CL4-E-1-12, CL4-E-2-12, CL4-E-1-12-5VDI, CL4-E-2-12-5VDI

Short instructions
Original: de

Version 1.1.0

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

Tel. +49 89 900 686-0
Fax +49 89 900 686-50
info@nanotec.de

Introduction

The *CL4-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 of the product*.

Copyright

CE

Intended use

The *CL4-E controller* is used to control stepper and BLDC motors and is used as a component in a wide range of industrial applications.

You use this product as intended within the limits defined by the technical data (see in particaular **Permissible operating voltage**) and under the approved **Environmental conditions**.

Under no circumstances may this 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.

Target group and qualification

The product and this documentation are directed towards technically trained specialists staff such as:

- Development engineers
- Plant engineers
- Installers/service personnel
- Application engineers

Only specialists may install, program and commission the product. Specialist staff are persons who

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

Warranty and disclaimer

Nanotec assumes no liability for damages and disruptions of operation that arise from assembly mistakes, non-observance of the technical manual or improperly carried out repairs. 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 product in the end system.

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

Note

Changes or modifications to the product are not permitted.

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

Damage to the electronics through improper handling of ESD-sensitive components!

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

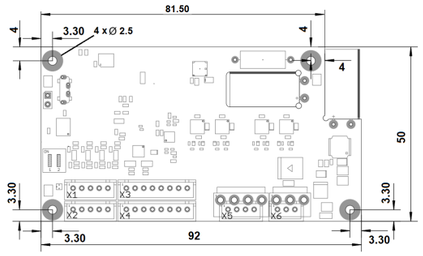
Environmental condition	Value
Protection class	No IP protection
Ambient temperature (operation)	-10 … +40°C
Air humidity (non-condensing)	0 … 95 %
Altitude of site above *sea level* (without drop in performance)	1500 m
Ambient temperature (storage)	-25 … +85°C

Electrical properties and technical data

Property	Description / value
Operating voltage	12 …58 V DC
Voltage range for logic supply	12 …30 V DC
Rated current	CL4-E-1-xx, *low current*: 3 A_{rms} CL4-E-2-xx, *high current*: 6 A_{rms}
Peak current	CL4-E-1-xx, *low current*: 6 A_{rms} for 5 seconds CL4-E-2-xx, *high current*: 18 A_{rms} for 5 seconds
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, RS-485 (Modbus RTU), USB (configuration interface)
Inputs	• 4 digital inputs: - 24 V for the variants with article numbers *CL4-E-1-12* and *CL4-E-2-12* - 5 V for the variants with article numbers *CL4-E-1-12-5VDI* and *CL4-E-2-12-5VDI* • 1 analog input, 10-bit resolution, 0-10 V
Outputs	2 digital outputs, positive switching (typical output voltage corresponds to the connected logic supply – 0.6 V)
Protection circuit	Overvoltage and undervoltage protection Overtemperature protection (> 75° Celsius on the power board)

Dimensioned drawing

All dimensions are in millimeters.



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.

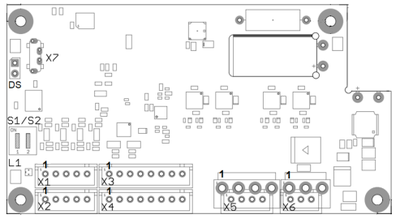
Flash rate	Error
1	General
2	Voltage
3	Temperature
4	Overcurrent
5	Controller
6	Watchdog-Reset

Note

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

Pin assignment

Pin 1 is marked with an "1".



Connector	Function	Pin assignment / description
X1/X2	CANopen/RS485 IN and OUT	**1.** RS-485- **2.** RS-485+ **3.** CAN-Low **4.** CAN-High **5.** GND
X3	Encoder and Hall sensor Max. 5 V DC, 1 MHz Switching thresholds: • On: >2 V • Off: <0.8 V	**1.** Supply: +5 V output voltage, max. 200 mA **2.** A **3.** B **4.** Index **5.** Hall 1 **6.** Hall 2 **7.** Hall 3 **8.** GND
X4	Inputs and outputs Switching thresholds for digital inputs: **CL4-E-1-12 / CL4-E-2-12** (24 V): On: >9 V; Off: <3.7 V **CL4-E-1-12-5VDI / CL4-E-2-12-5VDI** (5 V): On: >2 V; Off: <0.8 V	**1.** **Digital Input 1** **2.** **Digital Input 2** **3.** **Digital Input 3**: max. 1 MHz; *Direction input* in clock-direction mode **4.** **Digital Input 4**: max. 1 MHz; *clock input* in clock-direction mode **5.** **Analog Input 1**: 10 Bit, 0-10 V **6.** **Digital Output 1**: positive switching (*high-side switch*), max. 100 mA **7.** **Digital Output 2**: positive switching (*high-side switch*), max. 100 mA **8.** **GND**
		Note To use the digital outputs, you must connect a voltage (12…30 V DC) to pin 2 of **X6** (logic supply). The typical output voltage corresponds to the connected logic supply – 0.6 V. The current should not exceed 100 mA.
X5	Motor	**1.** **A** (Stepper) **U** (BLDC) **2.** **A** (Stepper) **V** (BLDC) **3.** **B** (Stepper) **W** (BLDC) **4.** **B** (Stepper)
X6	Voltage supply Ub: 12-58 V DC	**1.** **+Ub** **2.** **+UB Logic** (12-30 V DC, input voltage for the optional logic supply and the digital outputs **3.** **GND**
X7	USB connection	Micro USB
S1	DIP switch for 150 Ω termination for RS-485	**OFF**: The RS-485 Network is not terminated. **ON (up)**: The RS-485 Network is terminated.
S2	DIP switch for 120 Ω termination for CAnopen bus	**OFF**: The CANopen bus is not terminated. **ON (up)**: The CANopen bus is terminated.
DS	By short circuiting (e.g., with a wire bridge) the two copper openings, the address and baud rate are reset. The controller restarts automatically after approx. 3 seconds.	The following objects are affected: **Object** / **Function** / **Factory settings** **2009**$_h$ CANopen node-ID 127 **2005**$_h$ CANopen baud rate 136 (1 MBaud) **2028**$_h$ Modbus slave address 5 **202A**$_h$ Modbus baud rate 19200 **202D**$_h$ Modbus parity 04$_h$ (Even)

Note

- EMC: For a DC power supply line longer than 30 m or when using the motor on a DC bus, additional interference-suppression and protection measures are necessary.
- An EMI filter is to be inserted in the DC supply line as close as possible to the controller/motor.
- Long data or supply lines are to be routed through ferrites.
- A capacitor of at least 4700 µF is to be connected to the supply voltage (in parallel), as close to the controller as possible.

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 separately.

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.

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).

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 (max. motor 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
```

or only set bit 3

```
3202:00.3=1
```

Establishing communication via CANopen

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

   The controller is set to slave address 5 ex works, baud rate 19200 baud, 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 00 2F A7` to the controller (you can find a detailed description of the Modbus function codes in chapter **Modbus RTU** of the *technical manual*).
   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.
- Object **2031**$_h$:00_h: maximum permissible motor current (motor protection) in mA (see motor data sheet)
- Object **6075**$_h$:00_h: rated current of the motor in mA (see motor data sheet), limited by **2031**$_h$
- Object **6073**$_h$:00_h: maximum current (for a stepper motor, generally corresponds to the rated current, bipolar) in tenths of a percent of the set rated current (see motor data sheet). Factory settings: "1000", which corresponds to 100% of the value in **6075**$_h$. Is limited by **2031**$_h$.
- Object **203B**$_h$:02_h Maximum duration of the maximum current (**6073**$_h$) in ms (for initial commissioning, Nanotec recommends a value of 100 ms; this value is to be adapted later to the specific application).
- Setting the motor type:
  - Stepper motor:
    - 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:
    - Object **3202**$_h$:00_h (Motor Drive Submode Select): Defines motor type BLDC: 00000040h
- Motor with encoder without index: You must set the encoder parameters after the **Auto setup**, see chapter **Configuring the sensors**.

Note

Due to the sine commutation and the sinusoidal current flow, the current of a motor winding can achieve an alternating current value that is briefly greater (by max. √2 times) than the set current.

At especially slow speeds or while at a standstill with full load, one of the windings can therefore be supplied with overcurrent for a longer period of time. Take this into account when dimensioning the motor and, if necessary, select a motor with larger torque reserve if required by the application.

Auto setup

To determine a number of parameters related to the motor and the connected sensors (encoders/Hall sensors), you must perform an auto setup.

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.

Note

Note the following prerequisites for performing the auto setup:

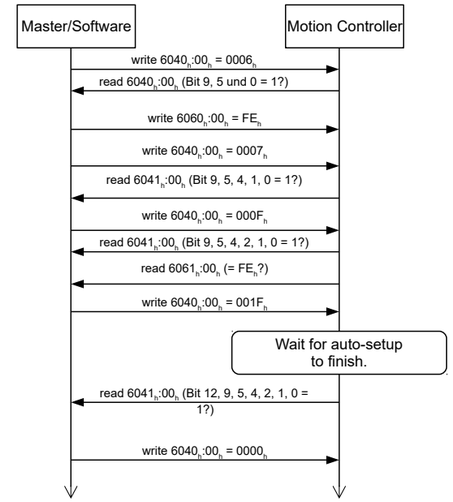
► The motor must be load-free.
► The motor must not be touched.
► The motor must be able to turn freely in any direction.
► No NanoJ programs may be running (object 2300_h:00_h bit 0 = "0", see **2300h NanoJ Control**).

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).

   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

Uncontrolled motor movements!

After the auto setup, the internal coordinate system is no longer valid. Unforeseen reactions can result.

► Restart the device after an auto setup. Homing alone does not suffice.

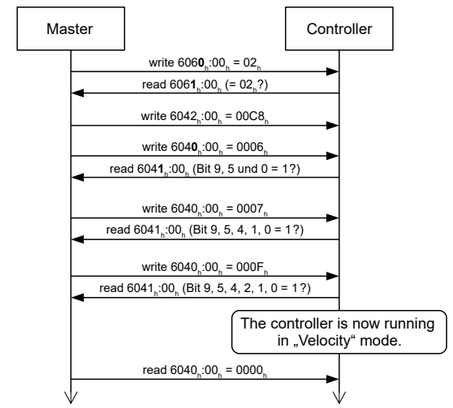
Test run

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*.

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



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