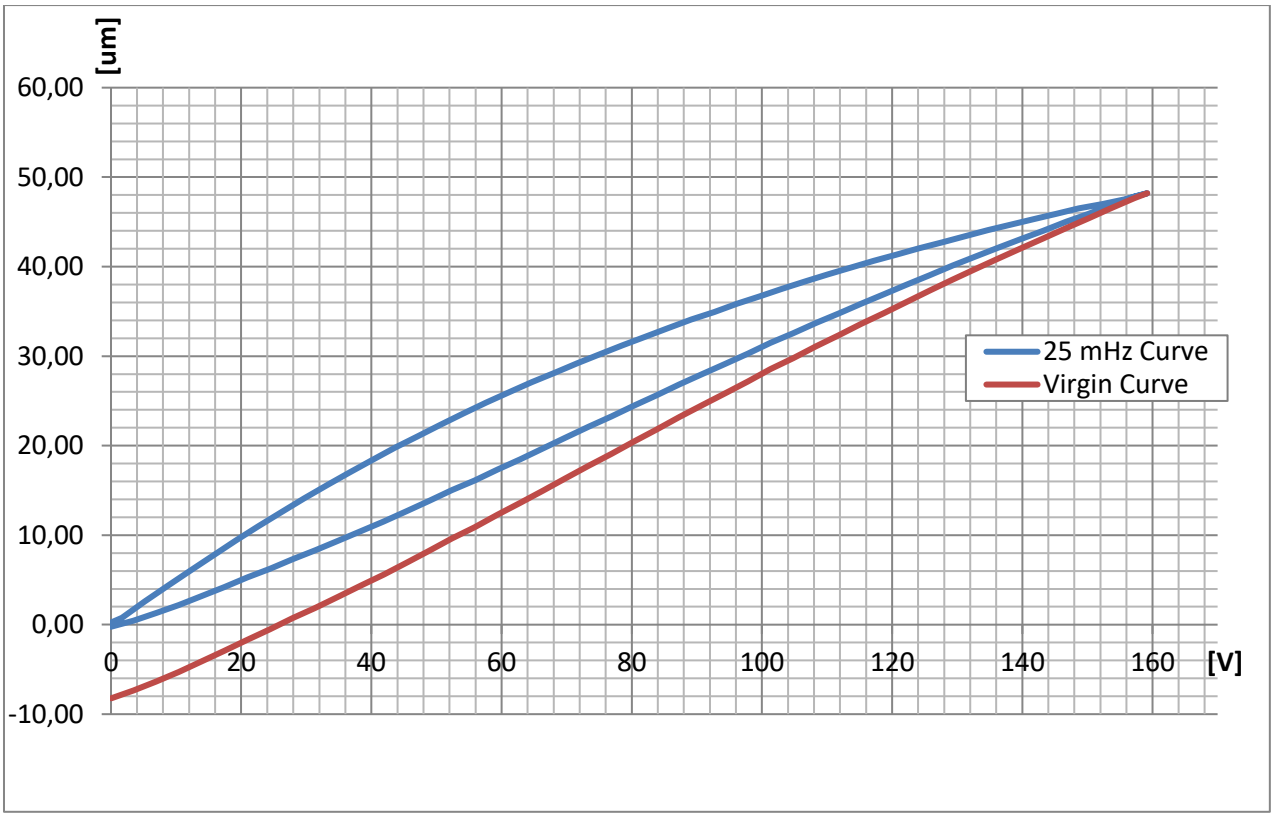
Typical behaviour of a $d_{33}$ PZT-actuator at an electrical field of ± 3 KV/mm (counter-polarization occurs at the low turning points):



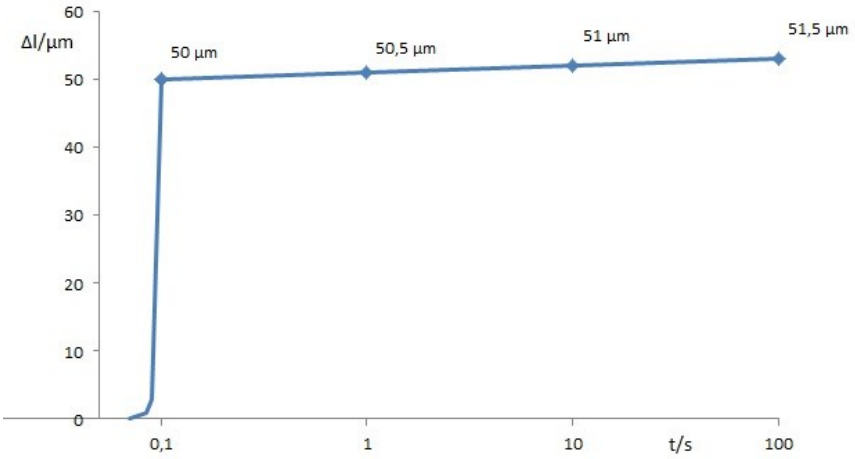

The symmetry of this butterfly-curve for plus and minus voltages and losses at repolarization lead to only unipolar usage making sense. Contraction at counter-fields can be added to 20%-30% of the nominal voltages. Maximum voltages and maximum counter-voltages depend on the kind of actuator. Follow manufacturer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

## 5.2. Voltage-Behavior of a $d_{33}$- PZT

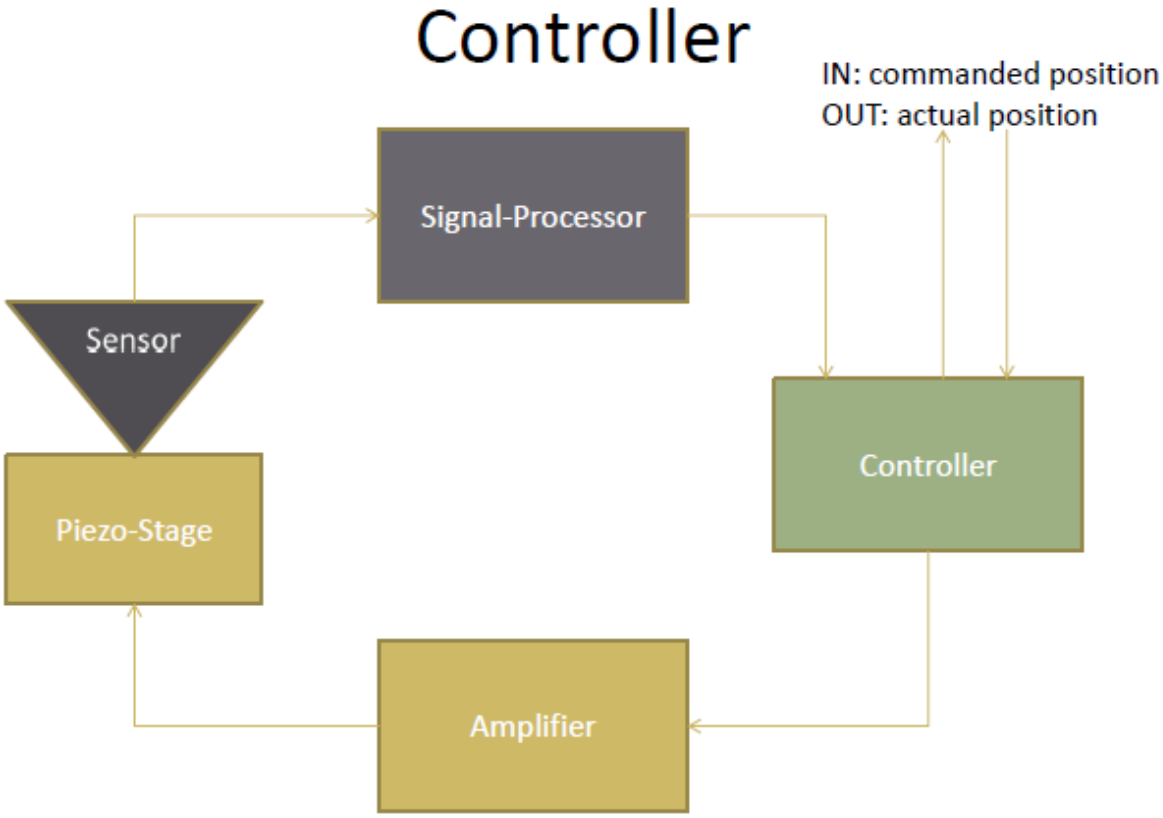



The curve above shows the typical behaviour of a nanoFaktur MPO-050050 PZT-multilayer stack. The stack was driven at voltages from 0 to 160 V, expansion monitored with a probe. When a PZT actuator is reactivated after a long rest, it performs a virgin curve. The Y-axis departure of the virgin-curve depends 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.



## 5.3. 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 50 µs for 2nd generation EBD-controllers.



# 6. Troubleshooting

Please first update firmware to the newest stand.

| Problem | Reason | Solution |
|---|---|---|
| USB connection | | Install the dedicated driver |
| TCP/IP connection: no interface found | IP address not configured | Enable DHCP when used in a sub-net<br>(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 |
| status LED red | Stage connecter not connected<br>Unknown stage type<br>System temperature problem<br>Controller is busy | Read stage status with "? 0x204F"<br>To reset status: " 0x204F"<br>Information: "?0xFFF9 0x204F" |
| Channel LED status | Off: Open loop<br>Blinking: Stage is Moving<br>Green: On-target | |
| Channel LED yellow | Overflow: Target can't reach and the voltage is at the limit. | Check soft-limit<br>Run auto-zero function |
| No movement | Voltage limit reached<br>Position limit reached<br>Small velocity value and Trajectory enabled<br>Target selection | Check open-loop soft-limit<br>Check close-loop soft-limit<br>Disable trajectory controlling<br>Check "?0x2041" |
| Monitor output | | Check the pin-definition |
| Oscillation | Notch-filter setting | Disable the filter |