Quick Start Guide PNDS3

For Plug & Drive Studio 3 Version 1.2.0

Quick Start Guide Version: 1.0.0

Contents

1 Document aim and conventions

Beside technical data, this document describes product use and function. For possible combinations with other Nanotec products, please ask your Nanotec sales partner. Before using the product, please note document font styles and conventions.

Underlined text marks a cross reference or hyperlink.

Example 1: Observe our safety notes.

Example 2: Download needed code templates from our website for EMEA / APAC or AMERICA.

Gray bold italics call out **menu paths, buttons, tab** and **file names**.

Example 1: Select **Home > Connect controller > CANopen**.

Example 2: In the **NanoJ** tab, select **NanoJ project** and open **Analog Input.cpp**.

Plain italics mark *Freehand entries* and *foreign-language* expressions. They also emphasize words of critical weight. Alternatively, bracketed exclaim marks(!) give critical weight.

Example 1: Enter *Plug & Drive Studio*. In addition to users (= *Nutzer; usuario; utente; utilisateur; utente* etc.), this document also addresses:

- Third-party users (= *Drittnutzer; tercero usuario; terceiro utente; tiers utilisateur; terzo utente* etc.).
- End users (= *Endnutzer; usuario final; utente final; utilisateur final; utente finale* etc.).

Example 2: Protect yourself, others and your equipment. Follow our *general* safety notes that are generally applicable to *all* Nanotec products. Also follow the *specific* safety notes that apply to *this* specific product.

`Courier` marks `code blocks` or `programming commands`.

Example 1: Via Bash, call `sudo make install` to copy shared objects; then call `ldconfig`.

Example 2: Use the following NanoLibAccessor function to change the logging level in NanoLib:

```
//
        ***** C++ variant *****
 void setLoggingLevel(LogLevel level);
```

The verb *to co-click*

Co-clicking means a mouse click by secondary key to open context menus etc.

Example 1: Co-click the file, select **Rename**, and rename the file.

Example 2: Co-click the file to check and select **Properties**.

Numerical values

Numbers appear in decimal. Hexadecimal notation ends in subscript *h*. Objects in the object dictionary notate in hexadecimal as `<Index>:<Subindex>`, non-notated subindices as 00_h. Example: $1003_h:05_h$ is subindex 5 in object 1003_h. And 6040_h is subindex 00 in object 6040_h.

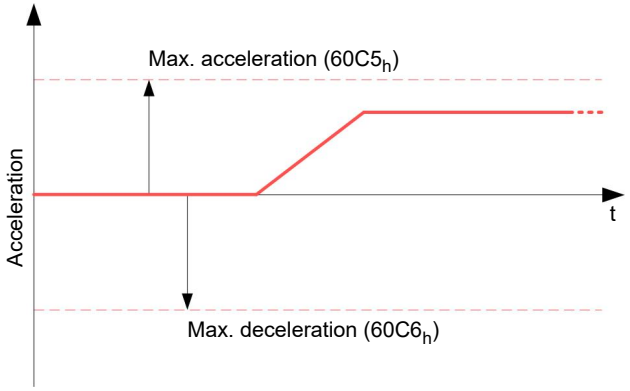
Bits

Each object bit counts up from LSB (bit number 0), such as data type *UNSIGNED8*:

	MSB							LSB	
Bit Nummer	7	6	5	4	3	2	1	0	
Bits	0	1	0	1	0	1	0	1	≙ 55_{hex} ≙ 85_{dec}

Count direction (arrows)

Illustrations always count arrow-wards; both example objects 60C5$_h$ and 60C6$_h$ are thus positive.



Versions

Document version	Date	Changes	Product release
1.0.0	12/2021	Edition	1.2.0

2 For your safety

Before product use, please ensure that all users read, understand and follow the instructions in this document fully.

2.1 Warn and risk levels

Please note that our hazard warnings, alert symbols and signal words mark different risk levels.

WARNING

WARNING warns of *possible danger of death!*

Grave injury / death **possible**.

► Instruction against **life-threatening** user errors.

CAUTION

CAUTION warns of *possible physical danger!*

Minor / moderate injury possible.

► Instruction against **unhealthy** user errors.

NOTICE

A NOTICE warns of wrong operation.

Material or ecological damage possible (not strictly injury).

► Instruction against **destructive** user errors (= mere material risks).

Note: Explains or simplifies a process by additional information.

2.2 General safety notes

Protect yourself, others, and your equipment. Observe our **general** warning messages that apply to **all** Nanotec products.

WARNING

Grave injury or death from product misuse!

► Handle the product as a qualified expert only.

► For due safety, follow valid OEM instructions.

2.3 Specific safety notes

For due protection, observe our **specific** warning messages that apply to **this** specific product.

WARNING

Grave injury or death from product abuse for safety purposes!

- ► **Never** integrate / use the product as a safety component.
- ► If you install the product: Warn customers expressly by safety note against risks and abuse.
- ► **Instantly** share our safety notes / instructions with your customers.

2.4 Additional safety notes

If handling **third-party** products, observe our **additional** warning messages and valid OEM instructions.

WARNING

Grave injury or death from unfit third-party equipment!

- ► Fulfill the system and hardware requirements.
- ► Use compatible equipment only.
- ► Follow valid OEM instructions.

3 Before you start

Before product use, you need to prepare the PC and verify product intent / limits. Via online help, you can learn how to install and set up projects and how PNDS3 runs. A reference chapter explains each object dictionary item for Nanotec controllers. Observe the safety notes in the manual (www.nanotec.de).

3.1 System and hardware requirements

WARNING

Grave injury or death from realtime-*in*capable environment (Windows PC, etc.)!

► Keep the product away from time-sensitive applications or synchronous multi-axis movement.

► **Never** integrate it as a safety component in a product or system.

Plug & Drive Studio 3 (PNDS3) needs 64-bit operating systems and supports all Nanotec products with CANopen, Modbus RTU or USB interface. Nanotec recommends controller firmware *FIR-v2139* or newer. Please find the current version in the **Firmware** folder. PNDS3 offers a special control for firmware update.

PNDS3

v1.2.0

64-bit OS requirements

- Windows 10
- .NET Framework 4.8
- Display resolution 1920x1080

Fieldbus adapters / cables

- **CANopen:**
  - IXXAT USB-to-CAN V2
  - Nanotec ZK-USB-CAN-1
- **Modbus RTU:**
  - Nanotec ZK-USB-RS485-1 or equivalent USB-RS485 adapter
  - USB cable via virtual comport (VCP)

3.2 Intended use and audience

NOTICE

Damage from unskilled staff!

► Use the product only for the purpose described in this document.

► Restrict use to expert staff only.

► Follow valid OEM and system prescriptions for all equipment involved.

As free software for easy Nanotec drive commissioning, Plug & Drive Studio 3 (PNDS3) supports Nanotec CANopen and USB products. The underlying operating system / hardware (PC) is **not** real-time capable. **Never** use PNDS3 for time-critical or synchronous multi-axis motion **nor** integrate it as a safety component in a product or system.

Add proper warnings and instructions for safe use / operation to each end user product with a Nanotec-produced component. Submit any Nanotec warning directly to end users. The product addresses skilled experts in industry use cases alone. Expert means:

- Training / experience in motor and controller handling
- Understanding this document plus Nanotec drive manuals
- Knowledge of valid regulations

3.3 Scope of delivery and warranty

PNDS3 comes as a **.zip* folder from our download website for either EMEA / APAC or AMERICA. Duly store and unzip your download before setup. The product package contains:

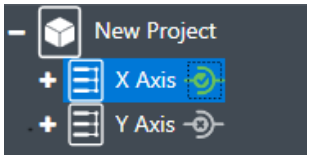
- Software as an executable file
- Project templates
- Current firmware release
- Online help file

For scope of warranty, please observe our terms and conditions for either EMEA / APAC or AMERICA. **Note:** Nanotec is not liable for wrong quality, handling, installation, operation, use, and maintenance of third-party equipment! Follow valid OEM instructions.

4 Your product

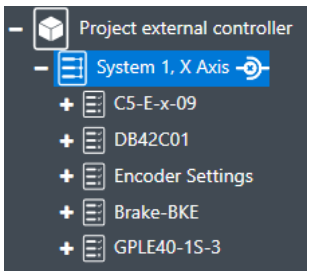
With PNDS3, you parametrize and commission Nanotec drives. Using templates for various Nanotec drives, you can add your own projects, systems and modules to the modular user interface. The software comes with a default folder structure (*Project, System, Module Group, Module,* etc.).

Project (default folder *Projects*)



You manage all settings and device parameters in projects, save these as a file and im- / export them, say, as a template. Such a reusable **Project** can have multiple systems, say, the axes of a machine.

System (default folder *Systems*)

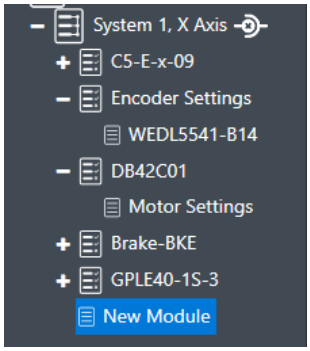


In a project (here: external controller), you create and store drive systems (here: X-axis). Each is im- / exportable as template.

You can extend such a reusable **System**, of at least motor and controller, by modules or module groups for encoder, gearbox, brake, settings, parameters, etc.

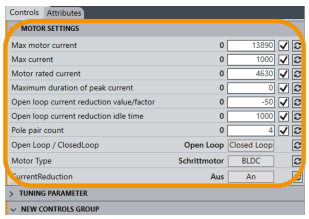
By parameters, sortable / poolable into several modules or module groups, you quickly control all system elements.

Module (default folder *Modules* / *Module Groups*)



A module contains parameters or controls (groups) and is im- / exportable, single or grouped, as template.

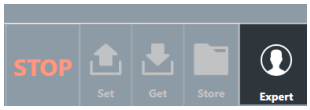
Controls Group (default folder *Groups*)



A **Controls group** pools single device parameters (objects from the dictionary in the controller) and / or **Special controls**.

You im- / export such a control group together with set values, say, as template.

User Level

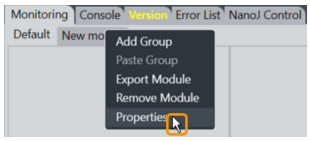


Via **User level**, you govern user rights for the following roles:

- **Expert**: Project owner with all rights. May create and edit projects, rights, visibilities, etc. Governs via **Properties**, for each single parameter up to a complete **Controls group**, *who* may see and edit exactly *what*.
- **Engineer**: May change device parameters, but can't edit a project.

- **Mechanic**: Similar to **Engineer**, but often gets fewer editing rights from **Expert**.

Property editing



Simply co-click an element, select **Properties**, insert a visible name, version number, and description: This way you create your individual user interface.

5 Installation and adapter

Install the software, set up the adapter – and PNDS3 is ready to go. You find PNDS3 software online as a zip download.

1. Open the website **Nanotec > Products > Software > Plug & Drive Studio 3**.
2. Download and extract the product zip file.
3. Run the executable file **PNDS3.exe**.
4. Only with PNDS3 installed: Prepare your fieldbus adapter (see below).

CANopen

1. Decide: **Ixxat USB-to-CAN**? Or **Nanotec ZK-USB-CAN-1**?
2. For **Ixxat USB-to-CAN:** Download the driver (www.ixxat.com/); install it by hand.
3. Connect the adapter to the computer. For **Nanotec ZK-USB-CAN-1:** Wait for self-installation.
4. Via correct cable (see product manual): Connect the installed adapter to the controller.

USB: Nanotec Virtual COM-Port (VCP)

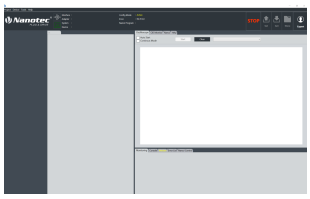
1. Connect the voltage supply to the controller and switch it on.
2. Via correct USB cable: Connect the PC to the controller (= "mass storage device").
3. In Explorer > Controller directory: Select `cfg.txt` (= `pd4ccfg.txt`for a PD4C).
4. Open the file via text editor (Notepad etc.).
5. Add the lines `2102|=0x100000` and `4015:01=0`. Save the file.
6. Restart the controller and check if its COM port appears in the device manager.

Modbus RTU

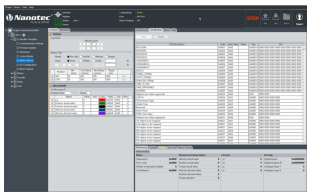
1. For **Nanotec ZK-USB-RS485-1**: Connect the adapter to the computer and wait for self-installation.
2. For **other equivalent adapters**: Follow valid OEM instructions to install the driver.

6 User interface (UI)

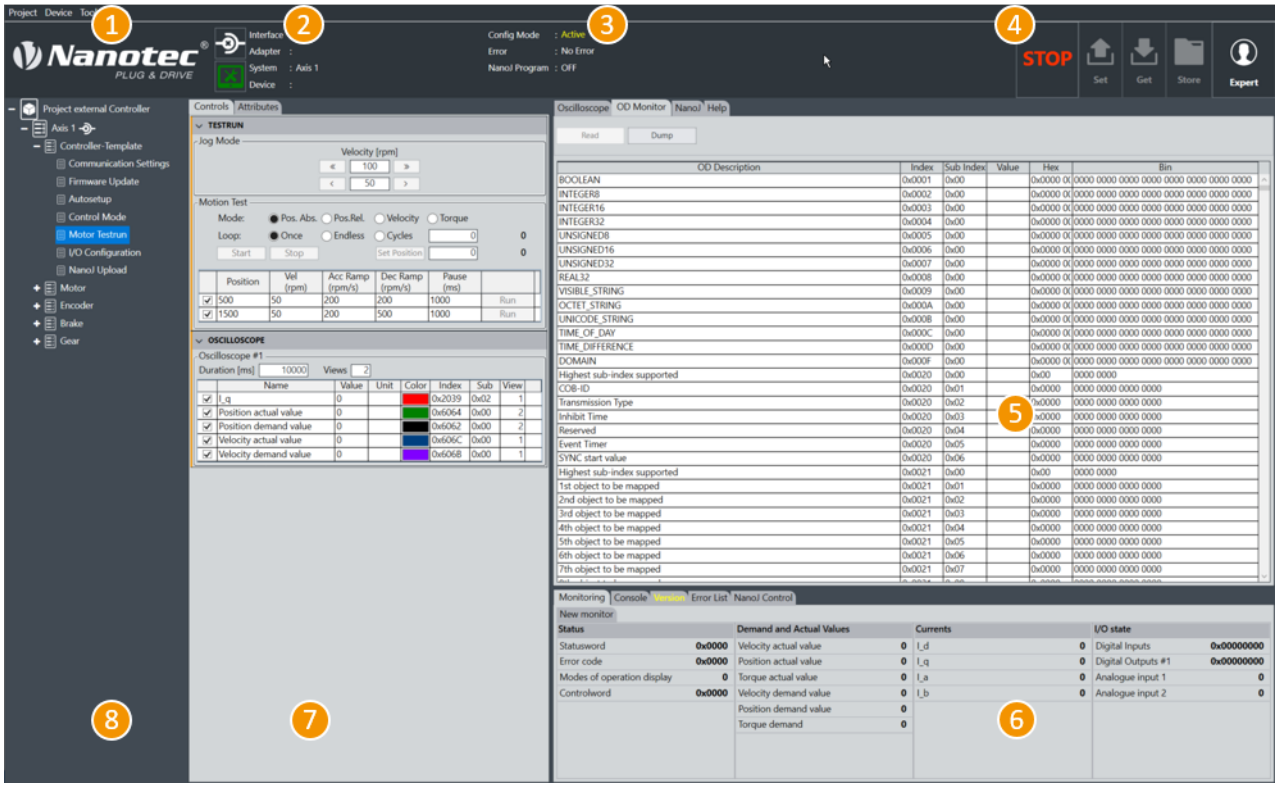
Thanks to flexible areas and windows, fitted into the main window or usable stand-alone, you can master a wide range of tasks. Before product use, please familiarize yourself with the UI structure.



When PNDS3 starts for the first time, the user interface appears rather empty. But that is intentional ...



Once you create or load a project, the interface fills up according to your needs. This way, you design your own UI.

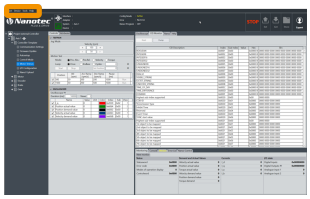


- Header for main menu (1), adapter linkage (2), brief status (3), and operation buttons (4).
- Display wall (5) for oscilloscope, object directory, help etc.
- Status board (6) with tabs for monitors, error lists etc.
- Work desk (7) for user controls etc.
- Project (or side) bar (8) for systems etc.

6.1 Header (1-4)

As a prominent layout bracket on top in the user interface, the UI header contains all basic functions and commands relevant to operation.

Main menu (1)



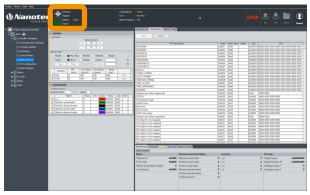
Leftmost above the header, you find the main menu for projects, devices, tools and help.

Project: Loads new – and saves, reopens, edits – existing projects.

Device: Reads, writes, and saves device parameters. Governs NanoJ programs and fieldbus network (with CANopen).

Help: Opens the online help and PNDS3 version info.

Linkage buttons (2)

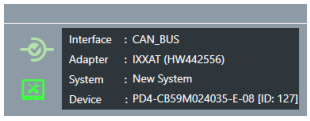


With these buttons / icons in the header's left, you link or unlink current interfaces / adapters / systems / devices.

Connect: Powers / unpowers the connected adapter (cf. Connecting to adapter).

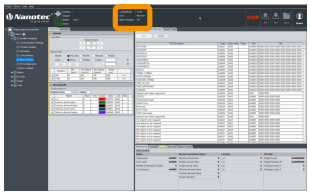
Configuration mode: Toggles as follows:

- **Configuration mode:** In this mode you may edit the project, add or delete systems and modules.
- **Commissioning mode:** In this mode you may change parameters and configuration values, but may *not* edit a project.

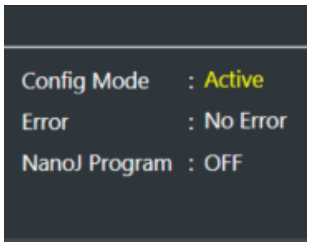


Connection information: Displays information on connected adapter, system and device.

Brief status (3)

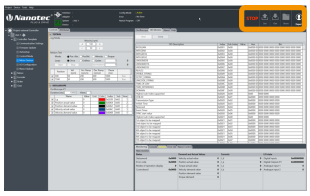


In mid-header, you can see at a glance if config mode and a NanoJ program are involved and which errors are found.

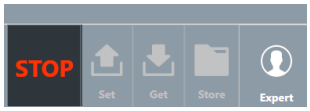


Example: An external controller in configuration mode (= *active*) with a correct run (= *no error*) and no NanoJ (= *off*).

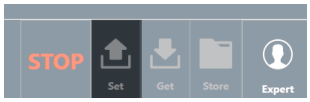
Operation buttons (4)



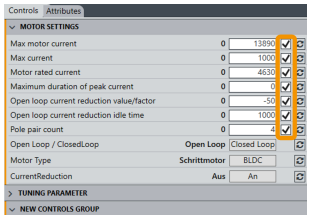
Rightmost in the header, these buttons toggle motor current, operation parameters and controller values.



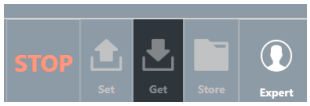
STOP: Cuts the motor current **without(!)** safe shutdown.



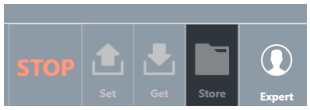
Set: Transmits *the selected* parameter values to system-connected controllers.



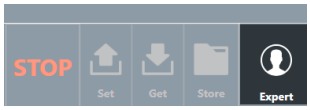
Note: You can select parameters to be set by ticking them.



Get: *Reads* the values of system-connected controllers.



Store: *Stores* **Set**-transmitted values of system-connected controllers.



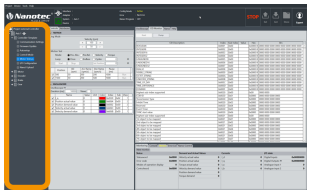
User level: Grants project rights for either:

- **Expert**: Project owner with all rights. May create and edit projects, rights, visibilities, etc. Governs via **Properties**, for each single parameter up to a complete **Controls group**, *who* may see and edit exactly *what*.
- **Engineer**: May change device parameters, can't edit projects.
- **Mechanic**: Similar to **Engineer**, often with fewer editing rights.

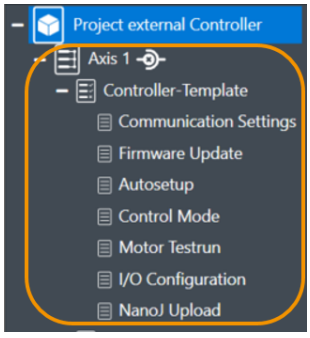
6.2 Project bar (8)

This side bar diplays your loaded project as a tree list by which you create the user interface. **Note:** Depending on assembly, you can check connections and attributes of all tree list items in the work desk (7).

Tree list



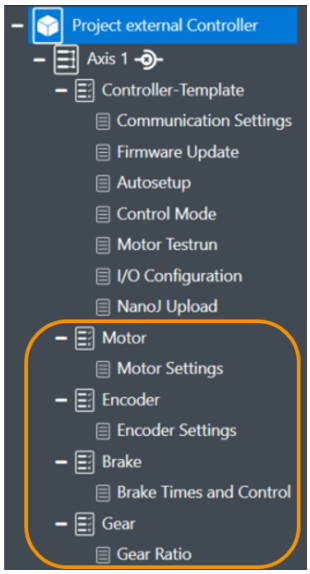
You find the project bar in the very left of the user screen.



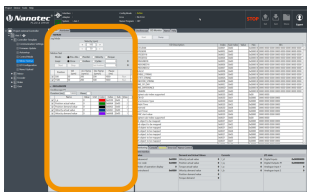
A project (here: for an *external controller*) tree-lists all systems and the items therein (see also Project setup). *One* project and *one* system are minimum; further items are optional and later on define the entire UI layout.

Our example project (= blue) contains the system *Axis 1* (= orange).

System *Axis 1* contains the module group *Controller template* with individual modules for *communication settings, firmware update, auto-setup,* etc.

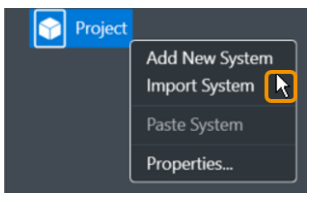


Further module groups are, say, *Motor, Encoder, Brake* and *Gear*, with their respective modules for settings, controls and parameters.



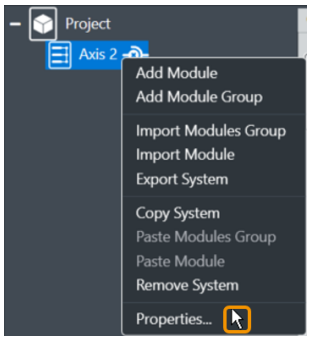
For each module, you may add one or more controls groups to the work desk (7) further to the right.

Project > System

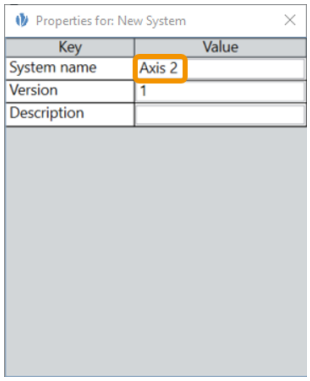


A system represents a motor with controller, that is, one per motor in a multi-axis application.

1. To set up a system: Co-click the project.
2. In the context menu: Either create a new system via **Add new system**.
3. Or fetch an existing one via **Import system**.

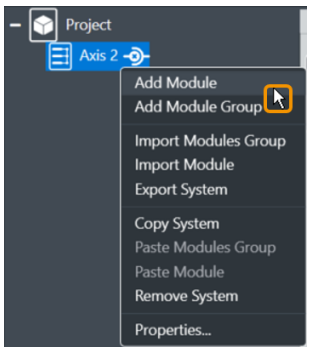


4. A new node (= blue) appears in the tree list.
5. To name it: Co-click the node, select **Properties**. **Note:** You can edit *any* object via **Properties**.



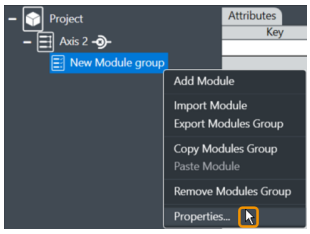
6. In the pop-up: Name the system as needed (here: *Axis 2*).
7. If needed: Versionize and describe the system.
8. After last entry: Set a tab stop (so that all is stored).
9. Only then: Close the pop-up.
10. Assemble the system with module groups (see below).

Project > System > Module group

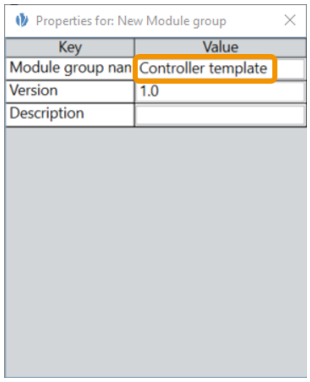


A module group bundles *several* motor functions (= modules). Depending on assembly, you can check its connections and attributes in the work desk (7).

1. To set up a module group: Co-click the system (here: *Axis 2*).
2. In the context menu: Either create a new module group via **Add module group**.
3. Or fetch an existing one via **Import module group**.

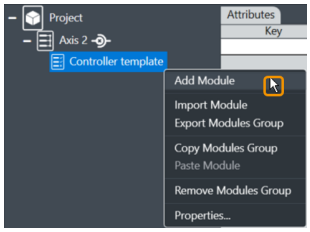


4. A new node appears in the tree list.
5. To name it: Co-click the node, select **Properties**. **Note:** You can edit *any* object via **Properties**.



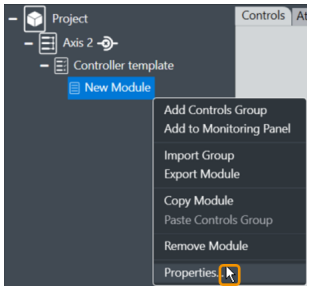
6. In the pop-up: Name the module group as needed (here: *Controller template*).
7. If needed: Versionize and describe the module group.
8. After last entry: Set a tab stop (so that all is stored).
9. Only then: Close the pop-up.
10. Assemble the module group with modules (see below).

Project > System > Module group > Module

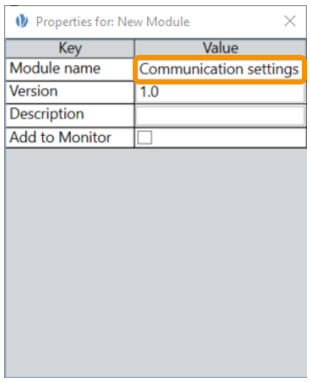


A module allows you to add a *single* motor function (= parameter set etc.). Depending on assembly, you can check its connections and attributes in the work desk (7).

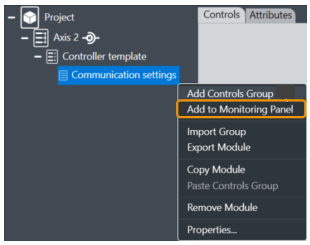
1. To set up a module: Co-click the module group (here: *Controller template*).
2. In the context menu: Either create a new module via **Add module**.
3. Or fetch an existing one via **Import module**.



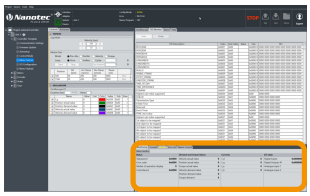
4. A new node appears in the tree list.
5. To name it: Co-click the node, select **Properties**. **Note:** You can edit *any* object via **Properties**.



6. In the pop-up: Name the module as needed (here: *Communication settings*).
7. If needed: Versionize and describe the module.
8. After last entry: Set a tab stop (so that all is stored).
9. Only then: Close the pop-up.

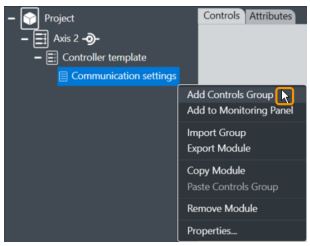


10. If needed: Add the module to the status board (6) as a monitor, via the context menu **Add to monitoring panel**.



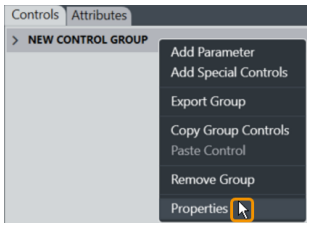
11. The module appears in the status board's **Monitoring** tab.
12. Assemble the module with controls groups (see below).

Project > System > Module group > Module > Controls group

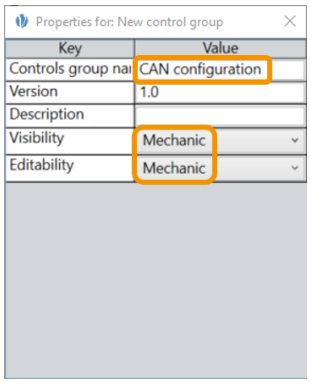


A controls group bundles individual operating elements or parameter sets.

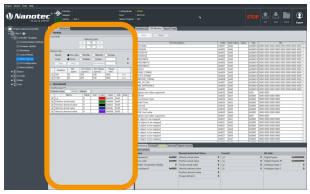
1. To set up a controls group: Co-click the module (here: *Communication settings*).
2. In the context menu: Either create a new controls group via **Add controls group**.
3. Or fetch an existing one via **Import group**.



4. In any case, the controls group appears in the work desk (7).
5. Right there: Co-click the group and its **Properties**. **Note:** You can edit *any* object via **Properties**.



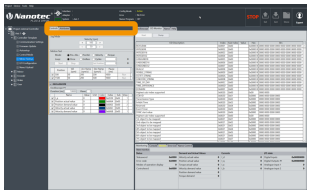
6. In the pop-up: Name the controls group as needed (here: *CAN configuration*).
7. If needed: Versionize and describe the group. **Note** the pull-downs for granted viewing and editing rights (here: both *Mechanic*).
8. After last entry: Set a tab stop (so that all is stored).
9. Only then: Close the pop-up.



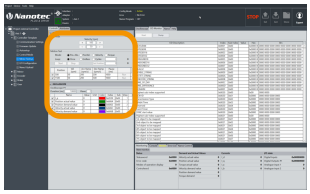
10. The controls group appears in the work desk.
11. **Note:** To delete a controls group, you must first delete each object therein.

6.3 Work desk (7)

At the workdesk, in the user screen's half-left, you edit the properties / contents / controls of your project tree. Depending on assembly, different tabs are above the worktable:



An **Attributes** tab accompanies all items (also module groups); **Bus settings**, by contrast, only the project itself. The **Connection settings** tab, finally, is for systems only; and **Controls** is only for modules. Each tab opens different aspects:

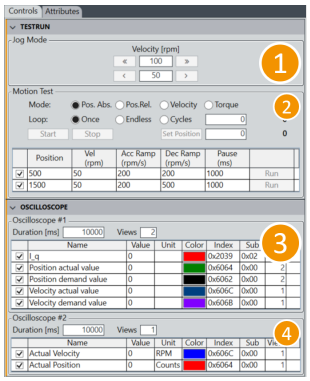


Controls groups Operator clusters

Parameters Operator values

Special controls Feature operators

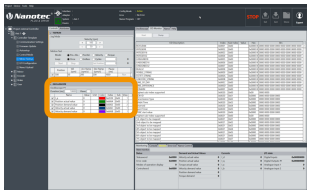
Complex controls Multi-level operators



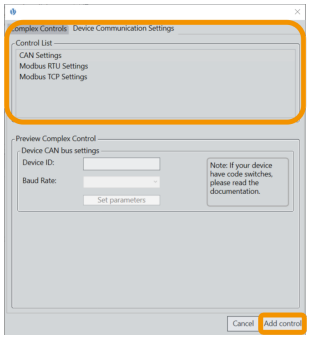
A workdesk for, say, motor testing could include the controls group *Testrun*, with special controls for *Jog mode* (1) and *Motion test* (2).

Plus: the controls group *Oscilloscope*, with special controls for *Oscilloscope #1* (3) and *Oscilloscope #2* (4).

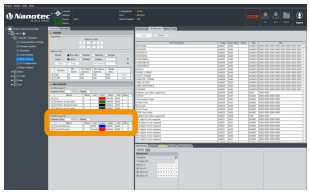
Basic principle



1. Co-klick a controls goup to open its context menu.
2. Either select **Add parameter** to parametrize the group.
3. Or select **Add special controls** to open a **Complex controls** list.

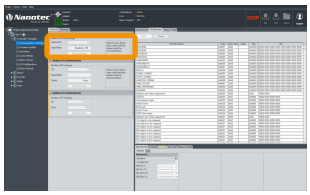


4. In the pop-up: Select the needed item by mouse / filter.
5. To confirm: Click **Add ...** / **Next** (depending on menu).
6. If possible: Edit the item. **Note:** No storage on **Cancel**.
7. Confirm each step with **Add ...** / **Next** (final step: **Finish**).

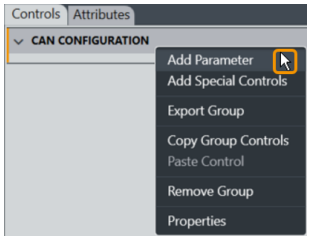


8. The item appears in the controls group. Parametrize there (if possible).

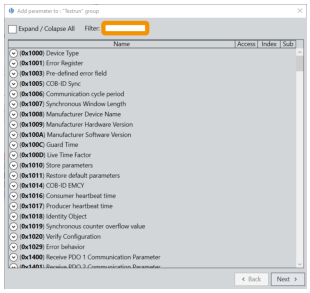
Controls group > Parameter (example: *device ID*)



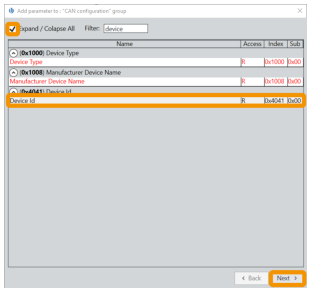
A parameter defines and monitors the system behavior. Depending on assembly, you can check its connections / attributes in the workdesk.



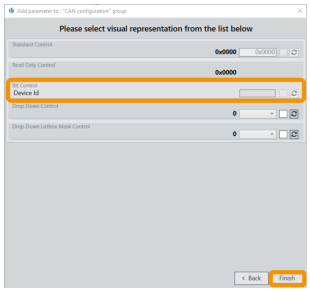
1. To set up, say, a device ID: Co-click the controls group (here: *CAN configuration*) and **Add parameter**.



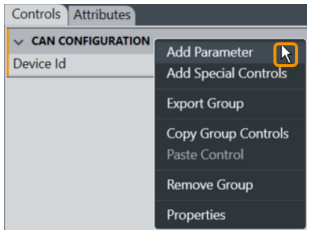
2. In the pop-up: Enter *device* or *0x4041* to filter for the **Device Id** object.



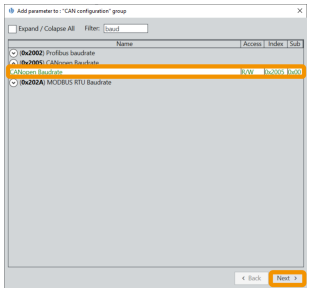
3. You may expand objects by mouse (or tick at **Expand all**).
4. Click **Device Id** and **Next** (if wrong: step **Back**).



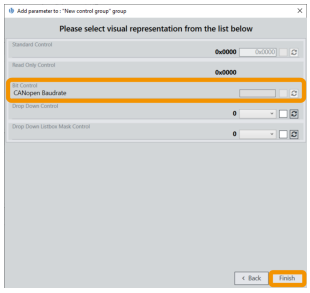
5. In the next pop-up: Edit the object and click **Finish**.



6. To set up, say, a baud rate: Co-click the controls group (here: *CAN configuration*) and **Add parameter** again.



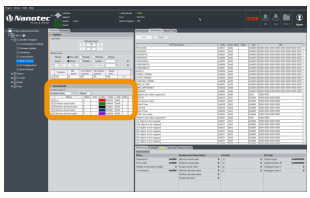
7. In the pop-up: Do filter, say, for *baudrate* or *0x2005*.
8. Click, say, **CANopen baudrate**, then **Next** and **Finish**.



9. In the next pop-up: Edit the object and click **Finish**.

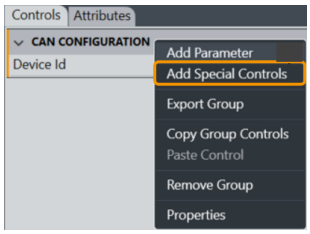
10. The baud rate appears in the controls group.
11. If needed: Assemble the controls group with special controls (see below).

Controls group > Special controls (example: *oscilloscope*)

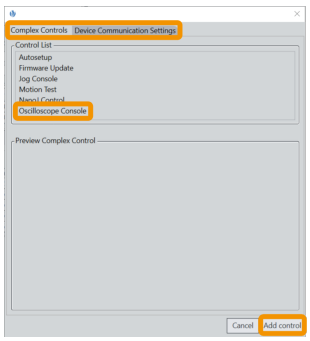


Special controls define and monitor (as macro collections) the system behavior. Depending on assembly, you can check their connections and attributes here in the work desk.

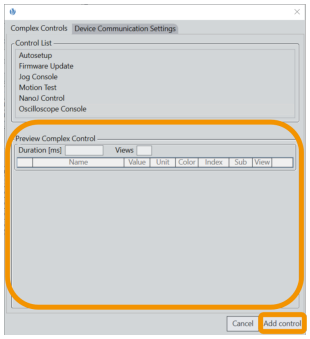
1. To set up a special control: Co-click the controls group.



2. In the context menu: Select **Add special controls**.

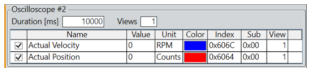


3. Decide: **Complex controls** tab? Or **Device communication settings**?
4. In the tab of choice: Filter / Select the needed item and **Add control**.



5. If possible: Check the manual setting options in **Preview complex control**.
6. Click **Add control**.

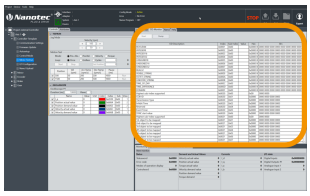
7. If possible: Edit the item and click **Finish**.



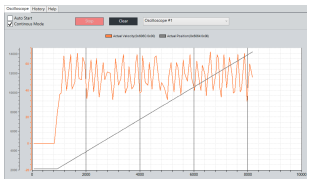
8. The special control (here: *Oscilloscope #2*) appears in the controls group.
9. For details: See special controls setup.

6.4 Display wall (5)

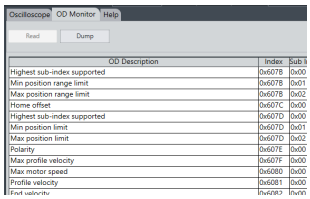
The display wall shows oscilloscope data, current OD values, and the help.



Several tabs facilitate navigation in the display wall, in the user screen's upper right.

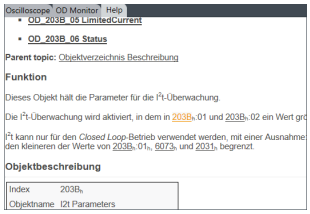


Oscilloscope: See **Special controls > Oscilloscope console**.



OD monitor: Lists all objects from the controller's dictionary, together with their current values. For updates: Click **Read**.

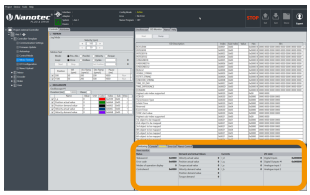
To save the list as a text file on the hard drive: Click **Dump**. Keep the text file with current values ready in case of support enquiries.



Help: Displays the description of the currently chosen element (OD object).

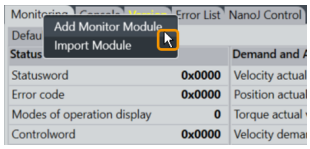
6.5 Status board (6)

Here, at the bottom right of the user screen, you track your system by monitors, error lists, NanoJ, etc.



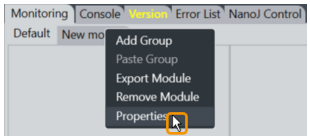
The status board is in the bottom right of the user screen. It has tabs for **Monitoring, Console, Version, Error list**, and **NanoJ control**.

Monitoring > Setting up a monitor (example: *Test*)

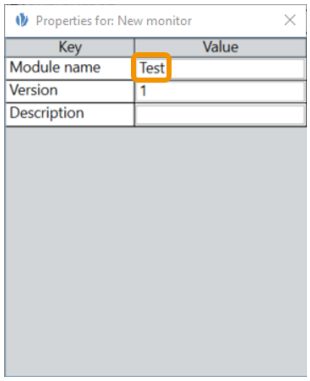


In the **Monitoring** tab, you combine either single or grouped monitors to track individual system behavior in real time.

1. To set up a monitor: Co-click the tab **Monitoring**.
2. In the context menu: Either create a new monitor via **Add Monitor module**.
3. Or fetch an existent one via **Import module**.

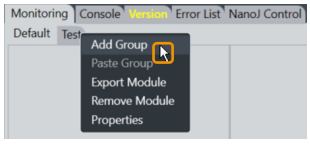


4. Co-click the new monitor tab and **Properties**. **Note:** You can edit *any* object via **Properties**.

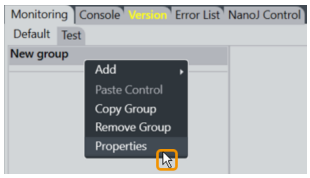


5. In the pop-up: Name the new monitor as needed (here: *Test*).
6. If needed: Versionize and describe the monitor.
7. After last entry: Set a tab stop (so that all is stored).
8. Only then: Close the pop-up.

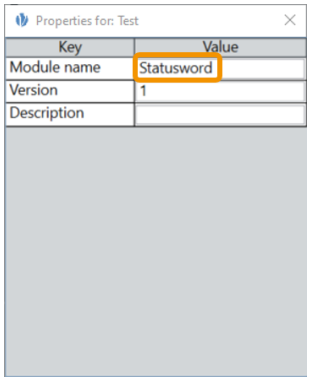
Monitoring > Setting up a group (example: *Statusword*)



1. Co-click the monitor of choice and **Add group**.

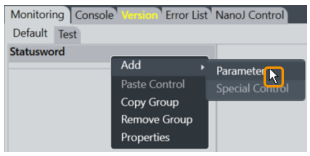


2. Co-click **New group > Properties**. **Note:** You can edit *any* object via **Properties**.

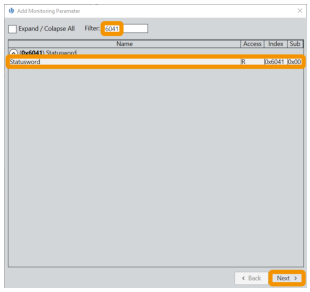


3. In the pop-up: Name the new group as needed (here: *Statusword*).
4. If needed: Versionize and describe the group.
5. After last entry: Set a tab stop (so that all is stored).
6. Only then: Close the pop-up.

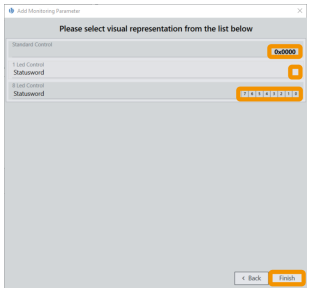
Monitoring > Setting up parameters (example: *Statusword* object format)



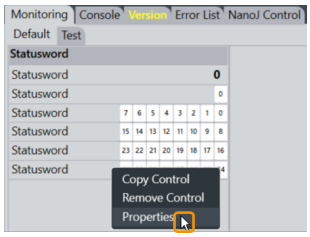
1. Co-click the group of choice and **Add > Parameter**.



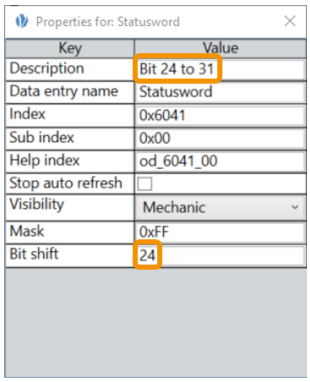
2. By filter entry: Find the object of choice (here *0x6041 Statusword*).
3. If needed: Select a sub-object (here: *Statusword*).
4. Click **Next**.



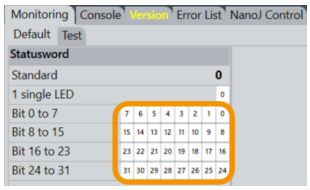
5. For (visual) object format: Select either *Standard 0x0000*.
   - Or Single *1-LED*.
   - Or *8-LED* row.
6. Click **Finish**.
7. Add all object formats of choice.



8. For 16 (or even 32) bits, you may combine several 8-LED rows.
9. By default, each LED row shows bit ***0*** to ***7***.
10. To show, say, bit ***24*** to ***31***: Co-click the LED row and **Properties**. **Note:** You can edit *any* object via **Properties**.



11. In the pop-up: Change the bit shift, say, to ***24***.
12. Set a tab stop (so that all is stored).
13. If useful: Add a description, say, *Statusbits 24 - 31*.

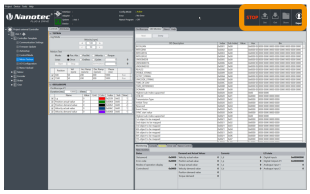


14. Change the bit shift for each 8-bit row needed.
15. Set a tab stop after each entry (so that all is stored).

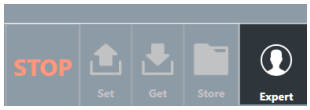
7 Project setup

Only in a project, you go first software steps and manage your devices, settings, connections, etc. **Note:** Ex works, in the software's Projects folder, there is a sample project each for motor-external and for NanoJ-integrated controllers. Nanotec recommends using these templates.

Grant expert rights

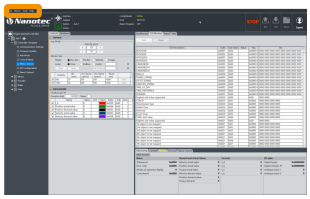


1. Connect the power supply and the cables to the controller.
2. In the user screen: Visit the operation buttons (4).

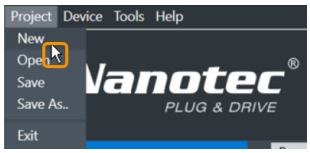


3. Via Button: Select **User > Expert**.

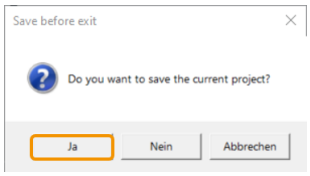
Load / Create a project



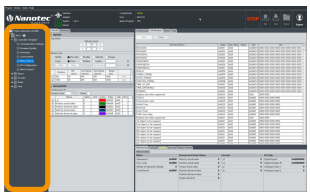
1. In the user screen: Visit the main menu (1)



2. Preferably use **Project > Open** to select an existent sample project for template.
3. Or, for a new one instead: Select **Project > New**.

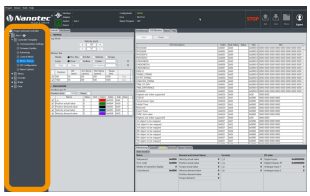


4. If a pop-up wants to store the current project: Click **Yes**.
   - **No** will close the project unstored and without backup.
   - **Cancel** will just close the pop-up.

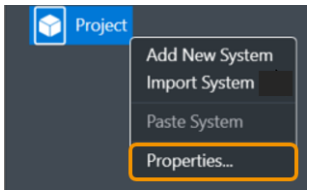


5. The newly loaded selection appears in the project bar (8)

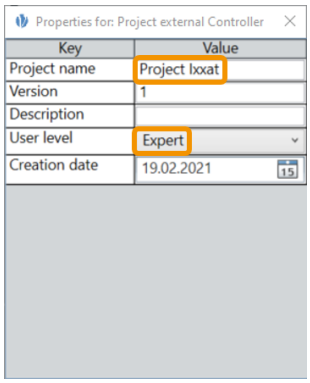
If needed: Name the project



1. Visit the project bar (8).

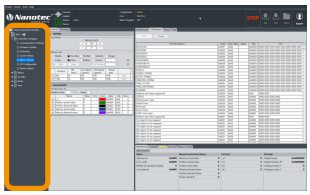


2. Co-click the current project and **Properties**.

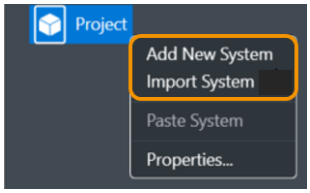


3. In the pop-up: Name, versionize, and describe the project (here: *Project Ixxat*). **Note** the pull-down for granted user rights (here: *Expert*).
4. After last entry: Set a tab stop (so that all is stored).
5. Only then: Close the pop-up.

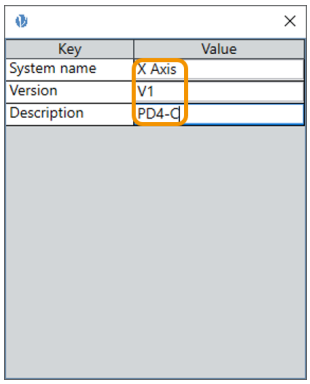
Load / Create a system



1. Visit the project bar (8).

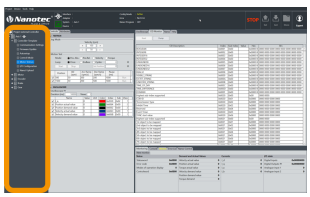


1. In the project: Preferably use **Import System** to select an existent sample system for template.
2. Or, for a new one instead: Select **Add new system**.

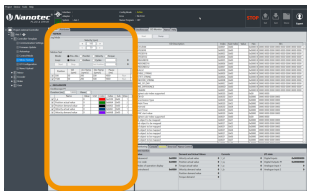


3. In the pop-up: Name, versionize, describe the system as needed (here: *X Axis*, version *V1*, for *PD4-C*).
4. After last entry: Set a tab stop (so that all is stored).
5. Only then: Close the pop-up.
6. Repeat for each additional system.

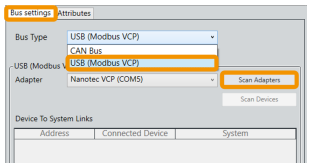
Connect to adapter



1. In the project bar (8): Open your project.



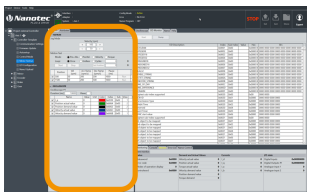
2. In the work desk (7): Open the **Bus settings** tab.



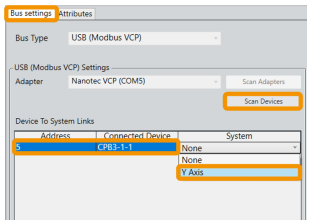
3. In the **Bus settings** tab: Select the **Bus type**.
4. Check setup by **Scan adapters**. If no result: Set up an adapter and check again.
5. Select the needed adapter.

6. You can link / unlink the adapter via **Connect** icon (here: green).

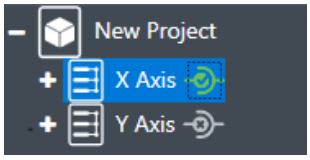
Connect to system



1. Visit the work desk (7).



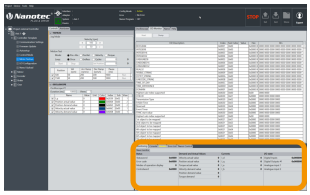
2. In the **Bus settings** tab: With the adapter linked, you can see all connected devices.
3. Click **Scan devices**. Check **Connected device**.
4. By Pulldown: Select a **system** to link your device to.



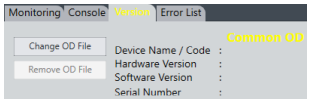
5. You can link / unlink the system via **Connect** icon (here: green).
6. If linked, you reach the device parameters.

Select the OD file

PNDS3 shows objects that match the controller firmware with correct OD file only (object dictionary). If the system is linked, a **Version** tab shows if the correct OD file is loaded. Otherwise, the generic file *Common OD* loads, by which you reach available objects of all Nanotec products.



1. Visit the status board (6).
2. Open the **Version** tab.



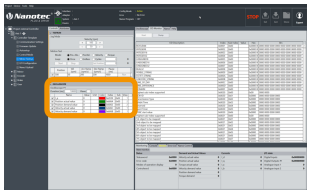
- *Common OD*: Reloadable via **Remove OD file**
- OD file of choice: Loadable via **Change OD file**
- Firmware-correct OD files for all Nanotec controllers: In PNDS3's **Firmware** folder

Wrong OD files report an error (= red).

8 Special controls setups

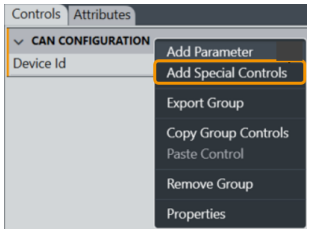
Via **Special controls**, you add **Complex controls** and **Device communication settings** to the user interface. Both help you to use advanced controller functions.

Basic principle

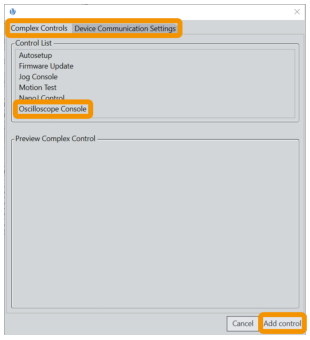


Special controls define and monitor (as macro collections) the system behavior. Depending on assembly, you can check their connections and attributes here in the work desk.

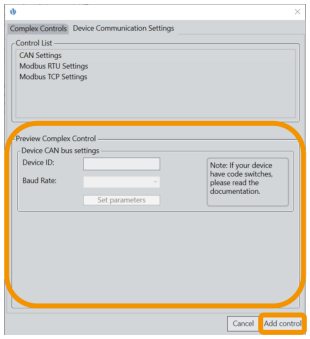
1. To set up complex controls or device communication: Co-click the controls group.



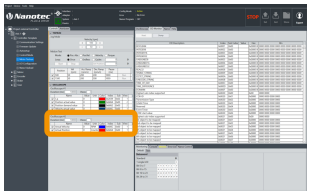
2. In the context menu: Select **Add special controls**.



3. **Complex controls**? **Device communication settings**? Open the tab of choice.
4. Select the needed item and **Add ...** / **Next** (depending on menu).



5. If possible: Check the manual setting options in **Preview complex control**.
6. If possible: Edit the item. **Note:** No storage on **Cancel**.
7. Confirm each step with **Add ...** / **Next** (if wrong: step **Back**).
8. Approve with **Finish**.



→ The control / setting of choice appears in the work desk.

8.1 Complex controls

CAUTION

Injury from erratic motor movement after auto-setup (= parameter loss)!

► For motors with integrated controllers: Avoid auto-setup (since it comes factory-run already).

► Otherwise: Restart the motor after auto-setup (homing alone won't suffice).

NOTICE

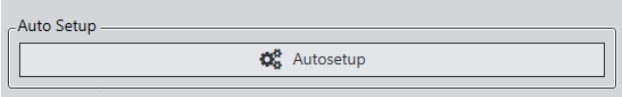
Motor malfunction from auto-setup user error!

- ► Close possible NanoJ programs (object 2300_h:00_h Bit 0 = "0"; cf. 2300h NanoJ Control).
- ► Keep the motor load-free, and freely rotable in any direction.
- ► **Don't** touch the motor.

With the **Complex controls** macro collection, you create your own controller functions. Next to **Autosetup** and **Firmware update**, these include **Jog console, Motion test, NanoJ control** and **Oscilloscope console**, etc.

Autosetup

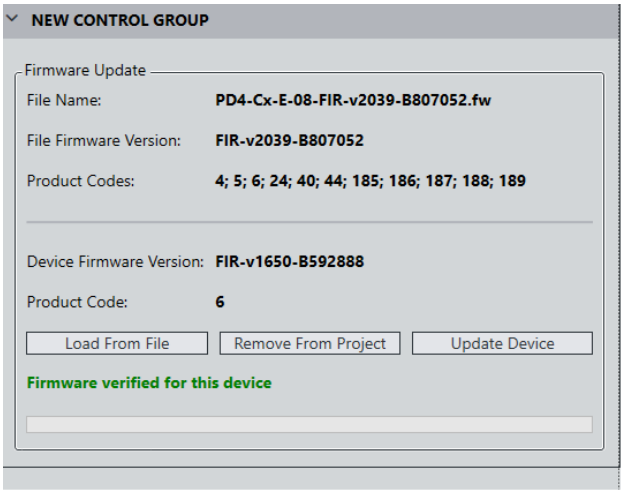
You need **Autosetup** to parametrize the motor and connected sensors (encoder / Hall sensors).



As long as the motor on the controller or the feedback sensors (encoder / Hall) remain the same: Run **Autosetup** only once, on initial commissioning.

Firmware update

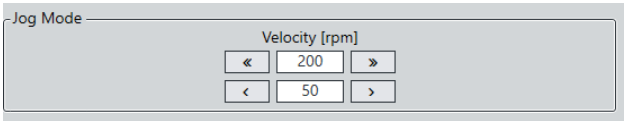
Nanotec recommends controller firmware *FIR-v2139* or newer. Please find the current version in the **Firmware** folder. You start **Firmware update** in the work desk (7):



1. In the control group of choice: Open the **Controls** tab and search for **Firmware update**.
2. There: Click **Load from file**.
3. Select a firmware file and click **Open**.
4. PNDS3 checks via product code if the chosen file fits to the product.
5. Click **Update device**.
6. Firmware updates itself.

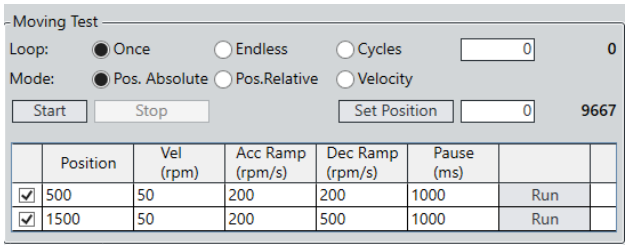
Note: The chosen firmware file will be stored as part of the project the next time the latter is stored. If you don't want this to happen, click **Remove from project** before.

Jog console



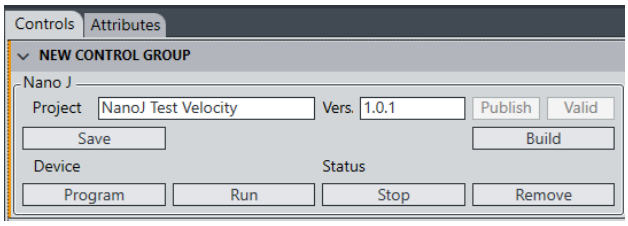
Via **Jog console**, you test the motor in velocity mode. You can select two target speeds. The motor runs as long as you use the mouse to press the button for left / right rotation.

Motion test



In **Motion test**, you check the motor in position / velocity / torque mode. Your options include target values, acceleration / deceleration ramps, repetition cycles, test run duration etc.

NanoJ control > Controls



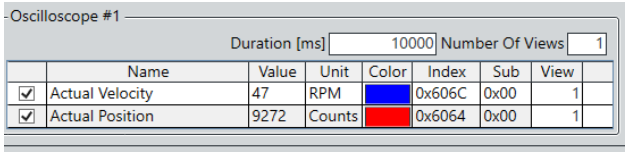
In **NanoJ control > Controls**, you create a NanoJ new project (= **New**); you name / version it by hand entry and save it by **Save**. The button **Build** compiles the project.

By **Import**, you import existing NanoJ projects of choice).

You can export the complete module. If you have selected **Publish**, the project is exported without the souce code.

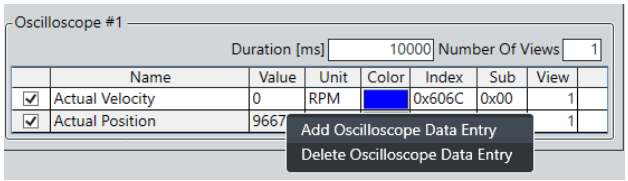
Note: The next time you store the project, the selected NanoJ file merges into the project. If you don't wish this to happen, click **Remove** before. You can change the active NanoJ program on the controller via **Program** and control it via **Run** or **Stop**.

Oscilloscope console

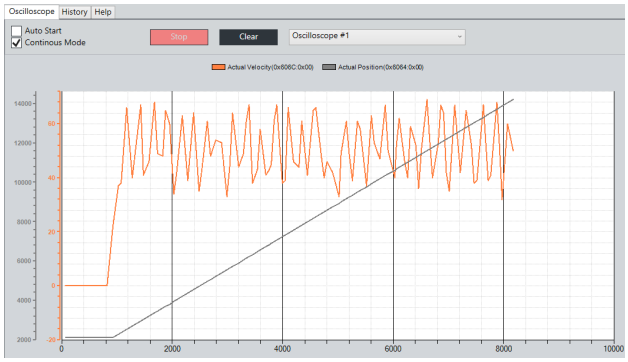


Via **Oscilloscope**, you monitor and control in real time the current value of device parameters from the object dictionary, say, for recording.

- Duration of recording
- Number of oscilloscope windows



You co-click the data table in the console and select **Add oscilloscope data entry**. From the list, you click an entry type and **Select**. Repeat for each entry type of choice.



For recording, you open the **Oscilloscope** tab in the display wall (5).

Click **Start** and select *one* option:

- **Auto Start**: Recording starts as soon as a parameter of choice changes its value. You start, say, a **Motion test** and check the speed / position curve in the oscilloscope.
- **Continuous Mode**: Recording starts as soon as you click **Start**.

8.2 Device communication

With these controls, you parametrize the device communication. **Note:** Coding switches for setting the communication parameters overwrite the software settings on some devices. For details: Follow valid OEM instructions.

9 Imprint, contact

Nanotec Electronic GmbH & Co. KG │ Kapellenstraße 6 │ 85622 Feldkirchen │ Germany

Tel. +49 (0)89 900 686-0 │ Fax +49 (0)89 900 686-50 │ info@nanotec.de │ www.nanotec.com