User Manual

Version 1.2

2025/04/21

CTRL1-48-5-G4

Single-Phase Controller Board

This page intentionally left blank

1. Document History

Ver.	Chapter	Description / Changes	Date
1.0		Initial release	2024.11.11
1.1	5.6.1 Command Codes, 5.6.2 Configurations, 5.6.3 Macros and Trajectory Design, 10.3 Dimensions	Updated configuration and dimensions, added new macros, and introduced the Firmware Update command.	2025.04.15
1.2	5.6.3 Macros and Trajectory Design	Macro value precision changed to 3.	2025.04.21

Table of Contents

2. Introduction

The CTRL1-48-5-G4 is a standalone, single-axis integrated controller designed for precise control of single-phase linear actuators. This controller is fine-tuned to be used with VC48 series actuators. Its operational parameters are optimized for indoor environments, requiring conditions free from contaminants like dirt, oil, and lubricants.

The CTRL1-48-5-G4 has a continuous current rating of 5 A. It operates with a single 48 V power supply, while an optional 5 V to 24 V power source is required for the digital inputs and outputs. This controller is equipped with 4 opto-isolated digital inputs and 5 opto-isolated digital outputs. Additionally, it features 2 analog outputs, along with 1 single-ended and 1 differential analog input. The controller is equipped with a Safe Torque Off (STO) protection system. It is also compatible with communication protocols including ADMORS, Modbus, and CANopen. Online and offline data acquisition capabilities have been added to the CTRL1-48-5-G4 firmware. The controller board offers macro and motion trajectory design options. It is also equipped with an advanced "Soft-Touch™" algorithm applicable to the VC48 series actuators. The soft-touch™ algorithm allows the actuator rod to touch the surface of a component with a low programmed force, which is very useful when handling delicate objects. The CTRL1-48-5-G4 can only be utilized for its intended purpose when properly installed and connected to a suitable actuator (p. 9).

3. Product Description

3.1. Product View

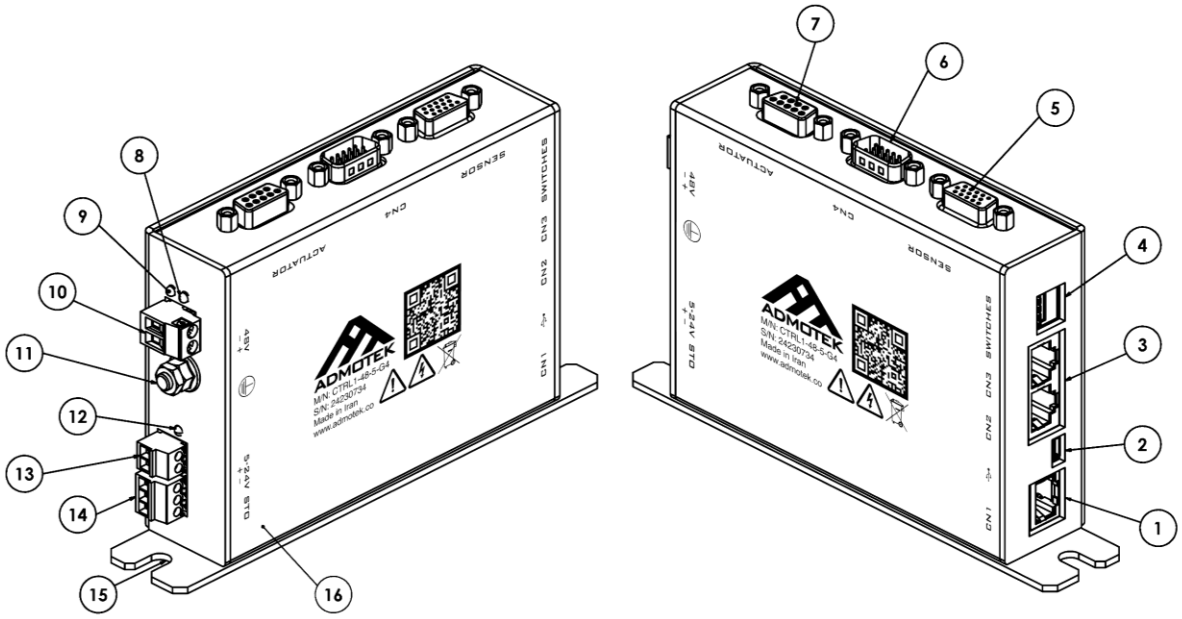

Fig. 1. CTRL1-48-5-G4 product view.

1	RS232 connection	9	48V LED indicator
2	Mini-USB connection	10	48 V connection
3	RS485/422 & CAN connection	11	M5 grounding screw
4	Configuration Switches	12	Isolated power supply LED indicator
5	Sensor connection (DB15 (f))	13	Isolated power supply (5V-24V)
6	I/O connection (DB15 (m))	14	STO inputs and feedback
7	Actuator connection (DB9 (f))	15	Mounting slots
8	Error LED indicator	16	Housing

3.2. Labeling

The housing of the CTRL1-48-5-G4 is marked with laser-printed information that includes the following details:

Labeling	Description
CTRL1-48-5-G4	Product name
24137861	Serial number (example). Meaning of each position (from the left): 1 and 2: year of manufacture, 3 to 8: consecutive number
Made in Iran	Country of origin
www.admotek.co	Manufacturer website address
	Manufacturer logo
	QR code containing the serial number. It can be scanned to download this manual.
	Warning sign “Pay attention to the manual!”
	Warning sign “High Voltage”
	Old equipment disposal

3.3. Features

- Compatible controller board for driving the VC-48-xx-xx-x actuator.
- Closed-loop current, position, velocity, and force control (25 kHz).
- Differential signal inputs for encoder and actuator flash memory.
- Digital and analog I/O
- STO inputs and feedback
- RS485, RS422, RS232, and USB FS interfaces
- CANopen, Modbus, and ADMORS communication protocols
- Overvoltage, overcurrent, short-circuit, over-temperature, and reverse polarity protection
- Internally-stored motion trajectories and configs
- Permanent and Temporary Configs
- Online and Offline Data acquisition
- Monitoring and Error Control
- Sensor-less force estimation
- Soft-Touch™ routine

3.4. Scope of Delivery

Model Number	Description
CTRL1-48-5-G4	Single-phase controller.

3.5. Suitable Actuators

Model Number	Description
VC-48-xx-xx-x	Linear voice coil actuator

4. Installation

4.1. General Notes on Installation

- Install the CTRL1-48-5-G4 close to the power source so that the power plug can be quickly and easily disconnected from the mains.
- Only use cables and connectors that meet local safety regulations.

4.2. Unpacking

NOTICE

Electrostatic hazard!

The CTRL1-48-5-G4 contains electrostatic-sensitive components (ESD) and can be damaged if handled improperly.

- Avoid touching assemblies, pins, and PCB traces.
- Before you touch the CTRL1-48-5-G4, discharge your body appropriately (e.g., by using an antistatic wrist strap).
- Only handle and store the CTRL1-48-5-G4 in environments that dissipate existing static charges to earth in a controlled way and prevent electrostatic charges (ESD workplace or electrostatic discharge protected area, abbreviated to EPA).

1- Unpack the CTRL1-48-5-G4 controller board with care.
2- Inspect the controller board for signs of damage. If any parts are damaged or missing, contact our customer service department immediately.
3- Keep all packaging materials in case the product needs to be returned.

4.3. Ensuring Ventilation

High temperatures can overheat the CTRL1-48-5-G4.

- Set up the CTRL1-48-5-G4 with a distance of at least 10 cm to the top and rear sides and at least 5 cm to the sides. If this is not possible, make sure that the area is cooled sufficiently.
- Ensure sufficient ventilation at the place of installation.
- Keep the ambient temperature to a non-critical level (< 50°C).

4.4. Mounting on a Surface

Fig. 2 shows the mounting option.

Tools and Accessories

- M5 screws of suitable length
- 4 mm hex key

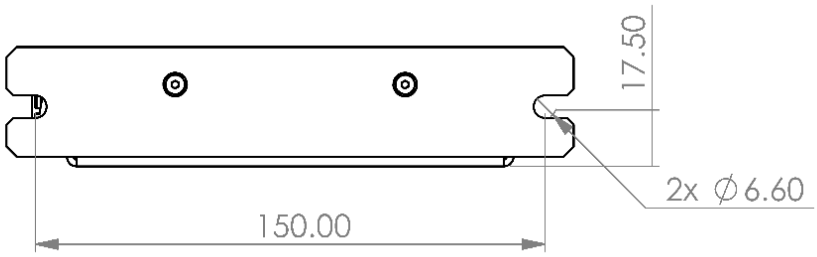

Fig. 2. Mounting the back plate with two M5 screws.

4.5. Connecting Protective Earth Conductor

Connect the threaded pin with the protective earth conductor symbol on the housing of the CTRL1-48-5-G4 to the protective earth conductor.

Requirements

- You have read and understood the general notes on installation (p. 9).
- No other connectors are connected to the controller.

Tools and Accessories

- 8-mm wrench
- Protective earth conductor with a 5-mm stud copper lug

4.6. Connecting Actuator Cable

Connect the connector of the actuator cable to the 9-pin D-sub panel plug of the CTRL1-48-5-G4. Secure the connector of the motor cable using the integrated screws against being accidentally pulled out.

Requirements

- You have read and understood the general notes on installation (p. 9).
- Only the protective earth conductor is connected to the controller.

Tools and Accessories

- ADMOTEK CBL1-9-1-0-0-1 cable
- Flat-head screwdriver

4.7. Connecting Sensor Cable

Connect the connector of the sensor cable to the 15-pin D-sub panel plug of the CTRL1-48-5-G4. Secure the connector of the sensor cable using the integrated screws against being accidentally pulled out.

Requirements

- You have read and understood the general notes on installation (p. 9).
- Only the protective earth conductor and the actuator cable are connected to the controller.

Tools and Accessories

- ADMOTEK CBL1-15-0-1-6-1 cable
- Flat-head screwdriver

4.8. Connecting the Serial Interface

Connect the USB cable to the mini-USB panel plug of the CTRL1-48-5-G4 and the computer.

Requirements

- You have read and understood the general notes on installation (p. 9).
- Only the protective earth conductor, the actuator cable, and the sensor cable are connected to the controller.

Tools and Accessories

- Mini-USB cable

4.9. Connecting the 48 V Power Supply

First, remove the male connector of the CTRL1-48-5-G4 terminal block and connect the power supply outputs to it. Then, insert the male connector into the CTRL1-48-5-G4 terminal block. Connect the power supply to the mains and check the green 48 V indicator LED to make sure the power is connected.

Requirements

- You have read and understood the general notes on installation (p. 9).
- Only the protective earth conductor, the actuator cable, the sensor cable, and the USB serial cable are connected to the controller.

Tools and Accessories

- 48 V power supply
- Flat-head screwdriver

4.10. Connecting the Isolated Power Supply (5 V-24 V)

This item is optional and is required for using the Safe Torque Off (STO) or digital input/output functions. First, remove the male connector of the CTRL1-48-5-G4 terminal block and connect the power supply's outputs to it. Then, insert the male connector into the CTRL1-48-5-G4 terminal block. Connect the power supply to the mains and check the green indicator LED to make sure the power is connected.

Requirements

- You have read and understood the general notes on installation (p. 9).
- Only the protective earth conductor, the actuator cable, the sensor cable, the USB serial cable, and the 48V power supply are connected to the controller.
- 48 V adapter is connected to the mains and its indicator LED is on.

Tools and Accessories

- 5 V - 24 V power supply
- Flat-head screwdriver

5. Startup and Operation

5.1. General Notes

NOTICE

Electromagnetic disturbances!

If the CTRL1-48-5-G4 board is operated without housing, live parts are accessible. Electrical, magnetic, and electromagnetic fields emitted by live parts can disturb the CTRL1-48-5-G4 and/or the environment.

- Install the CTRL1-48-5-G4 board in a suitable housing before startup.
- Make sure that the CTRL1-48-5-G4 board fulfills all requirements for electromagnetic compatibility after being installed in a housing.

5.2. Switching On

Connect the power supply to the 48 VDC terminal block.

Notes

The CTRL1-48-5-G4 performs the following actions when switched on or rebooted:

- Initializes the actuator.
- Loads the last permanent configuration settings (p. 27).
- Sets the error LED indicator and waits for the encoder initialization.

Prerequisites

- You have read and understood the general notes on startup (p. 13).
- The CTRL1-48-5-G4 has been installed properly (p. 9).

5.3. Establishing Communication

The CTRL1-48-5-G4 provides full-speed USB OTG, RS232, and RS485/422 interfaces for serial communication. The full-speed USB OTG is available on the mini-USB connector; the RS232 interface can be accessed through the CN1 connector, while the RS485/422 interfaces are available on the CN2 and CN3 connectors.

Notes

- CN2 and CN3 are pin-to-pin connect.
- RS485 is selected when the first key of the configuration switch is set to the "ON" position; otherwise, RS422 is selected.
- The third key of the configuration switch is used to enable or disable the 120 Ω termination resistor for the RS485 and transmitter lines of RS422.

- The fourth key of the configuration switch is used to enable or disable the 120 Ω termination resistor for the receive lines of RS422.
- By default, the RS485/422 interface is configured to operate with ADMORS protocol. It can be switched to Modbus using the "SCON" command (p. 27).
- The baud rate for RS232 communication is fixed at 115200 bits/s.
- The RS485/422 baud rate is set to 115200 bits/s by default and can be adjusted using the "SCON" command (p. 27).
- Configuring the baud rate is not required for full-speed USB communication.
- The USB interface is shown as a virtual COM port.
- Each serial command should be terminated with a Line Feed (\n) or a Carriage Return (\r) when the ADMORS protocol is used over the USB port or the RS485/422 interface.

5.4. Safe Torque Off (STO)

Safe Torque Off (STO) is a protection algorithm designed to prevent unwanted actuator motion. Activating the STO is essential for starting the actuator in normal mode; however, it can be bypassed using the "BSTO" configuration (p. 27), which is not recommended due to safety concerns.

Three digital pins are designated for the STO function in normal mode, which are isolated from the rest of the board. The STO configuration consists of two input pins and one feedback pin. To activate the STO, the input pins must be set to a "HIGH" state, which switches the feedback pin to the "HIGH" state. All of these pins can operate with logic voltages between 5V and 24V, as specified by the isolated input supply.

5.5. Starting Motion

The specified motion can be started using command codes through the serial interfaces or CAN bus based on CANopen, Modbus, and ADMORS protocols or with the trigger pins on the I/O connector.

Starting Motion Using Serial Interface and CAN bus

Immediately after power-up and establishing serial communication, CTRL1-48-5-G4 is ready to accept command codes from the serial interfaces.

- Commands can be transmitted via the USB port, RS232 interface (CN1 connector), and RS422/485 interface (CN2/CN3 connectors) using the ADMORS protocol. The response is received through the same interface that sends the command.
- The commands can be transmitted using Modbus protocol through the RS422/485 interfaces (CN2/CN3 connectors), if this protocol is enabled in the configuration (p. 27).
- The commands can be transmitted using the CANopen protocol through the CAN interfaces (CN2/CN3 connectors), if this protocol is enabled in the configuration (p. 27).

- The desired configuration and macros must be set before starting the motion (p. 27 and p. 46).

Starting Motion Using Trigger Pins

The digital inputs of the CTRL1-48-5-G4 are accessible via the I/O connector (p. 148). To start motion using these pins:

1- Configure the controller board:
Set the " DI#T" parameter in the controller configuration to "RMCR" (p. 27).
2- Set the desired macro using the "SMCR" command or load a macro from saved trajectories (p. 46). This step is required only the first time after rebooting the controller or powering it on.
3- Apply the signal. The motion starts with the rising edge of the trigger signal.

The actuator and encoder must be initialized first (p. 16). If any error occurs during the motion, the "GERR" command can be used to display the error code (p. 58).

5.6. ADMORS protocol

ADMORS is a communication protocol developed by ADMOTEK and is accessible via RS232, USB, and RS485/422 interfaces. This protocol is essential for setting the configurations and macros to initiate the actuator motion.

5.6.1 Command Codes

Overview

Command	Description
SCON	Set Configuration
GCON	Get Configuration
LCON	Load Configuration
SMCR	Set Macro
GMCR	Get Macro
LMCR	Load Macro
RMCR	Run Macro
PMCR	Stop Macro
DATA	Get Offline Data
DINF	Get Offline Data Information
CTMP	Get Controller Board Temperature
ATMP	Get Actuator Temperature
ARIN	Actuator Re-Init
GERR	Get Error
RERR	Reset Error
CINF	Get Controller Board Information
AINF	Get Actuator Information
CBRS	Reset Controller Board
ASCA	Actuator Self Calibration
EAIN	Encoder Auto-Init
FWUP	Firmware Update

Command: SCON (Set Configuration)

Description	This command is used to set controller board and actuator configurations. These settings are available in the configurations section (p. 27).
Format	SCON,<PERM_TEMP>,<Config_Command>,<Value> SCON,SAVE Example: SCON,TEMP,BSTO,1 Example: SCON,SAVE
Response	SCON,<PERM_TEMP_SAVE>,<Response_Code> The set configuration command (SCON) is displayed first, followed by the configuration mode (temporary, permanent, or save). The response code is reported at the end. A response code of "DONE" indicates that no errors have occurred; otherwise, an error code will be displayed (p. 58). Example: SCON,TEMP,DONE
Notes	• If the "PERM" mode is used after the "SCON" command, the controller first switches to the permanent configuration (setting the permanent configurations as the current ones), and then the values for the desired configurations are changed. The permanent configurations are stored in the flash memory of the controller board, so there is no need to apply this command each time the controller reboots or powers on. • If the "TEMP" mode is used, the controller first switches to the temporary configuration (setting the temporary configurations as the current ones), and then the values for the desired configurations are changed. The temporary configurations are stored in the RAM of the controller board and will be lost if the controller board is reset or powered off. • If the "SAVE" command is used followed by the "SCON" command, the current configurations used by the controller board will be saved as the permanent ones. • When the controller board is powered on, the temporary configurations are initialized with values from the permanent configurations. • The "LCON" command can be used to switch between the "PERM" and "TEMP" configurations. (p. 27). • Multiple configurations can be set simultaneously using the "SCON" command, and a response code will be reported for each setting. Example: SCON,TEMP,BSTO,1,COID,7. Response: SCON,TEMP,DONE,DONE. • It is highly recommended to verify the applied configuration using the "GCON" command to confirm that the settings are successfully implemented.

Command: GCON (Get Configuration)

Description	This command is used to read the configurations of the controller board and the actuator.
Format	GCON,<PERM_TEMP_CURR>,<Config_Command> Example: GCON,TEMP,BSTO
Response	GCON,<PERM_TEMP_CURR>,<Value> The get configuration command (GCON) is displayed first, followed by the configuration mode (temporary, permanent, or current). Finally, the desired value is reported.
Notes	• If the "PERM" mode is used after the "GCON" command, the value of the configurations saved in the flash memory of the controller board is reported. • If the "TEMP" mode is used, the value of the temporary configurations stored in the RAM of the controller board will be reported. • If the "CURR" mode is used with the "GCON" command, the current value being used by the controller board will be displayed, which may be either temporary or permanent. • Multiple configurations can be read simultaneously using the "GCON" command. Example: GCON,CURR,BSTO,COID. Response: GCON,CURR,1,7.

Command: LCON (Load Configuration)

Description	This command is used to establish the permanent or temporary configurations as the current configuration for the controller board and the actuator.
Format	LCON,<PERM_TEMP> Example: LCON,TEMP
Response	LCON,<PERM_TEMP>,<Response_Code> The load configuration command (LCON) is displayed first, followed by the configuration mode (temporary, permanent). Finally, the response code is reported. A response code of "DONE" indicates that no errors have occurred; otherwise, an error code will be displayed (p. 58).
Notes	None

Command: SMCR (Set Macro)

Description	This command is used to set macros. The available macro types can be found in the macro and trajectory design section (p. 46).
Format	SMCR,TEMP,<Macros> SMCR,PERM,<NUM#>,<Macros> SMCR,SAVE,<NUM#> Example: SMCR,TEMP,PHOL,1 Example: SMCR,PERM,NUM1,PHOL,1 Example: SMCR,SAVE,NUM1
Response	SMCR,TEMP,<Response_Code>,<Response_Code>,… SMCR,PERM,<NUM#>,<Response_Code>,<Response_Code>,… SMCR,SAVE,<NUM#>,<Response_Code>,<Response_Code>,… The set macro command (SMCR) is displayed first, followed by its mode (temporary, permanent, or save). If the permanent mode or save command is used, the desired number to save the macro is shown. Finally, the response code is reported for each parameter that defines the macro. A response code of "DONE" indicates that no errors have occurred; otherwise, an error code will be displayed (p. 58). Example: SMCR,PERM,NUM1,DONE,DONE.
Notes	• If the "PERM" mode is used after the "SMCR" command, the desired macro will be saved to the flash memory of the controller board. The macro number must also be specified in the permanent mode. There is no need to reapply this command each time the controller reboots or powers on; only the load macro command is required to switch to this macro. • A maximum of 9 permanent macros can be stored in the controller board. • If the "TEMP" mode is used, the desired macro will be stored in the RAM of the controller board and will be lost if the controller board is reset or powered off. • Using the "SAVE" mode followed by "SMCR" allows the current macro to be saved as a permanent one. The macro number must also be specified in this mode. • Once a macro is set, it is designated as the current macro used by the controller board, regardless of whether it is permanent or temporary. • The "LMCR" command can be used to switch between the "PERM" and "TEMP" macros (p. 46). • A maximum of 20 macro types can be configured as temporary or permanent macros (p. 46); otherwise, the "0xFF18" error is displayed.

Command: GMCR (Get Macro)

Description	This command is used to read macros. These macros are available in the macro and trajectory design section (p. 46).
Format	GMCR,<TEMP_CURR> GMCR,PERM,<NUM#> Example: GMCR,TEMP Example: GMCR,PERM,NUM1
Response	GMCR,<TEMP_CURR>,<Value> GMCR,PERM,<NUM#>,<Value> The get macro command (GMCR) is displayed first, followed by its mode (temporary, permanent, or current). Subsequently, the macro type (e.g., hold position, timed-based sine position, etc.) and the desired values used to define the macro (such as time, amplitude, etc.) are reported (p. 46). In the permanent mode, the desired macro number is also displayed after the "GMCR" command. Example: GMCR,PERM,NUM1,PHOL,1,PARK,-300.
Notes	• If the "PERM" mode is used after the "GMCR" command, the values of the macro saved in the flash memory of the controller board are reported. The macro number must also be specified. • If the "TEMP" mode is used, the values of the temporary macro stored in the RAM of the controller board will be reported. • If the "CURR" mode is used with the "GMCR" command, the current macro values being used by the controller board are displayed, which may be either temporary or permanent. • If no macro has been assigned, "0xFF03" is reported as the <Value>.

Command: LMCR (Load Macro)

Description	This command is used to load either a temporary macro or one of the nine permanent macros.
Format	LCON,TEMP LCON,PERM,<NUM#> Example: LCON,TEMP Example: LCON,PERM,NUM1
Response	LCON,<PERM_TEMP>,<Response_Code> The load macro command (LCON) is displayed first, followed by the macro mode (temporary, permanent). The response code is reported at the end, with "DONE" indicating no errors have occurred; otherwise, an error code will be displayed (p. 58).
Notes	None

Command: RMCR (Run Macro)

Description	This command is used to start the desired macro and repeat it according to a specified value.
Format	RMCR,<Macro_Repeat> Example: RMCR,1
Response	"RMCR" will be displayed first. If the macro completes without errors, "DONE" will be reported on a new line; otherwise, the macro will stop, and the error code or "EROR" will be displayed (p. 58).
Notes	• As the macro starts, the red LED on the controller board blinks to indicate that the macro is running correctly and the controller board is busy. • If an error occurs during the macro, the red LED on the controller board will remain on without blinking. • The <Macro_Repeat> value determines the repeat count for the macro. It must be set between 1 and 65535; otherwise, an illegal value error (0xFF14) will be displayed. • If online data recording is enabled in the controller configuration and the data record type is set correctly, the data will be reported at the desired frequency immediately after the macro starts.

Command: PMCR (Stop Macro)

Description	This command terminates the active macro.
Format	PMCR Example: PMCR
Response	"PMCR" will be displayed.
Notes	• If the current macro defines a motion trajectory and the actuator is not at the homing position, using this command causes the rotor to strike the stator, which can lead to damage with repeated use.

Command: DATA (Get Offline Data)

Description	This command sends the offline data from the last macro.
Format	DATA Example: DATA
Response	The data will be sent in CSV format if there is no error.
Notes	• The offline data recording settings and recorded data types must be configured correctly before using this command. • The maximum number of data points that can be stored during a macro is limited to 25,000. • Offline data recording can be started by using digital input. • Offline data recording can be started from a specified macro type number using the "DFMC" configuration. • If the macro repeats more than once, the reported data corresponds to the last cycle. • The recorded data number, along with the columns and rows of the reporting data, can be accessed using the "DINF" command. • If no data record type is set, the "0xFF15" error code will be displayed. • If no data has been recorded, the "0xFF17" error code will be reported.

Command: DINF (Get Offline Data Information)

Description	This command reports offline data information related to the last macro.
Format	DINF Example: DINF
Response	DINF,<Recorded_Data_Number>,<Row_Number>,<Columns_Number>.
Notes	• The <Row_Number> is calculated based on the desired frequency for offline data recording and macro duration. • <Columns_Number> is determined by the data recording types specified in the controller board configuration. • <Recorded_Data_Number> is the result of multiplying the <Columns_Number> by the <Row_Number>.

Command: CTMP (Get Controller Board Temperature)

Description	This command reports the controller board temperature.
Format	CTMP Example: CTMP
Response	CTMP,<Temperature>.
Notes	• Temperature is reported in degrees Celsius.

Command: ATMP (Get Actuator Temperature)

Description	This command sends the actuator temperature.
Format	ATMP Example: ATMP
Response	ATMP,<Temperature>.
Notes	• Temperature is reported in degrees Celsius.

Command: VBUS (Get Controller Board Bus Voltage)

Description	This command sends the controller board bus voltage.
Format	VBUS Example: VBUS
Response	VBUS,<Value>.
Notes	• The reported value should be multiplied by "0.0158" to calculate the controller board bus voltage in volts.

Command: ARIN (Actuator Reinitialize)

Description	This command reinitializes the actuator and loads the permanent configuration stored in its flash memory.
Format	ARIN Example: ARIN
Response	ARIN,<Response_Code> If no error occurs, "DONE" will be reported as the response code; otherwise, "0x5005" will be displayed.
Notes	• After using this command, the encoder should also be reinitialized using the "EAIN" command or by manually moving the rotor.

Command: GERR (Get Error)

Description	This command reports the last error code that occurred in the system.
Format	GERR Example: GERR
Response	GERR,<Error_Code> If no error has occurred, "0x0000" will be reported; otherwise, an error code will be displayed (p. 58).
Notes	None

Command: RERR (Reset Error)

Description	This command is used to reset the errors.
Format	RERR Example: RERR
Response	RERR,<Response_Code> If no errors remain, "DONE" will be reported as the response code; otherwise, the error code will be displayed (p. 58).
Notes	• If the reset error command completes successfully, the red LED on the controller board will turn off. • Actuator-init error and Encoder-init errors cannot be cleared using this command.

Command: CINF (Get Controller Board Information)

Description	This command reports the model, serial number, firmware version, and temperature of the controller board.
Format	CINF Example: CINF
Response	CINF,<model_number>,<serial_number>,<firmware_version>, <temperature>
Notes	• Temperature is reported in degrees Celsius.

Command: AINF (Get Actuator Information)

Description	This command reports the model, serial number, temperature, and the encoder initialization status of the actuator.
Format	AINF Example: AINF
Response	AINF,<model_number>,<serial_number>,<temperature>, <encoder_initialization_status>
Notes	• Temperature is reported in degrees Celsius. • If the actuator is not initialized, the "0x5005" error code will be displayed after the "AINF" command. • If the encoder initialization status is 0, it should be reinitialized using the "EAIN" command or by manually moving the rotor.

Command: CBRS (Reset Controller Board)

Description	This command resets the controller board.
Format	CBRS Example: CBRS
Response	None.
Notes	• If this command is sent via the USB peripheral, the serial port must be reopened.

Command: ASCA (Actuator Self-calibration)

Description	This command starts the actuator self-calibration process.
Format	ASCA,<Range> Example: ASCA,10.5
Response	ASCA\n<Response_Code> If no error occurs, "DONE" will be reported as the response code on a new line; otherwise, "EROR" will be displayed.
Notes	• The actuator and encoder must be initialized before using this command. • The actuator calibration range is restricted to 0 to 24.5 mm and should be set based on the actuator working stroke. • Self-calibration should be performed whenever the actuator orientation changes. • This command should be applied periodically, depending on the application requirements. • During the self-calibration process, the actuator must remain fixed, and the rotor should be free to move.

Command: EAIN (Encoder Auto-Init)

Description	This command initializes the encoder and finds the absolute value of the rotor position.
Format	EAIN,<Range> Example: EAIN
Response	EAIN\n<Response_Code> If no error occurs, "DONE" will be reported as the response code on a new line; otherwise, the error code will be displayed (p. 58).
Notes	• The actuator should be initialized before using this command. • The encoder initiation process can be performed by moving the rotor manually. • The encoder initialization status is available using the "AINF" command.

Command: FWUP (Firmware Update)

Description	This command initiates the Firmware Update.
Format	FWUP,<value> Example: FWUP,1
Response	FWUP,<Response_Code> If no error occurs, "DONE" will be reported as the response code; otherwise, the error code will be displayed (p. 58).
Notes	• The <value> parameter can be set to 0 or 1. • The controller board will restart after using this command. • If the <value> parameter is set to 1, the red LED on the controller board starts blinking, indicating that the controller is ready to receive new firmware. • It is highly recommended not to use this command unless instructed by customer service.

5.6.2 Configurations

5.6.2.1 Controller Board Configuration

These configurations are defined for the controller board and can be applied regardless of whether the actuator is connected or not.

5.6.2.1.1 General Configuration

Overview

Configuration	Description
CSNM	Controller Board Serial Number
BSTO	Bypass STO

Configuration: CSNM (Controller Board Serial Number)

Description	This configuration indicates the unique serial number of the controller board.
Access	Read Only
Notes	None.

Configuration: BSTO (Bypass STO)

Description	This configuration is used to bypass the STO inputs.
Access	Read and Write
Notes	• This configuration can be set to either 0 or 1. • If the BSTO configuration is set to 0, both STO inputs must be in the "HIGH" state (as defined by the isolated input power supply voltage) to allow the macro to start; otherwise, the STO Error (0x7004) will be displayed. • If the "BSTO" configuration is set to 1, the start command does not consider the states of the STO inputs. • The default value is 0.

5.6.2.1.2 Data Recording Configuration

Overview

Configuration	Description
DFEN	Offline Data Recording Enable
DFFP	Offline Data Recording Frequency Pre-scaler
DFTM	Offline Data Recording Trigger Mode
DFMC	Offline Data Recording Macro Counter
DNEN	Online Data Recording Enable
DNFP	Online Data Recording Frequency Pre-scaler
DNTM	Online Data Recording Trigger Mode
DNMC	Offline Data Recording Macro Counter
DTYP	Data Recording Type

Configuration: DFEN (Offline Data Recording Enable)

Description	This configuration is used to enable offline data recording.
Access	Read and Write
Notes	• This configuration can be set to either 0 or 1. • If the "DFEN" configuration is set to 0, no data will be recorded while running the macro. • If the "DFEN" configuration is set to 1, data can be recorded while running the macro, depending on the other offline data recording settings. • The default value is 0.

Configuration: DFFP (Offline Data Recording Frequency Pre-scaler)

Description	This configuration is used to set the data recording frequency.
Access	Read and Write
Notes	• The lower and upper limits for this configuration are 1 and 25000, respectively. • The offline data recording frequency is determined by dividing 25 kHz by the "DFFP" value. Example: using 25 as the "DFFP" value results in a data recording frequency of 1 kHz. • The default value is 1.

Configuration: DFTM (Offline Data Recording Trigger Mode)

Description	This configuration defines the offline data recording mode as hardware or software.
Access	Read and Write
Notes	• This configuration can be set to either 0 or 1. • If the "DFTM" configuration is set to 0, software mode activates, and offline data recording begins after the specified macro type number, which can be configured using the "DFMC" setting. • If the "DFTM" configuration is set to 1, hardware mode activates, and offline data recording occurs when isolated digital input pin is in a "HIGH" state. Note that the input type for the selected pin must be set to "DRFT" in advance. • The default value is 0.

Configuration: DFMC (Offline Data Recording Macro Counter)

Description	This configuration is used exclusively for offline data recording in software mode and establishes the starting macro type number for the data recording.
Access	Read and Write
Notes	• The lower and upper limits for this configuration are 0 and 20, respectively. • The default value is 0.

Configuration: DNEN (Online Data Recording Enable)

Description	This configuration is used to enable online data recording.
Access	Read and Write
Notes	• This configuration can be set to either 0 or 1. • If the "DNEN" configuration is set to 0, no data will be sent while running the macro. • If the "DNEN" configuration is set to 1, data can be sent while running the macro, depending on the other online data recording settings. • The default value is 1.

Configuration: DNFP (Online Data Recording Frequency Pre-scaler)

Description	This configuration is used to set the data recording frequency.
Access	Read and Write
Notes	• The lower and upper limits for this configuration are 1 and 25000, respectively. • The online data recording frequency is determined by dividing 25 kHz by the "DNFP" value. Example: using 25 as the "DNFP" value results in a data recording frequency of 1 kHz. • The default value is 50.

Configuration: DNTM (Online Data Recording Trigger Mode)

Description	This configuration defines the online data recording mode as hardware or software.
Access	Read and Write
Notes	• This configuration can be set to either 0 or 1. • If the "DNTM" configuration is set to 0, software mode activates, and online data transmission starts after the specified macro type number, which can be configured using the "DNMC" setting. • If the "DNTM" configuration is set to 1, hardware mode activates, and online data is transmitted when isolated digital input pin is in a "HIGH" state. Note that the input type for the selected pin must be set to "DRNT" in advance. • The default value is 0.

Configuration: DNMC (Online Data Recording Macro Counter)

Description	This configuration is used exclusively for online data recording in software mode and determines the starting macro type number for data transmission.
Access	Read and Write
Notes	• The lower and upper limits for this configuration are 0 and 20, respectively. • The default value is 0.

Configuration: DTYP (Data Recording Type)

Description	This configuration determines the data type to be recorded or sent in both offline and online modes.
Access	Read and Write
Notes	• The available data types are listed as follows:

Data type code	Description
ENCR	Rotor position
CCUR	Current through the rotor winding
MCEF	Mechanical control effort
MERR	Mechanical error
MSET	Mechanical set point

ECEF	Electrical control effort
EERR	Electrical error
ESET	Electrical set point
ATMP	Actuator temperature
CTMP	Controller board temperature
VBUS	Controller board bus voltage
ISRC	Control cycle count for each macro repeat
MCRC	Macro type counter
DIN1	Isolated digital input 1 status
DIN2	Isolated digital input 2 status
DIN3	Isolated digital input 3 status
DIN4	Isolated digital input 4 status
MCRR	Macro repeat counter
FRCE	Estimated external force
AIN1	Analog input 1
AIN2	Analog input 2

- Online and offline data require post-processing to extract relevant information as follows:
  - The "ENCR" data should be divided by the encoder interpolation gain and encoder LPI (available through the "ELPI" and "EINT" configurations), and then multiplied by "6.35" to obtain the rotor position in millimeters.
  - The "CCUR" data should be divided by "5241.6" to obtain the current through the rotor winding in amperes.
  - The "MCEF" data should be divided by "5,241,600" to obtain the mechanical control effort in amperes.
  - The "MERR" data must first be divided by "1000", and then divided by the encoder interpolation gain and encoder LPI, and finally multiplied by "6.35" to obtain the mechanical error in millimeters.
  - The "MSET" data should be divided by the encoder interpolation gain and encoder LPI, and then multiplied by "6.35" to determine the mechanical set point in millimeters.
  - The "ECEF" data should first be divided by "3,400,000," and then multiplied by "48" to calculate the electrical control effort in volts.

- The "EERR" data should be divided by "5,241,600" to determine the electrical error in amperes.
- The "ESET" data should be divided by "5241.6" to obtain the electrical set point in amperes.
- Actuator temperature data is obtained using the following formula in degrees Celsius:

$$Actuator\ Temp = \frac{1}{1.1 \times 10^{-4} \times ln\left(-\frac{6.1 \times 10^{-4} \times ATMP}{6.1 \times 10^{-4} \times ATMP - 3.3}\right) + 3.4 \times 10^{-3}} - 270.0$$

- Controller board temperature data is obtained using the following formula in degrees Celsius:

$$Controller\ Temp = \frac{1}{1.3 \times 10^{-4} \times ln\left(-\frac{6.1 \times 10^{-4} \times CTMP}{6.1 \times 10^{-4} \times CTMP - 2.5}\right) + 3.4 \times 10^{-3}} - 270.0$$

- The "VBUS" data should be multiplied by "0.0158" to calculate the controller board bus voltage in volts.
- The "FRCE" data should be divided by "1000" to determine the external force in newtons.
- The "AIN1" data should be divided by "81.9" and then added to "-25" to obtain the differential input voltage in volts.
- The "AIN2" data should be divided by "-163.8" and then added to "12.5" to obtain the input voltage in volts.

- The estimated external force data (FRCE) is valid only if the rotor is fixed or moving at a constant velocity.
- The analog input 1 and 2 data (AIN1 and AIN2) are valid only if "AIEN" configuration is enabled.
- The data type can be configured as either permanent or temporary. To set the desired data types, enter their codes followed by "DTYP".
  Example: SCON,TEMP,DTYP,ENCR,CCUR,FRCE,ATMP.
- The maximum number of allowed data types is 25. If more data types are assigned, the error code "0xFF19" will be displayed.
- Data will be sent in the same order as specified in the data type configuration settings.
- If an error occurs while setting the data type, the data type array resets to zero, and no data type will be configured.

5.6.2.1.3 Serial Interface Configuration Overview

Configuration	Description
RSEN	RS485/422 Enable
RSXP	RS485/422 Transmitter and Receiver Phase
RSSR	RS485/422 Slew-Rate Limit
RSBR	RS485/422 Baud Rate
MBEN	Modbus Enable
MBID	Modbus Unit ID

Configuration: RSEN (RS485/422 Enable)

Description	This configuration enables the RS485/422 serial interface.
Access	Read and Write
Notes	• This configuration can be set to either 0 or 1. • If "RSEN" is set to 0, the RS485/422 serial interface is disabled. • If "RSEN" is set to 1, the RS485/422 serial interface is enabled. • This configuration saves power by putting the RS485/422 IC into low-power shutdown mode. • The default value is 1.

Configuration: RSXP (RS485/422 Transmitter and Receiver Phase)

Description	This configuration inverts the polarity of the transmitter and receiver lines.
Access	Read and Write
Notes	• This configuration can be set to either 0 or 1. • If "RSXP" is set to 0, normal polarity is selected. • If "RSXP" is set to 1, normal polarity is inverted for both the transmitter and receiver lines. • The default value is 0.

Configuration: RSSR (RS485/422 Slew-Rate Limit)

Description	This configuration selects the communication rate.
Access	Read and Write
Notes	• This configuration minimizes EMI and reduces reflections caused by improperly terminated cables. • This configuration can be set to either 0 or 1. • If "RSSR" is set to 0, the communication rate of 16 Mbps is selected. • If "RSSR" is set to 1, the communication rate of 500 kbps is selected. • The default value is 1.

Configuration: RSBR (RS485/422 Baud Rate)

Description	This configuration selects the communication baud rate.
Access	Read and Write
Notes	• The lower and upper limits for this configuration are 2,595 bps and 10,625,000 bps, respectively. • The default value is 115,200. • If the desired baud rate exceeds 500 kbps, ensure that the "RSSR" configuration is set to 0. • If the desired baud rate is less than 500 kbps, ensure that the "RSSR" configuration is set to 1. • The default value is 115,200 bps.

Configuration: MBEN (Modbus Enable)

Description	This configuration switches between the ADMORS and Modbus protocols for the RS485/422 peripheral.
Access	Read and Write
Notes	• This configuration can be set to either 0 or 1. • If "MBEN" is set to 0, the ADMORS protocol is selected for the RS485/422 peripheral. • If "MBEN" is set to 1, the Modbus protocol is selected for the RS485/422 peripheral. • The default value is 0.

Configuration: MBID (Modbus Unit ID)

Description	This configuration assigns a unit ID to the controller board used in the Modbus network.
Access	Read and Write
Notes	• The lower and upper limits for this configuration are 1 and 247, respectively. • The default value is 7.

5.6.2.1.4 CANopen Configuration

Overview

Configuration	Description
COEN	CANopen Enable
COID	CANopen Unit ID
COPS	CANopen Frequency Pre-scaler
COS1	CANopen Frequency Time Segment 1
COS2	CANopen Frequency Time Segment 2

Configuration: COEN (CANopen Enable)

Description	This configuration enables the CANopen protocol.
Access	Read and Write
Notes	• This configuration can be set to either 0 or 1. • If " COEN " is set to 0, the CANopen protocol is disabled. • If " COEN " is set to 1, the CANopen protocol is enabled. • The default value is 0.

Configuration: COID (CANopen Unit ID)

Description	This configuration assigns a unit ID to the controller board used in the CANopen network.
Access	Read and Write
Notes	• The lower and upper limits for this configuration are 1 and 127, respectively. • The default value is "32" or "0x20h".

Configuration: COPS (CANopen Frequency Pre-scaler)

Description	This configuration is one of the parameters used to define the CANopen baud rate.
Access	Read and Write
Notes	• The lower and upper limits for this configuration are 1 and 512, respectively. • The default value is 4.

Configuration: COS1 (CANopen Frequency Time Segment 1)

Description	This configuration is one of the parameters used to define the CANopen baud rate.
Access	Read and Write
Notes	• The lower and upper limits for this configuration are 2 and 256, respectively. • The default value is 13.

Configuration: COS2 (CANopen Frequency Time Segment 2)

Description	This configuration is one of the parameters used to define the CANopen baud rate.
Access	Read and Write
Notes	• The lower and upper limits for this configuration are 2 and 128, respectively. • The default value is 2.

Notes

- The CANopen baud rate is defined by "COPS", "COS1", and "COS2" configurations.
- The default baud rate is set to 125 kbps.
- Here is a list of configuration values for typical baud rates in CANopen networks:

Baud Rate (kbps)	COPS	COS1	COS2
20	25	13	2
50	10	13	2
100	5	13	2
125	4	13	2
250	2	13	2
500	1	13	2
800	1	5	4
1000	1	5	2

The following formula can be used to calculate the CANopen baud rate:

$$CANopen_{Baud\ Rate} = \frac{8{,}000{,}000}{COPS\ (1 + COS1 + COS2)}$$

If the desired baud rate cannot be obtained through any combinations of "COPS", "COS1", and "COS2" values, contact customer service.

5.6.2.1.5 IO Configuration

Overview

Configuration	Description
AIEN	Analog Input Enable
DI1T	Digital Input 1 Type
DI2T	Digital Input 2 Type
DI3T	Digital Input 3 Type
DI4T	Digital Input 4 Type
MSEN	Macro Status Digital Output Enable

Configuration: AIEN (Analog Input Enable)

Description	This configuration enables both analog Input 1 and analog Input 2.
Access	Read and Write
Notes	• This configuration can be set to either 0 or 1. • If "AIEN" is set to 0, analog Inputs are disabled. • If "AIEN" is set to 1, analog Inputs are enabled. • The default value is 0.

Configuration: DI1T (Digital Input 1 Type)

Description	This configuration sets the digital Input type for isolated input 1.
Access	Read and Write
Notes	• This configuration can be set to "READ", "RMCR", "PMCR", "PARK", "DRFT", or "DRNT". • "READ" is the default type. If this type is used, the input state can be accessed using data recording. • If "RMCR" is set, the macro will start on the rising edge of the input pin. • If "PMCR" is set, the active macro will terminate on the rising edge of the input pin. Note that if the current macro defines a motion trajectory and the actuator is not at the homing position, this may cause the rotor to strike the stator, which can lead to damage if repeated. • If "PARK" is set, the rotor moves to its homing position on the rising edge of the input pin. This ensures a safe return to the homing position and prevents the rotor from striking the stator. • If "DRFT" is set, isolated digital input 1 is selected as the input pin for offline data recording in hardware mode. • If "DRNT" is set, isolated digital input 1 is selected as the input pin for online data recording in hardware mode. • The logic voltage level of the digital pin ranges from 5 V to 24 V and is defined by the isolated input supply.

Configuration: DI2T (Digital Input 2 Type)

Description	This configuration sets the digital Input type for isolated input 2.
Access	Read and Write
Notes	• This configuration can be set to "READ", "RMCR", "PMCR", "PARK", "DRFT", or "DRNT". • "READ" is the default type. If this type is used, the input state can be accessed using data recording. • If "RMCR" is set, the macro will start on the rising edge of the input pin. • If "PMCR" is set, the active macro will terminate on the rising edge of the input pin. Note that if the current macro defines a motion trajectory and the actuator is not at the homing position, this may cause the rotor to strike the stator, which can lead to damage if repeated. • If "PARK" is set, the rotor moves to its homing position on the rising edge of the input pin. This ensures a safe return to the homing position and prevents the rotor from striking the stator. • If "DRFT" is set, isolated digital input 2 is selected as the input pin for offline data recording in hardware mode. • If "DRNT" is set, isolated digital input 2 is selected as the input pin for online data recording in hardware mode. • The logic voltage level of the digital pin ranges from 5 V to 24 V and is defined by the isolated input supply.

Configuration: DI3T (Digital Input 3 Type)

Description	This configuration sets the digital Input type for isolated input 3.
Access	Read and Write
Notes	• This configuration can be set to "READ", "RMCR", "PMCR", "PARK", "DRFT", or "DRNT". • "READ" is the default type. If this type is used, the input state can be accessed using data recording. • If "RMCR" is set, the macro will start on the rising edge of the input pin. • If "PMCR" is set, the active macro will terminate on the rising edge of the input pin. Note that if the current macro defines a motion trajectory and the actuator is not at the homing position, this may cause the rotor to strike the stator, which can lead to damage if repeated. • If "PARK" is set, the rotor moves to its homing position on the rising edge of the input pin. This ensures a safe return to the homing position and prevents the rotor from striking the stator. • If "DRFT" is set, isolated digital input 3 is selected as the input pin for offline data recording in hardware mode.

	• If "DRNT" is set, isolated digital input 3 is selected as the input pin for online data recording in hardware mode. • The logic voltage level of the digital pin ranges from 5 V to 24 V and is defined by the isolated input supply.

Configuration: DI4T (Digital Input 4 Type)

Description	This configuration sets the digital Input type for isolated input 4.
Access	Read and Write
Notes	• This configuration can be set to "READ", "RMCR", "PMCR", "PARK", "DRFT", or "DRNT". • "READ" is the default type. If this type is used, the input state can be accessed using data recording. • If "RMCR" is set, the macro will start on the rising edge of the input pin. • If "PMCR" is set, the active macro will terminate on the rising edge of the input pin. Note that if the current macro defines a motion trajectory and the actuator is not at the homing position, this may cause the rotor to strike the stator, which can lead to damage if repeated. • If "PARK" is set, the rotor moves to its homing position on the rising edge of the input pin. This ensures a safe return to the homing position and prevents the rotor from striking the stator. • If "DRFT" is set, isolated digital input 4 is selected as the input pin for offline data recording in hardware mode. • If "DRNT" is set, isolated digital input 4 is selected as the input pin for online data recording in hardware mode. • The logic voltage level of the digital pin ranges from 5 V to 24 V and is defined by the isolated input supply.

Configuration: MSEN (Macro Status Digital Output Enable)

Description	This configuration enables the macro status digital output.
Access	Read and Write
Notes	• Enabling this configuration causes isolated output 1 to switch to the "HIGH" state for 4 milliseconds, then return to the "LOW" state for each macro type in the desired trajectory. • This configuration can be 0 or 1. • If "MSEN" is set to 0, the macro status digital output is disabled. • If "MSEN" is set to 1, the macro status digital output is enabled. • The default value is 0.

5.6.2.2 Actuator Configuration

These configurations are defined for the actuator and stored in its flash memory. They can be applied only when the actuator is connected to the controller; otherwise, an actuator initialization error (0x5005) will be displayed as a result.

5.6.2.2.1 General Configuration

Overview

Configuration	Description
ACMN	Actuator Model Number
ACSN	Actuator Serial Number

Configuration: ACMN (Actuator Model Number)

Description	This configuration indicates the actuator model number.
Access	Read Only
Notes	None.

Configuration: ACSN (Actuator Serial Number)

Description	This configuration indicates the unique serial number of the actuator.
Access	Read Only
Notes	None.

5.6.2.2.2 Controller Parameter Configuration

Overview

Configuration	Description
ECN0	Electrical Controller Nominator 0
ECN1	Electrical Controller Nominator 1
ECN2	Electrical Controller Nominator 2
ECD1	Electrical Controller Denominator 1
ECD2	Electrical Controller Denominator 2

SCN0	Position Step Controller Nominator 0
SCN1	Position Step Controller Nominator 1
SCN2	Position Step Controller Nominator 2
SCD1	Position Step Controller Denominator 1
SCD2	Position Step Controller Denominator 2
RCN0	Position Ramp Controller Nominator 0
RCN1	Position Ramp Controller Nominator 1
RCN2	Position Ramp Controller Nominator 2
RCD1	Position Ramp Controller Denominator 1
RCD2	Position Ramp Controller Denominator 2

Configurations: Controller Parameters

Description	These configurations specified the controller parameters used in the current and position control algorithm.
Access	Read and Write
Notes	• The lower and upper limits for all these configurations are -5000 and 5000, respectively. • Only 5 digits after the floating point are considered by the firmware for the controller parameter values. • It is highly recommended to not change these values, as they are designed for each actuator individually.

5.6.2.2.3 Actuator Encoder Parameter Configuration

Overview

Configuration	Description
ELPI	Encoder LPI
EINT	Encoder Interpolation Gain

Configuration: ELPI (Encoder LPI)

Description	This configuration specifies the encoder strip resolution.
Access	Read Only
Notes	• This configuration is reported as lines per inch.

Configuration: EINT (Encoder Interpolation Gain)

Description	This configuration specifies the encoder interpolation gain.
Access	Read Only
Notes	• The encoder resolution is four times the product of the encoder interpolation gain and the encoder LPI.

5.6.2.2.4 Actuator Parameter Configuration

Overview

Configuration	Description
EPRE	Coil Winding Resistance
EPIN	Coil Winding Inductance
MPMA	Mechanical Plane Mass
MPDA	Mechanical Plant Damping Coefficient
MPSP	Mechanical Plant Spring Coefficient
MPFC	Mechanical Plant Force Constant
SPFC	Spring Force

Configuration: EPRE (Coil Winding Resistance)

Description	This configuration indicates the coil winding resistance
Access	Read Only
Notes	• This value is reported in ohms.

Configuration: EPIN (Coil Winding Inductance)

Description	This configuration indicates the coil winding inductance.
Access	Read Only
Notes	• This value is reported in millihenries.

Configuration: MPMA (Mechanical Plane Mass)

Description	This configuration indicates the mass of the actuator moving parts.
Access	Read Only
Notes	• This value is reported in grams.

Configuration: MPDA (Mechanical Plant Damping)

Description	This configuration indicates the damping coefficient of the actuator plant.
Access	Read Only
Notes	• This value is obtained through a system identification algorithm. • This value is reported in kilograms per second (kg/s).

Configuration: MPSP (Mechanical Plant Spring Coefficient)

Description	This configuration indicates the spring coefficient of the actuator plant.
Access	Read Only
Notes	• This value is obtained through a system identification algorithm. • This value is reported in newtons per meter (N/m).

Configuration: MPFC (Mechanical Plant Force Constant)

Description	This configuration indicates the force constant of the actuator based on the rotor position.
Access	Read Only
Notes	• This configuration consists of 51 values that specify the actuator force constant with a rotor position step of 0.5 millimeters. • The reported values are in newtons per ampere (N/A).

Configuration: SPFC (Spring Force)

Description	This configuration represents the spring force applied to the actuator, depending on the rotor position.
Access	Read Only
Notes	• This configuration consists of 102 values that specify the spring force, with a rotor position step of 0.5 millimeters, intended for forward and backward movement. • These values are obtained by the actuator self-calibration algorithm.

5.6.3 Macros and Trajectory Design

Overview

Macro Type	Description
TVSI	Time-based Voltage Sine
TCST	Time-based Current Step
TCSI	Time-based Current Sine
CHOL	Time-based Current Hold
TPST	Time-based Position Step
TPSI	Time-based Position Sine
TPRA	Time-based Position Ramp
TPAC	Time-based Position Acceleration
PPAC	Position-based Position Acceleration
PHOL	Time-based Position Hold
TFST	Time-based Force Step
SOFT	Soft-Touch
MEAS	Measurement Only
PARK	Park
AOUT	Analog Output
DOUT	Digital Output
SREL	Soft-Release
TVSP	Time-based Voltage Setpoint
TCSP	Time-based Current Setpoint
TPSP	Time-based Position Setpoint

Macro Type: TVSI (Time-based Voltage Sine)

Description	This macro applies an open-loop voltage to the rotor winding.
Format	TVSI,<Amplitude>,<Frequency>,<Offset>,<Duration> Example: TVSI,5,10,2,1
Notes	• Amplitude is specified in volts. The lower and upper limits are -48 V and 48 V, respectively. • Frequency is specified in hertz; the lower and upper limits are 0 Hz and 5 kHz, respectively. • Offset is specified in volts. The lower and upper limits are set based on the amplitude value to ensure that the combination of the offset and amplitude does not exceed ±48V. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for all the inputs values. • This macro terminates when the desired time has elapsed or an unwanted error occurs.

Macro Type: TCST (Time-based Current Step)

Description	This macro passes a closed-loop current with a step wave shape through the rotor winding.
Format	TCST,<Set-point>,<Duration> Example: TCST,1,1
Notes	• Set-point is specified in amperes. The lower and upper limits are -5 A and 5 A, respectively. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for all the input values. • This macro terminates when the desired time has elapsed or an unwanted error occurs.

Macro Type: TCSI (Time-based Current Sine)

Description	This macro applies a closed-loop current with a sine wave shape through the rotor winding.
Format	TCSI,<Amplitude>,<Frequency>,<Duration> Example: TCSI,1,10 ,1
Notes	• Amplitude is specified in volts. The lower and upper limits are -5 A and 5 A, respectively. • Frequency is specified in hertz; the lower and upper limits are 0 Hz and 2.5 kHz, respectively. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • The measured current when the macro starts is considered as the offset value for the sine wave. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for all the inputs values. • This macro terminates when the desired time has elapsed or an unwanted error occurs.

Macro Type: CHOL (Time-based Current Hold)

Description	This macro maintains the current value passing through the rotor winding.
Format	CHOL,<Duration> Example: CHOL,1
Notes	• This command measures the current at the start of the macro and defines it as the set-point of the current control loop. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for all the input values. • This macro terminates when the desired time has elapsed or an unwanted error occurs.

Macro Type: TPST (Time-based Position Step)

Description	This macro moves the rotor in closed-loop mode to reach a specified position which is defined by a step wave.
Format	TPST,<Set-point>,<Duration> Example: TPST,10,1
Notes	• Set-point is specified in millimeters. The lower and upper limits are 0 mm and 24 mm, respectively. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for all the input values. • This macro terminates when the desired time has elapsed or an unwanted error occurs. • Since the step command requires a large control effort to reach the target quickly, it creates a high force on the actuator, which can damage it with repeated use. Therefore, it is recommended to use the Position-based Position Acceleration (PPAK) command instead of the step input.

Macro Type: TPSI (Time-based Position Sine)

Description	This macro moves the rotor in a sine wave shape using a closed-loop algorithm.
Format	TPSI,<Amplitude>,<Frequency>,<Duration> Example: TPSI,10,1,5
Notes	• Amplitude is specified in millimeters. The lower and upper limits are -12.5 mm and 12.5 mm, respectively. • Frequency is specified in hertz; the lower and upper limits are 0 Hz and 50 Hz, respectively. • The measured position when the macro starts is desired as the offset value for the sine wave. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for all the input values. • This macro terminates when the desired time has elapsed or an unwanted error occurs.

Macro Type: TPRA (Time-based Position Ramp)

Description	This macro moves the rotor with a constant velocity in closed-loop mode.
Format	TPRA,<Slope>,<Duration> Example: TPRA,10,1
Notes	• The position slope is specified in millimeters per second. The lower and upper limits are -180 mm/s and 180 mm/s, respectively. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • Only 3 digits after the floating point are considered by the firmware for all the input values. • This macro terminates when the desired time has elapsed or an unwanted error occurs. • The rotor moving range must be considered in setting the duration and slope parameters. These values must be set carefully to prevent the rotor from striking the stop barriers at the beginning and end of the stroke.

Macro Type: TPAC (Time-based Position Acceleration)

Description	This macro moves the rotor with a constant acceleration over the specified time duration.
Format	TPAC,<Acceleration>,<Duration> Example: TPAC,100,1
Notes	• This macro consists of two constant acceleration profiles that are inverted and calculates the final position to reach zero velocity at the end of the macro. • The measured position when the macro begins is desired as the starting point, which is needed to find the final position. • Acceleration is specified in millimeters per second squared. The lower and upper limits are -100,000 mm/s² and 100,000 mm/s², respectively. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • Only 3 digits after the floating point are considered by the firmware for all the inputs values. • This macro terminates when the desired time has elapsed or an unwanted error occurs. • The rotor moving range must be considered in setting the duration and acceleration parameters. These values must be set carefully to prevent the rotor from striking the stop barriers at the beginning and end of the stroke. The firmware checks the rotor moving range and reports an error if this condition is restricted.

Macro Type: PPAC (Position-based Position Acceleration)

Description	This macro moves the rotor with a constant acceleration to reach the desired final position.
Format	PPAC,<Acceleration>,<Final_position> Example: PPAC,100,10
Notes	• This macro consists of two inverted constant acceleration profiles and calculates the required time to reach zero velocity at the end of the macro. • The measured position at the beginning of the macro is desired as the starting point, which is needed to calculate the required time. • Acceleration is specified in millimeters per second squared. The lower and upper limits are -100,000 mm/s² and 100,000 mm/s², respectively. • The final position is specified in millimeters. The lower and upper limits are 0 mm and 24 mm, respectively. • Only 3 digits after the floating point are considered by the firmware for all the input values. • This macro ends when the desired final position is reached or if an unwanted error occurs.

Macro Type: PHOL (Time-based Position Hold)

Description	This macro maintains the rotor in its current position.
Format	PHOL,<Duration> Example: PHOL,1
Notes	• The measured position at the beginning of the macro is desired as position set-point in the control algorithm. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for all the inputs values. • This macro ends when the desired time has elapsed or an unwanted error occurs.

Macro Type: TFST (Time-based Force Step)

Description	This macro controls the external force to reach the desired set-point defined by a step signal.
Format	TFST,<Set-point>,<Duration> Example: TFST,5,1
Notes	• The force feedback signal is generated using advanced sensorless algorithms, removing the need for an external load cell. • Force set-point is specified in newtons. The lower and upper limits are -35 N and 35 N, respectively. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for all the input values. • This macro terminates when the desired time has elapsed or an unwanted error occurs. • The external force estimation is valid only at zero acceleration. This command should be used solely for constant velocity motions or fixed positions.

Macro Type: SOFT (Soft-Touch)

Description	This macro applies the Soft-Touch algorithm to recognize any surface while exerting minimal force.
Format	SOFT,<Slope>,<Saturation>,<Error-Threshold>, < No-Soft-Touch-Duration>,<Duration> Example: SOFT,2,0.1,0.1,0.2,5
Notes	• The Soft-Touch macro operates at a constant velocity. • The position slope is specified in millimeters per second. The lower and upper limits are -180 mm/s and 180 mm/s, respectively. • Saturation is specified in amperes and indicates the maximum current that can pass through the rotor winding. • The lower and upper limits for the "saturation" parameter are -5 A and 5 A, respectively. • Error-Threshold is specified in amperes. The lower and upper limits are -5 A and 5 A, respectively. • No-Soft-Touch-Duration is specified in seconds and indicates the required time of the velocity control algorithm. In this interval, the force estimation and soft-touch algorithms are not applied. • The lower and upper limits of "No-Soft-Touch-Duration" parameter are 0.01 and 10 seconds, respectively. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input.

	• Only 3 digits after the floating point are considered by the firmware for all the inputs values. • This macro ends when the specified duration elapses, the Soft-Touch algorithm recognizes a surface, or an unwanted error occurs. • Slope, saturation, and error-threshold parameters must be fine-tuned for each application.

Macro Type: MEAS (Measurement Only)

Description	This macro is used to record data without appalling any voltage to the rotor winding.
Format	MEAS,<Duration> Example: MEAS,1
Notes	• Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for all the input values. • This macro terminates when the desired time has elapsed or an unwanted error occurs.

Macro Type: PARK (Park)

Description	This macro moves the rotor to its homing position.
Format	PARK,<Acceleration> Example: PARK,-300
Notes	• This macro resembles the Position-based Position Acceleration (PPAC), but in this case, the final position is defined as zero. • Acceleration is specified in millimeters per second squared. The lower and upper limits are -100,000 mm/s² and -1 mm/s², respectively. • Only 3 digits after the floating point are considered by the firmware for all the input values. • This macro terminates when the rotor reaches the homing position or an unwanted error occurs. • It is highly recommended to set this macro type as the final step in any motion trajectory. This macro ensures the rotor returns safely to the homing position and prevents it from striking the stator after the trajectory is complete.

Macro Type: AOUT (Analog Out)

Description	This macro maps the desired data type to the analog output pins.
Format	AOUT,<PIN#>,<Data_Type> Example: AOUT,PIN1,ENCR
Notes	• This macro maps the selected data type to ±10 V. • Only PIN1 and PIN2 can be selected as analog output pins. • Available <Data_Type> options, along with their corresponding maximum and minimum values, are listed as follows:

Data type code	Description	Max (10V)	Min (-10V)
ENCR	Rotor position	24mm	0mm
CCUR	Current through the rotor winding	5A	-5A
MCEF	Mechanical control effort	5A	-5A
MERR	Mechanical error	5mm	-5mm
MSET	Mechanical set point	24mm	0mm
ECEF	Electrical control effort	48V	0V
EERR	Electrical error	0.5A	-0.5A
ESET	Electrical set point	5A	-5A
ATMP	Actuator temperature	4095	0
CTMP	Controller board temperature	4095	0
VBUS	Controller board bus voltage	55V	40V
FRCE	Estimated external force	25N	-25N

- To disable the analog output, set the <Data_Type> to "DOFF". This will set the pin to 0V. Example: AOUT,PIN1,DOFF

Macro Type: DOUT (Digital Output)

Description	This macro sets the digital pin state to a HIGH or LOW voltage level.
Format	DOUT,<PIN#>,<State> Example: DOUT,PIN1,PSET
Notes	• To set the digital output pin to a HIGH logic level, "PSET" should be used as <State>. • To set the digital output pin to a LOW logic level, "PRES" should be used as <State>. • The logic voltage level of the digital output pins ranges from 5 V to 24 V and is defined by the isolated input supply. • PIN1 and PIN2 are fast output, while PIN3, PIN4, and PIN5 are high-current output (400 mA). • If the "MSEN" configuration is enabled, the DOUT macro for PIN1 is automatically ignored.

Macro Type: SREL (Soft-Release)

Description	This macro applies the Soft Release algorithm to disengage from a surface smoothly.
Format	SREL,<Slope>,<Saturation>,<Error-Threshold>, < No-Soft-Release-Duration>,<Duration> Example: SREL,2,0.1,0.1,0.2,5
Notes	• The Soft-Release macro operates at a constant velocity. • The position slope is specified in millimeters per second. The lower and upper limits are -180 mm/s and 180 mm/s, respectively. • Saturation is specified in amperes and indicates the maximum current that can pass through the rotor winding. • The lower and upper limits for the "saturation" parameter are -5 A and 5 A, respectively. • Error-Threshold is specified in amperes. The lower and upper limits are -5 A and 5 A, respectively. • No-Soft-Release-Duration is specified in seconds and indicates the required time of the velocity control algorithm. In this interval, the force estimation and soft-release algorithms are not applied. • The lower and upper limits of "No-Soft-Release-Duration" parameter are 0.01 and 10 seconds, respectively. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for all the inputs values. • This macro ends when the specified duration elapses, the Soft-Release algorithm detects a release from the surface, or an error occurs. • Slope, saturation, and error-threshold parameters must be fine-tuned for each application.

Macro Type: TVSP (Time-based Voltage Setpoint)

Description	This macro sets the open-loop voltage value applied to the rotor winding.
Format	TVSP,<Setpoint Input>,<Duration> Example: TVSP,A1SP,10
Notes	• The voltage setpoint can be specified using CANopen, Modbus, or analog input pins. • To specify the setpoint with CANopen protocol, set <Setpoint_Input> to "CISP". The setpoint value is provided through object 0x6506. • To specify the setpoint with Modbus protocol, set <Setpoint_Input> to "MISP". The setpoint value is provided through register 0x40004.

	• To specify the setpoint with Analog input 1, set <Setpoint_Input> to "A1SP". In this case, the ±20 V input range is mapped to 0–48 V on the rotor winding. • To specify the setpoint with the Analog input 2, set <Setpoint_Input> to "A2SP". In this case, the ±10 V input range is mapped to 0–48 V on the rotor winding. • For analog input case, the "AIEN" configuration must be enabled in advance. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for the <Duration> value. • This macro terminates when the desired time has elapsed or an unwanted error occurs.

Macro Type: TCSP (Time-based Current Setpoint)

Description	This macro sets the closed-loop current value applied to the rotor winding.
Format	TCSP,<Setpoint_Input>,<Duration> Example: TCSP,A1SP,10
Notes	• The current setpoint can be specified using CANopen, Modbus, or analog input pins. • To specify the setpoint with CANopen protocol, set <Setpoint_Input> to "CISP". The setpoint value is provided through object 0x6506. • To specify the setpoint with Modbus protocol, set <Setpoint_Input> to "MISP". The setpoint value is provided through register 0x40004. • To specify the setpoint with Analog input 1, set <Setpoint_Input> to "A1SP". In this case, the ±20 V input range is mapped to ±5 A through the rotor winding. • To specify the setpoint with Analog input 2, set <Setpoint_Input> to "A2SP". In this case, the ±10 V input range is mapped to ±5 A through the rotor winding. • For analog input case, the "AIEN" configuration must be enabled in advance. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for the <Duration> value. • This macro terminates when the desired time has elapsed or an unwanted error occurs.

Macro Type: TPSP (Time-based Position Setpoint)

Description	This macro sets the closed-loop rotor position value.
Format	TPSP,<Setpoint_Input>,<Duration> Example: TPSP,A1SP,10
Notes	• The position setpoint can be specified using CANopen, Modbus, or analog input pins. • To specify the setpoint with CANopen protocol, set <Setpoint_Input> to "CISP". The setpoint value is provided through object 0x6506. • To specify the setpoint with Modbus protocol, set <Setpoint_Input> to "MISP". The setpoint value is provided through register 0x40004. • To specify the setpoint with Analog input 1, set <Setpoint_Input> to "A1SP". In this case, the ±20 V input range is mapped to 0–24.5 mm for the rotor position. • To specify the setpoint with Analog input 2, set <Setpoint_Input> to "A2SP". In this case, the ±10 V input range is mapped to 0–24.5 mm for the rotor position. • For analog input case, the "AIEN" configuration must be enabled in advance. • Duration is specified in seconds, with lower and upper limits of 0.01 and 300 seconds, respectively. • To set the macro for infinite time, use "INFT" as the "Duration" input. • Only 3 digits after the floating point are considered by the firmware for the <Duration> value. • This macro terminates when the desired time has elapsed or an unwanted error occurs.

Notes

- To design a motion trajectory, a series of macro types can be used together. The following example illustrates how to create a motion profile using this concept.
  Example: SMCR,PERM,NUM1,TPRA,100,2,PHOL,4,TPRA,-50,1,PARK,-500
- The maximum number of macro types that can be stacked is limited to 20. Exceeding this limit triggers "0xFF18" error.
- It is highly recommended to finish the trajectory with the "Park" macro type.

5.6.4 Error Codes

Overview

Error Code	Description
0x0000	No Error
0x2001	ADC Offset Error
0x400A	Actuator Temperature Error
0x400B	Controller Board Temperature Error
0x5005	Actuator Initialization Error
0x7004	STO Error
0x7006	Encoder Initialization Error
0x7007	Home Error
0xFF03	No Macro Error
0xFF08	Mechanical Set Point Error
0xFF09	Electrical Set Point Error
0xFF0F	Busy Error
0xFF13	Illegal Command Error
0xFF14	Illegal Value Error
0xFF15	No Data Record Type Error
0xFF16	No value Error
0xFF17	No Data Error
0xFF18	Too Much Macro Error
0xFF19	Too Much Data Record Type Error
0xFF1A	Read Only Configuration Error

Error: 0x2001 (ADC Offset Error)

Description	This error indicates that there is a problem with the current feedback.
Possible Causes	None.
Solution	Restart the controller board. Contact customer service.

Error: 0x400A (Actuator Temperature Error)

Description	This error indicates that the rotor temperature is too high.
Possible Causes	Macro requires high control effort demands within a short time interval. Ambient high temperature. Short circuit of the coil.
Solution	Stop and power off the controller board and actuator for a while. Contact the customer service if the problem persists.

Error: 0x400B (Controller Board Temperature Error)

Description	This error indicates that the controller board temperature is too high.
Possible Causes	Macro requires high control effort demands within a short time interval. Ambient high temperature. Short circuit in the controller board or actuator rotor winding.
Solution	Stop and power off the controller board for a while. Contact the customer service if the problem persists.

Error: 0x5005 (Actuator Initialization Error)

Description	This error indicates that the controller board cannot recognize the actuator.
Possible Causes	The actuator is not properly connected to the controller board.
Solution	Check the cable. Restart the controller board or apply "ARIN" command. Contact customer service.

Error: 0x7004 (STO Error)

Description	This error indicates that the safe torque off (STO) protection is active, and at least one of the STO inputs is not in the correct state.
Possible Causes	The isolated power supply is connected to an unsuitable source. The STO inputs are not in the "HIGH" state. The "BSTO" configuration is set to 0 by mistake.
Solution	Check the current configuration of "BSTO" setting. Check that the isolated power supply voltage is within the correct range (5V to 24V). Check the voltage of both STO inputs; they should be the same as the isolated power supply voltage. Check the STO feedback voltage; it should be in the "HIGH" state when both inputs are applied. Contact customer service.

Error: 0x7006 (Encoder Initialization Error)

Description	This error indicates that the encoder is not in absolute mode.
Possible Causes	The actuator is not properly connected to the controller board. The rotor was not moved, either manually or by using the "EAIN" command. The rotor is damaged. The encoder is damaged.
Solution	Check the cable. Apply "EAIN" command. Restart the controller board. Contact customer service.

Error: 0x7007 (Home Command Error)

Description	This error indicates that the rotor was not at the absolute position 0 when motion was requested.
Possible Causes	The actuator is not properly connected to the controller board. The rotor cannot return to the homing position due to a barrier. The actuator is damaged. The encoder is damaged.
Solution	Check the cable. Restart the controller board. Contact customer service.

Error: 0xFF03 (No Macro Error)

Description	This error indicates that the current macro array was empty when the "RMCR" command was applied.
Possible Causes	The controller was restarted or powered on, and the "RMCR" command was applied before the "SMCR" or "LMCR" commands.
Solution	Use the "SMCR" command to set a new macro. Use the "LMCR" command to load the desired macro.

Error: 0xFF08 (Mechanical Set Point Error)

Description	This error indicates that applying the desired macro causes the rotor to exceed its maximum or minimum limits, which may result in a hard strike against the stop barriers.
Possible Causes	The desired value for the acceleration, velocity, or duration parameters in the macro settings was incorrect.
Solution	Modify the desired macro. Contact customer service.

Error: 0xFF09 (Electrical Set Point Error)

Description	This error indicates that applying the desired macro causes the current passing through the rotor winding to exceed its maximum or minimum limits, which could damage the winding or lead to a hard strike against the stop barriers.
Possible Causes	The desired value for the acceleration, velocity, or duration parameters in the macro settings was incorrect. Problem with the position control loop.
Solution	Modify the desired macro. Contact customer service.

Error: 0xFF0F (Busy Error)

Description	This error indicates that a new "RMCR" command was applied while the previous macro was still running.
Possible Causes	The "RMCR" command was applied mistakenly.
Solution	Apply the "RERR" command to reset the error or directly request another "RMCR" command to restart the current macro.

Error: 0xFF13 (Illegal Command Error)

Description	This error indicates that an illegal command was applied.
Possible Causes	Typing error.
Solution	Check the wording. Check the command notation.

Error: 0xFF14 (Illegal Value Error)

Description	This error indicates that an illegal value is considered for setting.
Possible Causes	Typing error. The limits have been exceeded.
Solution	Check the wording. Check the upper and lower limits.

Error: 0xFF15 (No Data Record Type Error)

Description	This error indicates that no data type is specified, and the get data command (DATA) has been applied.
Possible Causes	The controller was reset or powered on, and no data type was set for permanent configuration. An error occurred while setting the data type configuration.
Solution	Reset the data type configuration, then apply the "RMCR" command to record the data.

Error: 0xFF16 (No value Error)

Description	This error indicates that no value has been set for the applied command.
Possible Causes	Typing error.
Solution	Check the wording. Check the command notation.

Error: 0xFF17 (No Data Error)

Description	This error indicates that no data has been recorded, and the "DATA" command has been applied.
Possible Causes	Offline data recording configurations were not set up correctly. The "DATA" command was applied before any prior "RMCR".
Solution	Check the offline data recording configurations.

Error: 0xFF18 (Too Much Macro Error)

Description	This error indicates that the macro type number applied in the "SMCR" step, exceeds the maximum limit.
Possible Causes	More than 20 macro types were tried to be applied.
Solution	Modify the desired macro.

Error: 0xFF19 (Too Much Data Record Type Error)

Description	This error indicates that the number of applied data record types exceeds the maximum limit.
Possible Causes	More than 25 data record types were tried to be applied.
Solution	Modify the desired data record type.

Error: 0xFF1A (Read Only Configuration Error)

Description	This error indicates that the "SCON" command was applied to a read-only configuration.
Possible Causes	Typing error.
Solution	Check the access level for the desired configuration.

5.7. CANopen protocol

CANopen is a robust communication protocol based on the Controller Area Network (CAN) bus, developed by the CiA (CAN in Automation) organization. It provides a standardized framework for communication between devices, facilitating interoperability and flexibility in system design. CTRL1-48-5-G4 controller boards are equipped with this protocol and support various communication services, including Process Data Objects (PDOs) for real-time data transfer and Service Data Objects (SDOs) for configuration and parameter management.

5.7.1. Connecting the CAN bus

The CAN bus can be connected through either CN2 or CN3 (p. 148). The controller board can be stacked in a CANopen network as shown in the following figure.

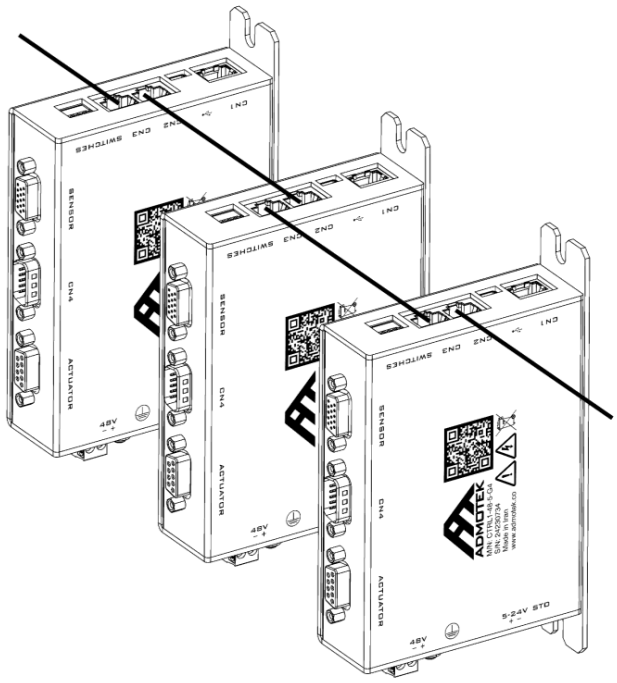

Fig. 3. Multi-Drop Connection Example.

The CAN bus must be terminated with two 120 Ω resistors at both ends of the bus line, as illustrated in the following figure.

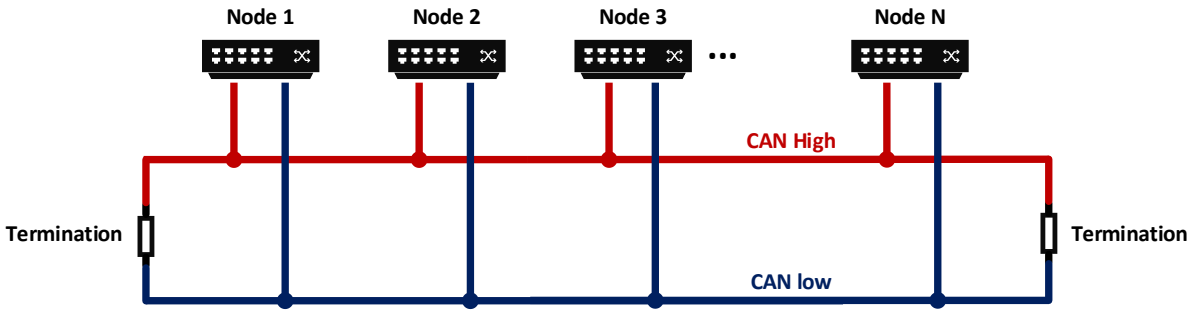

Fig. 4. Termination at the CAN bus lines.

For easy installation, the ADMOTEK CANopen controller board has built-in termination resistors, which can be activated, if the ADMOTEK drive is at the end of the bus line, and if there is no termination in the connector. The built-in termination resistor for the CAN bus can be activated by setting the second key of the configuration switch to the "ON" position.

5.7.2. CANopen Parameters

ADMOTEK CANopen drives have additional parameters such as unit ID and baud rate which can be configured with the ADMORS protocol. With these parameters, the CANopen behaviors can be defined (p. 27).

5.7.3. Object Structure

The ADMOTEK controller board uses objects (also known as registers) to perform functions and store operation results. The motion controller functionalities and parameters are accessed solely through these objects.

All objects are grouped into a dictionary and contain the following information:

Index / Subindex	Specifies the object number.
Name	Specifies the object name.
Object type	Specifies the object type.
Access	Specifies the type of object access.
Data type	Specifies the data type.
PDO mapping	Specifies if the object could be accessed via PDO.
Value range	Specifies the range of allowed values.
Default value	Specifies the default value of the object.

A 16-bit index is used to address all objects within the object dictionary. For simple variables, the index directly references the value of the variable. In case of records and arrays, the index addresses the whole data structure. To enable access to individual elements of data structures, an 8-bit subindex is defined.

Possible object types

VAR	Indicates the use of a basic type value (UINT8, INT16, STR, etc.).
ARRAY	Indicates a data set of the same basic type, such as UINT16. • In this case, subindex zero is always a UINT8 that indicates the length of the array.
RECORD	indicates a data set with different basic types. • In that case, subindex zero is always a UINT8 that indicates the number of items in the record.

Possible access types

Read Only	The object can only be consulted.
Write Only	The object can only be modified.
Read and Write	The object can be consulted and modified.

Possible data types

UINT8	1 byte size. Value range [0 to 255].
UINT16	2-byte size. Value range [0 to 65535]
UINT32	4-byte size. Value range [0 to 4294967295]
UINT64	8-byte size. Value range [0 to 18446744073709551616]
INT8	1 byte size. Value range [-128 to 127]
INT16	2-byte size. Range of values: [-32768 to 32767]
INT32	4-byte size. Range of values: [-2147483648 to 2147483647]
STR	Character sequence terminated with a null character. Variable size between 1 and 8 bytes.

Notes

Further information can be found at: http://www.can-cia.org/.

5.7.4. Object Dictionary

Overview

Name	Index
Device Type	0x1000
Error Registers	0x1001
Pre-defined Error Field	0x1003
COB-ID SYNC	0x1005
Communication Cycle Period	0x1006
Sync Window Length	0x1007
Manufacturer Device Name	0x1008
Manufacturer Hardware Version	0x1009
Software Version	0x100A
Guard Time	0x100C
Life Time Factor	0x100D
Store Parameters	0x1010
Restore Default Parameters	0x1010
COB-ID Emergency Message	0x1014
Producer Heartbeat Time	0x1017
Identity Object	0x1018
SDO Server Parameter	0x1200
RPDO1	0x1400
RPDO2	0x1401
RPDO3	0x1402
RPDO4	0x1403
RPDO1 Mapping Parameter	0x1600
RPDO2 Mapping Parameter	0x1601
RPDO3 Mapping Parameter	0x1602

RPDO4 Mapping Parameter	0x1603
TPDO1	0x1800
TPDO2	0x1801
TPDO3	0x1802
TPDO4	0x1803
TPDO1 Mapping Parameter	0x1A00
TPDO2 Mapping Parameter	0x1A01
TPDO3 Mapping Parameter	0x1A02
TPDO4 Mapping Parameter	0x1A03
Run Macro	0x6000
Macro Repeat	0x6001
Data	0x6003
Load Macro	0x6004
Controller Board Reset	0x6006
Encoder Auto Initialization	0x6007
Actuator Model Number	0x6008
Actuator Serial Number	0x6009
Actuator Reinitialize	0x600A
VBUS	0x600B
Actuator Temperature	0x600C
Controller Board Temperature	0x600D
Stop Macro	0x600E
Reset Error	0x600F
Macro Status	0x6011
Actuator Self-calibration	0x6012
Actuator Self-calibration Range	0x6013
Set Macro Number	0x6014

Macro Array	0x6015
Set Macro	0x6016
Error Codes and Error Behavior	0x603F
Actuator Encoder Resolution	0x608F
Drive Catalogue Web Address	0x6505
Macro Setpoint	0x6506

Object Dictionary: Device Type

Description	This object provides information about the device type.
Index	0x1000
Subindex	0x00
Data Type	UINT32
Access	Read Only
PDO mapping	No
Nonvolatile memory	Yes
Value range	-
Default value	0x00014854
Units	-
Notes	None

Object Dictionary: Error Registers

Description	This object provides error information.
Index	0x1001
Subindex	0x00
Data Type	UINT8

Access	Read Only
PDO mapping	No
Nonvolatile memory	No
Value range	-
Default value	-
Units	-
Notes	• The controller board maps internal errors into this object. • It is a part of an emergency object.

Object Dictionary: Pre-defined Error Field

Description	This object lists errors that occurred on the controller board and were signaled via the emergency object. It acts as an error history.				
Index	0x1003				
Subindex	0x01	0x02	0x03	0x04	0x05
Data Type	UINT32	UINT32	UINT32	UINT32	UINT32
Access	Read Only	Read Only	Read Only	Read Only	Read Only
PDO mapping	No	No	No	No	No
Nonvolatile memory	No	No	No	No	No
Value range	UINT32	UINT32	UINT32	UINT32	UINT32
Default value	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
Units	-	-	-	-	-

The object entry at subindex 0x00 contains the number of actual errors that are recorded in the array starting at subindex 0x01.
Every new error will be stored at subindex 0x01 and older errors will be moved to the next higher subindex.
Emergency objects are triggered by the occurrence of a CANopen device internal error situation and are transmitted from an emergency producer (normally a node) on the CANopen device.
An emergency object is sent only once per error event.
The data content of the emergency message uses the following structure:

Byte Number	1	2	3	4	5	6	7	8
	Emergency error codes (Object 0x603F)		Error registers (Object 0x1001)	Reserved (Zero values)				

Example of EMCY message:

COB-ID (hex)	Number of Bytes	Data field (hexadecimal)	Description
82	8	04 70 10 00 00 00 00 00	Node 2 sends an STO error (0x7004) emergency message

Notes

The following table lists the emergency error codes in hexadecimal format:

Error Code	Description
0x0000	No error
0x2001	ADC Offset Error
0x7002	Encoder Error
0xFF03	No Macro
0x7004	STO Error
0x5005	Actuator Initialization Error
0x7006	Encoder Initialization Error
0x7007	Home Error
0xFF08	Mechanical Set Point Error
0xFF09	Electrical Set Point Error
0x400A	Actuator Temperature Error
0x400B	Control Board Temperature Error
0x600C	Electrical Control Error
0x600D	Mechanical Control Error
0xFF0F	Busy Error
0xFF10	Macro Repeat Value Error

0xFF11	Macro Stopped
0xFF12	Self-Calibration Error
0x8110	CAN bus overrun
0x8130	Lifeguard or heartbeat error
0x8150	Transmit COBID collision
0x8210	PDO not processed due to length error
0x8220	PDO length exceeded

Further details are provided in section 5.6.4, Error Codes (p. 58).

Object Dictionary: COB-ID SYNC

Description	This object specifies the configured COB-ID of the synchronization object (SYNC) and indicates whether the controller board generates the SYNC.
Index	0x1005
Subindex	0x00
Data Type	UINT32
Access	Read and Write
PDO mapping	No
Nonvolatile memory	Yes
Value range	-
Default value	0x00000080
Units	-

Notes

- The format is as follows:

Bit number:	31	30	29	28	…	11	10	…	0
	x	generate	frame	0x00000			11-bit CAN-ID		
				29-bit CAN-ID					

Bit	Value	Description
x	x	Do not care
Generate	0	The CANopen device does not generate a SYNC message.
	1	The CANopen device generates a SYNC message.
Frame	0	11-bit CAN-ID valid (CAN base frame)
	1	29-bit CAN-ID valid (CAN extended frame)
29-bit CAN-ID	x	29-bit CAN-ID of the CAN extended frame
11-bit CAN-ID	x	11-bit CAN-ID of the CAN extended frame

Object Dictionary: Communication Cycle Period

Description	This object provides the communication cycle period.
Index	0x1006
Subindex	0x00
Data Type	UINT32
Access	Read and Write
PDO mapping	No
Nonvolatile memory	Yes
Value range	UINT32
Default value	0x00000000
Units	µs
Notes	• This period defines the SYNC interval. • If the value is set to 0x00000000 the transmission of SYNC messages is disabled.

Object Dictionary: Sync Window Length

Description	This object indicates the configured length of the time window for synchronous PDOs.
Index	0x1007
Subindex	0x00
Data Type	UINT32
Access	Read and Write
PDO mapping	No
Nonvolatile memory	Yes
Value range	UINT32
Default value	0x00000000
Units	µs
Notes	• If the synchronous window length expired: ➢ All synchronous TPDOs will be discarded and an EMCY message will be transmitted. ➢ All synchronous RPDOs will be discarded until the next SYNC message is received. ➢ Synchronous RPDO processing is resumed with the next SYNC message. • If the value is set to 0x00000000 the synchronous window is disabled.

Object Dictionary: Manufacturer Device Name

Description	This object provides the name of the device as specified by the manufacturer.
Index	0x1008
Subindex	0x00
Data Type	STR

Access	Read Only
PDO mapping	No
Nonvolatile memory	Yes
Value range	STR
Default value	ADMOTEK_CTRL1-48-5-G4
Units	-
Notes	None.

Object Dictionary: Manufacturer Hardware Version

Description	This object provides the manufacturer hardware version description.
Index	0x1009
Subindex	0x00
Data Type	STR
Access	Read Only
PDO mapping	No
Nonvolatile memory	Yes
Value range	STR
Default value	V1.0.0
Units	-
Notes	None.

Object Dictionary: Software Version

Description	This object provides the manufacturer software version description.
Index	0x100A
Subindex	0x00
Data Type	STR
Access	Read Only
PDO mapping	No
Nonvolatile memory	Yes
Value range	STR
Default value	V1.0.0
Units	-
Notes	None.

Object Dictionary: Guard Time

Description	The objects indicate the configured guard time.
Index	0x100C
Subindex	0x00
Data Type	UINT16
Access	Read and Write
PDO mapping	No
Nonvolatile memory	Yes
Value range	UINT16
Default value	0x0000
Units	ms
Notes	• The life time is calculated by multiplying the life time factor with the guard time in life guarding protocol.

Object Dictionary: Life Time Factor

Description	The objects indicate the configured life time factor.
Index	0x100D
Subindex	0x00
Data Type	UINT8
Access	Read and Write
PDO mapping	No
Nonvolatile memory	Yes
Value range	UINT8
Default value	0x00
Units	-
Notes	• The life time is calculated by multiplying the life time factor with the guard time in life guarding protocol.

Object Dictionary: Store Parameters

Description	This object manages saving parameters in non-volatile memory.		
Index	0x1010		
Subindex	0x01	0x02	0x03
Data Type	UINT32	UINT32	UINT32
Access	Read and Write	Read and Write	Read and Write
PDO mapping	No	No	No
Nonvolatile memory	No	No	No
Value range	UINT32	UINT32	UINT32
Default value	-	-	-
Units	-	-	-
Notes	• In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate subindex. The signature that must be written is "save".		

MSB			LSB
e	v	a	s
0x65	0x76	0x61	0x73

- Subindex 0x01 saves all parameters.
- Subindex 0x02 saves communication parameters.
- Subindex 0x03 saves application parameters.

Object Dictionary: Restore Default Parameters

Description	This object restores the default values of the communication profile, device profile, and application profile parameters.		
Index	0x1010		
Subindex	0x01	0x02	0x03
Data Type	UINT32	UINT32	UINT32
Access	Read and Write	Read and Write	Read and Write
PDO mapping	No	No	No
Nonvolatile memory	No	No	No
Value range	UINT32	UINT32	UINT32
Default value	-	-	-
Units	-	-	-

Notes

- Restoration of default parameters is executed only when a specific signature is written to the appropriate subindex, preventing any accidental overwrites. The signature that must be written is "load".

MSB			LSB
d	a	o	l
0x64	0x61	0x6F	0x6C

- Subindex 0x01 restores all parameters.
- Subindex 0x02 restores communication parameters.
- Subindex 0x03 restores application parameters.

Object Dictionary: COB-ID Emergency Message

Description	This object indicates the configured COB-ID for the emergency (EMCY) write service.
Index	0x1014
Subindex	0x00
Data Type	UINT32
Access	Read and Write
PDO mapping	No
Nonvolatile memory	Yes
Value range	UINT32
Default value	0x00000080 + Node ID
Units	-
Notes	• The format is as follows:

Bit number:	**31**	**30**	**29**	**28**	**...**	**11**	**10**	**...**	**0**
	Valid	Reserved always 0	frame	0x00000			11-bit CAN-ID		
				29-bit CAN-ID					

Bit	Value	Description
Valid	0	EMCY exists / is valid
	1	EMCY does not exist / is not valid
Frame	0	11-bit CAN-ID valid (CAN base frame)
	1	29-bit CAN-ID valid (CAN extended frame)
29-bit CAN-ID	x	29-bit CAN-ID of the CAN extended frame
11-bit CAN-ID	x	11-bit CAN-ID of the CAN extended frame

Object Dictionary: Producer Heartbeat Time

Description	This object indicates the configured cycle time of the heartbeat.
Index	0x1017
Subindex	0x00
Data Type	UINT16
Access	Read and Write
PDO mapping	No
Nonvolatile memory	Yes
Value range	UINT16
Default value	0x0000
Units	ms
Notes	• A value of 0x0000 disables the heartbeat.

Object Dictionary: Identity Object

Description	This object provides general identification information about the CANopen device.			
Index	0x1018			
Subindex	0x01	0x02	0x03	0x04
Name	Vendor ID	product code	revision number	serial number
Data Type	UINT32	UINT32	UINT32	UINT32
Access	Read Only	Read Only	Read Only	Read Only
PDO mapping	No	No	No	No

Nonvolatile memory	Yes	Yes	Yes	Yes
Value range	UINT32	UINT32	UINT32	UINT32
Default value	0x0000027D	0x00045841	0x00000000	-
Units	-	-	-	-
Notes	• The serial number is a unique identifier for each controller.			

Object Dictionary: SDO Server Parameter

Description	This object defines the SDO Parameters data type.	
Index	0x1200	
Subindex	0x01	0x02
Name	COB-ID Client -> Server	COB-ID Server-> Client
Data Type	UINT32	UINT32
Access	Read Only	Read Only
PDO mapping	No	No
Nonvolatile memory	No	No
Value range	UINT32	UINT32
Default value	0x00000600 + Node ID	0x00000580 + Node ID
Units	-	-
Notes	None.	

Object Dictionary: RPDO1

Description	This object contains the communication parameters for PDO1 that the CANopen device can receive.		
Index	0x1400		
Subindex	0x01	0x02	0x03
Name	COB-ID used	Transmission type	Inhibit time
Data Type	UINT32	UINT8	UINT16
Access	Read and Write	Read and Write	Read and Write
PDO mapping	No	No	No
Nonvolatile memory	Yes	Yes	Yes
Value range	UINT32	UINT8	UINT16
Default value	0x181 + Node ID	0xFE	0x0000
Units	-	-	100µs

Notes

- The format of COB-ID (subindex 0x01) is as follows:

Bit number:	**31**	**30**	**29**	**28**	**...**	**11**	**10**	**...**	**0**
	Valid	Reserved always 0	frame	0x00000			11-bit CAN-ID		
				29-bit CAN-ID					

Bit	Value	Description
Valid	0	PDO exists / is valid
	1	PDO does not exist / is not valid
Frame	0	11-bit CAN-ID valid (CAN base frame)
	1	29-bit CAN-ID valid (CAN extended frame)
29-bit CAN-ID	x	29-bit CAN-ID of the CAN extended frame
11-bit CAN-ID	x	11-bit CAN-ID of the CAN extended frame

	• Values from 0x00 to 0xF0 for subindex 0x02 indicates a synchronous reception (data will be update after the reception of the next SYNC). • 0xFE or 0xFF values for subindex 0x02 indicated asynchronous reception (the controller will update the value after PDO reception). • Subindex 0x03 is not used actually and reserved for future updates.

Object Dictionary: RPDO2

Description	This object contains the communication parameters for PDO2 that the CANopen device can receive.		
Index	0x1401		
Subindex	0x01	0x02	0x03
Name	COB-ID used	Transmission type	Inhibit time
Data Type	UINT32	UINT8	UINT16
Access	Read and Write	Read and Write	Read and Write
PDO mapping	No	No	No
Nonvolatile memory	Yes	Yes	Yes
Value range	UINT32	UINT8	UINT16
Default value	0x281 + Node ID	0xFE	0x0000
Units	-	-	100µs
Notes	• Similar to RPDO1.		

Object Dictionary: RPDO3

Description	This object contains the communication parameters for PDO3 that the CANopen device can receive.		
Index	0x1402		
Subindex	0x01	0x02	0x03
Name	COB-ID used	Transmission type	Inhibit time
Data Type	UINT32	UINT8	UINT16
Access	Read and Write	Read and Write	Read and Write
PDO mapping	No	No	No
Nonvolatile memory	Yes	Yes	Yes
Value range	UINT32	UINT8	UINT16
Default value	0x80000400 + Node ID	0xFE	0x0000
Units	-	-	100µs
Notes	• Similar to RPDO1.		

Object Dictionary: RPDO4

Description	This object contains the communication parameters for PDO4 that the CANopen device can receive.		
Index	0x1403		
Subindex	0x01	0x02	0x03
Name	COB-ID used	Transmission type	Inhibit time
Data Type	UINT32	UINT8	UINT16
Access	Read and Write	Read and Write	Read and Write
PDO mapping	No	No	No
Nonvolatile memory	Yes	Yes	Yes
Value range	UINT32	UINT8	UINT16
Default value	0x80000500 + Node ID	0xFE	0x0000
Units	-	-	100µs
Notes	• Similar to RPDO1.		

Object Dictionary: RPDO1 Mapping Parameter

Description	This object contains the mapping parameters for PDO1 that the CANopen device can receive.
Index	0x1600
Subindex	0x01
Name	1st application object
Data Type	UINT32
Access	Read and Write
PDO mapping	No
Nonvolatile memory	Yes
Value range	UINT32
Default value	0x60400010
Units	-
Notes	• Subindex zero indicates the number of objects mapped (up to 0x40). • The structure of the mapping for subindex 0x01 to 0xFF (application object) is as follows:

Bit number:

31	…	16	15	…	8	7	…	0
Index			Subindex			Length		

Object Dictionary: RPDO2 Mapping Parameter

Description	This object contains the mapping parameters for PDO2 that the CANopen device can receive.	
Index	0x1601	
Subindex	0x01	0x02
Name	1st application object	2nd application object
Data Type	UINT32	UINT32
Access	Read and Write	Read and Write
PDO mapping	No	No
Nonvolatile memory	Yes	Yes
Value range	UINT32	UINT32

Default value	0x60400010	0x60600008
Units	-	-
Notes	• Similar to the RPDO1 mapping parameter.	

Object Dictionary: RPDO3 Mapping Parameter

Description	This object contains the mapping parameters for PDO3 that the CANopen device can receive.	
Index	0x1602	
Subindex	0x01	0x02
Name	1st application object	2nd application object
Data Type	UINT32	UINT32
Access	Read and Write	Read and Write
PDO mapping	No	No
Nonvolatile memory	Yes	Yes
Value range	UINT32	UINT32
Default value	0x60400010	0x607A0020
Units	-	-
Notes	• Similar to the RPDO1 mapping parameter.	

Object Dictionary: RPDO4 Mapping Parameter

Description	This object contains the mapping parameters for PDO4 that the CANopen device can receive.	
Index	0x1603	
Subindex	0x01	0x02
Name	1st application object	2nd application object
Data Type	UINT32	UINT32

Access	Read and Write	Read and Write
PDO mapping	No	No
Nonvolatile memory	Yes	Yes
Value range	UINT32	UINT32
Default value	0x60400010	0x60FF0020
Units	-	-
Notes	• Similar to the RPDO1 mapping parameter.	

Object Dictionary: TPDO1

Description	This object contains the communication parameters for PDO1 that the CANopen device can transmit.			
Index	0x1800			
Subindex	0x01	0x02	0x03	0x05
Name	COB-ID used	Transmission type	Inhibit time	Event time
Data Type	UINT32	UINT8	UINT16	UINT16
Access	Read / Write	Read / Write	Read / Write	Read / Write
PDO mapping	No	No	No	No
Nonvolatile memory	Yes	Yes	Yes	Yes
Value range	UINT32	UINT8	UINT16	UINT16
Default value	0xC0000180 + Node ID	0xFE	1000	0x0000
Units	-	-	100µs	ms
Notes	• The format of COB-ID (subindex 0x01) is as follows:			

Bit number:	31	30	29	28	…	11	10	…	0
	Valid	Reserved always 0	frame	0x00000			11-bit CAN-ID		
				29-bit CAN-ID					

Bit	Value	Description
Valid	0	PDO exists / is valid
	1	PDO does not exist / is not valid
Frame	0	11-bit CAN-ID valid (CAN base frame)
	1	29-bit CAN-ID valid (CAN extended frame)
29-bit CAN-ID	x	29-bit CAN-ID of the CAN extended frame
11-bit CAN-ID	x	11-bit CAN-ID of the CAN extended frame

- The values of the transmission type (subindex 0x02) are as follows:

Value	Description
0x00	Synchronous (acyclic) – data will be sent only once after the next SYNC.
0x01	Synchronous (cyclic every SYNC) - data will be sent cyclically after receiving 1 SYNC.
0x02	Synchronous (cyclic every 2nd SYNC) - data will be sent cyclically after receiving 2 SYNC.
…	
0xF0	Synchronous (cyclic every 240th SYNC) - data will be sent cyclically after receiving 240 SYNC.
0xFC-0xFD	RTR-only – The PDO is sent after the reception of RTR.
0xFE-0xFF	Event-driven – the PDO is sent after a timeout of the event timer or when one of the mapped objects changes.

- The Inhibit time (subindex 0x03) is expressed as multiple of 100µs.
- The value of zero for the Inhibit time (subindex 0x03) will disable it.
- The time is the minimum interval for PDO transmission if transmission type is set to 0xFE or 0xFF.
- The Inhibit time value limits the transmission rate of the TPDOs, even if the event time is set to a smaller time or the mapped objects change faster than the inhibit time.
- The time is the maximum interval for PDO transmission if the transmission type is set to 0xFE or 0xFF.
- Event time is expressed as multiple of 1ms.
- If the event time is reached, the PDO will be transmitted automatically.
- The value of zero for the Event time (subindex 0x05) will disable the event-timer.

Object Dictionary: TPDO2

Description	This object contains the communication parameters for PDO2 that the CANopen device can transmit.			
Index	0x1801			
Subindex	0x01	0x02	0x03	0x05
Name	COB-ID used	Transmission type	Inhibit time	Event time
Data Type	UINT32	UINT8	UINT16	UINT16
Access	Read / Write	Read / Write	Read / Write	Read / Write
PDO mapping	No	No	No	No
Nonvolatile memory	Yes	Yes	Yes	Yes
Value range	UINT32	UINT8	UINT16	UINT16
Default value	0xC0000280 + Node ID	0xFE	1000	0x0000
Units	-	-	100µs	ms
Notes	• Similar to TPDO1.			

Object Dictionary: TPDO3

Description	This object contains the communication parameters for PDO3 that the CANopen device can transmit.			
Index	0x1802			
Subindex	0x01	0x02	0x03	0x05
Name	COB-ID used	Transmission type	Inhibit time	Event time
Data Type	UINT32	UINT8	UINT16	UINT16
Access	Read / Write	Read / Write	Read / Write	Read / Write
PDO mapping	No	No	No	No
Nonvolatile memory	Yes	Yes	Yes	Yes
Value range	UINT32	UINT8	UINT16	UINT16
Default value	0xC0000380 + Node ID	0xFE	1000	0x0000
Units	-	-	100µs	ms
Notes	• Similar to TPDO1.			

Object Dictionary: TPDO4

Description	This object contains the communication parameters for PDO4 that the CANopen device can transmit.			
Index	0x1803			
Subindex	0x01	0x02	0x03	0x05
Name	COB-ID used	Transmission type	Inhibit time	Event time
Data Type	UINT32	UINT8	UINT16	UINT16
Access	Read / Write	Read / Write	Read / Write	Read / Write
PDO mapping	No	No	No	No
Nonvolatile memory	Yes	Yes	Yes	Yes
Value range	UINT32	UINT8	UINT16	UINT16
Default value	0xC0000480 + Node ID	0xFE	1000	0x0000
Units	-	-	100µs	ms
Notes	• Similar to TPDO1.			

Object Dictionary: TPDO1 Mapping Parameter

Description	This object contains the mapping parameters for PDO1 that the CANopen device can transmit.
Index	0x1A00
Subindex	0x01
Name	1st application object
Data Type	UINT32
Access	Read and Write
PDO mapping	No
Nonvolatile memory	Yes
Value range	UINT32
Default value	0x60410010
Units	-

Notes	• Subindex zero indicates the number of objects mapped (up to 0x40). • The structure of the mapping for subindex 0x01 to 0xFF (application object) is as follows:

Bit number:

31	…	16	15	…	8	7	…	0
Index			Subindex			Length		

Object Dictionary: TPDO2 Mapping Parameter

Description	This object contains the mapping parameters for PDO2 that the CANopen device can transmit.	
Index	0x1A01	
Subindex	0x01	0x02
Name	1st application object	2nd application object
Data Type	UINT32	UINT32
Access	Read and Write	Read and Write
PDO mapping	No	No
Nonvolatile memory	Yes	Yes
Value range	UINT32	UINT32
Default value	0x60410010	0x60610008
Units	-	-
Notes	• Similar to the TPDO1 mapping parameter.	

Object Dictionary: TPDO3 Mapping Parameter

Description	This object contains the mapping parameters for PDO3 that the CANopen device can transmit.	
Index	0x1A02	
Subindex	0x01	0x02
Name	1st application object	2nd application object
Data Type	UINT32	UINT32
Access	Read and Write	Read and Write
PDO mapping	No	No
Nonvolatile memory	Yes	Yes
Value range	UINT32	UINT32
Default value	0x60410010	0x60640020
Units	-	-
Notes	• Similar to the TPDO1 mapping parameter.	

Object Dictionary: TPDO4 Mapping Parameter

Description	This object contains the mapping parameters for PDO4 that the CANopen device can transmit.	
Index	0x1A03	
Subindex	0x01	0x02
Name	1st application object	2nd application object
Data Type	UINT32	UINT32
Access	Read and Write	Read and Write
PDO mapping	No	No
Nonvolatile memory	Yes	Yes
Value range	UINT32	UINT32
Default value	0x60410010	0x606C0020
Units	-	-
Notes	• Similar to the TPDO1 mapping parameter.	

Object Dictionary: Run Macro

Description	This object starts the desired macro.
Index	0x6000
Subindex	0x00
Data Type	UINT8
Access	Write Only
PDO mapping	Yes
Nonvolatile memory	No
Value range	0 and 1.
Default value	0x00
Units	-
Notes	• The macro repeat value must be set using the Macro Repeat Object (index 0x6001) before starting the macro; otherwise, "0xFF10" error occurs. • The macro number value must be set using the Load Macro Object (index 0x6004) before starting the macro; otherwise, "0xFF03" error occurs.

Object Dictionary: Macro Repeat

Description	This object defines the macro repeat value.
Index	0x6001
Subindex	0x00
Data Type	UINT16
Access	Read and Write
PDO mapping	Yes
Nonvolatile memory	Yes
Value range	UINT16
Default value	0x0000
Units	-
Notes	• The "0xFF10" error occurs when the run macro is requested and the macro repeat object is set to zero.

Object Dictionary: Load Macro

Description	This object defines the macro number value.
Index	0x6004
Subindex	0x00
Data Type	UINT8
Access	Read and Write
PDO mapping	Yes
Nonvolatile memory	Yes
Value range	1-10
Default value	0x00
Units	-
Notes	• The "0xFF03" error occurs when the run macro is requested and the macro number object has not been set correctly. • The macro number object must be set to 10 to load the temporary macro. • To load a specific permanent macro, the corresponding macro number should be assigned to this object. • This object is similar to the "LMCR" command in the ADMORS protocol.

Object Dictionary: Data

Description	This object contains data obtained from the control cycle.
Index	0x6003
Object Type	Record
Access	Read Only
PDO mapping	Yes
Nonvolatile memory	No

Notes

- This object consists of the following data:

Name	Subindex	Data Type	Gain
Position	0x01	int32	1/1000.315
Current	0x02	int32	1/5241.6
Mechanical control effort	0x03	int32	1/5241600
Mechanical error	0x04	int32	1/1000315
Mechanical set point	0x05	int32	1/1000.315
Electrical control effort	0x06	int32	48/3400000
Electrical error	0x07	int32	1/5241600
Electrical set point	0x08	int32	1/5241.6
Actuator temperature	0x09	uint16	-
Controller temperature	0x0A	uint16	-
BUS voltage	0x0B	uint16	0.0158
Digital Input 1	0x0C	uint8	-
Digital Input 2	0x0D	uint8	-
Digital Input 3	0x0E	uint8	-
Digital Input 4	0x0F	uint8	-
Macro counter	0x10	uint8	-
ISR counter	0x11	uint32	-
Macro repeat counter	0x12	uint16	-
Force	0x13	int32	1/1000
Analog input 1	0x14	uint16	-
Analog input 2	0x15	uint16	-

- This data must be multiplied by the gain values specified in the table to determine the relevant information.
- Actuator temperature data is obtained using the following formula in degrees Celsius:

$$Actuator\ Temp = \frac{1}{1.1 \times 10^{-4} \times ln\left(-\frac{6.1 \times 10^{-4} \times ATMP}{6.1 \times 10^{-4} \times ATMP - 3.3}\right) + 3.4 \times 10^{-3}} - 270.0$$

- Controller board temperature data is obtained using the following formula in degrees Celsius:

$$Controller\ Temp = \frac{1}{1.3 \times 10^{-4} \times ln\left(-\frac{6.1 \times 10^{-4} \times CTMP}{6.1 \times 10^{-4} \times CTMP - 2.5}\right) + 3.4 \times 10^{-3}} - 270.0$$

- Analog input 1 and analog input 2 data (subindex 0x14 and 0x15) are valid only if "AIEN" configuration is enabled.
- Analog input 1 data (subindex 0x14) should be divided by "81.9" and then added to "-25" to obtain the differential input voltage in volts.
- Analog input 2 data (subindex 0x15) should be divided by "-163.8" and then added to "12.5" to obtain the input voltage in volts.
- The estimated external force data (subindex 0x13) is valid only if the rotor is fixed or moving at a constant velocity.
- Subindex zero indicates the recorded data value, which is fixed at 21.
- These values are updated at 25 kHz, which is the control cycle rate.

Object Dictionary: Controller Board Reset

Description	This object resets the controller board.
Index	0x6006
Subindex	0x00
Data Type	UINT8
Access	Write Only
PDO mapping	No
Nonvolatile memory	No
Value range	0-1
Default value	0x00
Units	-
Notes	• This object is similar to the "CBRS" command in the ADMORS protocol.

Object Dictionary: Encoder Auto Initialization

Description	This object starts the encoder auto-initialization process to find the absolute rotor position value.
Index	0x6007
Subindex	0x00
Data Type	UINT8
Access	Write Only
PDO mapping	No
Nonvolatile memory	No
Value range	0-1
Default value	0x00
Units	-
Notes	• The actuator must be initialized before setting this object. • The encoder initiation process can be performed by moving the rotor manually. • This object is similar to the "EAIN" command in the ADMORS protocol.

Object Dictionary: Actuator Model Number

Description	This object indicates the actuator model number.
Index	0x6008
Subindex	0x00
Data Type	STR
Access	Read Only
PDO mapping	No
Nonvolatile memory	No

Value range	STR
Default value	-
Units	-
Notes	None.

Object Dictionary: Actuator Serial Number

Description	This object indicates the actuator serial number.
Index	0x6009
Subindex	0x00
Data Type	UINT32
Access	Read Only
PDO mapping	No
Nonvolatile memory	No
Value range	UINT32
Default value	-
Units	-
Notes	• The serial number is a unique identifier for each actuator.

Object Dictionary: Actuator Reinitialize

Description	This object reinitializes the actuator and loads the permanent configuration stored in its flash memory.
Index	0x600A
Subindex	0x00
Data Type	UINT8
Access	Write Only
PDO mapping	No
Nonvolatile memory	No
Value range	0-1
Default value	0x00
Units	-
Notes	• After applying this object, the encoder must be reinitialized.

Object Dictionary: VBUS

Description	This object indicates the bus voltage of the controller board.
Index	0x600B
Subindex	0x00
Data Type	UINT16
Access	Read Only
PDO mapping	Yes
Nonvolatile memory	No
Value range	UINT16
Default value	-
Units	-
Notes	• This value should be multiplied by 0.0158 to get the bus voltage in volts.

Object Dictionary: Actuator Temperature

Description	This object indicates the actuator temperature.
Index	0x600C
Subindex	0x00
Data Type	UINT16
Access	Read Only
PDO mapping	Yes
Nonvolatile memory	No
Value range	UINT16
Default value	-
Units	-
Notes	• Actuator temperature data is obtained using the following formula in degrees Celsius: $Actuator\ Temp = \frac{1}{1.1 \times 10^{-4} \times ln\left(-\frac{6.1 \times 10^{-4} \times ATMP}{6.1 \times 10^{-4} \times ATMP - 3.3}\right) + 3.4 \times 10^{-3}} - 270.0$

Object Dictionary: Controller Board Temperature

Description	This object indicates the controller board temperature.
Index	0x600D
Subindex	0x00
Data Type	UINT16
Access	Read Only
PDO mapping	Yes
Nonvolatile memory	No

Value range	UINT16
Default value	-
Units	-
Notes	• Controller board temperature data is obtained using the following formula in degrees Celsius. $Controller\ Temp = \frac{1}{1.3 \times 10^{-4} \times ln\left(-\frac{6.1 \times 10^{-4} \times CTMP}{6.1 \times 10^{-4} \times CTMP - 2.5}\right) + 3.4 \times 10^{-3}} - 270.0$

Object Dictionary: Stop Macro

Description	This object terminates the active macro
Index	0x600E
Subindex	0x00
Data Type	UINT8
Access	Write Only
PDO mapping	Yes
Nonvolatile memory	No
Value range	UINT8
Default value	0x00
Units	-
Notes	• If the current macro defines a motion trajectory and the actuator is not in the homing position, using this object causes the rotor to strike the stator. • The emergency error "0xFF11" will be triggered when this object is used. • This object is similar to the "PMCR" command in the ADMORS protocol (p. 16).

Object Dictionary: Reset Error

Description	This object resets the errors.
Index	0x600F
Subindex	0x00
Data Type	UINT8
Access	Write Only
PDO mapping	Yes
Nonvolatile memory	No
Value range	UINT8
Default value	0x00
Units	-
Notes	• If the reset error command completes successfully, the red LED on the controller board will turn off. • Actuator-init error and Encoder-init errors cannot be cleared using this object. • This object is similar to the "RERR" command in the ADMORS protocol (p. 16).

Object Dictionary: Macro Status

Description	This object indicates the macro status.
Index	0x6011
Subindex	0x00
Data Type	UINT8
Access	Read Only
PDO mapping	Yes
Nonvolatile memory	No

Value range	UINT8
Default value	0x00
Units	-
Notes	• If the value of this object is reported as 0x00, the controller board is ready to start a macro. • If the value of this object is reported as 0x01, the controller board is busy with an unfinished macro. In this condition, a new run macro command causes the emergency error "0xFF0F". • A value of "0x02" for this object indicates an error occurred during the previous macro.

Object Dictionary: Actuator Self-calibration

Description	This object starts the actuator self-calibration process.
Index	0x6012
Subindex	0x00
Data Type	UINT8
Access	Write Only
PDO mapping	No
Nonvolatile memory	No
Value range	0-1
Default value	0x00
Units	-
Notes	• The actuator and encoder must be initialized before using this command. • The actuator self-calibration range (index 0x6013) must be configured before applying this object. • This object should be applied periodically, depending on the application requirements. • During the self-calibration process, the actuator must remain fixed, and the rotor should be free to move. • This object is similar to the " ASCA " command in the ADMORS protocol (p. 16).

Object Dictionary: Actuator Self-calibration range

Description	This object defines the actuator self-calibration range.
Index	0x6013
Subindex	0x00
Data Type	UINT16
Access	Read and Write
PDO mapping	No
Nonvolatile memory	Yes
Value range	0 to actuator stroke
Default value	0x00
Units	1/1000 mm
Notes	• This object must be configured before applying the actuator self-calibration object (index 0x6012). • The actuator calibration range is restricted to 0 to 24.5 mm and should be set based on the actuator working stroke. • This object is expressed in 1/1000 mm; so, 12000 should be set for a 12 mm moving range.

Object Dictionary: Set Macro Number

Description	This object specifies the macro array defined in object 0x6015 to be set as a temporary macro or saved as a permanent macro with its assigned number.
Index	0x6014
Subindex	0x00
Data Type	UINT8
Access	Read and Write
PDO mapping	No
Nonvolatile memory	No

Value range	1-10
Default value	0x00
Units	-
Notes	• This object must be set to 10 to define a temporary macro. • The corresponding macro number should be assigned to this object to specify a permanent macro.

Object Dictionary: Macro Array

Description	This object defines the macro array.
Index	0x6015
Object Type	Array
Data Type	INT32
Subindex	0x00 – 0x64
Access	Read and Write
PDO mapping	No
Nonvolatile memory	No
Value range	INT32
Default value	0x00
Units	-
Notes	• This object can be used to set or get a macro. • To set macro using CANopen protocol, this object should be filled with the macro types and their parameters starting from subindex 0x01. Additionally, the permanent or temporary type must be defined using the Set Macro Number object (index 0x6014), and then the Set Macro object (index 0x6016) should be applied. • To get the temporary or permanent macro, first, it should be loaded as the current macro using the Load Macro object (index 0x6004), and then the macro can be read from this object. • The macro types are defined by code numbers, which are listed as follows:

Macro type Code	Description
1	Time-based Voltage Sine
2	Time-based Current Step
3	Time-based Current Sine
4	Time-based Position Step
5	Time-based Position Hold
6	Time-based Position Sine
7	Reserved.
8	Time-based Position Ramp
9	Soft-Touch
10	Measurement Only
11	Time-based Current Hold
12	Reserved.
13	Reserved.
14	Reserved.
15	Reserved.
16	Park
17	Time-based Position Acceleration
18	Time-based Force Step
19	Reserved.
20	Position-based Position Acceleration
21	Analog Output
22	Digital Output
23	Soft-Release
24	Time-based Voltage Setpoint
25	Time-based Current Setpoint
26	Time-based Position Setpoint

- Each macro type is defined with specific parameters. Further details are provided in section 5.6.3, "Macros and Trajectory Design" (p. 46).
- The macro type parameters should be multiplied by 1000.
- To set the duration to infinite for a macro type that accepts this parameter, use 0 in place of the duration time.
- Example:
  Desired macro: TPRA,100,2,PHOL,4,PARK,-500.
  Macro array: 8,100000,2000,5,4000,16,-500000.
- In Analog Out macro, the data type should be specified using the following table:

Data type Code	Description
0	Data Off
1	Rotor position
2	Current through the rotor winding
3	Mechanical control effort
4	Mechanical error
5	Mechanical set point
6	Electrical control effort

7	Electrical error
8	Electrical set point
9	Actuator temperature
10	Controller board temperature
11	Controller board bus voltage
12 … 19	-
20	Estimated external force

- In Digital Out macro, 0 and 1 should be used to specify the pin state: 0 for LOW and 1 for logic HIGH.
- In Time-based Voltage Setpoint, Time-based Current Setpoint, and Time-based Position Setpoint, 1 should be set for CANopen input, 2 for Modbus input, 3 for Analog input 1, and 4 for Analog input 2.
- Setting macro using this method is not recommended, as the input values are not checked by the firmware. Users must pay close attention and consider the upper and lower limits specified for each parameter. The proper approach is to use ADMORS protocol.

Object Dictionary: Set Macro

Description	This object sets the desired macro.
Index	0x6016
Subindex	0x00
Data Type	UINT8
Access	Read and Write
PDO mapping	No
Nonvolatile memory	No
Value range	UINT8
Default value	0x00
Units	-
Notes	• The Macro Array (object 0x6015) and Set Macro Number (object 0x6014) must be configured before applying this object.

Object Dictionary: Error Codes and Error Behavior

Description	This object indicates the last error detected in the controller.
Index	0x603F
Subindex	0x00
Data Type	UINT16
Access	Read Only
PDO mapping	Yes
Nonvolatile memory	No
Value range	UINT16
Default value	0x0000
Units	-
Notes	• Error codes are reported in p. 72.

Object Dictionary: Actuator Encoder Resolution

Description	This object indicates the encoder resolution used in the actuator.
Index	0x608F
Subindex	0x00
Data Type	UINT32
Access	Read Only
PDO mapping	Yes
Nonvolatile memory	No
Value range	UINT32
Default value	-
Units	-
Notes	• This value is four times the product of the encoder interpolation gain and the encoder LPI values (p. 27). • The actuator should be initialized first before using this object.

Object Dictionary: Drive Catalogue Web Address

Description	This object indicates the assigned web address of the drive manufacturer.
Index	0x6505
Subindex	0x00
Data Type	STR
Access	Read Only
PDO mapping	No
Nonvolatile memory	Yes
Value range	STR
Default value	https://www.admotek.co
Units	-
Notes	None.

Object Dictionary: Macro Setpoint

Description	This object defines the setpoint value for the TVSP, TCSP, and TPSP macros when using CANopen input type.
Index	0x6506
Subindex	0x00
Data Type	INT16
Access	Read and Write
PDO mapping	Yes
Nonvolatile memory	No
Value range	INT16
Default value	0x00

Units	TVSP: 1/1000 V TCSP: 1/1000 A TPSP: 1/1000 mm
Notes	• This object is expressed in 1/1000 units. For example, 12000 should be set for a 12 mm position setpoint in the TPSP macro, or 30000 should be set for a 30 V open-loop input voltage.

Notes

- The default configuration sets the first RPDO to the Run Macro object (index 0x6000) and the Macro Repeat object (index 0x6001), respectively.
- The second RPDO is set to the Load Macro object (index 0x6004) by default.

5.8. Modbus protocol

Modbus is a communication protocol utilized in industrial automation and control systems, developed by Modicon. This protocol operates in a master-slave architecture, where a master device initiates requests and one or more slave devices respond. On CTRL1-48-5-G4 controller board, the Modbus protocol is implemented over the serial interface in RTU (Remote Terminal Unit) mode. This mode utilizes a binary-based structure that minimizes data errors during transmission and maximizes compatibility across diverse systems.

5.8.1. Connecting the Serial Lines

The ADMOTEK controller board supports the Modbus protocol via an RS485 or RS422 serial interface, accessible through the CN2 or CN3 connectors (p. 148). RS485 operates in a half-duplex mode, whereas RS422 provides full-duplex communication.

Communication Modes

Full-duplex implementation requires two signal pairs (four wires) and transceivers with separate lines for transmitting and receiving. This configuration enables simultaneous data transmission on one pair while receiving on the other. In the half-duplex mode, only one signal pair is used, allowing data transmission in both directions, but only in one direction at a time. The RS422 and RS485 modes are illustrated in Fig. 5 and Fig. 6.

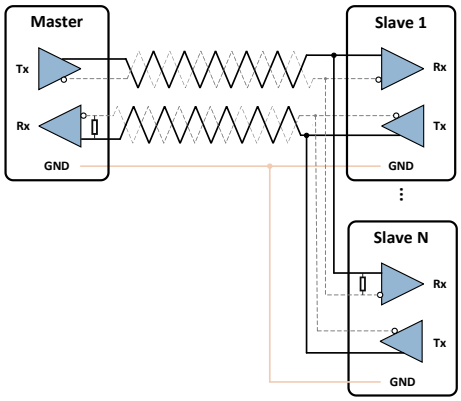

Fig. 5. RS422 mode

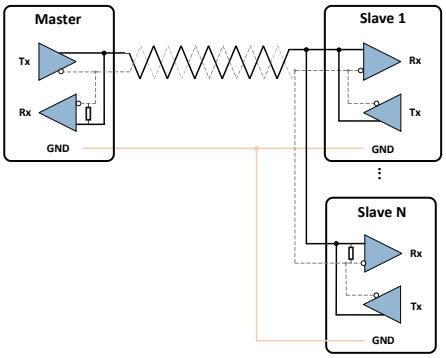

Fig. 5. RS485 mode

Notes

Half-duplex and full-duplex modes can be selected using the configuration switch on the controller board. When the first key of the configuration switch is in the "ON" position, RS485 is selected. In the "OFF" position, RS422 is activated.

Termination

The serial lines must be terminated at both ends. Proper termination requires matching the terminating resistors to the characteristic impedance of the transmission cable. Since the RS485 and RS422 standards recommend cables with a 120 Ω impedance, the cable trunk is typically terminated with 120 Ω resistors.

For easy installation, the ADMOTEK drive has built-in termination resistors, which can be activated using the configuration switches on the controller board:

- The third key of the configuration switch is used to enable or disable the 120 Ω termination resistor for the RS485 and transmitter lines of RS422.
- The fourth key of the configuration switch is used to enable or disable the 120 Ω termination resistor for the receive lines of RS422.

Topologies

Fig. 7 shows different methods for connecting the bus nodes.

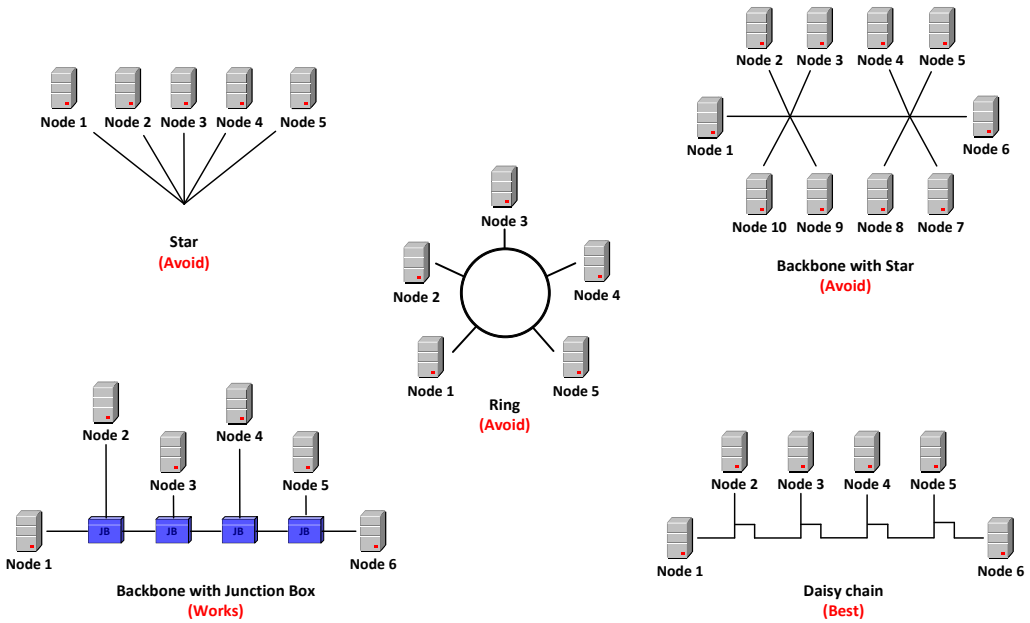

Fig. 7. Bus nodes connecting methods

It is highly recommended to connect the bus nodes in a daisy-chain configuration, as it minimizes the stub length. A stub is an unterminated piece of transmission line, whose electrical length should be less than 1/10 the fastest signal transition on the bus.

ADMOTEK controller board can be stacked in a daisy-chain configuration on the Modbus network, as shown in the following figure.

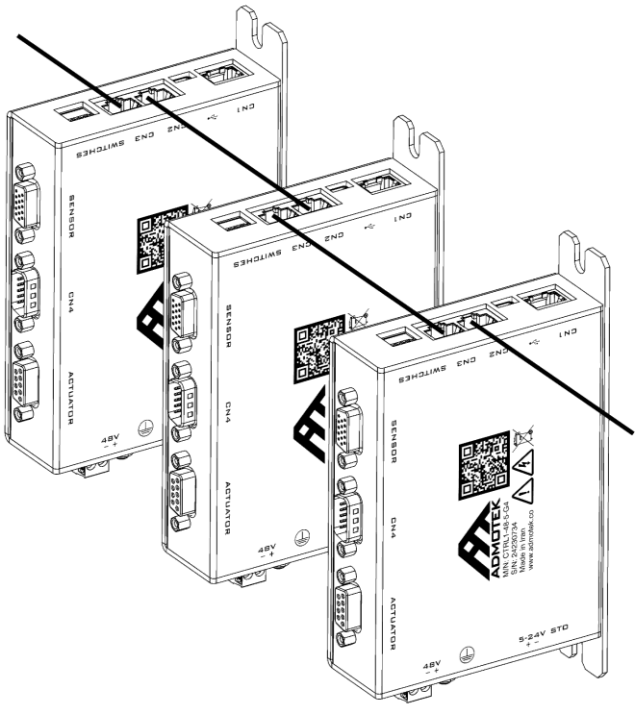

Fig. 8. Multi-Drop connection of the controller board

Bus Loading

In an RS485 network, the driver output depends on the current required to supply the load. Adding transceivers to the bus increases the total load current needed. To estimate the maximum number of bus loads possible, RS485 specifies a hypothetical term of a unit load (UL), which represents a load impedance of approximately 12 kΩ. Standard-compliant drivers must be capable of driving 32-unit loads. ADMOTEK controller board features a reduced unit load of 1/8 UL, allowing for the connection of up to 256 transceivers on the bus.

5.8.2. Modbus Parameters

ADMOTEK Modbus drives have additional parameters such as unit ID and baud rate which can be configured with ADMORS protocol. With these parameters, the Modbus behaviors can be defined (p. 27).

5.8.3. Modbus Register Types

Modbus devices use different types of registers to store and manage data. Each register type has a specific function and data format, allowing various data types, such as on/off states, sensor values, and configuration parameters, to be handled efficiently. These registers are listed as follows:

Resister Type	Description	Access	Address Range
Coil	Single-bit registers used to store on/off (binary) values for controlling digital outputs.	Read and Write	00001 – 09999
Discrete Input	Single-bit registers representing the status of digital inputs, such as sensors or push buttons.	Read Only	10001 – 19999
Input Register	16-bit registers used to capture analog or numerical data, such as sensor measurements.	Read Only	30001 – 39999
Holding Register	16-bit registers commonly used for configuration parameters, setpoints, or adjustable data.	Read and Write	40001 – 49999

5.8.4. Modbus Data Frame Structure

The Modbus message frame in RTU mode consists of several fields that contain data, addressing, and error-checking information. Each field plays a crucial role in ensuring accurate and reliable communication. The data frame of RTU modes is defined as follows:

Start	A silence interval for more than 10 milliseconds
Slave Address	Communication address
Function	Function code: 1-byte
Data (n-1) **…** **Data (0)**	Data content: n word= n * 2-byte, n≤10
CRC	Error checking: 1-byte
End 1	A silence interval for more than 10 milliseconds

Communication in RTU mode begins with a silent interval and concludes with another silent interval. Between these two intervals, the message includes the communication address, function code, data content, and error-checking CRC (Cyclic Redundancy Check), among other elements.

Address

The master initiates the communication by sending a byte with the address of the slave to which the message is destined. When sending the answer, the slave also initiates the telegram with its own address. The master can also send a message to the address 0 (zero), which means that the message is destined for all the slaves in the network (broadcast). In that case, no slave will answer to the master.

Function code

This field contains a single byte, where the master specifies the kind of service or function requested from the slave (such as reading or writing). According to the protocol, each function is used to access a specific type of data. Modbus uses function codes to specify the action requested by the master. Each function code identifies the operation type and directs the slave device to either read from or write to specific registers. Here are the supported function codes for this device:

Function Code	Description
0x01	Read one or more coils (binary outputs)
0x02	Read one or more discrete inputs (binary inputs)
0x03	Read one or more holding registers (16-bit values)
0x04	Read one or more input registers (16-bit values)
0x05	Write a single coil (binary output)
0x06	Write a single holding register (16-bit value)
0x0F	Write multiple coils (binary outputs)
0x10	Write multiple holding registers (16-bit)

Notes

- The functions for reading or writing multiple objects are useful when configuration or control changes need to be read or adjusted in a synchronized way.

Data

This is a variable-sized field, and its length varies based on the function code and the specific operation.

CRC

The Cyclic Redundancy Check (CRC) is a two-byte error-checking code appended to each message. It ensures data integrity by detecting transmission errors. The master and slave devices calculate the CRC independently, allowing them to validate message accuracy. If the calculated CRC values do not match, the frame is discarded, and no response is sent.

Exception Function Code

When the slave device encounters an issue while processing a request, it returns an error response containing an exception code. This mechanism enables the master to identify specific problems and address them appropriately. The most common exception codes supported by this device are listed as follows:

Exception Code	Name	Description
0x01	Illegal Function	The function code received is not supported by the slave.
0x02	Illegal Data Address	The address specified in the request is not accessible by the slave device.
0x03	Illegal Data Value	The data field contains an invalid value that the slave cannot process.

The function code in an error response is modified by adding 0x80 to the original function code. For example, if an error occurs with function code 0x03, the slave returns 0x83 to indicate an exception response.

Notes

- For further information, refer to the Modbus application protocol specification, available on the Modbus organization website at www.modbus.org.

5.8.5. Modbus Registers

Overview

Name	Index
Run Macro	00001
Stop Macro	00002
Actuator Reinitialization	00003
Encoder Auto Initialization	00004
Controller Board Reset	00005

Reset Error	00006
Actuator Self-calibration	00007
Set Macro	00008
Macro Status	30001
Rotor Position	30002
Winding Current	30004
Mechanical Control Effort	30006
Mechanical Error	30008
Mechanical Set Point	30010
Electrical Control Effort	30012
Electrical Error	30014
Electrical Set Point	30016
Actuator Temperature	30018
Controller Board Temperature	30019
Bus Voltage	30020
Digital Input 1 Status	30021
Digital Input 2 Status	30022
Digital Input 3 Status	30023
Digital Input 4 Status	30024
Macro Counter	30025
Control Cycle Counter	30026
Macro Repeat Counter	30028
Force	30029
Analog Input1	30031
Analog Input2	30032
Actuator Temperature Current Value	30034
Controller Board Temperature Current Value	30035

Bus Voltage Current Value	30036
Encoder Resolution	30037
Controller Board Serial Number	30039
Controller Board Version	30041 - 30046
Controller Board Model Number	30047 - 30059
Actuator Serial Number	30060
Actuator Model Number	30062 - 30073
Macro Repeat	40001
Load Macro	40002
Self-calibration Range	40003
Macro Setpoint	40004
Set Macro Number	40005
Macro Array	40006 - 40208

Register: Run Macro

Description	This register starts the macro.
Index (dec)	00001
Data Type	Boolean
Register Type	Coil
Value range	Boolean
Default value	0
Notes	• The macro repeat value must be set using the Macro Repeat register (40001) before starting the macro; otherwise, the Macro Status register (30001) changes to " Macro repeat value error". • The macro number value must be set using the Load Macro register (40002) before starting the macro; otherwise, the Macro Status register (30001) changes to "No macro".

Register: Stop Macro

Description	This register terminates the active macro.
Index (dec)	00002
Data Type	Boolean
Register Type	Coil
Value range	Boolean
Default value	0
Notes	• If the current macro defines a motion trajectory and the actuator is not in the homing position, using this object causes the rotor to strike the stator, which can damage it with repeated use. • When this register is set to 1, the Macro Status register (30001) changes to "Macro stopped". • This register is similar to the "PMCR" command in ADMORS protocol (p. 16).

Register: Actuator Reinitialization

Description	This register reinitializes the actuator and loads the permanent configuration stored in its flash memory.
Index (dec)	00003
Data Type	Boolean
Register Type	Coil
Value range	Boolean
Default value	0
Notes	• After applying this register, the encoder must be reinitialized. • This register is similar to the "ARIN" command in ADMORS protocol (p. 16).

Register: Encoder Auto Initialization

Description	This register starts the encoder auto-initialization process to find the absolute rotor position value.
Index (dec)	00004
Data Type	Boolean
Register Type	Coil
Value range	Boolean
Default value	0
Notes	• The actuator must be initialized before applying this register. • The encoder initiation process can be performed by moving the rotor manually. • This register is similar to the "EAIN" command in ADMORS protocol (p. 16).

Register: Controller Board Reset

Description	This register resets the controller board.
Index (dec)	00005
Data Type	Boolean
Register Type	Coil
Value range	Boolean
Default value	0
Notes	• This register is similar to the "CBRS" command in ADMORS protocol (p. 16).

Register: Reset Error

Description	This register resets the errors.
Index (dec)	00006
Data Type	Boolean
Register Type	Coil
Value range	Boolean
Default value	0
Notes	• If the reset error command completes successfully, the red LED on the controller board will turn off. • Actuator-init error and Encoder-init errors cannot be cleared using this register. • This register is similar to the "RERR" command in ADMORS protocol (p. 16).

Register: Actuator Self-calibration

Description	This register starts the actuator self-calibration process.
Index (dec)	00007
Data Type	Boolean
Register Type	Coil
Value range	Boolean
Default value	0
Notes	• The actuator and encoder must be initialized before using this command. • The actuator self-calibration range (register 40003) must be configured before applying this object. • This register should be applied periodically, depending on the application requirements. • During the self-calibration process, the actuator must remain fixed, and the rotor should be free to move. • This register is similar to the "ASCA" command in ADMORS protocol (p. 16).

Register: Set Macro

Description	This register sets the desired macro.
Index (dec)	00008
Data Type	Boolean
Register Type	Coil
Value range	Boolean
Default value	0
Notes	• The Macro Array (register 40006 – 40208) and Set Macro Number (register 40005) must be configured before applying this object.

Register: Macro Status

Description	This register indicates the macro status and errors.
Index (dec)	30001
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Default value	0

Notes

- The macro status and error codes are defied as follows:

Status Code	Description
0	No error
1	Busy (macro not finished yet)
2	ADC offset error
3	Encoder error
4	No macro
5	STO error
6	Actuator initialization error
7	Encoder initialization error
8	Home error
9	Mechanical set point error
10	Electrical set point error
11	Actuator temperature error
12	Controller board temperature error
13	Electrical control error (Not supported yet)
14	Mechanical control error (Not supported yet)
15	Hard fault error (Not supported yet)
16	Busy error
17	Macro repeat value error
18	Macro stopped
19	Actuator self-calibration error

- Further information on (p. 58).

Register: Rotor Position

Description	This register indicates the rotor position value obtained from the last control cycle.
Index (dec)	30002
Data Type	INT32
Register Type	Input register
Value range	INT32
Notes	• This value must be divided by the encoder resolution and then multiplied by "25.4" to obtain the rotor position in millimeters. • The encoder resolution is four times the product of the encoder interpolation gain and the encoder LPI specified in the actuator configuration (p. 27). • The encoder resolution can be obtained directly from register 30034.

Register: Winding Current

Description	This register indicates the current through the actuator winding obtained from the last control cycle.
Index (dec)	30004
Data Type	INT32
Register Type	Input register
Value range	INT32
Notes	• This value should be divided by "5241.6" to obtain the current through the rotor winding in amperes.

Register: Mechanical Control Effort

Description	This register indicates the mechanical control effort value obtained from the last control cycle.
Index (dec)	30006
Data Type	INT32
Register Type	Input register
Value range	INT32
Notes	• This value should be divided by "5,241,600" to obtain the mechanical control effort in amperes.

Register: Mechanical Error

Description	This register indicates the mechanical error value obtained from the last control cycle.
Index (dec)	30008
Data Type	INT32
Register Type	Input register
Value range	INT32
Notes	• This value must first be divided by "1000", and then divided by the encoder resolution and finally multiplied by "25.4" to obtain the mechanical error in millimeters. • The encoder resolution is four times the product of the encoder interpolation gain and the encoder LPI specified in the actuator configuration (p. 27). • The encoder resolution can be obtained directly from register 30034.

Register: Mechanical Set Point

Description	This register indicates the mechanical set point value obtained from the last control cycle.
Index (dec)	30010
Data Type	INT32
Register Type	Input register
Value range	INT32
Notes	• This value must be divided by the encoder resolution and then multiplied by "25.4" to determine the mechanical set point in millimeters. • The encoder resolution is four times the product of the encoder interpolation gain and the encoder LPI specified in the actuator configuration (p. 27). • The encoder resolution can be obtained directly from register 30034.

Register: Electrical Control Effort

Description	This register indicates the electrical control effort value obtained from the last control cycle.
Index (dec)	30012
Data Type	INT32
Register Type	Input register
Value range	INT32
Notes	• This value should first be divided by "3,400,000," and then multiplied by "48" to calculate the electrical control effort in volts.

Register: Electrical Error

Description	This register indicates the electrical error value obtained from the last control cycle.
Index (dec)	30014
Data Type	INT32
Register Type	Input register
Value range	INT32
Notes	• should be divided by "5,241,600" to determine the electrical error in amperes.

Register: Electrical Set Point

Description	This register indicates the electrical set point value obtained from the last control cycle.
Index (dec)	30016
Data Type	INT32
Register Type	Input register
Value range	INT32
Notes	• This value should be divided by "5241.6" to obtain the electrical set point in amperes.

Register: Actuator Temperature

Description	This register indicates the actuator temperature value obtained from the last control cycle.
Index (dec)	30018
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Notes	• Actuator temperature data is obtained using the following formula in degrees Celsius: $Actuator\ Temp = \frac{1}{1.1 \times 10^{-4} \times ln\left(-\frac{6.1 \times 10^{-4} \times ATMP}{6.1 \times 10^{-4} \times ATMP - 3.3}\right) + 3.4 \times 10^{-3}} - 270.0$

Register: Controller Board Temperature

Description	This register indicates the controller board temperature value obtained from the last control cycle.
Index (dec)	30019
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Notes	• Controller board temperature data is obtained using the following formula in degrees Celsius: $Controller\ Temp = \frac{1}{1.3 \times 10^{-4} \times ln\left(-\frac{6.1 \times 10^{-4} \times CTMP}{6.1 \times 10^{-4} \times CTMP - 2.5}\right) + 3.4 \times 10^{-3}} - 270.0$

Register: Bus Voltage

Description	This register indicates the bus voltage value of the controller board obtained from the last control cycle.
Index (dec)	30020
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Notes	• This value should be multiplied by "0.0158" to calculate the controller board bus voltage in volts.

Register: Digital Input 1 Status

Description	This register indicates the state of digital input 1 obtained from the last control cycle.
Index (dec)	30021
Data Type	UINT16
Register Type	Input register
Value range	0 - 1
Notes	• A value of 1 indicates that digital input 1 is in a "HIGH" state. • A value of 0 indicates that digital input 1 is in a "LOW" state.

Register: Digital Input 2 Status

Description	This register indicates the state of digital input 2 obtained from the last control cycle.
Index (dec)	30022
Data Type	UINT16
Register Type	Input register
Value range	0 - 1
Notes	• A value of 1 indicates that digital input 2 is in a "HIGH" state. • A value of 0 indicates that digital input 2 is in a "LOW" state.

Register: Digital Input 3 Status

Description	This register indicates the state of digital input 3 obtained from the last control cycle.
Index (dec)	30023
Data Type	UINT16
Register Type	Input register
Value range	0 - 1
Notes	• A value of 1 indicates that digital input 3 is in a "HIGH" state. • A value of 0 indicates that digital input 3 is in a "LOW" state.

Register: Digital Input 4 Status

Description	This register indicates the state of digital input 4 obtained from the last control cycle.
Index (dec)	30024
Data Type	UINT16
Register Type	Input register
Value range	0 - 1
Notes	• A value of 1 indicates that digital input 4 is in a "HIGH" state. • A value of 0 indicates that digital input 4 is in a "LOW" state.

Register: Macro Counter

Description	This register indicates the macro counter value obtained from the last control cycle.
Index (dec)	30025
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Notes	-

Register: Control Cycle Counter

Description	This register indicates the control loop counter value obtained from the last cycle.
Index (dec)	30026
Data Type	UINT32
Register Type	Input register
Value range	UINT32
Notes	-

Register: Macro Repeat Counter

Description	This register indicates the macro repeat counter value obtained from the last control cycle.
Index (dec)	30028
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Notes	-

Register: Force

Description	This register indicates the force value obtained from the last control cycle.
Index (dec)	30029
Data Type	INT32
Register Type	Input register
Value range	INT32
Notes	• This value should be divided by "1000" to determine the external force in newtons. • This value is valid only if the rotor is fixed or moving at a constant velocity.

Register: Analog Input 1

Description	This register indicates the analog input 1 value obtained from the last control cycle.
Index (dec)	30031
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Notes	• This value should be divided by "81.9" and then added to "-25" to obtain the differential input voltage in volts. • This value is valid only if "AIEN" configuration is enabled.

Register: Analog Input 2

Description	This register indicates the analog input 2 value obtained from the last control cycle.
Index (dec)	30032
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Notes	• This value should be divided by "-163.8" and then added to "12.5" to obtain the input voltage in volts. • This value is valid only if "AIEN" configuration is enabled.

Register: Actuator Temperature Current Value

Description	This register indicates the current value of the actuator temperature.
Index (dec)	30034
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Notes	• Actuator temperature data is obtained using the following formula in degrees Celsius: $Actuator\ Temp = \frac{1}{1.1 \times 10^{-4} \times ln\left(-\frac{6.1 \times 10^{-4} \times ATMP}{6.1 \times 10^{-4} \times ATMP - 3.3}\right) + 3.4 \times 10^{-3}} - 270.0$

Register: Controller Board Temperature Current Value

Description	This register indicates the current value of the controller board temperature.
Index (dec)	30035
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Notes	• Controller board temperature data is obtained using the following formula in degrees Celsius: $Controller\ Temp = \frac{1}{1.3 \times 10^{-4} \times ln\left(-\frac{6.1 \times 10^{-4} \times CTMP}{6.1 \times 10^{-4} \times CTMP - 2.5}\right) + 3.4 \times 10^{-3}} - 270.0$

Register: Bus Voltage Current Value

Description	This register indicates the current value of the controller board bus voltage.
Index (dec)	30036
Data Type	UINT16
Register Type	Input register
Value range	UINT16
Notes	• This value should be multiplied by "0.0158" to calculate the controller board bus voltage in volts.

Register: Encoder Resolution

Description	This register indicates the encoder resolution used in the actuator.
Index (dec)	30037
Data Type	UINT32
Register Type	Input register
Value range	UINT32
Notes	• This value is four times the product of the encoder interpolation gain and the encoder LPI values (p. 27). • The actuator should be initialized first before using this register.

Register: Controller Board Serial Number

Description	This register indicates the controller board serial number.
Index (dec)	30039
Data Type	UINT32
Register Type	Input register
Value range	UINT32
Notes	• The serial number is a unique identifier for each controller.

Register: Controller Board Version

Description	This register indicates the controller board version.
Index (dec)	30041 - 30046
Data Type	STR
Register Type	Input register
Value range	STR
Notes	• The controller board version length is 6 and is defined from register 30041 to register 30046.

Register: Controller Board Model Number

Description	This register indicates the controller board model number.
Index (dec)	30047 - 30059
Data Type	STR
Register Type	Input register
Value range	STR
Notes	• The controller board model number length is 13 and is defined from register 30047 to register 30059. • The default value is "CTRL1-48-5-G4".

Register: Actuator Serial Number

Description	This register indicates the actuator serial number.
Index (dec)	30060
Data Type	UINT32
Register Type	Input register
Value range	UINT32
Notes	• The serial number is a unique identifier for each actuator.

Register: Actuator Model Number

Description	This register indicates the actuator model number.
Index (dec)	30062 - 30073
Data Type	STR
Register Type	Input register
Value range	STR
Notes	• Controller version length is equal to 12 and is declared from 30062 to 30073.

Register: Macro Repeat

Description	This register defines the macro repeat value.
Index (dec)	40001
Data Type	UINT16
Register Type	Holding register
Value range	UINT16
Default value	0
Notes	• The Macro Status register (30001) changes to "Macro repeat value error" if the run macro command is requested while the Macro Repeat register is not set correctly.

Register: Load Macro

Description	This register loads the permanent macro number or switches to the temporary macro.
Index (dec)	40002
Data Type	UINT16
Register Type	Holding register
Value range	0-10
Default value	0
Notes	• The Macro Status register (30001) changes to " No macro" if the run macro command is requested while the Load Macro register is not set correctly. • This register must be set to 10 to load the temporary macro. • To load a specific permanent macro, the corresponding macro number should be assigned to this register. • This register is similar to the "LMCR" command in ADMORS protocol (p. 16).

Register: Self-calibration Range

Description	This register defines the actuator self-calibration range.
Index (dec)	40003
Data Type	UINT16
Register Type	Holding register
Value range	0 to actuator stroke
Default value	0
Notes	• This object must be configured before applying the actuator self-calibration register (register 00007). • The actuator calibration range is restricted to 0 to 24.5 mm and should be set based on the actuator working stroke. • This object is expressed in 1/1000 mm; so, 12000 should be set for a 12 mm moving range.

Register: Macro Setpoint

Description	This register defines the setpoint value for the TVSP, TCSP, and TPSP macros when using the Modbus input type.
Index (dec)	40004
Data Type	INT16
Register Type	Holding register
Value range	INT16
Default value	0
Notes	• This register is expressed in 1/1000 units. For example, 12000 should be set for a 12 mm position setpoint in the TPSP macro, or 30000 should be set for a 30 V open-loop input voltage.

Register: Set Macro Number

Description	This register specifies the macro array defined in registers 40006 - 40208 to be set as a temporary macro or saved as a permanent macro with its assigned number.
Index (dec)	40005
Data Type	UINT16
Register Type	Holding register
Value range	0-10
Default value	0
Notes	• This register must be set to 10 to define a temporary macro. • The corresponding macro number should be assigned to this register to specify a permanent macro.

Register: Macro Array

Description	This register defines the macro array.
Index (dec)	40006 - 40208
Data Type	INT32
Register Type	Holding register
Value range	INT32
Default value	0
Notes	• This register can be used to set or get macro. • To set macro using the Modbus protocol, this register should be filled with the macro types and their parameters. Additionally, the permanent or temporary type must be defined using the Set Macro Number register (register 40005), and then the Set Macro register (register 00008) should be applied. • To get the temporary or permanent macro, first, it should be loaded as the current macro using the Load Macro register (register 40002), and then the macro can be read from this register. • The macro types are defined by code numbers, which are listed as follows:

Macro type Code	Description
1	Time-based Voltage Sine
2	Time-based Current Step
3	Time-based Current Sine
4	Time-based Position Step
5	Time-based Position Hold
6	Time-based Position Sine
7	Reserved.
8	Time-based Position Ramp
9	Soft-Touch
10	Measurement Only
11	Time-based Current Hold
12	Reserved.
13	Reserved.
14	Reserved.
15	Reserved.
16	Park
17	Time-based Position Acceleration
18	Time-based Force Step
19	Reserved.
20	Position-based Position Acceleration
21	Analog Output
22	Digital Output
23	Soft-Release
24	Time-based Voltage Setpoint
25	Time-based Current Setpoint
26	Time-based Position Setpoint

- Each macro type is defined with specific parameters. Further details are provided in section 5.6.3, "Macros and Trajectory Design" (p. 46).
- The macro type parameters should be multiplied by 1000.
- To set the duration to infinite for a macro type that accepts this parameter, use 0 in place of the duration time.
- Example:
  Desired macro: TPRA,100,2,PHOL,4,PARK,-500.
  Macro array register: 8,100000,2000,5,4000,16,-500000.
- In Analog Out macro, the data type should be specified using the following table:

Data type Code	Description
0	Data Off
1	Rotor position
2	Current through the rotor winding

3	Mechanical control effort
4	Mechanical error
5	Mechanical set point
6	Electrical control effort
7	Electrical error
8	Electrical set point
9	Actuator temperature
10	Controller board temperature
11	Controller board bus voltage
12 … 19	-
20	Estimated external force

- In Digital Out macro, 0 and 1 should be used to specify the pin state: 0 for LOW and 1 for logic HIGH.
- In Time-based Voltage Setpoint, Time-based Current Setpoint, and Time-based Position Setpoint, 1 should be set for CANopen input, 2 for Modbus input, 3 for Analog input 1, and 4 for Analog input 2.
- Setting macro using this register is not recommended, as the input values are not checked by the firmware. Users must pay close attention and consider the upper and lower limits specified for each parameter. The proper approach is to use ADMORS protocol.

6. Maintenance

The CTRL1-48-5-G4 is maintenance-free.

7. Cleaning

- Prior to cleaning, ensure that the CTRL1-48-5-G4 has been disconnected from the power supplies, and other components (e.g., computers, I/O boards).
- If needed, use a cloth slightly moistened with a mild cleanser or disinfectant to clean the surfaces of the device.

NOTICE

Short circuits or flashovers!

The CTRL1-48-5-G4 contains electrostatic-sensitive devices that can be damaged by short-circuiting or flashovers when cleaning fluids penetrate the housing.

- Before cleaning, disconnect the controller from the power source by removing the main plug.
- Prevent cleaning fluid from penetrating the housing.

8. Troubleshooting

Actuator Does Not Move

Possible Causes	Solution
Cable not connected correctly	Check the cable connections.
Unsuitable cable used	• If unsuitable cables are used, interferences can occur in the signal transmission between the actuator and the controller board. • Only use original ADMOTEK parts to connect the actuator to the controller board.
Controller board is defective	• Send the "GERR" command and check the error code. • If the controller board is locked, wait for at least 7 seconds and reopen the serial interface. • Contact customer service.
Actuator is defective	• Apply "ARIN" command and check the response. • If available, replace the defective actuator with a different actuator of the same type and test the new combination.

Incorrect configuration	• Check the parameter settings of the controller using the "GCON" command. • Check the STO inputs and "BSTO" configuration.
Incorrect command or incorrect syntax	Make sure that the appropriate commands are used (p. 16).
Motion commands are ignored	Send the "GERR" command and check the error code.
Macro array is empty or incorrect	Send "GMCR,CURR" command and check the response.

Actuator Does Not initialize

Possible Causes	**Solution**
Cable not connected correctly	Check the cable connections**.**
Unsuitable cable used	Only use original ADMOTEK parts to connect the actuator to the controller board.
Actuator is defective	• Apply "ARIN" command and check the response. • If available, replace the defective actuator with a different actuator of the same type and test the new combination.
Controller board is defective	• Send the "GERR" command and check the error code. • If the controller board is locked, wait for at least 7 seconds and reopen the serial interface. • Contact customer service.

Communication with the Controller Does Not Work

Possible Causes	**Solution**
The wrong communication cable is used or it is defective	If necessary, check whether the cable works on a fault-free system.
Baud rate not configured correctly	The baud rate should be set to 115200 bits/s for successful communication using RS232 protocol.
Wrong COM port is selected	Check the COM port.

Another program is accessing the interface	Close the other program.
Loss of communication due to excessive utilization of the controller board processor	When the communication is faulty or has been terminated, wait for at least 7 seconds before reopening the serial interface (If the controller board processor is locked for 7 seconds, the microprocessor will reboot).

If the problem with your system is not listed in the table above or cannot be solved as described, contact our customer service.

9. Customer Service

If you have questions concerning your system, have the following information ready:

- Product codes and serial numbers of all products in the system.
- Firmware version of the controller (if available).
- The version of the driver or the software (if available).
- Operating system on the PC (if available).

If possible, take photographs or record videos of your system that can be sent to our customer service.

10. Technical Data

10.1. Specifications

	CTRL1-48-5-G4
Motion and control	
Controller type	Current, position, velocity.
Control cycle time	40 µs
Encoder input	Differential quadrature interface
Electrical properties	
Max. output voltage	48 V
Max. output current	±5 A
Interfaces and operation	
Communication interfaces	GPIO, RS485/422, RS232, CAN, USB
Actuator connection	D-sub 9 (f)
Sensor connection	D-sub 15 (f)
I/O connection	D-sub 15 (m)
I/O port	1x Differential analog input (±10V) 1x Sniggle-ended analog input (±10V) 2x Analog output (±10V) 4x Isolated digital input (5V-24V) 2x Fast isolated digital output (5V-24V) 3x High current (400 mA) isolated digital output (5V-24V) 2x STO input 1x STO feedback
Protection and monitoring	Reverse polarity protection ESD protection Overvoltage & undervoltage protection Overcurrent protection Controller board temperature monitoring Actuator temperature monitoring Short-circuit protection on actuator connector Bus voltage monitoring Safe torque off (STO)

Supported functions	Permanent and temporary configurations Permanent and temporary macros and trajectories Measurement and data recording Online and offline acquisition Protection and error control External force estimation Soft-touch
Miscellaneous	
Operating voltage	48 VDC for the power circuit 5-24 VDC for STO and isolated digital I/O
Max. current consumption	6.7 A from the 48 VDC 0.25 A from the 5-24 VDC
Operating temperature range	5 – 50℃
Mass	310 grams
Dimensions	166.8 mm x 97 mm x 33 mm
Body material	Aluminum

10.2. Maximum Ratings

Operating Voltage	Operating Frequency	Peak Current	Continuous Current
48 V	⎓	6.2 A	5 A

10.3. Dimensions

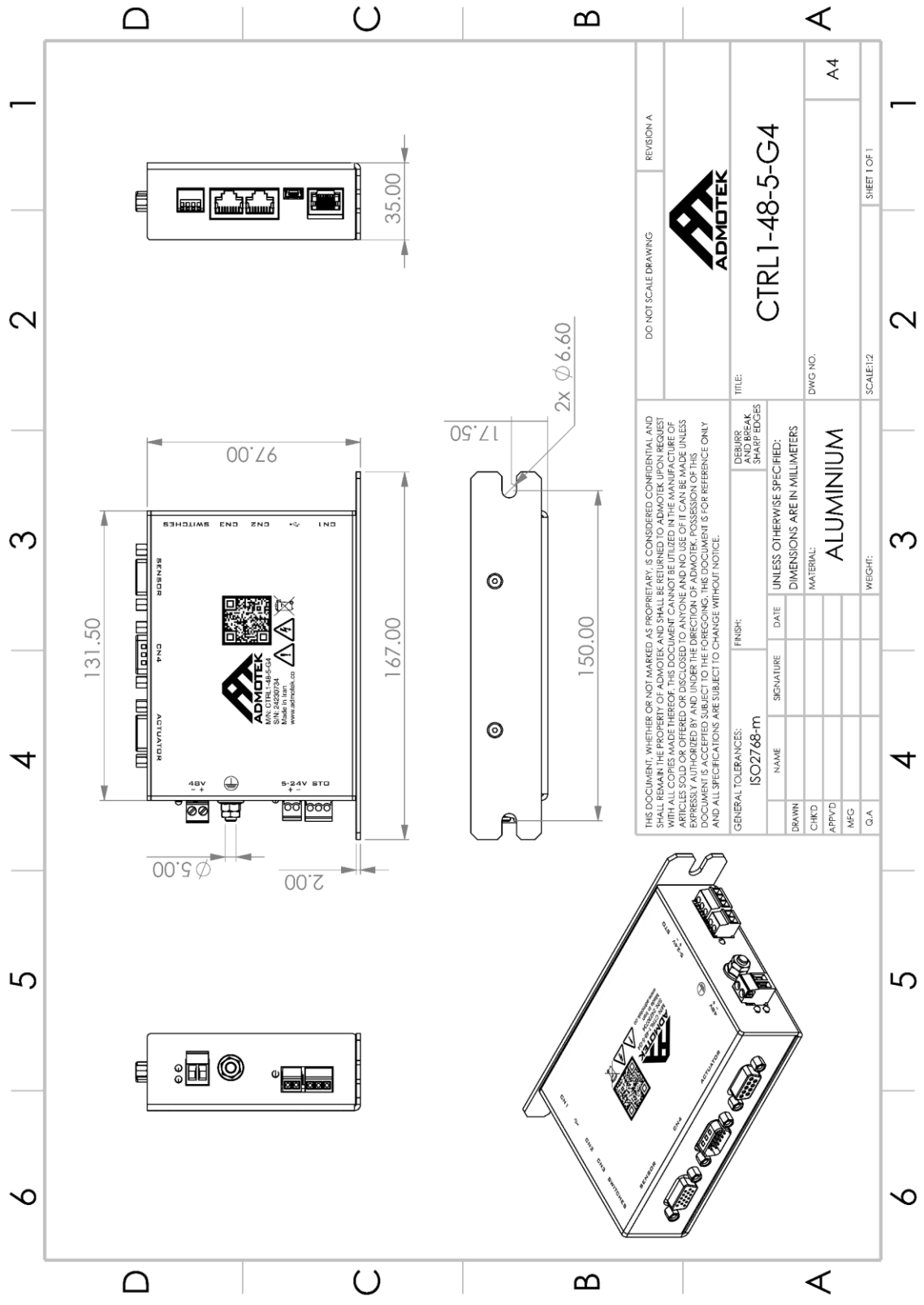

Fig. 9. CTRL1-48-5-G4 Dimensions.

10.4. Pin Assignment

Actuator (D-sub connector, 9-pin, female)

Pin	Description
1	Actuator P1
2	Actuator P1
3	NC
4	Actuator N1
5	Actuator N1

Pin	Description
6	Actuator P1
7	Actuator P1
8	Actuator N1
9	Actuator N1

Sensor (D-sub connector, 15-pin, high density, female)

Pin	Description
1	SPI CS–
2	SPI SCK–
3	SPI SCK+
4	SPI MOSI+
5	SPI MOSI–
6	Actuator Temperature
7	SPI CS+
8	Encoder A+

Pin	Description
9	Encoder B–
10	Encoder Z+ / SPI MISO+
11	GND
12	Encoder A–
13	Encoder B+
14	Encoder Z– / SPI MISO–
15	+5 VDC

CN4 (D-sub connector, 15-pin, male)

Pin	Description
1	Analog Input 1+
2	Analog Input 1–
3	Analog Output 1
4	Analog Output 2
5	Analog Input 2
6	Digital Input 4
7	Digital Input 1
8	Digital Input 2

Pin	Description
9	Digital Input 3
10	GND
11	Digital Output 2
12	Digital Output 1
13	Digital Output 5
14	Digital Output 4
15	Digital Output 3

48V (Terminal block header, 2-pin, 5.08 pitch)

Pin	Description
1	Power Supply Return
2	Power Supply Positive

5-24V (Terminal block header, 2-pin, 3.81 pitch)

Pin	Description
1	Power Supply Positive
2	Power Supply Return

STO (Terminal block header, 3-pin, 3.81 pitch)

Pin	Description
1	STO Input 1
2	STO Input 2
3	STO Feedback

CN1 (RJ45)

Pin	Description
1	RS232 RTS
2	NC
3	RS232 RX
4	GND

Pin	Description
5	GND
6	RS232 TX
7	NC
8	RS232 CTS

CN2, CN3 (RJ45 - Internally connected)

Pin	Description
1	CAN HIGH
2	CAN LOW
3	GND
4	RS422 TX+ RS485 A

Pin	Description
5	RS422 TX– RS485 B
6	GND
7	RS422 RX+
8	RS422 RX–

Configuration Switches (Dip switch)

Key number	Description
1	Half-Duplex and Full-Duplex Mode Selection for CN2 and CN3
2	CAN Bus Termination
3	RS485 Termination RS422-TX Termination
4	RS422 RX Termination

This page intentionally left blank


ADMOTEK