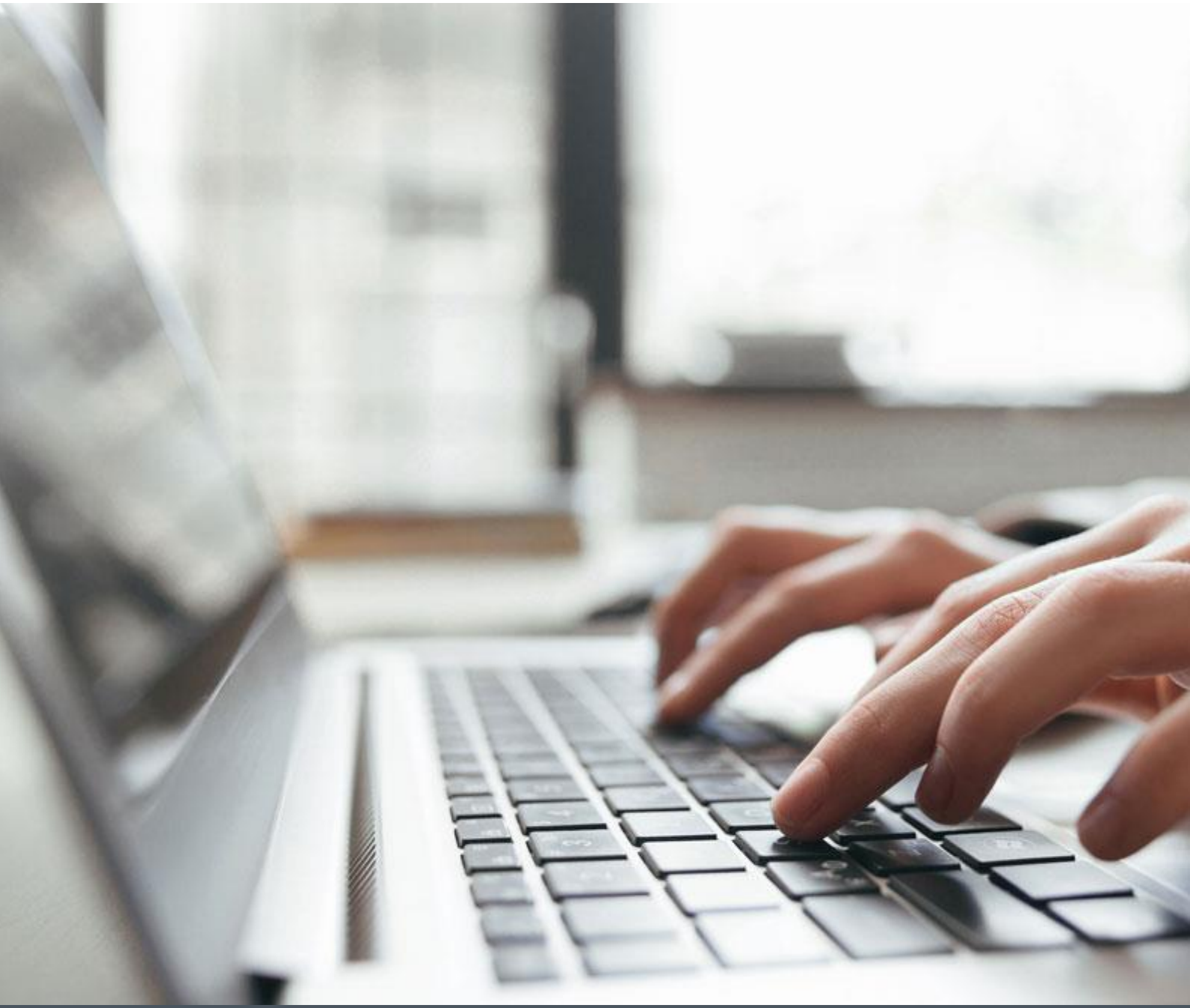


Application Note

How to use *Profile Position* in *NanoJ*

Version 1.0.0

www.nanotec.com

Contents

1 Intended use and audience

This application note shows you how to use the digital outputs of a Nanotec motor controller in a NanoJ program. You can find the corresponding NanoJ code template in the download folder.

Profile Position offers a NanoJ code template for setting target position and target velocity via digital inputs of an electronic Nanotec motor controller. To open and edit the template requires Plug & Drive Studio software. Both NanoJ and Plug & Drive Studio are for use with Nanotec products only, by trained specialists only.

2 Prerequisites

NOTICE
Malfunction from incompatibility! Plug & Drive Studio comes in various software versions. Find out and, if necessary, install the correct version for your Nanotec motor controller in advance.

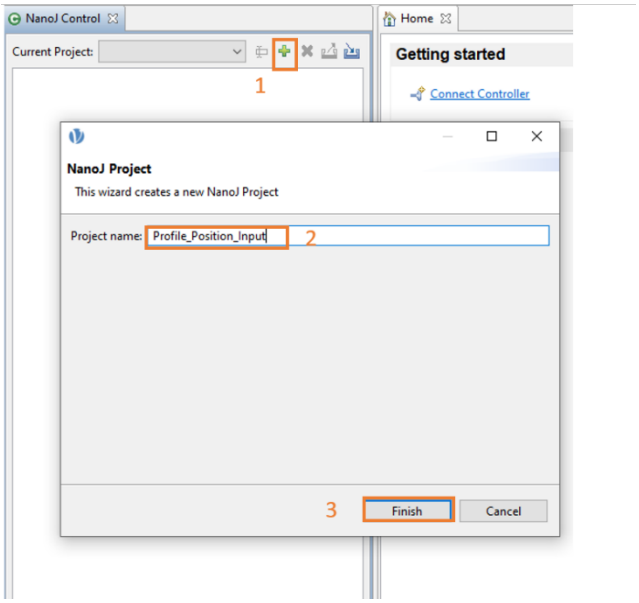
You must have the correct Plug & Drive Studio version installed on your computer:

1. Open the Nanotec software webpage.
2. Click on the *Plug & Drive Studio* buttons.
3. Browse *Compatible Products* to find out which version is compatible with your motor controller.
4. Download and install the latest compatible Plug & Drive Studio version on your computer.
5. If not done so yet: Also download the latest NanoJ V2 Library (`nanotec.h`).

3 Creating a new project in Plug & Drive Studio

Open the *NanoJ Control* tab and click on the "**+**" icon (1). A *NanoJ Project* tab pops up:

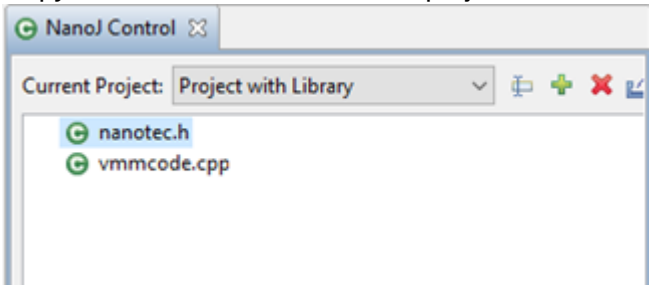
1. Assign a new project name (2).
2. Click on *Finish* (3) to close the tab.
3. Your new project is now created.



4 Including the nanotec.h library into your NanoJ project

The Plug & Drive Studio installation folder does include `wrapper.h`. But you must download the NanoJ V2 library (`nanotec.h`) from our knowledge base and copy it into NanoJ:

1. Generate a new NanoJ project or open an existing one.
2. Copy the `nanotec.h` file into the project tree via drag & drop:



3. To implement the NanoJ V2 library, add `#include wrapper.h` and `#include nanotec.h` to your code:

```
10
11 #include "wrapper.h"
12 #include "nanotec.h"
13
14
15 void user()
16 {
```

5 Using the code template for digital outputs in NanoJ

NOTICE

Variable signal level! Some Nanotec controllers provide digital input pins switchable between 5V and 24V. For correct digital inputs setup: Refer to the corresponding manual of your motor controller.

Profile Position code allows a routing in a NanoJ program depending on the input signals. In a first step, we include the libraries and mappings.

5.1 Including libraries, mappings

For our case, we use the Nanotec NanoJ V2 library `nanotec.h` to implement the code template and provide basic functions to control our motor. To include the `nanotec.h` library, we must at least add the object mappings in lines 25 to 32 to our code:

```
25 map U16 Controlword as inout 0x6040:00
26 map U16 Statusword as input 0x6041:00
27 map U32 Inputs as input 0x60FD:00
28 map U32 Outputs as inout 0x60FE:01
29 map S08 ModesOfOperation as output 0x6060:00
30 map S08 ModesOfOperationDisplay as input 0x6061:00
31 map S16 AnalogInput as input 0x3220:01
32 map S32 TargetVelocity as output 0x60FF:00
```

5.2 Main program loop: void user()

5.2.1 Implementing a release function (Input 1)

For Input 1 signals, we implement a release function. A high release signal **powers** the motor, a low signal **unpowers** it. The release function thus ensures the motor to run on a high release signal only.

- Line 54: With a release signal not set to high, you can stop the motor via `Quickstop()` function.
- Line 46: Input 1 also selects *Profile Position* via `ModesOperation(1)`.
- Line 48: `AbsoluteMovement()` defines the absolute movement operation mode.
- Line 50: With `ChangeSetPointImmediately()` set to `true`, you can activate *change setpoint immediately* to execute each new travel command immediately:

```
42    while(true)
43    {
44        if(DigitalInput(1))                    //if Input 1 is high.
45        {
46            ModesOfOperation(1);               //set the Mode to Profile Position
47            EnableOperation();                 //change the state to Operation Enabled
48            AbsoluteMovement();                //set to absolute movement
49            //RelativeMovement();              //set to relative movement
50            ChangeSetPointImmediately(true);   //change setpoint immediately
51        }
52        else                                   //else...
53        {
54            Quickstop();                       //stop the motor
55        }
```

5.2.2 Setting a target position (Input 2 & 3)

- Line 58, 60: By default, the code template selects *0* for target position if both Input 2 and 3 is low.
- Line 62, 64: If only Input 2 is high, the new target position is *2000*.
- Line 66, 68: With both inputs high, the third target position is *6000*.

```
57          // change the Target Position via digital input 2 and 3...
58          if(!DigitalInput(2) & !DigitalInput(3))         //if Input 2 and 3 low
59          {
60              Out.TargetPosition=0;                     //set the target position to 0
61          }
62          else if(DigitalInput(2) & !DigitalInput(3))     //if Input 2 high and 3 low
63          {
64              Out.TargetPosition=2000;                  //set the target position to 2000
65          }
66          else if (DigitalInput(3) & !DigitalInput(2))    //if Input 2 low and 3 high
67          {
68              Out.TargetPosition=4000;                  //set the target position to 4000
69          }
70          else if (DigitalInput(3) & DigitalInput(2))     //if Input 2 and 3 high
71          {
72              Out.TargetPosition=6000;                  //set the target position to 6000
73          }
```

5.2.3 Starting a position movement (Input 4)

Line 76: Via Input 4, we can start the position movement.

```
75          // start a movement to the selected Target Position...
76          if(DigitalInput(4))                             //if Input 4 is active.
77          {
78              NewSetPoint(true);                          //set new setpoint
79          }
80          else                                            //else
81          {
82              NewSetPoint(false);                         //reset new setpoint
83          }
```

5.2.4 Changing the target velocity (Input 5 & 6)

- Line 87: We use Inputs 5 and 6 to automatically select user-defined speeds.
- Line 89: With both Inputs low, the speed is set to 100 rpm.
- Line 91, 93: If only Input 5 is high, speed rises to 200 rpm.
- Line 95, 97: We reach 300 rpm if only Input 6 is high.
- Line 99, 101: Both inputs high make the motor run with 500 rpm.

```
86          // change the Profile Velocity via digital input 5 and 6...
87          if(!DigitalInput(5) & !DigitalInput(6))         //if Input 5 and 6 low
88          {
89              Out.ProfileVelocity=100;                  //set the velocity to 100
90          }
91          else if(DigitalInput(5) & !DigitalInput(6))     //if Input 5 high and 6 low
92          {
93              Out.ProfileVelocity=200;                  //set the velocity to 200
94          }
95          else if(!DigitalInput(5) & DigitalInput(6))     //if Input 5 low and 6 high
96          {
97              Out.ProfileVelocity=300;                  //set the velocity to 300
98          }
99          else if(DigitalInput(5) & DigitalInput(6))      //if Input 5 and 6 high
100         {
101             Out.ProfileVelocity=500;                  //set the velocity to 500
102         }
103         yield();
```

Your code is finally implemented.

6 Liability

This Application Note is based on our experience with typical user requirements in a wide range of industrial applications. The information in this Application Note is provided without guarantee regarding correctness and completeness and is subject to change by Nanotec without notice.

It serves as general guidance and should not be construed as a commitment of Nanotec to guarantee its applicability to all customer applications without additional tests under the specific conditions and – if and when necessary – modifications by the customer.

The provided information does not replace datasheets and other product documents. For the latest version of our datasheets and documentations please visit our website at www.nanotec.com.

The responsibility for the applicability and use of the Application Note in a particular customer application lies solely within the authority of the customer. It is the customer's responsibility to evaluate, investigate and decide, whether the Application Note is valid and suitable for the respective customer application, or not.

Defects resulting from the improper handling of devices and modules are excluded from the warranty. Under no circumstances will Nanotec be liable for any direct, indirect, incidental or consequential damages arising in connection with the information provided.

In addition, the regulations regarding the liability from our Terms and Conditions of Sale and Delivery shall apply.

7 Imprint

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