CL3-E-1-0F, CL3-E-2-0F

Nanotec®
PLUG & DRIVE

Short instructions
Original: de

Version 1.0.1

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

Phone: +49 (0)89-900 686-0
Fax: +49 (0)89-900 686-50
info@nanotec.de

Introduction

The *CL3-E* is a controller for the *open loop* or *closed loop* operation of stepper motors and the *closed loop* operation of BLDC motors.

This document describes the installation and commissioning of the controller.You can find the detailed documentation for the product on the Nanotec website **us.nanotec.com**. The short instructions do not replace the technical manual oft he product.

Copyright, marking and contact

CE

Intended use

The *CL3-E controller* is used to control stepper and BLDC motors and is designed for use under the approved **Environmental conditions**.

Any other use is considered unintended use.

Note

Changes or modification to the controller are not permitted.

Warranty and disclaimer

Nanotec produces component parts that are used in a wide range of industrial applications. The selection and use of Nanotec products is the responsibility of the system engineer and end user. Nanotec accepts no responsibility for the integration of the products in the end system.

Under no circumstances may a Nanotec product be integrated as a safety controller in a product or construction. All products containing a component part manufactured by Nanotec must, upon delivery to the end user, be provided with corresponding warning notices and instructions for safe use and safe operation. All warning notices provided by Nanotec must be passed on directly to the end user.

Our general terms and conditions apply: **en.nanotec.com/service/general-terms-and-conditions/**.

Specialist staff

Only specialists may install, program and commission the device:

- Persons who have appropriate training and experience in work with motors and their control.
- Persons who are familiar with and understand the content of this technical manual.
- Persons who know the applicable regulations.

EU directives for product safety

The following EU directives were observed:

- RoHS directive (2011/65/EU, 2015/863/EU)

Other applicable regulations

In addition to this technical manual, the following regulations are to be observed:

- Accident-prevention regulations
- Local regulations on occupational safety

Safety and warning notices

Note

- Damage to the controller.
- Changing the wiring during operation may damage the controller.
- Only change the wiring in a de-energized state. After switching off, wait until the capacitors have discharged.

Note

- Fault of the controller due to excitation voltage of the motor.
- Voltage peaks during operation may damage the controller.
- Install suitable circuits (e.g., charging capacitor) that reduce voltage peaks.

Note

- There is no polarity reversal protection.
- Polarity reversal results in a short-circuit between supply voltage and GND (earth) via the power diode.
- Install a line protection device (fuse) in the supply line.

Note

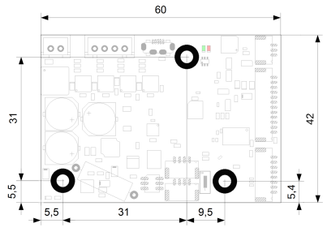
- The device contains components that are sensitive to electrostatic discharge.
- Improper handling can damage the device.
- Observe the basic principles of ESD protection when handling the device.

Technical details and pin assignment

Environmental conditions

Environmental condition	Value
Protection class	No IP protection
Ambient temperature (operation)	-10 … +40°C
Air humidity (non-condensing)	0 … 95 %
Altitude of site above *sea level* (without drop in performance)	1500 m
Ambient temperature (storage)	-25 … +85°C

Dimensioned drawing



Overtemperature protection

Above a temperature of approx. 75°C on the power board the power part of the controller switches off and the error bit is set . After cooling down and confirming the error , the controller again functions normally.

LED signaling

Power LED

Normal operation

In normal operation, the green power LED L1 flashes briefly once per second.

Case of an error

If an error has occurred, the LED turns red and signals an error number.

The following table shows the meaning of the error numbers.

Flash rate	Error
1	General
2	Voltage
3	Temperature
4	Overcurrent
5	Controller
6	Watchdog-Reset

Note

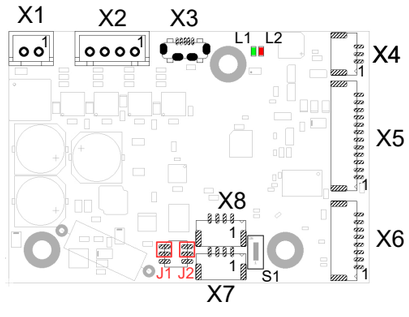
For each error that occurs, a more precise error code is stored in object **1003**$_h$.

Electrical properties and technical data

Property	Description / value
Operating voltage	12 V DC to 24 V DC +/-5%
Rated current	3 A_{rms}
Peak current	CL3-E-1-0F (*low current*): 3 A_{rms} CL3-E-2-0F (*low current*): 6 A_{rms}
Commutation	Stepper motor – open loop, stepper motor – closed loop with encoder, BLDC motor – closed loop with Hall sensor, and BLDC motor – closed loop with encoder
Operating modes	*Profile Position Mode*, *Profile Velocity Mode*, *Profile Torque Mode*, *Velocity Mode*, *Homing Mode*, *Interpolated Position Mode*, *Cyclic Sync Position Mode*, *Cyclic Sync Velocity Mode*, *Cyclic Synchronous Torque Mode*, *Clock-Direction Mode*
Set value setting / programming	*Clock-direction*, *analog*, *NanoJ program*
Interfaces	CANopen, USB, RS-485 (Modbus RTU), RS-232 (Modbus RTU)
Inputs	• 5 digital inputs 5 V • 1 analog input, 10-bit resolution, 0-10 V or 0-20 mA (switchable by means of software, default setting is 0-10 V) • 1 analog input, 10-bit resolution, 0-10 V
Outputs	3 outputs, (open drain, 0 switching, max. 24 V and 100 mA)
Protection circuit	Overvoltage and undervoltage protection Overtemperature protection (> 75° Celsius on the power board) Polarity reversal protection: In the event of a polarity reversal, a short-circuit will occur between supply voltage and GND over a power diode; a line protection device (fuse) is therefore necessary in the supply line. The values of the fuse are dependent on the application and must be dimensioned • greater than the maximum current consumption of the controller • less than the maximum current of the voltage supply. If the fuse value is very close to the maximum current consumption of the controller, a medium / slow tripping characteristics should be used.

Pin assignment

Pin 1 is marked with an asterisk "*".



Connector	Function	Pin assignment / description
X1	Voltage supply 12-24 V DC±5%	**1. +UB** **2. GND**
X2	Motor	**1. A** (Stepper) **U** (BLDC) **2. A** (Stepper) **V** (BLDC) **3. B** (Stepper) **W** (BLDC) **4. B** (Stepper)
X3	USB connection	Micro USB
X4	RS-232	**1.** RS-232-RX **2.** RS-232-TX **3.** GND
X5	Digital and analog inputs and outputs Switching thresholds for digital **inputs 1 - 5**: • On: >3 V • Off: <1 V	**1. 10V output**: +10 VDC, max. 200 mA **2. Digital input 1**: 5 V **3. Digital input 2**: 5 V **4. Digital input 3**: 5 V, max. 1 MHz; *direction input* in clock/direction mode **5. Digital input 4**: 5 V, max. 1 MHz; *clock input* in clock/direction mode **6. Digital input 5**: 5 V **7. Analog input 1**: 10 Bit, 0-10 V oder 0-20 mA, switchable with object **3221**$_h$ **8. Analog input 2**: 10 Bit, 0-10 V, not switchable per software **9. Digital output 1**: Open drain, max 24 V/100 mA **10. Digital output 2**: Open drain, max 24 V/100 mA **11. Digital output 3**: Open drain, max 24 V/100 mA **12. GND**
X6	Encoder and Hall sensor Max. 5 V DC, 1 MHz Switching thresholds: • On: >2.8 V • Off: <1.1 V	**1.** Vcc: +5 V DC output, max. 200 mA **2.** A **3.** B **4.** I **5.** Hall 1 **6.** Hall 2 **7.** Hall 3 **8.** GND
X7	CANopen/RS-485 IN	**1.** +UB Logic (24 V DC/approx. 36 mA, external logic supply for the communication) **2.** CAN/RS-485+: The changeover is performed via jumper J2. **3.** CAN/RS-485-: The changeover is performed via jumper J1. **4.** GND
X8	CANopen/RS-485 OUT	
S1	DIP switch for 120 Ω termination for the CAN bus or RS-485 interface	**OFF**: The bus termination is off. **ON**: The CAN bus termination is on.
J1 und J2	Jumpers to change between CANopen and RS-485	• Setting RS-485: Plug in J1 and J2 facing the middle of the board. • Setting CANopen: Plug in J1 und J2 facing the edge of the board.

Commissioning

The *Plug & Drive Studio* software offers you an option for performing the configuration and adapting the controller to the connected motor. You can find further information in document *Plug & Drive Studio: Quick Start Guide* at **us.nanotec.com**.

Observe the following note:

Note

- EMC: Current-carrying cables – particularly around supply and motor cables – produce electromagnetic alternating fields.
- These can interfere with the motor and other devices. Nanotec recommends the following measures:
- Use shielded cables and earth the cable shielding on both ends over a short distance.
- Use cables with cores in twisted pairs.
- Keep power supply and motor cables as short as possible.
- Earth motor housing with large contact area over a short distance.
- Lay supply, motor and control cables physically separate from one another.

Configuration via USB

General

The following options are available for configuring the controller via USB:

Configuration file

This file can be saved to the controller via the USB connection. For further information, read chapters **USB connection** and **Configuration file**.

NanoJ program

This program can be programmed, compiled and then transferred to the controller with *NanoJ* via USB. *NanoJ* is integrated in the *Plug & Drive Studio* software. You can find further information in document *Plug & Drive Studio: Quick Start Guide* at **us.nanotec.com**.

After connecting to a voltage supply, the controller reads out the configuration in the following order:

1. The configuration file is read out and processed.
2. The NanoJ program is started.

USB connection

If the controller is connected to a PC via a USB cable, the controller behaves like a removable storage device. No further drivers are required.

Three files are displayed: the configuration file (`cfg.txt`), the NanoJ program (`vmmcode.usr`) and the information file (`info.bin`), where the serial numbers and firmware version of the product can be found.

You can thereby store the configuration file or the NanoJ program on the controller. The voltage supply of the controller must also be connected during USB operation.

Configuration file

General

The `cfg.txt` configuration file is used to preset values for the object dictionary to a certain value during startup. This file uses a special syntax to make accessing the objects of the object dictionary as easy as possible. The controller evaluates all assignments in the file from top to bottom.

Reading and writing the file

How to access the file:

1. Connect and switch on the voltage supply.
2. Connect the controller to your PC using the USB cable.
3. After the PC has detected the device as a removable storage device, navigate in the Explorer to the directory of the controller. File `cfg.txt` (for a PD4C, the file is named `pd4ccfg.txt`) is stored there.
4. Open this file with a simple text editor, such as Notepad or Vi. Do not use any programs that use markup (LibreOffice or similar).

Tip

To be able to connect the controller with *Plug & Drive Studio* via the *virtual COM port* mit verbinden zu können, insert the following line:

```
2102:00=0x19000F
```

After you have made changes to the file, proceed as follows to apply the changes:

1. Save the file if you have not yet already done so.
2. Disconnect the USB cable from the controller.
3. Disconnect the voltage supply from the controller for approx. 1 second until the power LEDs stop flashing.
4. Reconnect the voltage supply. When the controller is now restarted, the values in the configuration file are read out and applied.

Structure of the configuration file

Comments

Lines that begin with a semicolon are ignored by the controller.

Assignments

Values in the object dictionary can be set with the following syntax:

```
<Index>:<Subindex>=<Value>
```

Example

Set object 2031_h:00 (rated current) to the value "258_h" (600 mA):

```
2031:00=0x258
```

Set object 3202_h:00 to the value "8" (activate current reduction while at a standstill in *open loop* mode):

```
3202:00=8
```

Set object 2057_h:00 to the value "512" and object 2058_h to the value "4" (*quarter step* step mode in clock-direction mode):

```
2057:00=512
```

```
2058:00=4
```

Establishing communication via CANopen

1. Connect the CANopen master to the controller via the CAN- and CAN+ cables. Check the connection of your CAN-GND and that the necessary **120 ohm termination resistor** is present between CAN+ and CAN-.
2. Supply the controller with voltage.
3. Change the configuration values if necessary.

   The controller is set per default to node-ID 127, baud rate 1 Mbaud.
4. To test the interface, send bytes `40 41 60 00 00 00 00 00` to the controller.
   Statusword (6041_h) was read; you receive this response: `4B 41 60 00 XX XX 00 00`.

Establishing communication via Modbus

1. Connect the *Modbus master* to the controller via the RS-485+ and RS-485- cables.

   If using RS-485, mount jumpers J1 and J2 in the correct position.
2. Supply the controller with voltage.
3. Change the configuration values if necessary.

   The controller is set at the factory to slave address 5, baud rate 19200 Kbaud, even parity, 1 stop bit.

   The following settings can be performed:

Configuration	Object	Value range	Factory settings
Slave address	**2028**$_h$	1 to 247	5
Baud rate	**202A**$_h$	7200 to 256000	19200
Parity	**202D**$_h$	• None: 0x00 • Even: 0x04 • Odd: 0x06	0x04 (Even)

4. To test the interface, send bytes `05 65 55 8A AE` to the controller (you can find a detailed description of the Modbus function codes in chapter **Modbus RTU** of the technical manual of the controller).
   The object dictionary is read out.

Setting the motor data

Prior to commissioning, the motor controller requires a number of values from the motor data sheet.

- Number of pole pairs: Object **2030**$_h$:00_h (pole pair count) The number of motor pole pairs is to be entered here. With a stepper motor, the number of pole pairs is calculated using the step angle, e.g., 1.8° = 50 pole pairs, 0.9° = 100 pole pairs (see step angle in motor data sheet). With BLDC motors, the number of pole pairs is specified directly in the motor data sheet.
- Setting the motor current / motor type:
  - Stepper motor only: Object **2031**$_h$:00_h: Rated current (bipolar) in mA (see motor data sheet)
    - Object **2031**$_h$:00_h: Rated current (bipolar) in mA (see motor data sheet)
    - Object **3202**$_h$:00_h (Motor Drive Submode Select): Defines motor type stepper motor, activates current reduction on motor standstill: 0000008h.
    - Object **2037**$_h$ (Open Loop Current Reduction Value/factor): the root mean square is specified to which the rated current is to be reduced if current reduction is activated in *Open Loop*.
  - BLDC motor only:
    - Object **2031**$_h$:00_h Peak current in mA (see motor data sheet)
    - Object **203B**$_h$:01_h Rated current in mA (see motor data sheet)
    - Object **203B**$_h$:02_h Maximum duration of the peak current in ms (for initial commissioning, a value of 100 ms is recommended; this value is to be adapted later to the specific application).
    - Object **3202**$_h$:00_h (Motor Drive Submode Select): Defines motor type BLDC: 00000041h

Auto setup

To determine a number of parameters related to the motor and the connected sensors (encoders/Hall sensors), an auto setup is performed. **Closed Loop** operation requires a successfully completed auto setup.

Note

- Note the following prerequisites for performing the auto setup:
- The motor must be load-free.
- The motor must not be touched.
- The motor must be able to turn freely in any direction.
- No NanoJ programs may be running (object 2300_h:00_h bit 0 = "0", see **2300h NanoJ Control**).

Tip

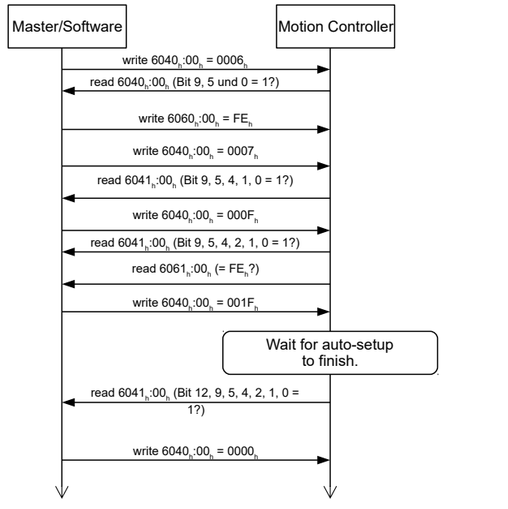
As long as the motor connected to the controller or the sensors for feedback (encoders/Hall sensors) are not changed, auto setup is only to be performed once during initial commissioning.

Execution

1. To preselect the *auto setup* operating mode, enter the value "-2" (="FE_h") in object 6060_h:00_h.
   The *power state machine* must now switch to the *Operation enabled* state.
2. Start *auto setup* by setting bit 4 *OMS* in object 6040_h:00_h (controlword). While the auto setup is running, the following tests and measurements are performed in succession:

   To determine the values, the direction of the measurement method is reversed and edge detection re-evaluated.

   Value 1 in bit 12 *OMS* in object 6041_h:00_h (statusword) indicates that the auto setup was completely executed and ended. In addition, bit 10 *TARG* in object 6041_h:00_h can be used to query whether (= "1") or not (= "0") an encoder index was found.



CAUTION

- After executing auto setup mode, the internal coordinate system is no longer valid.
- *Homing* alone does not suffice! If the controller is not restarted, unexpected reactions may result.
- Restart the device after an auto setup!

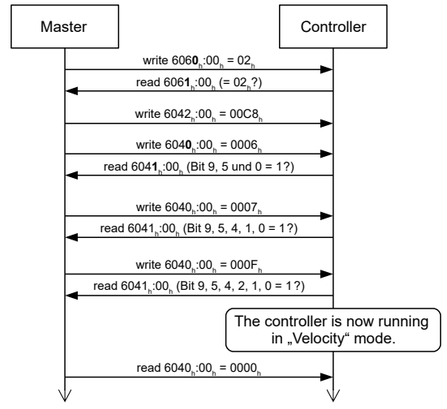
Test run

As an example, the **Velocity** operating mode is used.

The values are transferred from your *CANopen master* or *Modbus master* to the controller. After every transfer, the *master* should use the status objects of the controller to ensure successful parameterization.

1. Select the *Velocity* mode by setting object **6060**$_h$ (Modes Of Operation) to the value "2".
2. Write the desired speed in **6042**$_h$.
3. Switch the *power state machine* to the *Operation enabled* state.

   The following sequence starts *Velocity* mode; the motor turns at 200 rpm.



4. To stop the motor, set controlword (**6040**$_h$) to "0".