User Manual

MS251E, valid for C-891.130300

BRO, 2020-06-30

C-891 PIMag® Motion Controller

for magnetic direct drives, 1 axis, 24/48 V, 5 A

Physik Instrumente (PI) GmbH & Co. KG, Auf der Roemerstrasse 1, 76228 Karlsruhe, Germany
Phone +49 721 4846-0, Fax +49 721 4846-1019, Email info@pi.ws, www.pi.ws

Contents

The following company names and brands are registered trademarks of Physik Instrumente (PI) GmbH & Co. KG:
PI®, NanoCube®, PICMA®, PILine®, NEXLINE®, PiezoWalk®, NEXACT®, Picoactuator®, PInano®, PIMag®, Q-Motion®

Notes on brand names and third-party trademarks:
Microsoft® and Windows® are registered trademarks or trademarks of Microsoft Corporation in the USA and/or other countries.
LabVIEW, National Instruments and NI trademarks of National Instruments. Neither the driver software nor the software programs offered by PI or other goods and services are connected to or sponsored by National Instruments.
Python® is a registered trademark of Python Software Foundation.
BiSS is a registered trademark of iC-Haus GmbH.
The following designations are protected company names, trademarks or registered trademarks of other owners:
Linux, MATLAB, MathWorks

The patents owned by PI can be found in our patent list.

The software products provided by PI are subject to the General Software License Terms of Physik Instrumente (PI) GmbH & Co. KG and may contain and/or use third-party software components. Further information can be found in the General Software License Terms and in the Third-Party Software Notes on our website.

© 2019-2020 Physik Instrumente (PI) GmbH & Co. KG, Karlsruhe, Germany. The text, photographs, and drawings in this manual are protected by copyright. With regard thereto, Physik Instrumente (PI) GmbH & Co. KG retains all the rights. The use of any text, images and drawings is permitted only in part and only when indicating the source.

Original instructions
First printing: 2020-06-30
Document number: MS251E, BRo

Subject to change without notice. This manual is superseded by any new release. The latest respective release is available for download on our website (http://www.pi.ws).

User Manual

MS251E, valid for C-891.130300

BRo, 2019-02-08

About this Document

This user manual contains information on the intended use of the C-891.130300 PIMag® motor controller, hereinafter referred to as „C-891“.

It assumes that the reader has a fundamental understanding of basic servo systems as well as motion control concepts and applicable safety procedures.

The latest versions of user manuals and user manuals are available for download on our website (www.pi.ws).

Symbols and Typographic Conventions

The following symbols and markings are used in this user manual:

Symbol	Meaning
	Notice If not avoided, the dangerous situation will result in damage to the equipment.
1. 2.	Action consisting of several steps whose sequential order must be observed
➢	Action consisting of one or several steps whose sequential order is irrelevant
▪	List item
p. 5	Cross-reference to page 5
`SVO?`	Command line or command from PI's General Command Set (GCS) (example: command to get the servo mode)
RS-232	Operating element labeling on the product (example: socket of the RS-232 interface)
Device S/N	Parameter name (example: parameter where the serial number is stored)
Start > Settings	Menu path in the PC software (example: to open the menu, the Start and Settings buttons must be clicked in succession)
5	Value that must be entered or selected via the PC software

Physik Instrumente (PI) GmbH & Co. KG, Auf der Roemerstrasse 1, 76228 Karlsruhe, Germany
Phone +49 721 4846-0, Fax +49 721 4846-1019, Email info@pi.ws, www.pi.ws

Other Applicable Documents

For the following functionality of the C-891, see the MS205E User Manual of the C-863.11 Mercury DC Motor Controller:

- Data recorder
- Controller macros
- Variables

For details regarding the PC software, see the following documents:

Description	Document
GCS Data	SM146E Software Manual
PI GCS 2 driver set for use with NI LabVIEW software	SM158E Software Manual
Merge Tool for use with driver sets for NI LabVIEW software	SM154E Software Manual
PI GCS 2 DLL	SM151E Software Manual
PIPython	SM157E User Manual
PI MATLAB Driver GCS 2.0	SM155E Software Manual
PIMikroMove	SM148E Software Manual
PIStages3Editor	SM156E Software Manual
PI Update Finder	A000T0028 User Manual

Downloading Manuals

INFORMATION

If a manual is missing or problems occur with downloading:

➢ Contact our customer service department (p. 121).

Downloading manuals

1. Open the website www.pi.ws.
2. Search the website for the product number (“C-891”).
3. Click the corresponding product to open the product detail page.
4. Click ***Downloads***.

   The manuals are shown under ***Documentation***.
5. Click the desired manual and fill out the inquiry form.

   The download link will then be sent to the email address entered.

Safety

Intended Use

The C-891 is a laboratory device according to DIN EN 61010-1. It is intended to be used in interior spaces and in an environment which is free of dirt, oil and lubricants.

According to its design, the C-891 is intended for operating stages with PIMag® magnetic direct drive (hereinafter referred to as „3-phase motor“) or with stepper motor.

The C-891 is intended for closed-loop operation. For closed-loop operation, position sensor signals must be provided. Furthermore, the C-891 can read the reference switch and limit switch signals of the connected stage and process them further.

The C-891 may only be used in compliance with the technical specifications and instructions in this user manual. The user is responsible for process validation.

General Safety Instructions

The C-891 is built according to state-of-the-art technology and recognized safety standards. Improper use can result in personal injury and/or damage to the C-891.

- Only use the C-891 for its intended purpose, and only use it if it is in a good working order.
- Read the user documentation (user manuals, Technical Notes).
- Immediately eliminate any faults and malfunctions that are likely to affect safety.

The operator is responsible for the correct installation and operation of the C-891.

- Install the C-891 near the power source so that the power plug can be quickly and easily disconnected from the mains.
- Use the supplied components to connect the C-891 to the power source.
- If one of the supplied components for connecting to the power source has to be replaced, use a sufficiently dimensioned component.
- Only use cables and connections that meet local safety regulations.

Organizational Measures

User documentation (user manual, Technical Notes):

- Always keep this user documentation available by the C-891.
- The latest versions of the user documentation are available from PI.
- Add all information given by the manufacturer to the user documentation, for example supplements or Technical Notes.
- If you give the C-891 to other users, also turn over the user documentation as well as other relevant information provided by the manufacturer.
- Only use the device on the basis of the complete user documentation. Missing information due to an incomplete user documentation can result in property damage.
- Only install and operate the C-891 after having read and understood this user manual.

Personnel Qualification

The C-891 may only be installed, started up, operated, maintained and cleaned by authorized and qualified staff.

Product Description

Product View

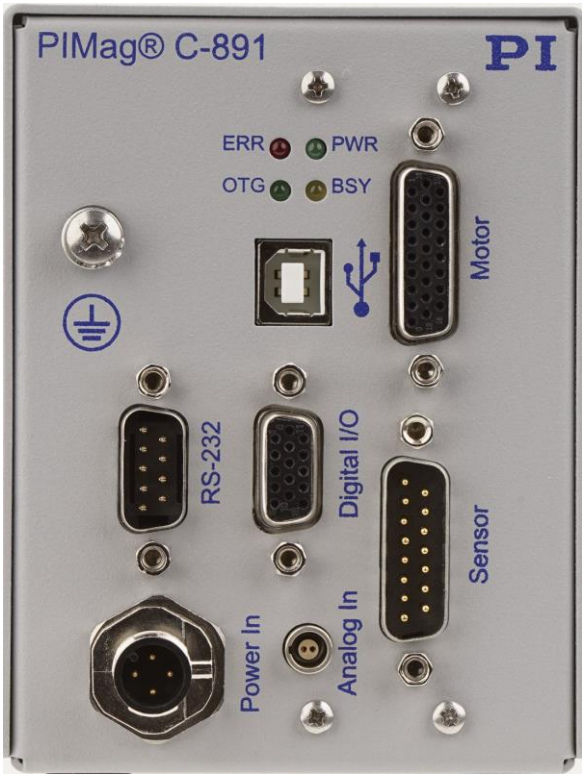


Figure 1: Front panel of C-891 PIMag® motor controller

Labeling	Type	Function
⏚	M4 hole with fastening material for protective earth conductor	Protective earth connection A protective earth conductor can be connected to the C-891 via the M4 hole and the fastening material. Note that the C-891 is not grounded via the power supply connector.
RS-232	Sub-D 9 (m) (p. 120)	Serial connection to PC Baud rate of the C-891: 115200
Power In	M12 panel plug (m), 4-pin (p. 120)	Connection for the supply voltage. The supply voltage can be 24 or 48 V DC. The BUS Voltage Setting parameter (0x3056) must be set accordingly. Default setting: 24 V DC. For details, see “Parameter Descriptions” (p. 99).

Labeling	Type	Function
ERR	LED red/off	Error indicator: ▪ Continuously lit: Error (error code ≠ 0) ▪ Continuously off: No error (error code = 0) The error code can be queried with the ERR? command. The query resets the error code to zero and the LED is switched off.
		During firmware update: ▪ The LED is lit while the current firmware is deleted on the C-891, and is switched off when the new firmware is written to the C-891. ▪ **ERR** and **PWR** LED flashing alternately: No or wrong firmware has been written to the C-891.
PWR	LED green/off	Power-on and ready indicator: ▪ Continuously lit: C-891 is ready for normal operation ▪ Continuously off: C-891 is not connected to the supply voltage
		During firmware update: ▪ Flashing with 2 Hz: C-891 is in firmware update mode ▪ Flashing with 1 Hz: New firmware has been successfully written to the C-891 ▪ **ERR** and **PWR** LED flashing alternately: No or wrong firmware has been written to the C-891.
OTG	LED green/off	On-target indicator for the axis: ▪ Continuously lit: The axis has reached the target. ▪ Continuously off: The axis has not reached the target. The on-target state of the axis can be queried with the ONT? command.
		During firmware update: The LED is lit when the new firmware has been successfully saved on the C-891.
BSY	LED yellow/off	Safety state and ready indicator: ▪ Continuously on: C-891 is busy with a lengthy operation, e.g. reference move or motion for commutation angle adjustment ▪ Continuously off: C-891 is ready for normal operation ▪ Flashing: A safety feature is activated, for example: – Phase current has exceeded the I^2t limit, the servo mode and the motor have been switched off (p. 38) – Coil temperature of the motor has exceeded the limit, the servo mode and the motor have been switched off (p. 38)
		During firmware update: Flashing alternating with PWR LED while the new firmware is written to the C-891.
	USB-B socket	Universal serial bus for connection to the PC

Labeling	Type	Function
Digital I/O	HD Sub-D 15 (f) (p. 119)	Digital lines: ▪ 4 outputs: Can be set with DIO, e.g. for use in macros ▪ 4 inputs: For future use ▪ Safety pin (p. 35)
Analog In	2-pin LEMO socket (p. 120)	Analog input For use, e.g., with an external sensor (p. 45).
Motor	HD Sub-D 26 (f) (p. 118)	Connection for motor (3-phase motor or stepper motor). Can also be used for connection of: ▪ Sensor data (A/B) ▪ Reference switch signals* ▪ Limit switch signals * The connection to be used for reference signal input is selectable via parameter (ID 0x3090; for details, see “Reference Moves” (p. 13)).
Sensor	Sub-D 15 (m) (p. 119)	Connection for: ▪ Sensor data (BiSS, SIN/COS, A/B) ▪ Reference switch signals* ▪ Limit switch signals * The connection to be used for reference signal input is selectable via parameter (ID 0x3090; for details, see “Reference Moves” (p. 13)).

Scope of Delivery

Item number	Description
C-891	PIMag® motion controller, including 2 mounting brackets and 4 screws that can be used to mount the C-891, for example, in a control cabinet
C-501.24120M12	24 V wide input range power supply (120 W/5 A) for use with line voltages from 100 to 240 V AC and voltage frequencies of 50 or 60 Hz, with 4-pin M12 connector (f)
3763	Power cord
C-815.34	RS-232 Null-Modem Cable, 3 m, 9/9-pin
000011448	USB cable (type A to type B) for connection to the PC
C-990.CD1	PI software CD for digital electronics
MS251E	User manual for the C-891, this document

Commandable Items

The following table contains the items that can be accessed with commands of the PI General Command Set (GCS).

Item	Number	Identifier	Description
Logical axis	1	1 (modifiable)	The logical axis represents the motion of the stage in the firmware of the C-891. It corresponds to the axis of a linear coordinate system. Motion commands for the axis are, for example, MOV and MVR (for closed-loop operation). The axis identifier can be queried with the SAI? command and modified with the SAI command. It can comprise up to 8 characters; valid characters are 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ-_ The new axis identifier is saved automatically in the nonvolatile memory and is thus still available even after a reboot or after the next switch-on.
Digital outputs	4	1 to 4	1 to 4 identify digital output lines 1 to 4 of the **Digital I/O** socket (p. 119). For further details, see “Digital Output Signals” (p. 39).
Wave generators	1	1	The wave generator is permanently allocated to the logical axis. For further details, see “Wave Generator” (p. 46).
Wave tables	1	1	The wave table contains the (temporarily) saved definition for the waveform that is output by the wave generator. The value of the ***Number Of Waves*** parameter (ID 0x1300010A) indicates the number of wave tables. The total number of available points for waveforms is 2,000,000, as indicated by the ***Max Number of Points in Wave Table*** parameter (ID 0x13000004).
Data recorder tables	≤8	1, 2, …	The data recorder tables contain the recorded data (a total of 16384 points; indicated by the ***Number Of Data Recorder Points*** parameter, ID 0x16000200). The number of data recorder tables can be set with the ***Data Recorder Chan Number*** parameter (ID 0x16000300). The data points are equally distributed among the available data recorder tables. Possible values: **Number of tables** / **Number of points per table** 1 / 16384 2 / 8192 4 / 4096 8 / 2048 The ***Max Number Of Data Recorder Channels*** parameter (ID 0x16000100) indicates the maximum number of data recorder tables.
Overall system	1	1	C-891 as an overall system.

Supported Motor and Sensor Types

Motor Types

The C-891 supports the following motor types:

- 3-phase motor
- Stepper motor

Via the ***Motor Type*** parameter (ID 0x3062), the C-891 is adapted to the motor type to be connected. The motor type selection determines the following:

- Relevant parameters: See the response to the HPA? command
- Suitable commands: OSM, SMO and FPH can only be used for a specific motor type, see below for details
- Relevant record options of the data recorder: See the response to the HDR? command

The value of the ***Motor Type*** parameter can only be changed in nonvolatile memory (using the SEP command), because the adaptation to a new motor type requires a reboot of the C-891. For more information, see “Adapting Settings” (p. 95).

Specific details for 3-phase motors:

3-phase motors require a commutation angle adjustment procedure that can be performed using the FPH command (p. 75). If the C-891 and the stage are supplied as a pre-configured system, the commutation offset is already set to a suitable value, and an adjustment procedure is to be performed only in the following cases:

- One of the system components has been replaced
- Stages with incremental sensor: The reference signal type to be used for reference moves has been changed, and therefore the reference position has changed. For further details, see “Reference Moves” (p. 13).

Open-loop motion can be commanded using the SMO command which sets the motor current. The OSM command, which starts steps in open-loop operation, cannot be used with 3-phase motors.

Specific details for stepper motors:

Because stepper motors do not require a commutation angle adjustment, the FPH command cannot be used.

Open-loop motion can be commanded using the OSM command which sets the number of steps to be performed. The SMO command cannot be used with stepper motors.

In open-loop operation, positions can be commanded and queried (using the MOV, MVR and POS? commands), provided that the value of the ***Encoder Type*** parameter is set to “step counter” (see “Sensor Types” and “Reference Moves” below for details). Note that the control algorithm is not used in this case.

Sensor Types

The C-891 supports the following sensor types:

- Analog SIN/COS signals (1 Vpp), interpolation in the C-891 (via **Sensor** connector): incremental sensor
- Digital BiSS interface (via **Sensor** connector): absolut measuring sensor
- A/B (TTL) signals via **Sensor** connector: incremental sensor
- A/B (TTL) signals via **Motor** connector: incremental sensor
- Step counter: with stepper motors only; motor steps are counted to determine the current position; if a sensor is present, it will be ignored. Only open-loop operation possible; positions can be commanded and queried using the MOV, MVR and POS? commands.

Via the ***Encoder Type*** parameter (ID 0x3032), the C-891 is adapted to the sensor type to be connected. If the "main" sensor signal is fed in via the **Sensor** connector, the sensor type setting has no effect on a second sensor which may be connected to the encoder lines of the **Motor** connector (details on p. 17).

The value of the ***Encoder Type*** parameter can only be changed in nonvolatile memory (using the SEP command), because the adaptation to a new sensor type requires a reboot of the C-891. For more information, see "Adapting Settings" (p. 95).

Reference Moves

The sensor type selection determines if a reference move is necessary:

- Incremental sensors: The servo mode can be switched on with the SVO command only if a reference move has been performed before.
- "Step counter" sensor type is set for a stepper motor: Target positions can be commanded with the MOV and MVR commands in open-loop operation only if a reference move has been performed before.
- Absolut measuring sensor: A reference move is not necessary.

The reference move is started using the FRF command. The sequence of the reference move depends on the reference signal type that is provided by the stage. The C-891 supports the following reference signal types:

- Homing track with one signal transition low/high or high/low
- Impulse
- Impulse, approach via negative limit
- Impulse, approach via positive limit
- Negative limit switch
- Positive limit switch

Via the ***Reference Type*** parameter (ID 0x70), the C-891 is adapted to the reference signal type that is provided by the connected stage. Via the ***Reference Input Selection*** parameter (ID 0x3090), the connector to be used for the input of the reference signal can be selected (**Motor** or **Sensor**). For more information, see "Adapting Settings" (p. 95).

The reference signal determines where the reference move ends. At the end of the reference move, the current position of the axis is set to the value of the ***Position Offset*** parameter (ID 0x48). Using this parameter, the zero position of the axis can be permanently shifted by adding an offset to the sensor position.

Travel Range and Soft Limits

The permissible travel range is established by the “soft limit” parameters ***Maximum Travel In Positive Direction [Phys. Unit]*** (ID 0x15) and ***Maximum Travel In Negative Direction [Phys. Unit***] (ID 0x30).

Motion commands are executed only if the commanded position is within these soft limits.

Note that changing the value of the ***Position Offset*** parameter (ID 0x48) does **not** change the soft limits. This means that if you change parameter 0x48, you also have to adapt the values of parameters 0x15 and 0x30.

Brake Handling

The C-891 provides parameters for handling of an electromagnetic brake. ***Has Brake?*** (ID 0x1A) indicates if a brake is present for the axis. Two supply voltage levels can be set for the brake: ***Brake Voltage [V]*** (ID 0x3095) is set at brake deactivation for one second to release the brake. One second after brake deactivation, ***Brake Continuous Voltage [V]*** (ID 0x3096) is applied to keep the brake released. The continuous supply voltage (0x3096) should be smaller than the release voltage (0x3095) to prevent heat from building up too much.

INFORMATION

An electromagnetic brake is activated (applied) if its supply voltage is zero. If the supply voltage is high enough, the brake is deactivated (released).

The brake can only be used if the ***Has Brake?*** parameter has the value 1 ("yes").

If ***Has Brake?*** has the value 1 ("yes"), the following applies:

- The brake can be activated or deactivated with BRA only if the servo mode is switched off. Secure the stage against unintentional motions before you deactivate the brake with BRA!
- Setting the servo mode with the SVO command influences the activation state of the brake:
  - Switching on the servo mode deactivates the brake.
  - Switching off the servo mode activates the brake.
- Switching off the motor with the EAX command switches off the servo mode too and therefore also activates the brake.
- When one of the safety features of the C-891 has been triggered (p. 35), the servo mode and the motor are switched off and the brake is activated.

Velocity, Acceleration and Deceleration

Velocity (in physical units/s), acceleration and deceleration (in physical units/s^2) can be set with the VEL, ACC and DEC commands and queried with VEL?, ACC? and DEC?. See also the descriptions of parameters 0x49, 0xA, 0xB, 0x4A, 0xC and 0x4B in “Adapting Settings” (p. 95). The settings are valid for motion in closed-loop **and** open-loop operation.

In closed-loop operation, the target generator of the C-891 performs calculations to specify the target position, velocity and acceleration of the axis for any point in time (dynamics profile). See “Generation of Target Values for Closed-Loop Operation” below for more information.

Generation of Target Values for Closed-Loop Operation

Basic Structure of the Control Loop

The structure of the control algorithm for closed-loop operation can be selected via the ***Controller Structure*** parameter (ID 0x301D). Possible values:

- 0 = POS/VEL I = parallel structure of position and velocity control, with feed forward (default setting)
- 1 = POS-VEL II = serial structure of position and velocity control, with feed forward

INFORMATION

The optimum values of the servo-control parameters of the C-891 depend on the selected closed-loop control algorithm and the application.

- Change the value of the ***Controller Structure*** parameter only when you are familiar with servo system basics as well as motion control concepts.
- Check the values of the servo-control parameters when the closed-loop control algorithm is changed. If necessary: Optimize the servo-control parameters.
- See “Adapting Settings” (p. 99) for parameter handling and a parameter overview.

INFORMATION

When the axis is on target, the velocity control is deactivated. The on-target state is true when the current position is inside the settling window and stays there at least for the duration of a delay time. The delay time is given by the ***Settling Time [s]*** parameter (ID 0x3F), and the settling window is given by the ***Settling Window [Phys. Unit]*** parameter (ID 0x7000900). If the settling window is set too large, the control loop may become instable.

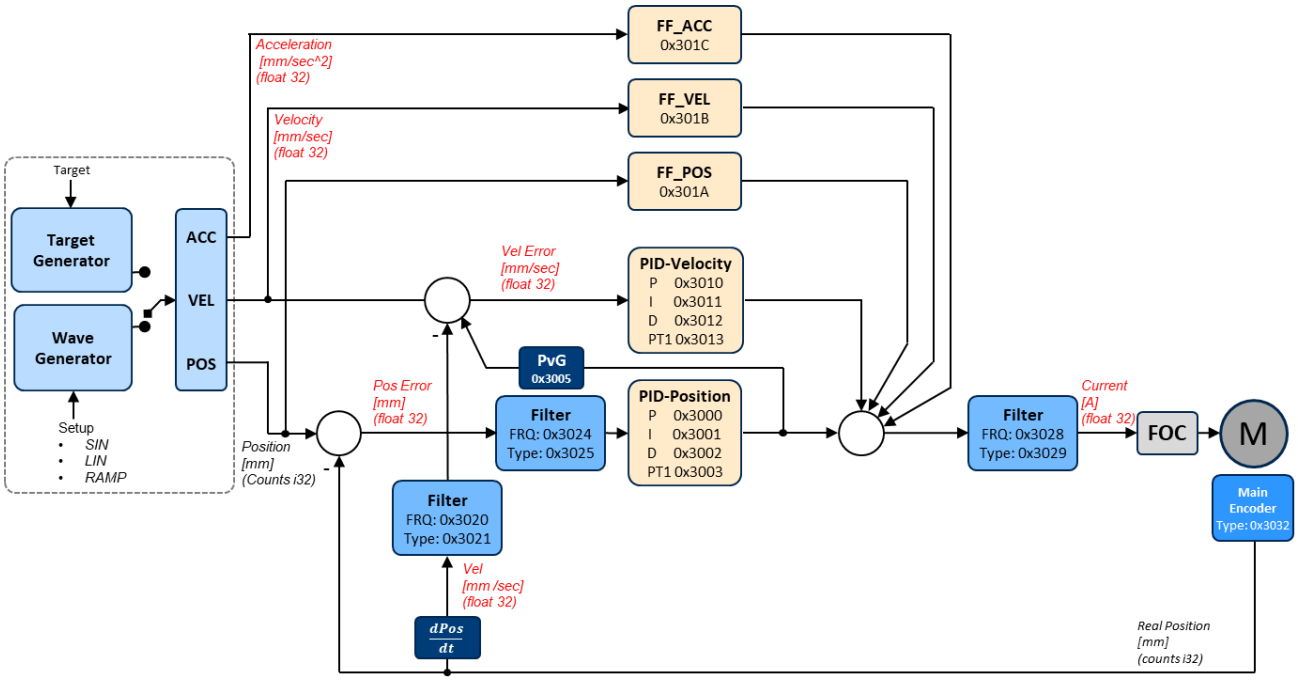


Figure 1: Default control algorithm: Block diagram for parallel structure of position and velocity control

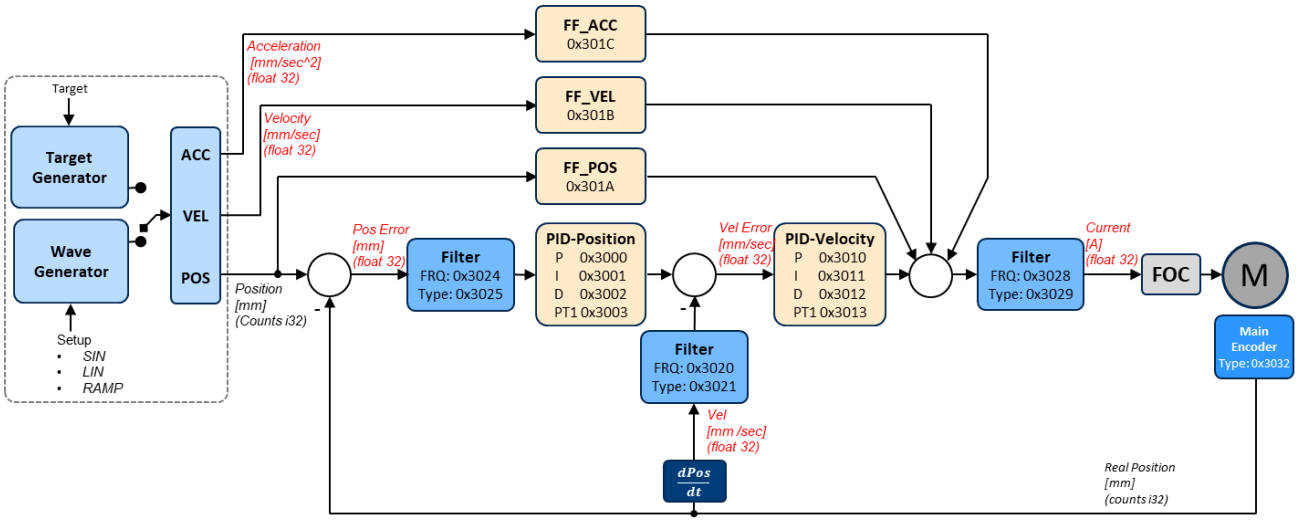


Figure 2: Alternative control algorithm: Block diagram for serial structure of position and velocity control

INFORMATION

The target generator of the C-891 performs calculations to specify the target position, velocity and acceleration of the axis for any point in time (dynamics profile).

- For a description of the target generator see “Generation of Dynamics Profile” in the MS205E User Manual of the C-863.11 Mercury DC Motor Controller. Note that the target generator is referred to as “profile generator” in the MS205E User Manual.
- For the motion parameters used by the target generator of the C-891, see the parameters of the *Motion* group in “Parameter Descriptions” (p. 99).

INFORMATION

The wave generator output is especially suited to dynamic applications with periodic axis motions.

- For a description of the wave generator see “Wave Generator” (p. 46).

Optional Use of a Second Sensor for the Velocity Control Loop

The velocity control loop can use its own sensor. The signal of this second sensor must be fed in via the **Motor** connector. This means that the “main” sensor must be connected via the **Sensor** connector and the value of the ***Encoder Type*** parameter (ID 0x3032) must be different from 3.

The second sensor must provide A/B (TTL) signals with one of the following signal processing options:

- Velocity (incremental sensor)
- Angle (incremental sensor)
- Angle (index reset). Reset angle every rotation after receive an index signal

Via the ***Second Encoder Signal Processing*** parameter (ID 0x30D0), the C-891’s velocity control loop is adapted to use its own sensor with the desired signal processing.
The value of the ***Second Encoder Signal Processing*** parameter can only be changed in nonvolatile memory (using the SEP command), because the adaptation of the control loop requires a reboot of the C-891. For more information, see “Adapting Settings” (p. 95).

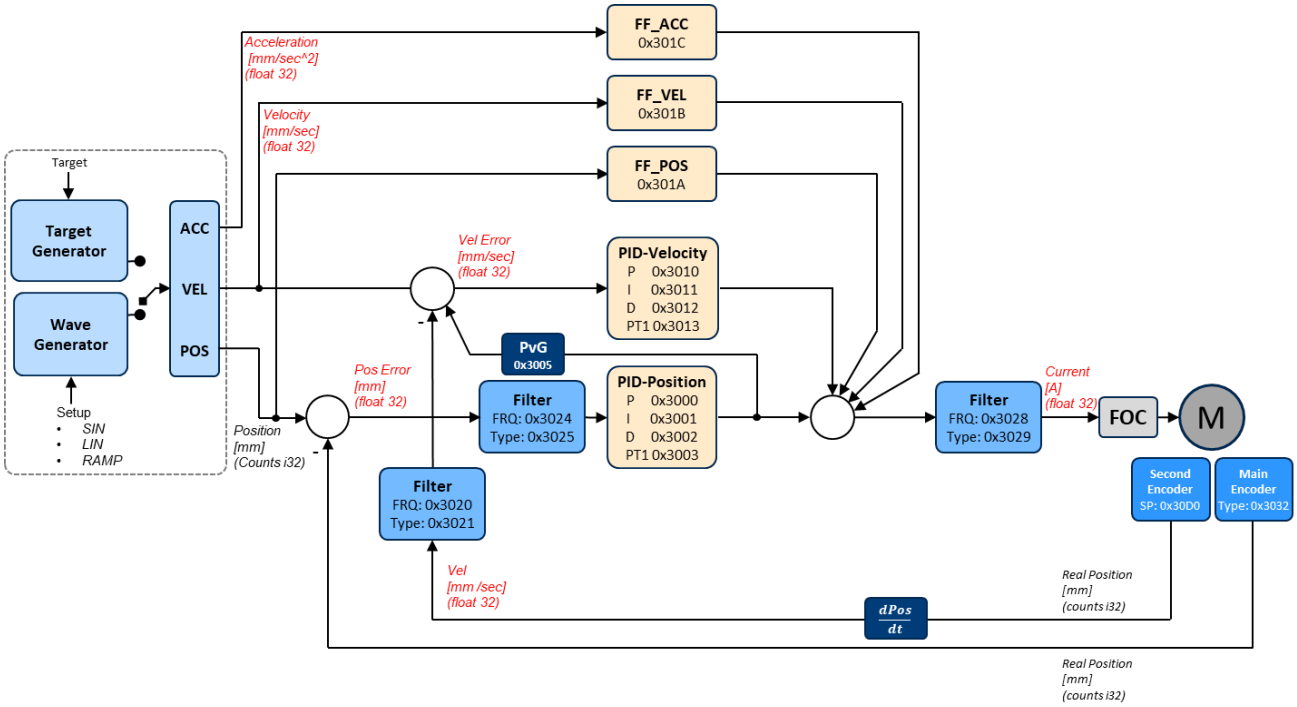


Figure 3: Control algorithm: Block diagram for parallel structure of position and velocity control with second sensor

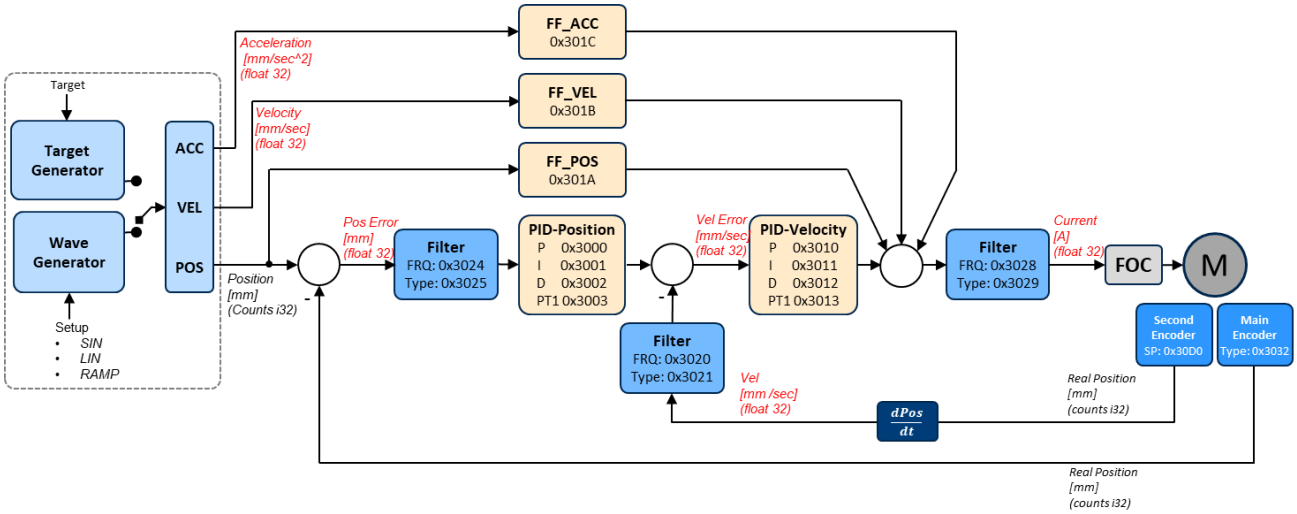


Figure 4: Alternative control algorithm: Block diagram for serial structure of position and velocity control with second sensor

Optional Use of Target Control Window

The C-891 can automatically adapt the servo-control parameters when the axis is close to the target position during movement. Automatic adaptation is activated via the ***Target Control Window*** parameter (ID 0x30B0) which can take the following values:

- 0 = Disabled: The control algorithm only uses the general servo-control parameters (ID 0x3000 to 0x3002 and ID 0x3010 to 0x3012)
- 1 = Enabled: The control algorithm uses the general servo-control parameters (ID 0x3000 to 0x3002 and ID 0x3010 to 0x3012) when the current position is outside a window which is centered around the target position ("target control window"). When the current position is inside the target control window, the C-891 automatically adapts the servo-control parameters.

INFORMATION

When the current position is inside the target control window, the C-891 adapts the servo-control parameters linearly between the values of general parameters (ID 0x3000 to 0x3002 and ID 0x3010 to 0x3012) and values which were specifically defined for the target position (parameter ID 0x30B2 to 0x30B7).

- The size of the target control window is given by the ***Target Control Window (Phys. Unit)*** parameter (ID 0x30B1). Note that the parameter value gives half the window width.
- The current value of the P term is given by ***Kp(x)***. I term and D term are calculated in the same way.

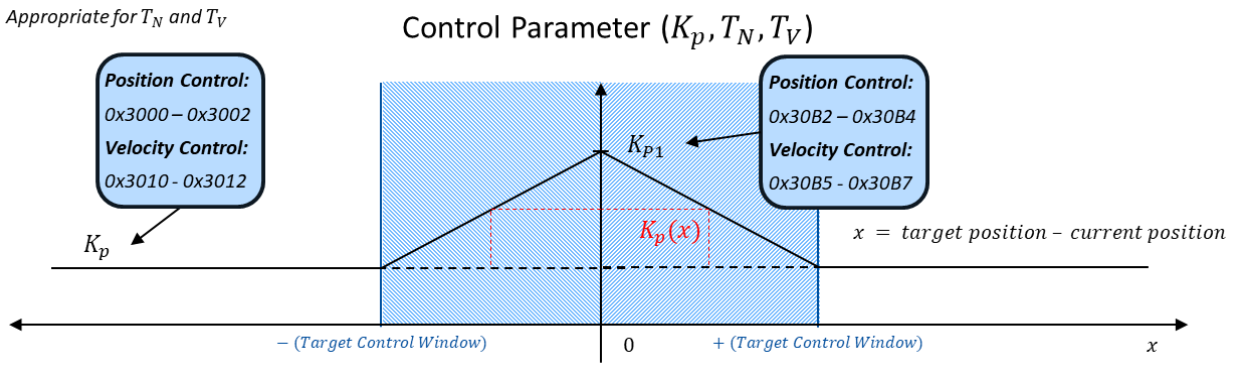


Figure 5: Adaptation of P term in the target control window; target position = 0 and x = target position – current position

Communication Interfaces

The C-891 can be controlled with ASCII commands (PI General Command Set) from a PC via the following communication interfaces:

- Serial RS-232 connection, baud rate: 115200
- USB connection

Firmware updates (p. 114) require a RS-232 connection.

Stage Databases

You can select a suitable parameter set for your stage from a stage database in the PC software from PI. The PC software transfers the values of the selected parameter set to the nonvolatile memory of the C-891 and reboots the C-891.

The values in the nonvolatile memory are loaded to the volatile memory as default values when the C-891 is switched on or rebooted and take effect immediately.

After loading a parameter set, a commutation angle adjustment must be performed for 3-phase motors; for details, see p. 75.

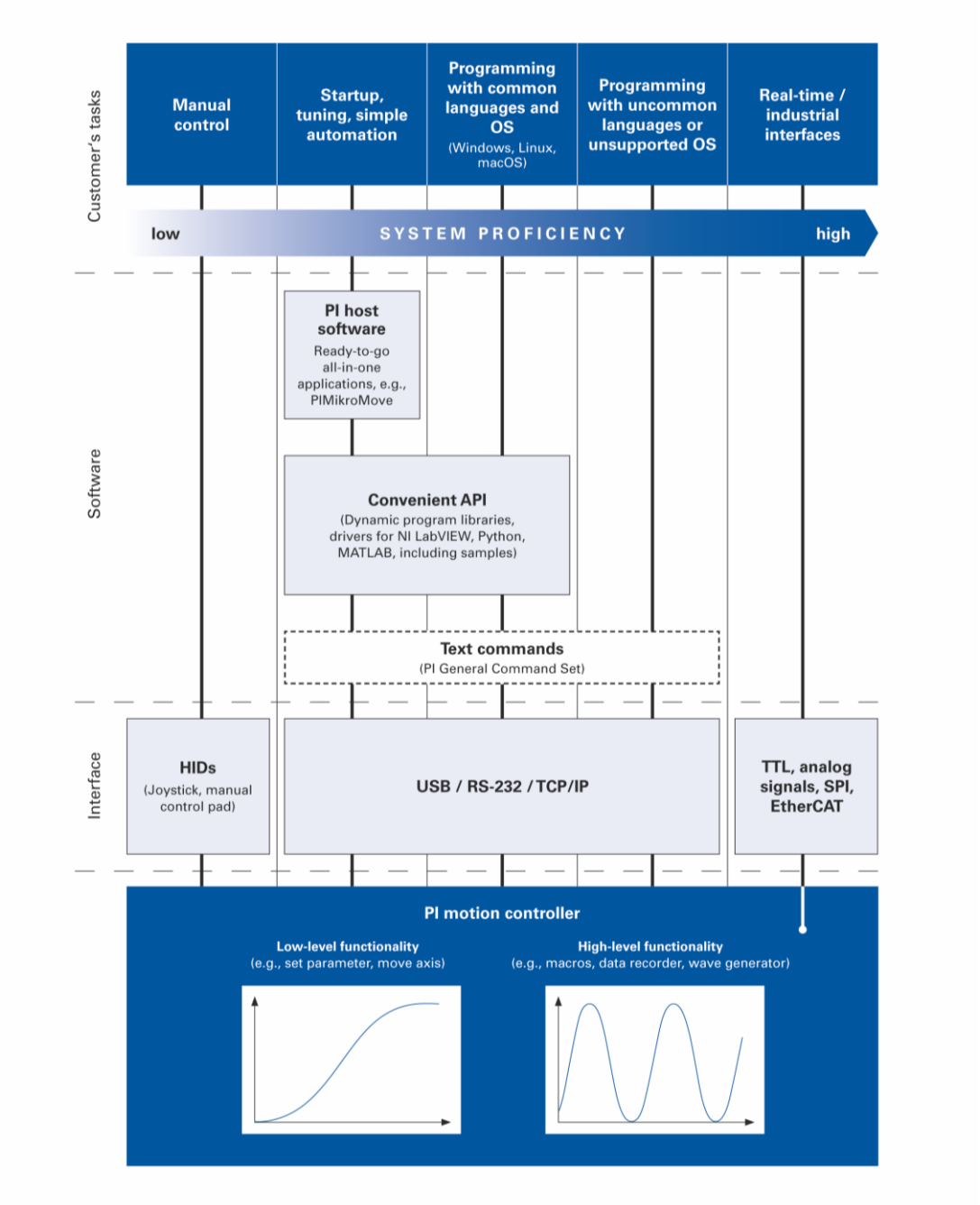
Database file name	Description
PISTAGES3.DB	Delivery includes parameter sets for all standard stages from PI and PI miCos; is automatically saved to the PC when the PC software is installed New parameter sets can be created, edited, and saved.
<Produkt>.db e.g.: V-xxxxxxx.db	Includes the parameter set for a custom stage. In order for the parameter set to be selected in the PC software, it must be added to the PISTAGES3.DB first, see "Installing a Custom Stage Database" (p. 25).

For a list of the parameters supported by the C-891, see “Parameter Descriptions” (p. 99). Note that the stage databases do **not** contain values for system parameters (Item Type = System in the parameter list).

For more information on the stage database, see the manuals for the PIStages3Editor and the PI GCS program library.

Overview of PC Software

PI's systems can generally be controlled as follows:



The following table shows the PC software that is included in the PI software CD. The given operating systems stand for the following versions:

- Windows: Windows 8.1, 10 (32 bit, 64 bit)
- Linux: Kernel 2.6, GTK 2.0, from glibc 2.15

PC software	Operating system	Short description	Recommended use
Dynamic program library for GCS	Windows, Linux	Allows software programming for the C-891 with programming languages such as e. g. C++. The functions in the dynamic program library are based on the PI General Command Set (GCS).	For users who would like to use a dynamic program library for their application. Is required for PIMikroMove. Is required for the drivers for NI LabVIEW software.
Drivers for use with NI LabVIEW software	Windows	NI LabVIEW is a software for data acquisition and process control (must be ordered separately from National Instruments). The driver library is a collection of virtual instrument drivers for PI electronics. The drivers support the PI General Command Set.	For users who want to use NI LabVIEW to program their application.
Merge Tool for use with NI LabVIEW drivers	Windows	The Merge Tool allows you to combine product-specific NI LabVIEW drivers from PI with each other.	For users who want to operate several products from PI at the same time while using NI LabVIEW.
PIPython	Windows, Linux, OS X	Collection of Python modules for convenient use of PI devices and GCS data. The modules can be used with 2.7+ and 3.4+ in the specified operating systems as well as via sockets on any other operating system of a PC.	For users who want to use Python for operating the controller. The use of Python as programming language offers a variety of possibilities and extends the scope of functions of the GCS commands considerably. Furthermore, debugging of macros is possible.
MATLAB drivers	Windows	MATLAB is a development environment and programming language for numerical calculations (must be ordered separately from MathWorks). The PI MATLAB driver consists of a MATLAB class that can be included in any MATLAB script. This class supports the PI General Command Set. The PI MATLAB driver does not require any additional MATLAB toolboxes.	For users who want to use MATLAB to program their application.
PIMikroMove	Windows	Graphic user interface for Windows with which the C-891 and other controllers from PI can be used: ▪ The system can be started without programming effort ▪ Graph of motions in open-loop and closed-loop operation ▪ Macro functionality for storing command sequences on the PC (host macros) ▪ Support of HID devices ▪ Complete environment for command entry, for trying out different commands No command knowledge is necessary to operate PIMikroMove. PIMikroMove uses the dynamic program library to supply commands to the controller.	For users who want to perform simple automation tasks or test their equipment before or instead of programming an application. A log window showing the commands sent makes it possible to learn how to use the commands.

PC software	Operating system	Short description	Recommended use
PITerminal	Windows	Terminal program that can be used for nearly all PI controllers (see the description of the ***Command Entry*** window in the PIMikroMove user manual).	For users who want to send GCS commands directly to the controller.
PIStages3Editor	Windows	Program for opening and editing stage databases in .db format.	For users who want to deal intensively with the contents of stage databases.
PISTAGES3.DB	Windows	Delivery includes parameter sets for all standard stages from PI and PI miCos. New parameter sets can be created, edited, and saved.	For users who want to load a parameter set for their stage.
PI Update Finder	Windows	Checks the PI software installed on the PC. If more current versions of the PC software are available on the PI server, downloading is offered.	For users who want to update the PC software.
PI Firmware Updater	Windows	Program for user support when updating firmware of the C-891.	For users who want to update the firmware.
USB driver	Windows	Driver for the USB interface	For all users.

Installation

General Notes on Installation

- Install the C-891 near the power source so that the power plug can be quickly and easily disconnected from the mains.
- Only use cables and connections that meet local safety regulations.

Installing the PC Software

The communication between the C-891 and a PC is necessary to configure the C-891 and send motion commands using the commands of the GCS. Various PC software applications are available for this purpose.

Doing the Initial Installation

Accessories

- PC with a Windows operating system (8.1, 10) or Linux operating system
- PI software CD (included in the scope of delivery)

Installing the PC software in Windows

1. Start the installation wizard by double-clicking PISoftwareSuite.exe in the installation directory (root directory on the CD).
   The ***InstallShield Wizard*** window opens for installing the PC software.
2. Follow the instructions on the screen.
   The PI software suite includes the following components:
   - Driver for use with NI LabVIEW software

- Dynamic program library for GCS
- PIMikroMove
- PC software for updating the firmware of the C-414
- PI Update Finder for updating the PC software
- USB driver

Installing the PC software on Linux

1. Unpack the tar archive from the /linux directory of the PI software CD to a directory on your PC.
2. Open a terminal and go to the directory to which you have unpacked the tar archive.
3. Log on as a superuser (root rights).
4. Enter ./INSTALL to start the installation.
   Pay attention to lower and upper case when entering commands.
5. Follow the instructions on the screen.

You can select individual components for installation.

Installing Updates

PI is constantly improving the PC software.

➢ Always install the latest version of the PC software.

Prerequisite

➢ Active connection to the Internet.

➢ If your PC uses a Windows operating system:

- You have downloaded the PI Update Finder manual (A000T0028) from the PI website. The link is in the "Downloading_Manuals_from_PI_EN_A000T0081" file in the \Manuals directory on the PI software CD.

➢ If your PC uses a Linux operating system:

- You have the access data (user name and password) for the PI software CD. Information regarding the access data can be found in the file "C-990.CD1_Releasenews.pdf" in the \Manuals folder on the PI software CD.

Updating the PC software and PISTAGES3.DB on Windows

➢ Use the PI Update Finder:

- Follow the instructions in the PI Update Finder manual (A000T0028).

Updating the PC software on Linux

INFORMATION

If software is missing in the ***Downloads*** area or problems occur with downloading:

➢ Contact our customer service department (p. 121).

1. Open the website https://www.physikinstrumente.com/en/products/motion-control-software/.
2. Click ***Login***.
3. Log in with the user name and password for the PI software CD.
4. Scroll down to ***Downloads***.
5. Click the archive file "CD Mirror" or the associated download link.
6. Select the option in the following request to save the file to your PC.

   If you do not specify anything else, the "CD Mirror" archive file is stored in the default download directory of your PC.
7. Unpack the archive file into a separate installation directory.
8. Go to the ***linux*** subdirectory in the directory with the unpacked files.
9. Unpack the archive file in the ***linux*** directory by entering the command tar -xvpf <name of the archive file> on the console.
10. Read the accompanying information on the software update (readme file and/or "C-990.CD1_Releasenews.pdf" file) and decide whether the update makes sense for your application.
    - If no: Stop the update procedure.
    - If yes: Go through the following steps.
11. Log into the PC as superuser (root privileges).
12. Install the update.

Updating PISTAGES3.DB on Linux

1. Contact the customer service department (p. 121) to receive the latest version of the stage database PISTAGES3.DB.
2. Log onto the PC as a superuser (root account).
3. Install the update, which you have received from our customer service department, on your PC.

Installing a Custom Stage Database

If the C-891 and a custom stage are **not** supplied as a pre-configured system, you will receive a CD from PI with the following contents:

- Program Import PI CustomStage
- Custom stage database with the parameter set for the stage

In order for the parameter set to be selected in the PC software, it must first be inserted into the stage database PIStages3 with the Import PI Custom Stage program.

➢ Install the customer stage database by double-clicking the file ***Import_PI_CustomStage.exe*** in the root directory of the CD.

The parameter set from the customer stage database is inserted into PIStages3.

If a message appears that the installation of the customer stage database failed:

a) Update the PIStages3 stage database on your PC, see "Installing Updates" (p. 24).
b) Repeat the installation of the customer stage database.

Ensuring Ventilation

High temperatures can overheat the C-891.

- ➢ Set up the C-891 with a distance of at least 5 cm to the top and 2 mm to the bottom sides. If this is not possible, make sure that the area is cooled sufficiently.
- ➢ Set up the C-891 so that the ventilation holes in the housing are freely accessible.
- ➢ Ensure sufficient ventilation at the place of installation.
- ➢ Keep the ambient temperature to a non-critical level (<40° C).

Operation

General Notes on Operation

NOTICE

Unexpected motions from lack of self-locking!
A PIMag® magnetic direct drive (3-phase motor) has no self-locking. If no brake is present, a connected stage that is equipped with such a drive can unexpectedly move in the following cases:

- Switching off the C-891
- Rebooting the C-891 with the RBT command or with the corresponding functions of the PC software
- Switching off the servo mode for the axis.
- Switching off the motor for the axis.

Unexpected motions can result in damage to the stage and/or the load attached to it.

- ➢ Before switching off or rebooting the C-891, take suitable measures to ensure that no unexpected motions are possible due to the lack of self-locking of the PIMag® magnetic direct drive. Examples of measures:
  - – Approaching a "safe" position
  - – Installation of a mechanical device to catch the moving part

INFORMATION

If the C-891 and the stage are supplied as a pre-configured system:

- ➢ If a connection assignment is given on the labels of the C-891 and/or stage, observe this assignment when connecting the stage.
- ➢ Make sure to use the supply voltage for which the C-891 is preset (parameter 0x3056, can be 24 or 48 V DC).

INFORMATION

For the following functionality of the C-891, see the MS205E User Manual of the C-863.11 Mercury DC Motor Controller:

- Data recorder
- Controller macros
- Variables

INFORMATION

The C-891 is switched off by disconnecting the power source. Options:

- ➢ Pull the power cord out of the power socket.
- ➢ Pull the power cord out of the power supply.
- ➢ Pull the M12 connector of the power supply out of the M12 panel plug on the C-891.

Starting the System

NOTICE

Damage from incorrect installation!

The stages that are intended for operation with the C-891 are able to generate high forces and high accelerations. Damage can occur to the stage, the load or the environment during operation if the stage and the load are **not** installed correctly.

- ➢ Make sure that no collisions are possible between the stage, the load to be moved and the environment in the motion range of the stage.
- ➢ Start up the system only if the stage and the load have been installed properly. Follow the instructions in the documentation of the stage.
- ➢ If present, remove the transport lock from the stage before you start up the system.

INFORMATION

If the C-891 and the stage are **not** supplied as a pre-configured system, or if you have replaced one of the system components, you can load a suitable parameter set from a stage database (p. 20). Note that the stage database does **not** contain values for system parameters (Item Type = System in the parameter list on p. 99). It is strongly recommended that you create a backup copy of the current parameter values before you load a parameter set.

To load a parameter set, proceed as follows in PIMikroMove when you start up the C-891 with the stage for the first time:

1. Establish communication between the C-891 and the PC in PIMikroMove.
2. Create a backup copy of the parameter values from nonvolatile memory, see “Saving Parameter Values in a Text File” (p. 95).
3. In the ***Start up controller*** window, click ***2. Select connected stages*** in the left column (use ***C-891 > Select connected stages …*** if necessary).
4. Select the matching stage type from the ***Stage database entries*** list. Make sure that the selection is suitable for the supply voltage used (24 or 48 V DC).
5. Click on ***Assign ->***.
6. Confirm selection with ***OK*** to load the parameter settings for the selected stage type from the stage database into the nonvolatile memory of the C-891. If a message box opens indicating that the settings will be stored permanently in the controller, confirm again with ***OK***.
   Loading takes some time and includes a reboot of the C-891.
7. Perform a commutation angle adjustment using the FPH command (p. 75).
8. Save the result of the adjustment procedure to nonvolatile memory. See "Adapting Settings" (p. 95), or use the WPA command.

INFORMATION

The C-891 can be configured with the ***Motor Power At Startup*** parameter (ID 0x3080) so that the motor is automatically switched on upon switch-on or rebooting. Default setting of the parameter value: Motor is **not** automatically switched on.

INFORMATION

For stages with incremental sensors, the servo mode can be switched on for the axis only after a reference move has been successfully finished. For further details, see “Reference Moves” (p. 13).

INFORMATION

The available motion commands depend on the motor type and sensor type of the connected stage; see “Motor Types” (p. 12) and “Sensor Types” (p. 13) for details. PIMikroMove appropriately provides the corresponding controls. Possible special cases:

- Motor type = stepper motor; sensor type = step counter (motor steps are counted to determine the current position):
  - Controls for switching on the servo mode are hidden.
  - Controls for open-loop steps **and** for target positions are available.
- Motor type = stepper motor; sensor type = incremental or absolute measuring position sensor:
  - Controls for switching on the servo mode are available.
  - Controls for open-loop steps **or** for target positions are available, depending on the activation state of the servo mode.

➢ If necessary, add the columns for open-loop steps to the ***Axes*** card of the PIMikroMove main window. To add columns, follow step 9 of the instructions below.

Proceed as follows to start the C-891 with the stage in PIMikroMove:

1. Install the following on the PC:
   - The PC software and the USB drivers from the product CD
   - Updates for PC software

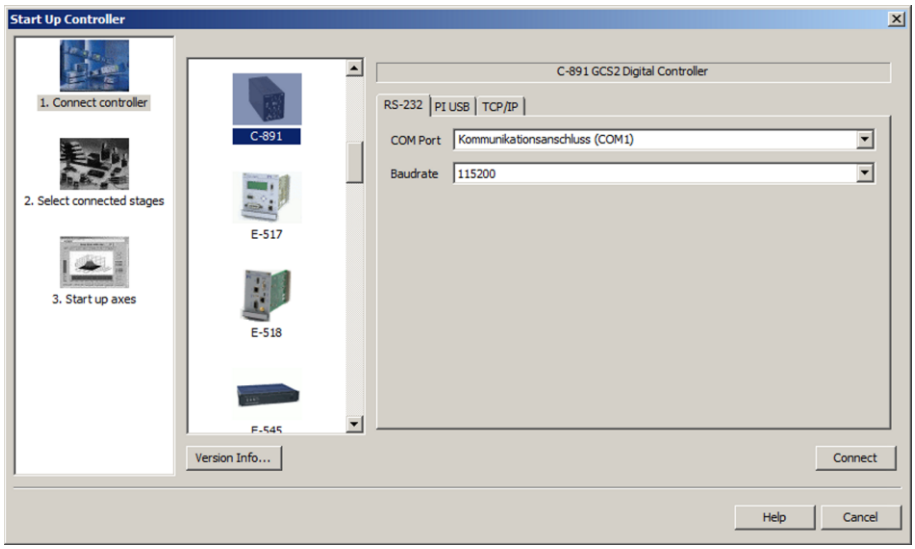
   Details see "Installing the PC Software" (p. 23).
2. Install the C-891:
   - Observe the general information on installation (p. 23).
   - Ensure the ventilation (p. 26).
3. Connect the following to the C-891:
   - The stage to the **Motor** socket and, if a separate sensor connection is required, to the **Sensor** panel plug.
   - The PC via the RS-232 interface (**RS-232** panel plug) or via the USB interface (USB type B).
   - A suitable power supply (output: 24 or 48 V DC) to the **Power In** panel plug. Make sure that the power supply is **not** connected to the power socket yet.
4. Switch on the C-891 by connecting the power cord of the wide-range-input power supply to the power socket.

   The C-891 is ready for operation when the **PWR** LED is continuously lit.
5. Start PIMikroMove on the PC.
6. Establish communication between the C-891 and the PC in PIMikroMove via RS-232 or USB. The baud rate to be used for RS-232 communication is 115200.

   Here, the communication is established via RS-232 (the procedure for USB is similar):

   a) Select ***C-891*** in the field for controller selection.
   b) Select the ***RS-232*** tab on the right side of the window.
   c) In the ***COM Port*** field, select the COM port of the PC to which you have connected the C-891.

d) Set the value 115200 in the ***Baudrate*** field to adapt the baud rate of the PC to the baud rate of the C-891.
e) Click ***Connect*** to establish communication.



The ***Start up controller*** window changes to the ***Start up axes*** step.

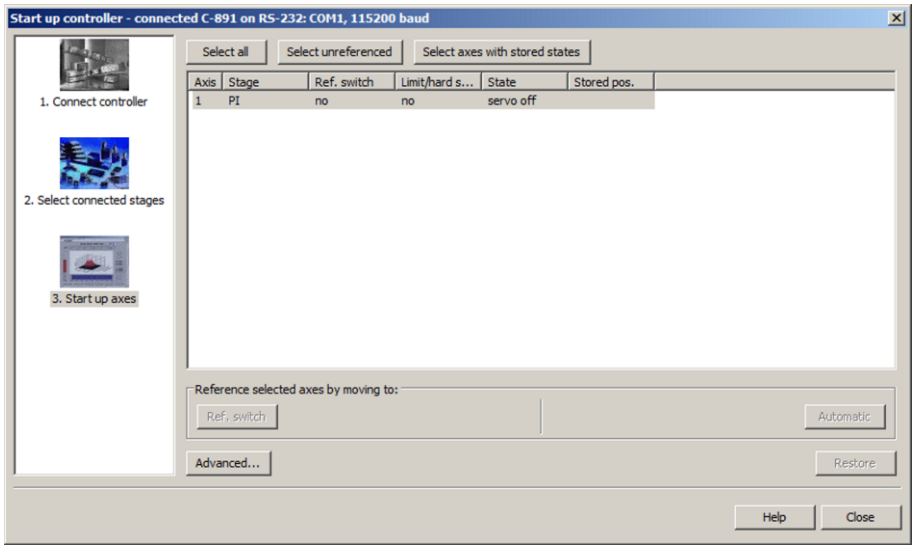
7. If necessary, execute the reference move for the axis in the ***Start up axes*** step so that the controller knows the absolute axis position:

   Is a reference move necessary?

   - Stage with absolute measuring position sensor: Reference move is **not** necessary. All buttons in the ***Reference selected axes by moving to:*** area are deactivated.
   - Stage with incremental position sensor: Reference move is necessary.
   - "Step counter" sensor type is set for a stage with stepper motor, and target positions are to be commanded with the MOV or MVR command in open-loop operation: Reference move is necessary.

   If a reference move is **not** necessary:

   a) Click ***Close*** to go to the main window of PIMikroMove.
   b) Continue with step 9.



In the example shown in the figure above, a V-551 stage with absolute measuring sensor is connected to the C-891. Therefore, a reference move is **not** necessary.

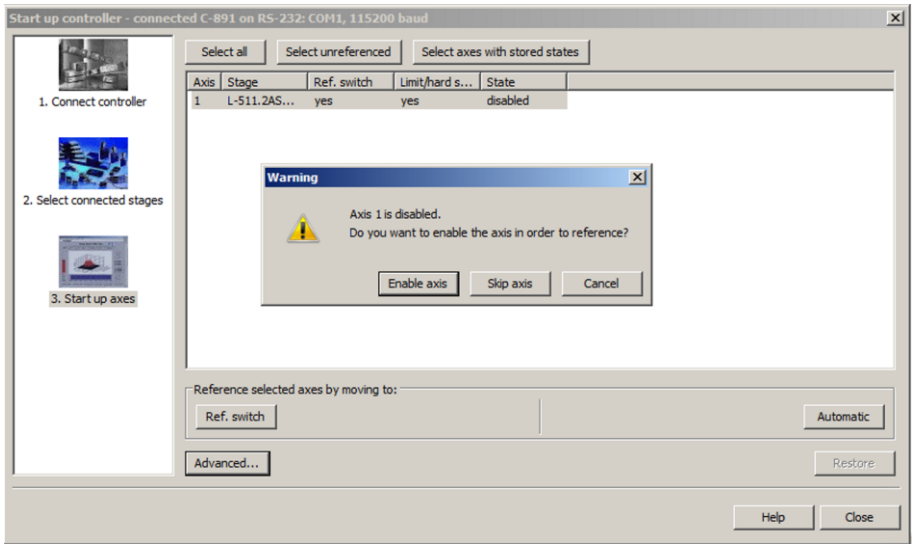
If a reference move is necessary:

- Click on ***Ref. switch*** to start the reference move.

If a warning message appears indicating that the axis is disabled:

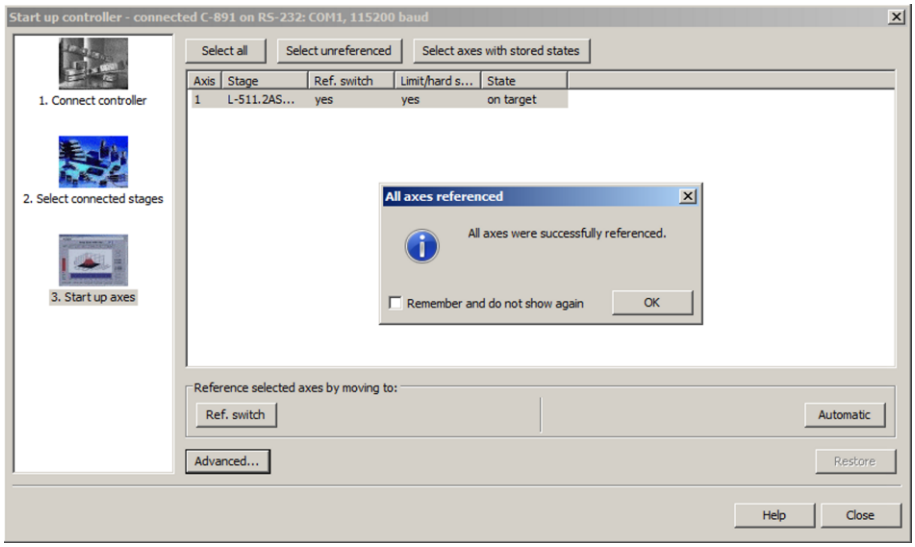
- Enable the axis by clicking on the ***Enable axis*** button (switches the motor on).

The axis executes the reference move.



In the example shown in the figure above, an L-511 stage with incremental sensor is connected to the C-891. Therefore, a reference move is necessary.

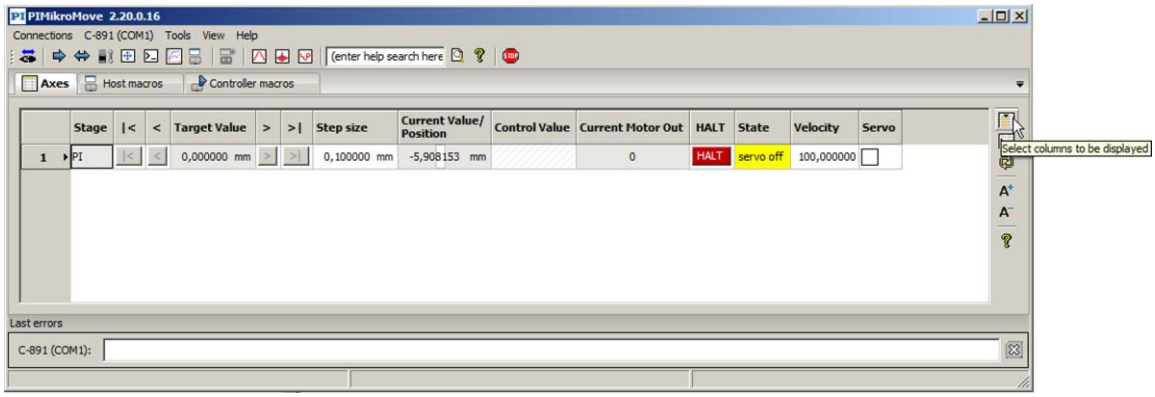
8. After a successful reference move, click on ***OK > Close***.



The main window of PIMikroMove opens.

9. If the ***Axes*** card of the main window does not show the ***Enable Axis*** column, add this column to the display:

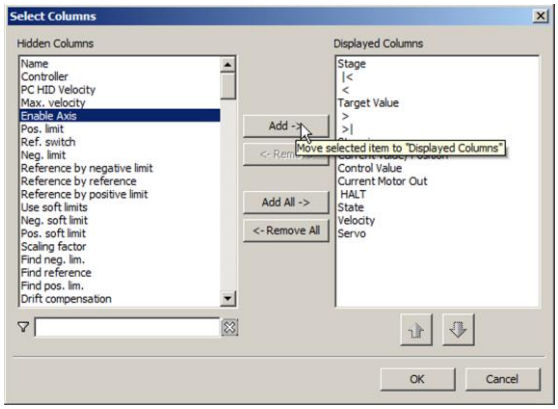
   a) On the right margin of the ***Axes*** card, click  (***Select columns to be displayed***).



The ***Select Columns*** dialog opens.

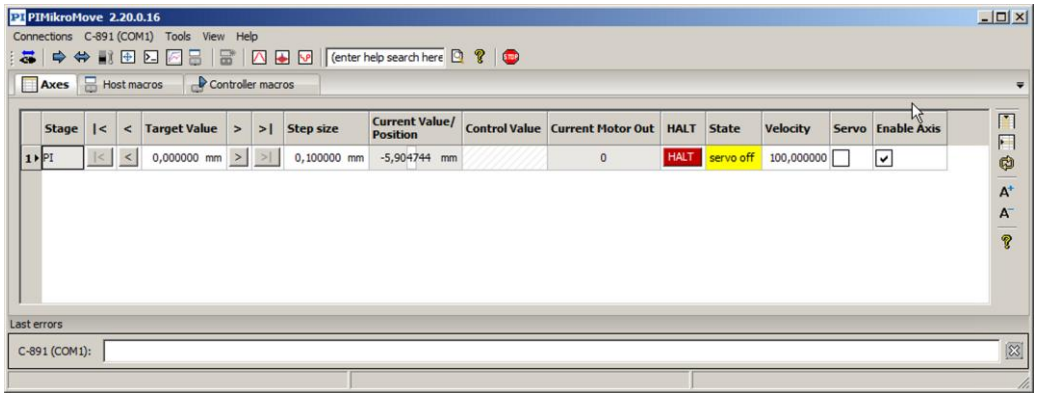
   b) In the ***Select Columns*** dialog, go to the ***Hidden Columns*** area and select the ***Enable Axis*** line.

   c) In the ***Select Columns*** dialog, click ***Add ->***.

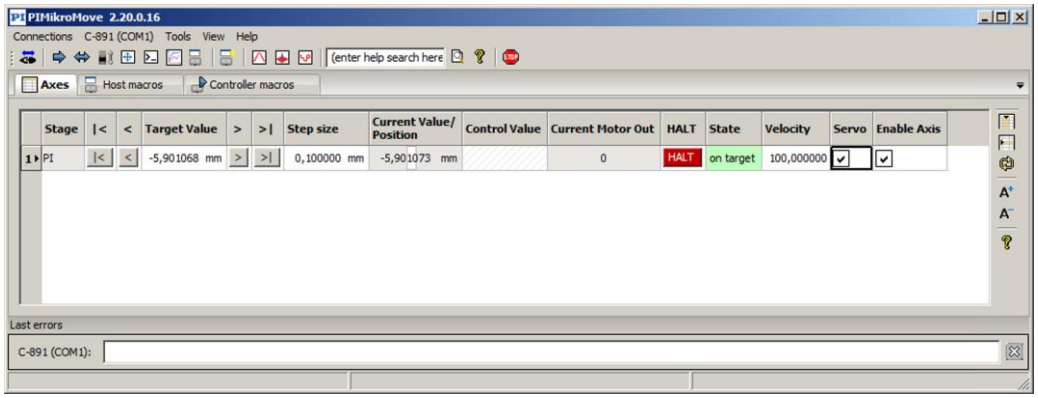


d) Close the ***Select Columns*** dialog by clicking ***OK***.

10. If the axis is not yet enabled: On the ***Axes*** card in the main window of PIMikroMove, switch on the motor for the axis by marking the check box in the ***Enable Axis*** column.



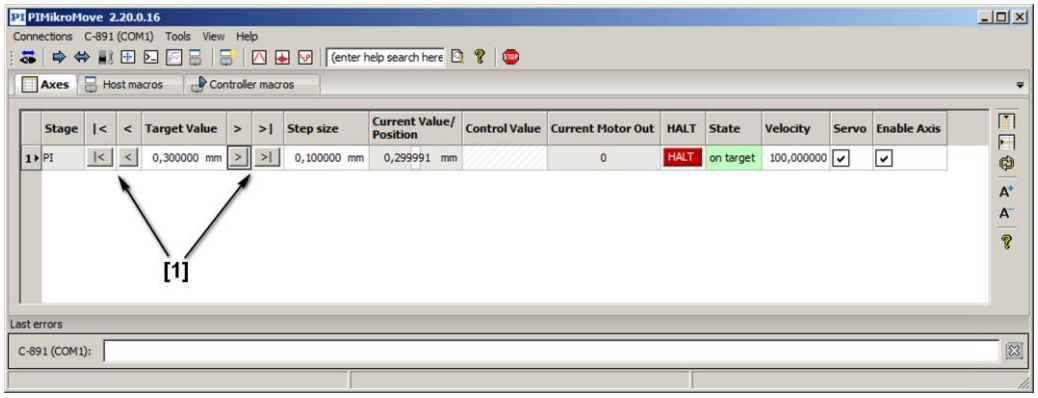
11. If the stage has an incremental or absolute measuring position sensor: On the ***Axes*** card in the main window of PIMikroMove, switch on the servo mode for the axis by marking the check box in the ***Servo*** column.



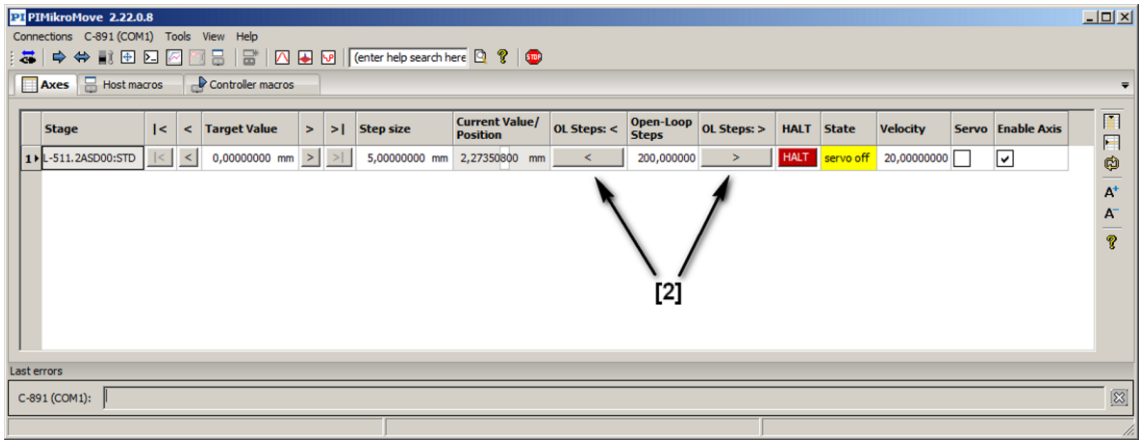
12. Start a few test motions to position the axis.

    Depending on the motor type and sensor type of the connected stage, controls for target positions and/or for open-loop steps are available on the ***Axes*** card in the main window of PIMikroMove.

    Using the controls for target positions, you can execute, for example, motions of a particular distance (specification in ***Step size*** column) or to the limits of the travel range by clicking the corresponding arrow keys [1] for the axis. Furthermore, you can enter a new target value in the ***Target Value*** column (apply with the ENTER key).



    Using the controls for open-loop steps, you can enter the number of steps to perform in the ***Open-Loop Steps*** column (apply with the ENTER key) and start the motion by clicking the corresponding arrow keys [2] for the axis.



Safety Features of the C-891

NOTICE

Unexpected motions from lack of self-locking!
A PIMag® magnetic direct drive (3-phase motor) has no self-locking. If no brake is present, a connected stage that is equipped with such a drive can unexpectedly move when the servo mode or the motor are switched off for the axis.

➢ Take suitable measures to ensure that unexpected motions cannot cause damage to the stage, the load affixed to it or the environment.

The safety features described in this section can be configured with parameters of the C-891. See „Adapting Settings“ (p. 95) for how to check and change parameter values.

When one of the safety features has been triggered, proceed as follows to re-establish readiness for operation:

1. Send the `ERR?` command to read out the error code.

   `ERR?` resets the error code to zero during the query.
2. Check your system and make sure that the axis can be moved safely.
3. Switch on the motor for the axis with the `EAX` command.
4. Switch on the servo mode for the axis with the `SVO` command.
   When switching on the servo mode, the target position is set to the current axis position and the brake is deactivated, if present. Now the axis can move again and you can command a new target position.

Motor Deactivation

The motor can be switched on / off for the axis with the EAX command (p. 74). Switching off the motor also switches off the servo mode and – if present – activates the brake. As long as the motor is switched off, the following applies:

- No move commands are accepted.
- The servo mode cannot be switched on.
- The brake can be deactivated: Secure the stage against unintentional motions before you deactivate the brake with the BRA command!

In the following cases, the motor is switched off automatically by the C-891:

- A motion error occurred (error code -1024, see p. 36)
- Safety stop is configured appropriately and was activated (see p. 37)
- Motor temperature too high, see “Overtemp Protection of the Motor” (p. 38)
- Motor current to high, see “Overcurrent Protection of the System” (p. 38)

The C-891 can be configured with the ***Motor Power At Startup*** parameter (ID 0x3080) so that the motor is automatically switched on upon switch-on or rebooting. Default setting of the parameter value: Motor is **not** automatically switched on.

Motion Error

A motion error occurs when the position error (i.e. the absolute value of the difference between the current position and the commanded position) exceeds the specified maximum value in closed-loop operation. The range in which the deviation may lie is specified by the ***Maximum Position Error (Phys. Unit)*** parameters (ID 0x8) on the C-891.

If a motion error occurs, the C-891 reacts as follows to protect the system from damage:

- Servo mode is switched off for the axis.
- Motor is switched off for the axis.
- If present, the brake is activated for the axis.
- All motions are stopped.
- Error code -1024 is set on the C-891.

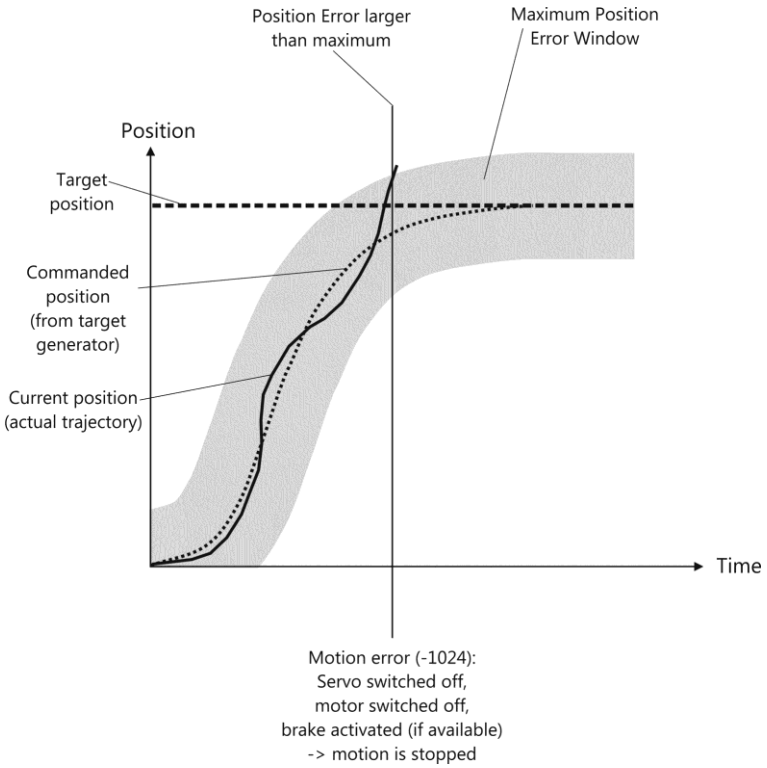


Figure 2: Behavior in case of motion errors (here, the target position was given by a MOV command)

Stop Axis by External Hardware Switch

The „Safety Logic“ settings of the C-891 can be configured so that a suitable hardware switch connected to pins 9 („Safety Pin“) and 10 (GND) of the **Digital I/O** socket (p. 119) can be used to stop the axis motion.

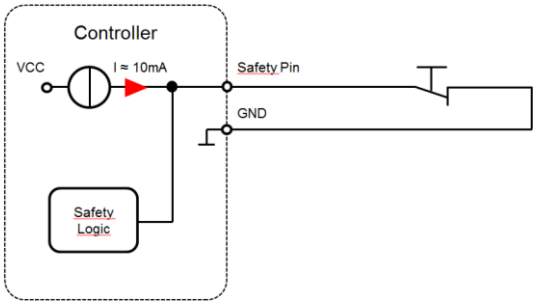


Figure 6: Signal path for connection of external hardware switch;
“Safety Pin” = pin 9, and “GND” = pin 10 of **Digital I/O**

If the „Safety Logic“ of the C-891 is configured appropriately, it evaluates the signal on the „Safety Pin“. Depending on the state of the external hardware switch, the connection between pins 9 and 10 is open or closed, and the signal on pin 9 is 0 V or 5 V. The action to be triggered by a transition between the signal levels and the signal polarity (active high or active low) depends on the configuration settings.

Requirements for use of an external hardware switch:

- Suitable hardware switch connected between pins 9 and 10 of the **Digital I/O** socket. R_{ON} < 30 ohm (C-891 feeds the safety pin with ca. 10 mA).
- Suitable configuration of the „Safety Logic“ settings: The parameters ***Safety Pin Action*** (ID 0x3081) and ***Safety Stop*** (ID 0x3082) of the C-891 must be set to suitable values, see below.

Parameter	Description and Possible Values
Safety Pin Action (ID 0x3081)	Configuration of safety pin action: 0 = OFF: Safety pin deactivated (default) 1 = Fast stop with maximum deceleration (active low) 2 = Fast stop with maximum deceleration (active high) 3 = Stop with deceleration (active low) 4 = Stop with deceleration (active high)
Safety Stop (ID 0x3082)	Configuration of axis behavior after safety pin action: 0 = “Runner Free”: Servo mode and motor are switched off (default; SVO? and EAX? both will return 1 = 0) → **No self-locking of the axis, if no brake is present!** 1 = “Runner Clamped”: The servo mode is switched off, and the motor remains on (SVO? will return 1 = 0, and EAX? will return 1 = 1)

Overload Protection of the Motor Driver

The motor driver of the C-891 can be protected against overload by limiting the phase current. For that purpose, the maximum current per phase can be set via the ***Current Max Trip Zone [A]*** parameter (ID 0x3073).

INFORMATION

The limitation of the phase current can decrease the performance of the stage.

Overtemp Protection of the Motor

If present in the stage, a thermistor can be used to protect the motor coils against overheating. The usage of the thermistor signal, and hence the overtemp protection of the motor, can be enabled in the C-891 via the ***Temperature Fuse*** parameter (ID 0x3077), see table below.

If the overtemp protection is enabled in the C-891, and the motor temperature exceeds the maximum specified via the ***Coil Temperature Maximum (C)*** parameter (ID 0x3078), the C-891 reacts as follows:

- Servo mode is switched off for the axis.
- Motor is switched off for the axis.
- If present, the brake is activated for the axis.
- All motions are stopped.
- The BSY LED flashes as long as the motor temperature exceeds the maximum.
- Error code 603 is set on the C-891.

Parameter	Description and Possible Values
Temperature Fuse (ID 0x3077)	Activation state of motor overtemp protection: 0 = None: Disabled 1 = Version A: Enabled (PT100; default) 2 = Version B: For future use
Coil Temperature Maximum (C) (ID 0x3078)	Temperature threshold for motor overtemp protection

Overcurrent Protection of the System

The C-891 provides an I^2t overcurrent protection to protect the system against overcurrent. From the following three parameters, the C-891 calculates an I^2t overcurrent threshold.

Parameter	Description and Possible Values
Continuous Phase Current [A] (ID 0x3074)	Nominal current of the drive (unit: A) Refer to the datasheet for the mechanics.
Peak Phase Current [A] (ID 0x3075)	Peak current Ip of the drive (unit: A) Refer to the datasheet for the mechanics.
Peak Phase Current Duration [s] (ID 0x3076)	Maximum duration of peak current (unit: s) Refer to the datasheet for the mechanics.

If the I^2t overcurrent threshold is exceeded, the C-891 reacts as follows:

- Servo mode is switched off for the axis.
- Motor is switched off for the axis.
- If present, the brake is activated for the axis.
- All motion is stopped.
- The BSY LED flashes as long as the I^2t overcurrent threshold is exceeded.
- Error code 655 is set on the C-891.
- Error code 656 is set on the C-891 when the actual current exceeds the value of the ***Peak Phase Current [A]*** parameter (ID 0x3075).

Digital Output Signals

The digital outputs of the C-891 are available at the **Digital I/O** socket (p. 119).

➢ Get the number of the output lines available on the C-891 with the `TIO?` command.

External devices can be triggered via the digital outputs of the C-891. Potential applications:

- Linking the trigger output to the motion of the axis. Details and examples can be found in this section.
- Direct switching of output lines, e. g., in macros. Details and examples of macros can be found in "Controller Macros" in the MS205E User Manual of the C-863.11 Mercury DC Motor Controller.

Commands for Digital Outputs

The following commands are available for the use of digital outputs:

Command	Syntax	Function
`CTO`	CTO {<TrigOutID> <CTOPam> <Value>}	Configures the conditions for the trigger output. Couples the trigger output to the axis motion.
`DIO`	DIO {<DIOID> <OutputOn>}	Switches digital output lines directly to the low or high state, either separately or all lines at once. Should not be used for output lines on which the trigger output is enabled with `TRO`.
`TRO`	TRO {<TrigOutID> <TrigMode>}	Enables or disables the trigger output conditions set with `CTO`. Default: Trigger output disabled.

One configuration setting can be made per `CTO` command:

`CTO <TrigOutID> <CTOPam> <Value>`

- `<TrigOutID>` is one digital output line of the controller.
- `<CTOPam>` is the CTO parameter ID in decimal format.
- `<Value>` is the value to which the CTO parameter is set.

The following trigger modes (<Value>) can be set for <CTOPam> = 3:

<Value>	Trigger mode	Short description
0 (default)	Position Distance	Once the axis has moved a specified distance, a trigger pulse is output (p. 40). Optionally, start and stop values can be defined to limit triggering to one position range and one particular direction of motion (negative or positive).
2	On Target	The on-target state of the axis selected is output at the selected trigger output (p. 42).
3	MinMax Threshold	The selected digital output line is active when the position of the selected axis is within a specified band (p. 43).
5	Motion Error	The selected digital output line becomes active when a motion error occurs (p. 43). The line stays active until the error code is reset to 0 (by a query with `ERR?`).
6	In Motion	The selected digital output line is active as long as the selected axis is in motion (p. 44).
10	Trigger Servo	The signal level of the selected digital output line changes with every servo cycle (p. 44).

In addition, the polarity (active high / active low) of the signal at the digital output can be set (p. 45).

INFORMATION

The settings for the configuration of the digital output lines can only be modified in the volatile memory of the C-891. After the C-891 has been switched on or rebooted, factory default settings are enabled, provided a configuration has not already been carried out with a start-up macro.

Setting Up "Position Distance" Trigger Mode

The *Position Distance* trigger mode lends itself to scanning applications. Once the axis has moved along the distance that was set with CTO parameter ID = 1 (TriggerStep), a trigger pulse is output. The pulse width is one servo cycle (50 µs).

The distance (TriggerStep) is to be specified in the axis unit. Default is mm.

1. Configure the digital output line (<TrigOutID>) that is to be used as the trigger output:
   - Send `CTO <TrigOutID> 2 A`, where *A* indicates the axis to be moved.
   - Send `CTO <TrigOutID> 3 0`, where 0 specifies the *Position Distance* trigger mode.
   - Send `CTO <TrigOutID> 1 S`, where *S* indicates the distance.
2. If you want to enable the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example:

A pulse on digital output line 1 is output every time the axis 1 of the stage has covered a distance of 0.1 µm.

➢ Send:

```
CTO 1 2 1
CTO 1 3 0
CTO 1 1 0.0001
TRO 1 1
```

"Position Distance" trigger mode with start and stop values for positive motion direction of the axis

Optionally, you can define start and stop values for limiting the range and for specifying the motion direction of the axis (positive or negative).

INFORMATION

If start and stop values have the same value, they are ignored.
If the direction of motion is reversed before the axis position has reached the stop value, trigger pulses continue to be output.

1. Configure the digital output line (<TrigOutID>) that is to be used as the trigger output:
   - Send `CTO <TrigOutID> 2 A`, where *A* indicates the axis to be moved.
   - Send `CTO <TrigOutID> 3 0`, where 0 specifies the *Position Distance* trigger mode.
   - Send `CTO <TrigOutID> 1 S`, where *S* indicates the distance.
   - Send `CTO <TrigOutID> 8 Start`, where *Start* indicates the start value.
   - Send `CTO <TrigOutID> 9 Stop`, where *Stop* indicates the stop value.
2. If you want to enable the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example

A pulse on digital output line 1 is output every time the axis 1 of the stage has covered a distance of 0.1 µm, as long as axis 1 is moving in positive direction of motion within the range of 0.2 µm to 0.55 µm (start value < stop value).

➢ Send:

```
CTO 1 2 1
CTO 1 3 0
CTO 1 1 0.0001
```

```
CTO 1 8 0.0002
CTO 1 9 0.00055
TRO 1 1
```

"Position Distance" trigger mode with start and stop values for negative motion direction of the axis

The above example is presented with interchanged start and stop values in the following. Triggering occurs in negative motion direction of the axis (stop value < start value) in the range between 0.55 µm and 0.2 µm.

Example:

➢ Send:

```
CTO 1 2 1
CTO 1 3 0
CTO 1 1 0.0001
CTO 1 8 0.00055
CTO 1 9 0.0002
TRO 1 1
```

Setting Up "On Target" Trigger Mode

In the *On Target* trigger mode, the on-target state of the axis selected is output at the selected trigger output.

The on-target state is true when the current position is inside the settling window and stays there at least for the duration of a delay time. The delay time is given by the ***Settling Time [s]*** parameter (ID 0x3F), and the settling window is given by the ***Settling Window [Phys. Unit]*** parameter (ID 0x7000900). If the settling window is set too large, the control loop may become instable.

1. Configure the digital output line (<TrigOutID>) that is to be used as the trigger output:
   - Send `CTO <TrigOutID> 2 A`, where *A* indicates the axis to be moved.
   - Send `CTO <TrigOutID> 3 2`, where *2* specifies the *On Target* trigger mode.
2. If you want to enable the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example:

The on-target state of axis 1 is to be output on the digital output line 1.

➢ Send:

```
CTO 1 2 1
CTO 1 3 2
TRO 1 1
```

Setting Up "MinMax Threshold" Trigger Mode

In *MinMax Threshold* trigger mode, the selected digital output line is active when the position of the selected axis is within a specified band.

The position limits of the band are to be specified in the axis unit. Default is mm.

1. Configure the digital output line (<TrigOutID>) that is to be used as the trigger output:
   - Send `CTO <TrigOutID> 2 A`, where *A* indicates the axis to be moved.
   - Send `CTO <TrigOutID> 3 3`, where 3 specifies the *MinMax Threshold* trigger mode.
   - Send `CTO <TrigOutID> 5 Min`, where *Min* indicates the position value for the lower limit of the band.
   - Send `CTO <TrigOutID> 6 Max`, where *Max* indicates the position value for the upper limit of the band.
2. If you want to enable the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example:

Digital output line 1 is to be active if the current position of axis 1 of the stage is in the range between 0.2 mm and 0.55 mm. The unit of the position is millimeters.

➢ Send:

```
CTO 1 2 1
CTO 1 3 3
CTO 1 5 0.2
CTO 1 6 0.55
TRO 1 1
```

Setting Up "Motion Error" Trigger Mode

The *Motion Error* trigger mode lends itself to monitoring motions. The selected digital output line becomes active when a motion error occurs on one of the connected axes. The line stays active until the error code is reset to 0 (by a query with `ERR?`).

INFORMATION

A motion error occurs when the current position differs too much from the commanded position during the motion. For further information, see "Motion Error" (p. 36).

1. Configure the digital output line (<TrigOutID>) that is to be used as the trigger output:
   - Send `CTO <TrigOutID> 3 5`, where 5 specifies the *Motion Error* trigger mode.
2. If you want to enable the conditions for trigger output, send `TRO <TrigOutID> 1`.

Setting Up "In Motion" Trigger Mode

In the *In Motion* trigger mode, the motion state of the selected axis is output at the selected trigger output. The line is active, as long as the selected axis is in motion.

The motion state can also be read with the `#5`, `#4` and `SRG?` commands.

INFORMATION

If the axis is in motion, then bit 13 of the state register 1 of the axis is set.

1. Configure the digital output line (<TrigOutID>) that is to be used as the trigger output:
   - Send `CTO <TrigOutID> 2 A`, where *A* indicates the axis to be moved.
   - Send `CTO <TrigOutID> 3 6`, where 6 specifies the *In Motion* trigger mode.
2. If you want to enable the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example:

Digital output line 1 is to be active if axis 1 of the stage is in motion.

➢ Send:

```
CTO 1 2 1
CTO 1 3 6
TRO 1 1
```

Setting Up "Trigger Servo" Mode

In the *Trigger Servo* mode, the signal level of the selected digital output line changes with every servo cycle; i.e., the signal level toggles between High and Low synchronously with the servo cycles of the C-891.

INFORMATION

To count the servo cycles of the C-891, you have to count all state transitions of the digital output line (low --> high **and** high --> low).

1. Configure the digital output line (<TrigOutID>) that is to be used as the trigger output:
   - Send `CTO <TrigOutID> 3 10`, where 10 specifies the *Trigger Servo* trigger mode.
2. If you want to enable the conditions for trigger output, send `TRO <TrigOutID> 1`.

Setting Signal Polarity

The polarity of the signal at the digital output which is used for triggering can be selected with the *Polarity* CTO parameter. The polarity can have the following values:

- active high = 1 (default setting)
- active low = 0

➢ Configure the digital output line (<TrigOutID>) that is to be used as the trigger output:

– Send `CTO <TrigOutID> 7 P`, where *P* indicates the polarity.

Example:

The signal polarity for digital output line 1 is to be set to active low.

➢ Send:

```
CTO 1 7 0
```

Analog Input Signal

The analog input of the C-891 is available at the **Analog In** socket (p. 120).

The analog input signal can be recorded using the data recorder with the “Analog In Value” record option. Data recorder details can be found in "Data Recorder" in the MS205E User Manual of the C-863.11 Mercury DC Motor Controller.

This way, you can, for example, record the signal of an external sensor. The analog input signal **cannot** be used as feedback for the control algorithm in closed-loop operation.

The analog input signal can be adjusted to the permissible input range (-10 V to 10 V) via the parameters ***Analog Input Amplification*** (ID 0x30A0) and ***Analog Input Offset*** (ID 0x30A1). For more information, see “Adapting Settings” (p. 95).

Wave Generator

Functionality of the Wave Generator

The wave generator of the C-891 is intended to be used as control source for the axis motion in closed-loop operation.

The wave generator outputs the target values for the axis motion on the basis of defined waveforms. The wave generator output is especially suited to dynamic applications with periodic axis motions.

The following block diagram shows the integration of the wave generator in the C-891.

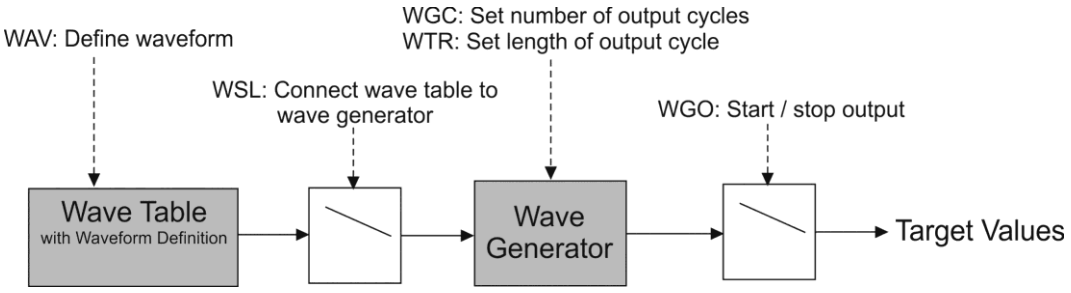


Figure 7: Block diagram of the wave generator

Wave table

Waveform definitions can be temporarily stored in a wave table in the volatile memory of the C-891. The wave table contains the definition of one waveform. A waveform can consist of up to 40 segments which are each to be defined using a separate WAV command (p. 84).

The wave table has to be assigned to the wave generator and thus to the axis (p. 56).

Changing the output cycles

The number of output cycles (p. 57) and the output rate (p. 57) of the wave generator can be set with commands.

Wave generator output

The wave generator outputs absolute target values.

When the wave generator is running for the axis, switching off the servo mode or switching off the motor stops the output.

INFORMATION

It is recommended to use the PI Wave Generator Tool for work with the wave generator (available in PIMikroMove). No command knowledge is necessary to work with PIMikroMove. The use of the PI Wave Generator Tool is described in the PIMikroMove manual (SM148E).

You can permanently save the settings of the wave generator in the C-891 with the macro functionality of the C-891. You can also use a startup macro to configure the wave generator and start the output each time that the C-891 is switched on or rebooted. For the macro functionality of the C-891, see the MS205E User Manual of the C-863.11 Mercury DC Motor Controller.

Commands and Parameters for the Wave Generator

Commands

The following commands are available for using the wave generator:

Command	Syntax	Function
TWG?	TWG?	Gets the number of wave generators (= number of axes).
WAV	WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>	Defines the waveform.
WAV?	WAV? [{<WaveTableID> <WaveParameterID>}]	Gets the current wave table length (number of points).
WGC	WGC {<WaveGenID> <Cycles>}	Sets the number of output cycles.
WGC?	WGC? [{<WaveGenID>}]	Gets the number of output cycles.
WGO	WGO {<WaveGenID> <StartMode>}	Starts and stops the wave generator output.
WGO?	WGO? [{<WaveGenID>}]	Gets the start mode that was last commanded for the wave generator.
WGR	WGR	Starts the data recording again while the wave generator is running.
WSL	WSL {<WaveGenID> <WaveTableID>}	Creates the connection between the wave table and the wave generator.
WSL?	WSL? [{<WaveGenID>}]	Gets the connection between the wave table and the wave generator.
WTR	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}	Sets the wave generator table rate (thus influencing the duration of an output cycle).
WTR?	WTR? [{<WaveGenID>}]	Gets the wave generator table rate.
#9	#9	Gets the current activation state of the wave generator.

Parameter

The following parameters are available for configuring the wave generator:

Parameter	Description and Possible Values
Maximum Number Of Wave Points (ID 0x13000004)	Total number of available points for waveforms The total number of available points for waveforms is 2,000,000. This parameter is write-protected.
Number of Waves (ID 0x1300010A)	Number of wave tables This parameter is write-protected.

Defining the Waveform

Waveforms are defined with the following steps:

- Optional: Getting information on wave table (p. 48)
- Creating a waveform in the wave table (p. 48)

This manual contains examples for creating waveforms (p. 50).

INFORMATION

The wave table content (= waveform definitions) is only present in the volatile memory of the C-891 and is lost when the C-891 is switched off or rebooted.

Optional: Getting information on the wave table

- Send the `SPA? 1 0x13000004` command to get the total number of points that the C-891 provides for defining waveforms.
- Send the `SPA? 1 0x1300010A` command to get the number of wave tables available in the C-891.
- Get the current number of already defined waveform points with the WAV? command (p. 88).

Defining a waveform in the wave table

1. If the wave table is connected to the wave generator, make sure that the wave generator output is not running. For details see "Configuring the Wave Generator" (p. 56) and "Stopping the Wave Generator Output" (p. 58).
2. Define the waveform in the wave table from individual segments with the WAV command (p. 84). Supported curve types:

   “PNT” (user-defined curve)

   "SIN_P" (inverted cosine curve)

   "RAMP" (ramp curve)

   "LIN" (curve in the form of a single scan line)

The waveform definition is written in the wave table in the volatile memory. For details, see "Examples for creating waveforms" (p. 50).

INFORMATION

The waveform points give absolute target positions for closed-loop operation.

- When you define a waveform with `WAV` (p. 84), make sure that the resulting target values do not exceed the travel range limits (can be queried with `TMN?` and `TMX?`).

INFORMATION

The length of the waveform influences the frequency of the wave generator output.

- Define the waveform so that the following conditions are met:
  - The frequency of the wave generator output is lower than the maximum permissible operating frequency of the connected mechanical system (see specifications of the stage).
  - The frequency of the wave generator output is selected so that the motor driver in the C-891 does not overheat (see also “Safety Features of the C-891”, p. 35).
- When you use the PI Wave Generator Tool of PIMikroMove to define waveform segments, the “***Use physical units (s)***” box should be checked. This way, the waveform parameters can be entered in seconds (instead of a number of points), and you do not have to calculate the waveform duration from the servo cycle time and number of points. Note that for the duration of an output cycle, the wave table rate must also be taken into account, see p. 57.

INFORMATION

When the C-891 is switched off or rebooted, the wave table content is automatically deleted.

Examples for creating waveforms

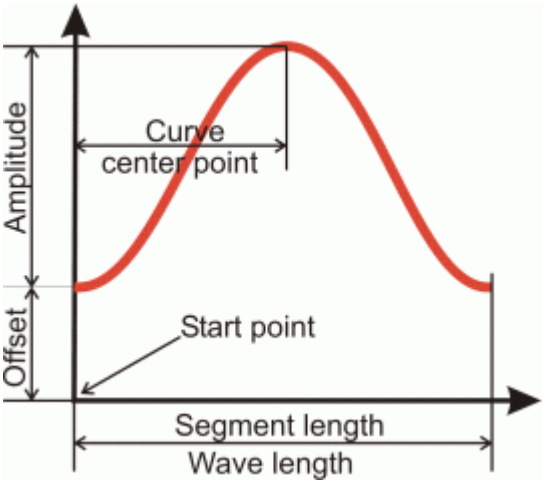
The following examples will help you to define the waveform.

Sine curve 1

- Symmetrical sine curve with offset
- Segment overwrites the wave table content

Command: `WAV 1 X SIN_P 20000 20 10 20000 0 10000`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 20000
<Amp> = 20
<Offset> = 10
<WaveLength> = 20000
<StartPoint> = 0
<CurveCenterPoint> = 10000

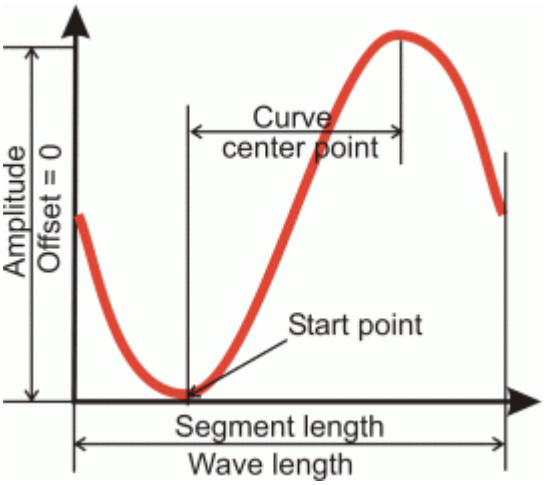


Sine curve 2

- Symmetrical sine curve without offset
- Segment overwrites the wave table content

Command: `WAV 1 X SIN_P 20000 30 0 20000 4999 10000`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 20000
<Amp> = 30
<Offset> = 0
<WaveLength> = 20000
<StartPoint> = 4999
<CurveCenterPoint> = 10000

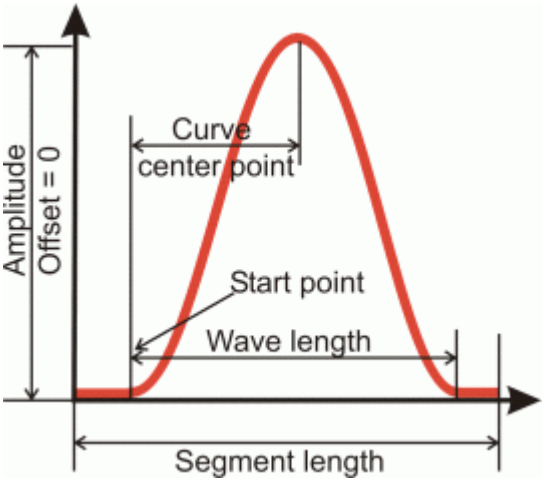


Sine curve 3

- Symmetrical sine curve without offset
- Segment is attached to the wave table content

Command: `WAV 1 & SIN_P 20000 25 0 18000 1000 9000`

<WaveTableID> = 1
<AppendWave> = &
<WaveType> = SIN_P
<SegLength> = 20000
<Amp> = 25
<Offset> = 0
<WaveLength> = 18000
<StartPoint> = 1000
<CurveCenterPoint> = 9000

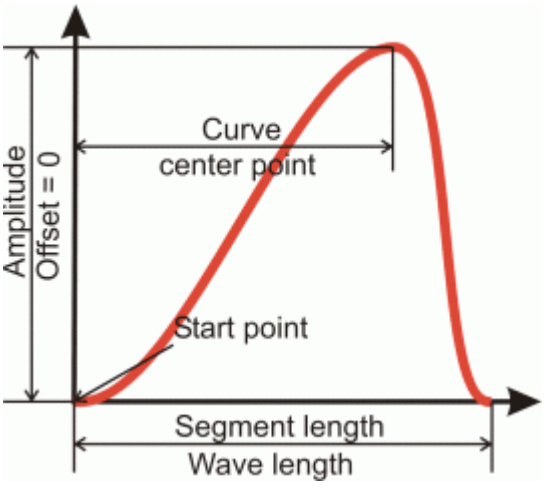


Sine curve 4

- Asymmetrical curve without offset
- Segment overwrites the wave table content

Command: `WAV 1 X SIN_P 40000 20 0 40000 0 31000`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 40000
<Amp> = 20
<Offset> = 0
<WaveLength> = 40000
<StartPoint> = 0
<CurveCenterPoint> = 31000

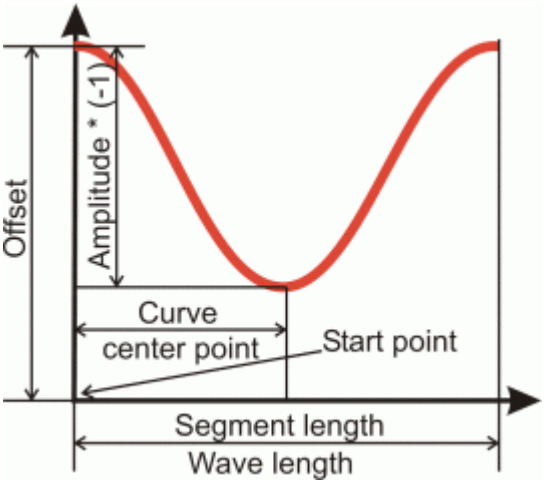


Sine curve 5

- Symmetrical curve with negative amplitude
- Segment overwrites the wave table content

Command: `WAV 1 X SIN_P 10000 -30 45 10000 0 5000`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 10000
<Amp> = –30
<Offset> = 45
<WaveLength> = 10000
<StartPoint> = 0
<CurveCenterPoint> = 5000

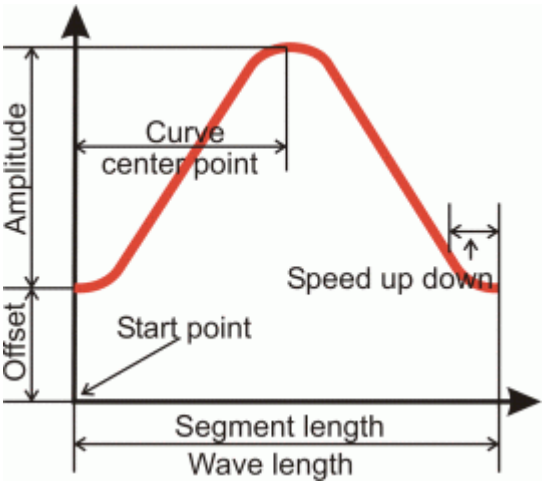


Ramp curve 1

- Symmetrical ramp curve with offset
- Segment overwrites the wave table content

Command: `WAV 1 X RAMP 20000 20 10 20000 0 1000 10000`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = RAMP
<SegLength> = 20000
<Amp> = 20
<Offset> = 10
<WaveLength> = 20000
<StartPoint> = 0
<SpeedUpDown> = 1000
<CurveCenterPoint> = 10000

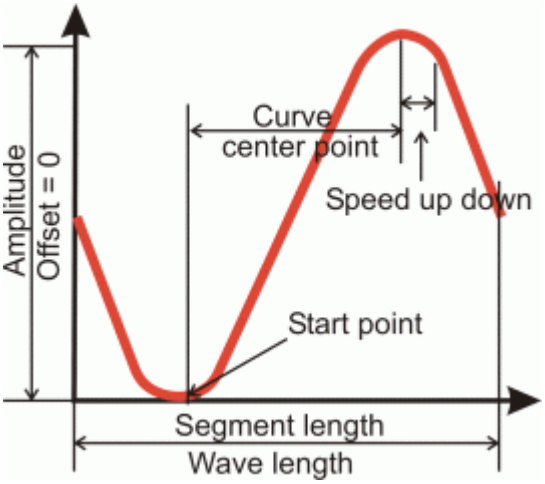


Ramp curve 2

- Symmetrical ramp curve without offset
- Segment overwrites the wave table content

Command: `WAV 1 X RAMP 20000 35 0 20000 4999 1000 10000`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = RAMP
<SegLength> = 20000
<Amp> = 35
<Offset> = 0
<WaveLength> = 20000
<StartPoint> = 4999
<SpeedUpDown> = 1000
<CurveCenterPoint> = 10000

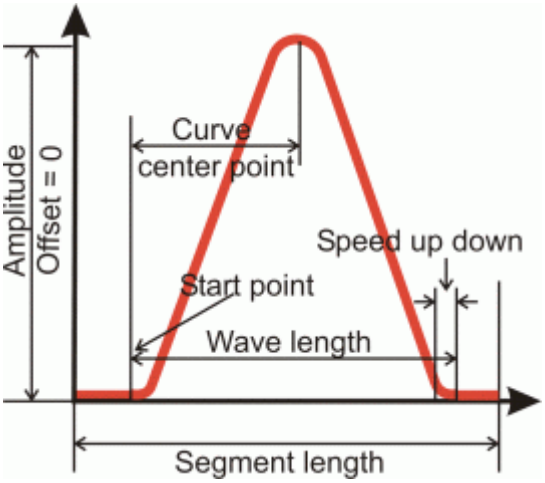


Ramp curve 3

- Symmetrical ramp curve without offset
- Segment overwrites the wave table content

Command: `WAV 1 X RAMP 20000 15 0 18000 1200 500 9000`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = RAMP
<SegLength> = 20000
<Amp> = 15
<Offset> = 0
<WaveLength> = 18000
<StartPoint> = 1200
<SpeedUpDown> = 500
<CurveCenterPoint> = 9000

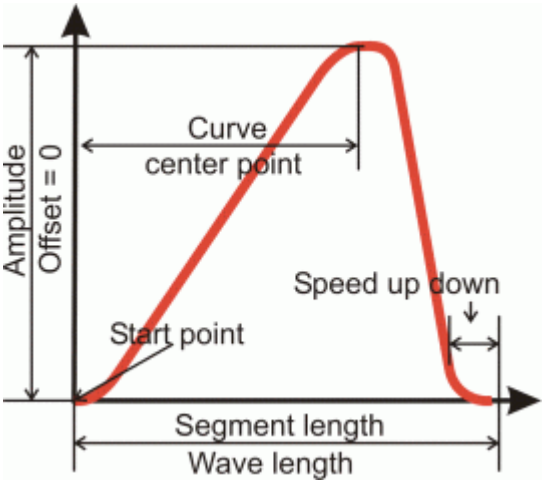


Ramp curve 4

- Asymmetrical ramp curve without offset
- Segment is attached to the wave table content

Command: `WAV 1 & RAMP 30000 35 0 30000 0 2000 22500`

<WaveTableID> = 1
<AppendWave> = &
<WaveType> = RAMP
<SegLength> = 30000
<Amp> = 35
<Offset> = 0
<WaveLength> = 30000
<StartPoint> = 0
<SpeedUpDown> = 2000
<CurveCenterPoint> = 22500

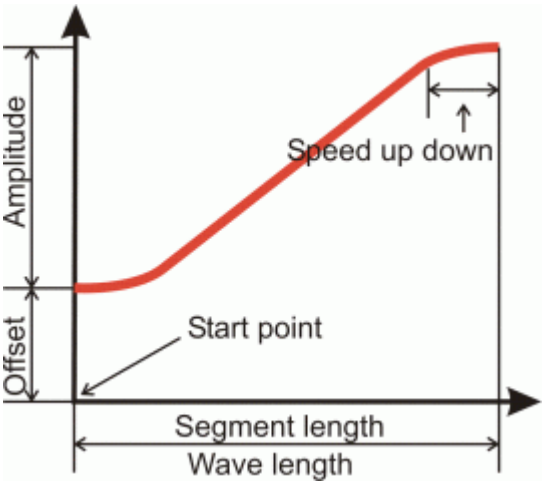


Single scan line 1

- Scan line with offset
- Segment overwrites the wave table content

Command: `WAV 1 X LIN 15000 30 15 15000 0 3700`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = LIN
<SegLength> = 15000
<Amp> = 30
<Offset> = 15
<WaveLength> = 15000
<StartPoint> = 0
<SpeedUpDown> = 3700

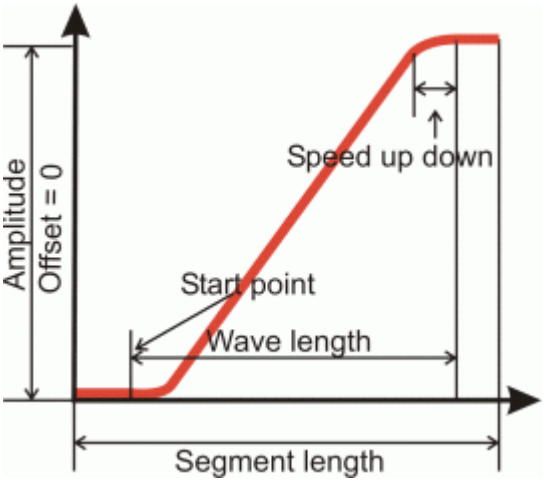


Single scan line 2

- Scan line without offset
- Segment overwrites the wave table content

Command: `WAV 1 X LIN 15000 40 0 11000 2100 1800`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = LIN
<SegLength> = 15000
<Amp> = 40
<Offset> = 0
<WaveLength> = 11000
<StartPoint> = 2100
<SpeedUpDown> = 1800

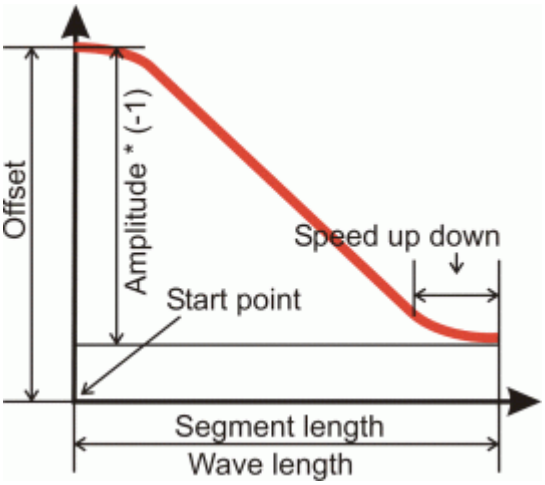


Single scan line 3

- Scan line with negative amplitude
- Segment is attached to the wave table content

Command: `WAV 1 & LIN 30000 -40 50 30000 0 6500`

<WaveTableID> = 1
<AppendWave> = &
<WaveType> = LIN
<SegLength> = 30000
<Amp> = –40
<Offset> = 50
<WaveLength> = 30000
<StartPoint> = 0
<SpeedUpDown> = 6500

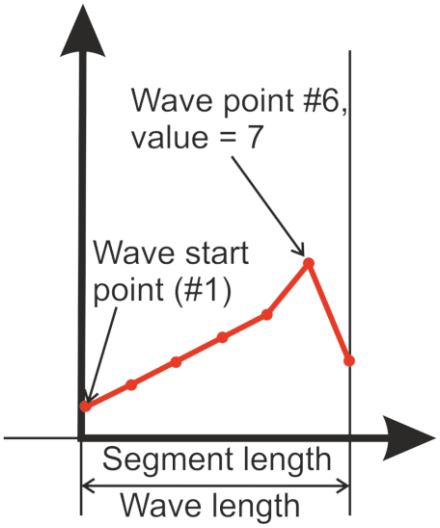


User-defined form

- User-defined curve
- Segment overwrites the wave table content

Command: `WAV 1 X PNT 1 7 1 2 3 4 5 7 3`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = PNT
<WaveStartPoint> = 1
<WaveLength> = 7
<WavePoint> = 1, 2, 3, 4, 5, 7, 3



Configuring the Wave Generator

The wave generator is configured with the following steps:

- Connecting or disconnecting the wave generator and the wave table (p. 56)
- Optional: Setting the number of output cycles (p. 57)
- Optional: Setting the output rate (p. 57)

This manual contains an example for setting the output rate (p. 57).

INFORMATION

The configuration settings for the wave generator can only be changed in the volatile memory of the C-891 and are lost when the C-891 is switched off or rebooted.

Connecting or disconnecting wave generator and wave table

➢ Get the current connection of the wave generator and wave table with the `WSL?` command (p. 93).

➢ Connect or disconnect the wave generator and the wave table:

a) Make sure that the output has not been started for the wave generator. For details see "Stopping the wave generator output" (p. 58).
b) Use the `WSL` command (p. 92) to connect the wave table with the wave generator or terminate the connection of the generator to the wave table.

Optional: Setting the number of output cycles

The factory default setting for the number of output cycles is 0. With the factory setting, the waveform is output without a time limitation until it is stopped with the `WGO` or `#24` or `STP` command.

- Send the `WGC?` command (p. 89) to get the current setting for the number of output cycles of the wave generator.
- Set the number of output cycles of the wave generator with the `WGC` command (p. 89).

INFORMATION

When the number of output cycles is set during the wave generator output, the counting of the output cycles starts when the `WGC` command is sent.

Optional: Setting the wave table rate

The duration of an output cycle for the waveform can be calculated as follows:

Output duration = servo cycle time * wave table rate * number of points

where

the servo cycle time for the C-891 is given by the parameter 0x0E000200 (in seconds),
the wave table rate is the number of servo cycles that the output of a waveform point lasts; the default is 1,
the number of points is the length of the waveform (i.e. the wave table length).

- Send the WTR? command (p. 94) to get the current setting for the wave generator table rate.
- Set the output rate with the WTR command (p. 93)

INFORMATION

Since the individual target points of a waveform are only created when the waveform is output by the wave generator, a wave table rate of 1 is recommended.

Starting and Stopping Output

The wave generator outputs absolute target values for the axis.

- Starting the wave generator output (p. 58)
- Stopping the wave generator output (p. 58)
- Optional: Getting the activation state of the wave generator (p. 59)
- Optional: Starting data recording during the wave generator output (p. 59)

This manual contains examples for starting/stopping the wave generator output (p. 59).

INFORMATION

The wave generator output can only be started in closed-loop operation. When the wave generator is running for the axis, switching off the servo mode or switching off the motor stops the output.

INFORMATION

Depending on the current position of the axis at wave generator start, the axis may perform a jerky movement to reach the start point of the waveform. The jerky movement can cause a motion error of the axis.

- If a motion error occurred, re-establish readiness for operation; see "Safety Features of the C-891" (p. 36) for details.
- Avoid jerky movement: Before you start the wave generator output, move the axis to the position of the waveform start point with an appropriate `MOV` command.

INFORMATION

When the wave generator output is active, motion commands such as MOV are not allowed for the associated axis.

Prerequisites

- ✓ You have defined the desired waveform (p. 48).
- ✓ You have connected the wave generator with the wave table (p. 56).

Starting the wave generator output

1. Switch on the servo mode for the axis with the `SVO` command.
2. Move the axis to the position of the waveform start point with an appropriate `MOV` command. This is strictly recommended to avoid jerky movement to the start point.
3. Start the wave generator output with the `WGO` command (p. 90).

   The output takes place synchronously with the servo cycles of the C-891.

   When the wave generator output is started, a data recording cycle automatically starts.

Stopping the wave generator output

- Stop the wave generator output by sending one of the following commands:
  - `WGO F 0`, where *F* specifies the wave generator and 0 causes stopping (p. 90).
  - `STP`
  - `#24`
  - `HLT`

When the wave generator output is stopped by sending `STP`, `#24` or `HLT`, the C-891 sets the error code 10 (get with the `ERR?` command).

When the number of output cycles has been limited (p. 57), the wave generator output is automatically stopped when the specified number of cycles is reached.

INFORMATION

Exiting the PC software does **not** stop the wave generator output.

Optional: Getting the activation state of the wave generator

- Get whether the wave generator output is running with the `#9` command (p. 67).
- Get the last-commanded start settings of the wave generator with the `WGO?` command (p. 91).

  Stopping the wave generator output with `#24`, `STP` or `HLT` sets the start mode value to zero. The start mode value is also set to zero when the servo mode or the motor are switched off.

Optional: Starting data recording during the wave generator output

- Start the data recording during the wave generator output by sending the `WGR` command (p. 92).

When the wave generator is started, the first data recording cycle will automatically start.

The recorded data can be read out with the `DRR?` command. For more information see "Data Recorder" (MS205E User Manual of the C-863.11 Mercury DC Motor Controller).

Example for starting/stopping the wave generator output

Action	Command	Result
Define a sine curve for wave table 1.	`WAV 1 X SIN P 2000 2 0 2000 0 1000`	The length of the waveform is 2000.
Connect wave generator 1 with wave table 1.	`WSL 1 1`	Prerequisite for wave generator output fulfilled: No wave generator output is possible without allocation of the wave table.
Start wave generator 1.	`WGO 1 1`	The waveform defined in the wave table is output.
Stop wave generator 1.	`WGO 1 0`	The output of the waveform points is stopped.

GCS Commands

Notation

The following notation is used in this chapter to define the GCS syntax and to describe the commands:

<...>	Angle brackets indicate an argument of a command, can be an item identifier or a command-specific parameter
[…]	Square brackets indicate an optional entry
{…}	Curly brackets indicate a repetition of entries, i.e. that it is possible to access more than one item (e.g. several axes) in one command line.
LF	LineFeed (ASCII char #10), is the default termination character (character at the end of a command line)
SP	Space (ASCII char #32), indicates a space character
"..."	Quotation marks indicate that the characters enclosed are returned or to be entered.

GCS Syntax for Syntax Version 2.0

A GCS command consists of 3 characters, e.g. CMD. The corresponding query command has a question mark added to the end, e. g. CMD?.

Command mnemonic:

CMD ::= character1 character2 character3 [?]

Exceptions:

- Single-character commands, e. g. fast query commands, consist only of one ASCII character. The ASCII character is written as combination of # and the character code in decimal format, e. g. as `#24`.
- `*IDN?` (for GPIB compatibility).

The command mnemonic is not case-sensitive. The command mnemonic and all arguments (e. g. axis identifiers, channel identifiers, parameters, etc.) must be separated from each other by a space (SP). The command line ends with the termination character (LF).

CMD[{SP}<Argument>}]LF

CMD?[{SP}<Argument>}]LF

Exception:

- Single-character commands are not followed by a termination character. The response to a single-character commands is followed by a termination character, however.

The argument <AxisID> is used for the logical axes of the controller. Depending on the controller, an axis identifier can consist of up to 16 characters. All alphanumeric characters and the underscore are allowed. See "Commandable Items" (p. 11) for the identifiers supported by the C-891.

Example 1:

Axis 1 is to be moved to position 10.0. The unit depends on the controller (e. g. µm or mm).

Send: `MOV SP 1 SP 10.0 LF`

More than one command mnemonic per line is not allowed. Several groups of arguments following a command mnemonic are allowed.

Example 2:

Two axes which are connected to the same controller are to be moved:

Send: `MOV SP 1 SP 17.3 SP 2 SP 2.05 LF`

When a part of a command line cannot be executed, the line is not executed at all.

When all arguments are optional and are omitted, the command is executed for all possible argument values.

Example 3:

The position of all axes is to be queried.

Send: `POS? LF`

The response syntax is as follows:

[<Argument>[{SP<Argument>}]"="]<Value>LF

With multi-line replies, the space preceding the termination character is omitted in the last line:

{[<Argument>[{SP<Argument>}]"="]<Value>SP LF}

[<Argument>[{SP<Argument>}]"="]<Value>LF for the last line!

In the response, the arguments are listed in the same order as in the query command.

Query command:

CMD?SP<Arg3>SP<Arg1>SP<Arg2>LF

Response to this command:

<Arg3>"="<Val3>SP LF

<Arg1>"="<Val1>SP LF

<Arg2>"="<Val2>LF

Example 4:

Send: `TSP?`SP`2`SP`1`LF

Receive: `2=-1158.4405`SP LF

`1=+0000.0000`LF

INFORMATION

With the C-891 only a single element per command line can be addressed (e.g. axis or parameter).
Example:
By sending command line
`SEP 100 1 0x32 0`
a new value of parameter 0x32 is saved in nonvolatile memory for axis 1,
sending command line
`SEP 100 1 0x32 0 1 0x14 1`
is not possible, however, because two parameters are to be changed.

If the command supports this, all elements can be addressed by omitting the element identifier.
Example:
By sending command line
`SEP?`
all parameters from the nonvolatile memory are queried.

Command Overview

The table below lists the commands supported by the C-891 in alphabetical order. Commands highlighted in grey are described in this user manual. For descriptions of all other commands see the MS205E User Manual of the C-863.11 Mercury DC Motor Controller.

Command	Format	Short description	Details see
#4	#4	Request Status Register	MS205E
#5	#5	Request Motion Status	MS205E
#7	#7	Request Controller Ready Status	MS205E
#8	#8	Query If Macro Is Running	MS205E
#9	#9	Get Wave Generator Status	p. 67
#24	#24	Stop All Axes	MS205E
*IDN	*IDN?	Get Device Identification	MS205E
ACC	ACC {<AxisID> <Acceleration>}	Set Closed-Loop Acceleration	MS205E
ACC?	ACC? [{<AxisID>}]	Get Closed-Loop Acceleration	MS205E
ADD	ADD <Variable> <FLOAT1> <FLOAT2>	Add and Save To Variable	MS205E
BRA	BRA {<AxisID> <BrakeState>}	Set Brake Activation State	MS205E
BRA?	BRA? [{<AxisID>}]	Get Brake Activation State	MS205E
CCL	CCL <Level> [<PSWD>]	Set Command Level	p. 68
CCL?	CCL?	Get Command Level	p. 69
CPY	CPY <Variable> <CMD?>	Copy Into Variable	MS205E
CST?	CST? [{<AxisID>}]	Get Assignment Of Stages To Axes	MS205E
CSV?	CSV?	Get Current Syntax Version	MS205E
CTO	CTO {<TrigOutID> <CTOPam> <Value>}	Set Configuration Of Trigger Output	p. 69
CTO?	CTO? [{<TrigOutID> <CTOPam>}]	Get Configuration Of Trigger Output	p. 72
DEC	DEC {<AxisID> <Deceleration>}	Set Closed-Loop Deceleration	MS205E
DEC?	DEC? [{<AxisID>}]	Get Closed-Loop Deceleration	MS205E
DEL	DEL <uint>	Delay The Command Interpreter	MS205E
DIA?	DIA? [{<MeasureID>}]	Get Diagnosis Information	p. 72
DIO	DIO {<DIOID> <OutputOn>}	Set Digital Output Lines	MS205E
DIO?	DIO? [{<DIOID>}]	Get Digital Input Lines	MS205E
DRC	DRC {<RecTableID> <Source> <RecOption>}	Set Data Recorder Configuration	MS205E
DRC?	DRC? [{<RecTableID>}]	Get Data Recorder Configuration	MS205E
DRL?	DRL? [{<RecTableID>}]	Get Number Of Recorded Points	p. 73

Command	Format	Short description	Details see
DRR?	DRR? [<StartPoint> <NumberOfPoints> [{<RecTableID>}]]	Get Recorded Data Values	MS205E
DRT	DRT {<RecTableID> <TriggerSource> <Value>}	Set Data Recorder Trigger Source	MS205E
DRT?	DRT? [{<RecTableID>}]	Get Data Recorder Trigger Source	MS205E
EAX	EAX {<AxisID> <MotorEnableState>}	Set Motor Enable State	p. 74
EAX?	EAX? [{<AxisID>}]	Get Motor Enable State	p. 75
ERR?	ERR?	Get Error Number	MS205E
FPH	FPH {<AxisID>}	Start Commutation Angle Adjustment Note: Only suitable for 3-phase motors.	p. 75
FPH?	FPH? [{<AxisID>}]	Get Result Of Commutation Angle Adjustment Note: Only suitable for 3-phase motors.	p. 77
FRF	FRF [{<AxisID>}]	Reference Move To Reference Switch Note: Starts also reference moves to limit switches, if parameter 0x70 is set appropriately.	MS205E
FRF?	FRF? [{<AxisID>}]	Get Referencing Result	MS205E
HDI?	HDI?	Get Help For Interpretation Of DIA? Response	p. 78
HDR?	HDR?	Get All Data Recorder Options Note: The response only contains the record options that are relevant for the selected motor type (see p. 12).	MS205E
HLP?	HLP?	Get List of Available Commands	MS205E
HLT	HLT [{<AxisID>}]	Halt Motion Smoothly	MS205E
HPA?	HPA?	Get List Of Available Parameters Note: The response only contains the parameters that are relevant for the selected motor type (see p. 12).	MS205E
HPV?	HPV?	Get List Of Possible Parameter Values	p. 78
JRC	JRC <Jump> <CMD?> <OP> <Value>	Jump Relatively Depending On Condition	MS205E
LIM?	LIM? [{<AxisID>}]	Indicate Limit Switches	MS205E

Command	Format	Short description	Details see
MAC	MAC <keyword> {<parameter>} in particular: MAC BEG <macroname> MAC DEF <macroname> MAC DEF? MAC DEL <macroname> MAC END MAC ERR? MAC NSTART <macroname> <uint> [<String1> [<String2>]] MAC START <macroname> [<String1> [<String2>]]	Call Macro Function	MS205E
MAC?	MAC? [<macroname>]	List Macros	MS205E
MAN?	MAN? {<CMD>}	Get Help String For Command	p. 79
MAT	MAT <Variable> "=" <Float1><OP> <Float2>	Calculate And Save To Variable	p. 80
MEX	MEX <CMD?> <OP> <Value>	Stop Macro Execution Due To Condition	MS205E
MOV	MOV {<AxisID> <Position>}	Set Target Position	MS205E
MOV?	MOV? [{<AxisID>}]	Get Target Position	MS205E
MVR	MVR {<AxisID> <Distance>}	Set Target Relative To Current Position	MS205E
ONT?	ONT? [{<AxisID>}]	Get On-Target State	MS205E
OSM	OSM {<AxisID> <Steps>}	Open-Loop Step Moving Note: Only suitable for stepper motors.	p. 81
OSM?	OSM? [{<AxisID>}]	Get Commanded Steps for Open-Loop Step Moving Note: Only suitable for stepper motors.	p. 81
OSN?	OSN? [{<AxisID>}]	Get Number Of Remaining Open-Loop Steps Note: Only suitable for stepper motors.	p. 82
POS	POS {<AxisID> <Position>}	Set Real Position	MS205E
POS?	POS? [{<AxisID>}]	Get Real Position	MS205E
RBT	RBT	Reboot System	MS205E
RMC?	RMC?	List Running Macros	MS205E
RON	RON {<AxisID> <ReferenceOn>}	Set Reference Mode	MS205E
RON?	RON? [{<AxisID>}]	Get Reference Mode	MS205E
RTR	RTR <RecordTableRate>	Set Record Table Rate	MS205E

Command	Format	Short description	Details see
RTR?	RTR?	Get Record Table Rate	MS205E
SAI	SAI {<AxisID> <NewIdentifier>}	Set Current Axis Identifiers	MS205E
SAI?	SAI? [ALL]	Get List Of Current Axis Identifiers	MS205E
SEP	SEP <Pswd> {<ItemID> <PamID> <PamValue>}	Set Non-Volatile Memory Parameters	MS205E
SEP?	SEP? [{<ItemID> <PamID>}]	Get Non-Volatile Memory Parameters	MS205E
SMO	SMO {<AxisID> <ControlValue>}	Set Open-Loop Control Value Note: Only suitable for 3-phase motors.	MS205E
SMO?	SMO? [{<AxisID>}]	Get Control Value Note: Only suitable for 3-phase motors.	MS205E
SPA	SPA {<ItemID> <PamID> <PamValue>}	Set Volatile Memory Parameters	MS205E
SPA?	SPA? [{<ItemID> <PamID>}]	Get Volatile Memory Parameters	MS205E
SRG?	SRG? {<AxisID> <RegisterID>}	Query Status Register Value	MS205E
STE	STE <AxisID> <Amplitude>	Start Step And Response Measurement	MS205E
STP	STP	Stop All Axes	MS205E
SVO	SVO {<AxisID> <ServoState>}	Set Servo Mode	MS205E
SVO?	SVO? [{<AxisID>}]	Get Servo Mode	MS205E
TCV?	TCV? [{<AxisID>}]	Get Commanded Closed-Loop Velocity	p. 82
TIO?	TIO?	Tell Digital I/O Lines	MS205E
TMN?	TMN? [{<AxisID>}]	Get Minimum Commandable Position	MS205E
TMX?	TMX? [{<AxisID>}]	Get Maximum Commandable Position	MS205E
TNR?	TNR?	Get Number Of Record Tables	MS205E
TRO	TRO {<TrigOutID> <TrigMode>}	Set Trigger Output State	p. 83
TRO?	TRO? [{<TrigOutID>}]	Get Trigger Output State	p. 83
TRS?	TRS? [{<AxisID>}]	Indicate Reference Switch	MS205E
TVI?	TVI?	Tell Valid Character Set For Axis Identifiers	MS205E
TWG?	TWG?	Get Number of Wave Generators	p. 84
VAR	VAR <Variable> <String>	Set Variable Value	MS205E
VAR?	VAR? [{<Variable>}]	Get Variable Value	MS205E
VEL	VEL {<AxisID> <Velocity>}	Set Closed-Loop Velocity	MS205E
VEL?	VEL? [{<AxisID>}]	Get Closed-Loop Velocity	MS205E
VER?	VER?	Get Versions Of Firmware And Drivers	MS205E

Command	Format	Short description	Details see
WAC	WAC <CMD?> <OP> <Value>	Wait For Condition	MS205E
WAV	WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>	Set Waveform Definition	p. 84
WAV?	WAV? [{<WaveTableID> <WaveParameterID>}]	Get Waveform Definition	p. 88
WGC	WGC {<WaveGenID> <Cycles>}	Set Number Of Wave Generator Cycles	p. 89
WGC?	WGC? [{<WaveGenID>}]	Get Number Of Wave Generator Cycles	p. 89
WGO	WGO {<WaveGenID> <StartMode>}	Set Wave Generator Start/Stop Mode	p. 90
WGO?	WGO? [{<WaveGenID>}]	Get Wave Generator Start/Stop Mode	p. 91
WGR	WGR	Starts Recording In Sync With Wave Generator	p. 92
WPA	WPA <Pswd> [{<ItemID> <PamID>}]	Save Parameters To Non-Volatile Memory	MS205E
WSL	WSL {<WaveGenID> <WaveTableID>}	Set Connection Of Wave Table To Wave Generator	p. 92
WSL?	WSL? [{<WaveGenID>}]	Get Connection Of Wave Table To Wave Generator	p. 93
WTR	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}	Set Wave Generator Table Rate	p. 93
WTR?	WTR? [{<WaveGenID>}]	Get Wave Generator Table Rate	p. 94

Command Descriptions

This section describes the commands which are **not** contained in the MS205E User Manual of the C-863.11 Mercury DC Motor Controller. For descriptions of all other commands listed in „Command Overview“ (p. 63), see the MS205E user manual.

#9 (Get Wave Generator Status)

Description: Requests the status of the wave generator(s).

The #9 command can be used to query the current activation state of the wave generators. WGO? gets the last start options commanded for the wave generator.

Format: #9

Arguments: None

Response:	The <uint> response is bit-mapped and output as the hexadecimal sum of the following codes: 1 = Wave generator 1 is active, 2 = Wave generator 2 is active, 4 = Wave generator 3 is active, etc. "Active" = Wave generator output is running
Examples:	0 indicates that no wave generator is running 5 indicates that wave generators 1 and 3 are running

CCL (Set Command Level)

Description:	Changes the active "command level" and thus determines the availability of commands and of write access to system parameters.
Format:	CCL <Level> [<PSWD>]
Arguments:	<Level> is a command level of the controller <PSWD> is the password required for changing to the appropriate command level The following command levels and passwords are valid: Level = 0 is the default setting, all commands provided for "normal" users are available, read access to all parameters, no password required. Level = 1 adds additional commands and write access to level-1 parameters (commands and parameters from level 0 are included). The required password is "advanced". Level > 1 is provided for PI service personnel only. Users cannot change to a level > 1. Contact the customer service department if there seem to be problems with level 2 or higher parameters.
Response:	None
Troubleshooting:	Invalid password
Notes:	With C-891, the command levels only determine the write permission for the parameters. The availability of the commands of the C-891 is independent of the active command level. HPA? lists the parameters including the information about which command level allows write access to them. For

further information on changing parameters, see "Adapting Settings" (p. 95).

After controller switching-on or reboot, the active command level is always level 0.

CCL? (Get Command Level)

Description:	Get the active "command level".
Format:	CCL?
Arguments:	none
Response:	<Level> is the currently active command level; uint.
Notes:	<Level> should be 0 or 1. <Level> = 0 is the default setting, write access is given for level 0 parameters, read access is given for all parameters. <Level> = 1 allows write access for level 1 parameters (parameters from level 0 are included).

CTO (Set Configuration Of Trigger Output)

Description:	Configures the trigger output conditions for the given digital output line.
Format:	CTO {<TrigOutID> <CTOPam> <Value>}
Arguments:	<TrigOutID> is one digital output line of the controller, see below for details. <CTOPam> is the CTO parameter ID in decimal format, see below for the available IDs. <Value> is the value to which the CTO parameter is set, see below.
Response:	None
Notes:	The trigger output conditions will become active when enabled with TRO (p. 83). Do not use DIO on digital output lines for which the trigger output is enabled with TRO. The CTO settings are lost when you power down or reboot the C-891. An easy way to keep them is to save them to a macro.

Output lines and trigger conditions available:	<TrigOutID> corresponds to digital output lines 1 to 4, IDs = 1 to 4; see "Digital I/O" (p. 119). <CTOPam> parameter IDs available for C-891: 1 = TriggerStep 2 = Axis 3 = TriggerMode 5 = MinThreshold 6 = MaxThreshold 7 = Polarity 8 = StartThreshold 9 = StopThreshold <Value> available for the appropriate <CTOPam> ID: for TriggerStep: Distance for Axis: The identifier of the axis to be connected to the digital output line. Irrelevant for the MotionError trigger mode. for TriggerMode (default value is 0): 0 = PositionDistance; a trigger pulse is written whenever the axis has covered the TriggerStep distance (<CTOPam> ID 1). Optionally, values for StartThreshold and StopThreshold (<CTOPam> IDs 8 and 9) can be defined to enable the trigger output for a limited position range and a certain direction of motion only (negative or positive; Note: If the motion direction is reversed before the axis position has reached the stop threshold, trigger pulses will continue to be generated). When StartThreshold and StopThreshold are set to the same value, they will not be used. 2 = OnTarget; the on-target state of the selected axis is transferred to the selected digital output line (this state can also be read with the ONT? command). 3 = MinMaxThreshold; when the position of the selected axis is within the band that is defined by MinThreshold and MaxThreshold (<CTOPam> IDs 5 and 6), the selected digital output line is active. 5 = MotionError; the selected digital output line becomes active when a motion error occurs. The line will stay active until the error code is reset to 0 (by a query).

6 = InMotion;
the selected digital output line is active as long as the selected axis is in motion (the motion state can also be read with commands, e.g. SRG? or #5).

10 = TriggerServo;
the signal level of the selected digital output line changes with every servo cycle; i.e., the signal level toggles between High and Low synchronously with the servo cycles of the C-891.

for MinThreshold/MaxThreshold: Position value; used for the MinMaxThreshold trigger mode; both values must be set to form a band

for Polarity (default value is 1): sets the signal polarity for the digital output line
0 = Active Low
1 = Active High

for StartThreshold/StopThreshold: Position value;
used for the PositionDistance trigger mode; both thresholds must be set in order to determine the position range and the direction of motion for the trigger output

For application examples and further details, see "Digital Output Signals" (p. 39) and the lines below.

Example 1:

A pulse is to be generated on digital output line 1 (ID 1) whenever axis 1 has covered a distance of 0.05 µm. The following parameters must be set:

TrigOutID = 1
Axis = 1
TriggerMode = 0
TriggerStep = 0.05
Send: `CTO 1 2 1`
Send: `CTO 1 3 0`
Send: `CTO 1 1 0.00005`

Example 2: In this example, the digital output line 1 shall be set from low to high when axis A starts its motion. The following parameters must be set:
TrigOutID = 1
Axis = A (axis identifier was changed with `SAI`)
TriggerMode = 6
Polarity = Active High
So you have to send:

```
CTO 1 2 A
CTO 1 3 6
CTO 1 7 1
```

CTO? (Get Configuration Of Trigger Output)

Description: Gets the values set for specified trigger output lines and parameters.

Format: CTO? [{<TrigOutID> <CTOPam>}]

Arguments: <TrigOutID>: is a digital output line of the controller; see CTO.

<CTOPam>: parameter ID; see CTO.

If all arguments are omitted, the response contains the values for all parameters and all output lines.

Response: {<TrigOutID> <CTOPam>"="<Value> LF}

For <Value> see CTO.

DIA? (Get Diagnosis Information)

Description: Gets the current value of a specified measurand.

If all arguments are omitted, the current value of all measurands is queried.

Format: DIA? [{<MeasureID>}]

Arguments: <MeasureID> is the identifier of one measurand, see below for details.

Response: {<MeasureID>"="<MeasuredValue> LF}

where

<MeasuredValue> gives the current value of the measurand, see below for details.

Notes: Use the response to HDI? (p. 78) to get descriptions and physical units of the supported measurands.

C-891 supports the following measurands:

<MeasureID>	<Description> (get with HDI?)
1	Intermediate circuit voltage in volts
2	Servo load in percent (processor utilization caused by servo control)
3	Coil temperature in °C (only shown if overtemp protection of the motor is enabled, see p. 38 for details)
4	I^2T value phase I in percent
5	I^2T value phase II in percent
6	I^2T value phase III in percent
7	I^2T value phase IV in percent
8	Bootloader version
9	Hardware revision
10	TW8 Interpolation Factor
11	Amplifier temperature in °C
12	EtherCAT status (only available with C-891.132300)
13	EtherCAT DSM information (only available with C-891.132300)

DRL? (Get Number of Recorded Points)

Description: Reads the number of points comprised by the last recording.

Format: DRL? [{<RecTableID>}]

Arguments: <RecTableID> is one data recorder table of the controller

Response:	{<RecTableID>"="<uint> LF} where <uint> gives the number of points recorded with the last recording
Notes:	The number of points is reset to zero for a data recorder table when changing its configuration with DRC.

EAX (Set Motor Enable State)

Description:	Sets the motor enable state the of the given axis (motor on/off switch).
Format:	EAX {<AxisID> <MotorEnableState>}
Arguments:	<AxisID> is one axis of the controller <MotorEnableState> can have the following values: 0 = motor off (axis motion disabled) 1 = motor on (axis motion enabled)
Response:	None
Troubleshooting:	Illegal axis identifier
Notes:	Switching off the motor also switches off the servo mode and – if present – activates the brake. As long as the motor is switched off, the following applies: ▪ No move commands are accepted. ▪ The servo mode cannot be switched on. ▪ The brake can be deactivated: Secure the stage against unintentional motions before you deactivate the brake with the BRA command! In the following cases, the motor is switched off automatically: ▪ A motion error occurred (error code -1024, see p. 36) ▪ Safety stop is configured appropriately and was activated (see p. 37) ▪ Motor temperature too high, see “Overtemp Protection of the Motor” (p. 38) ▪ Motor current to high, see “Overcurrent Protection of the Motor” (p. 38) The C-891 can be configured with the ***Motor Power At Startup*** parameter (ID 0x3080) so that the motor is automatically switched on upon switch-on or rebooting.

EAX? (Get Motor Enable State)

Description:	Gets the motor enable state (motor on/off state) for the axis specified. If all arguments are omitted, gets the state of all axes.
Format:	EAX? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response:	{<AxisID>"="<MotorEnableState> LF} where <MotorEnableState> is the current motor enable state for the axis: 0 = motor off (axis motion disabled) 1 = motor on (axis motion enabled)
Troubleshooting:	Illegal axis identifier

FPH (Start Commutation Angle Adjustment)

Description: Starts a commutation angle adjustment. As a result, a commutation offset is set to match the sensor position and electrical motor position to one another.

Depending on the value of the ***Phase Finding (PMSM)*** parameter (ID 0x3053), the axis moves as follows during the adjustment procedure:

- 0 = Type A (default): The axis moves one full electrical motor cycle (distance specified by parameter 0x3050).
- 1 =Type B: The motion can extend over the entire travel range.
- 2 = Type C: Only small motion, i.e., the axis moves by a maximum of a few millimeters.

The resulting commutation offset is set as parameter in volatile memory, details see below, and can be queried with FPH? (p. 77).

The adjustment procedure can be stopped with #24 or STP.

The adjustment procedure can take several seconds.

Format:	FPH {<AxisID>}
Arguments:	<AxisID> is one axis of the controller.
Response:	none
Notes:	FPH can only be used for stages with 3-phase motors (for further details on motor type selection, see p. 12).

If the C-891 and the stage are supplied as a pre-configured system, the commutation offset is already set to a suitable value, and an adjustment procedure is to be performed only in the following cases:

- One of the system components has been replaced
- Stages with incremental sensor: The reference signal type to be used for reference moves has been changed, and therefore the reference position has changed. For further details, see “Reference Moves” (p. 13).

The servo mode must be switched off when a commutation angle adjustment procedure is to be performed. The sequence of steps is as follows:

1. Make sure that the motor is switched on (EAX? returns 1 = 1).
2. If a reference move is necessary: Start the reference move with FRF.
3. Start the adjustment procedure with FPH.
4. Switch the servo mode on with SVO.

FPH configures the data recorder so that the following signals are recorded during the adjustment procedure:

- Record option 2: Current position of the axis
- Record option 111: Amplitude of the phase voltage
- Record option 112: Electrical angle of the rotating field
- Record option 113: Angle of the rotating field, calculated from the sensor position

FPH analyzes the recorded data and sets the value of the ***Phase Correction Offset [Phys. Unit]*** parameter (ID 0x3051) to an appropriate value in volatile memory. If a second encoder is used (***Second Encoder Signal Processing*** (ID 0x30D0) has a value different from 0), the value of the ***Second Encoder Phase Correction Offset (Phys. Unit)*** parameter (ID 0x30D4) is also adapted. If the values are to be preserved when the C-891 is switched off or rebooted,

they have to be saved to nonvolatile memory. See "Adapting Settings" (p. 95), or use the WPA command.

Keep in mind the following when checking the recorded data: The aim of the adjustment procedure is to minimize the phase difference between the sensor position and electrical motor position and hence between the two angle curves (record options 112 and 113). But the data recorded during the procedure may still show a phase difference since the resulting commutation offset is set only at the end of the procedure.

The motion during the adjustment procedure is configured using the following parameters:

- ***Velocity For Reference Moves [mm/s]*** (ID 0x50): Velocity during the move
- ***Motion Motor Voltage [V]*** (ID 0x3060): Amplitude of the phase voltage during the move
- ***Invert Motor Direction (Rotating Field Direction)?*** (ID 0x3066): Direction of rotating field for 3-phase motors. Has to be changed only if the direction of the electrical angle differs from that of the calculated angle of the rotating field. Possible values:
  - 0 = No
  - 1 = Yes

FPH? (Get Result Of Commutation Angle Adjustment)

Description:	Gets the commutation offset that has been set during the last adjustment procedure.
Format:	FPH? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response:	{<AxisID>"="<Offset> LF}

where

<Offset> is the current valid commutation offset value, float. Depending on the value of the ***Commutation Angle Encoder*** parameter (ID 0x30D5), this offset is taken from the volatile value of one of the following parameters:

- 0x30D5 = 0 (main encoder): The response is the value of the ***Phase Correction Offset [Phys. Unit]*** parameter (ID 0x3051).
- 0x30D5 = 1 (second encoder): The response is the value of the ***Second Encoder Phase Correction Offset (Phys. Unit)*** parameter (ID 0x30D4).

Possible offset values are in the range of 0 up to the travel caused by one electrical cycle of the motor. The value -1 indicates that the commutation offset has not been determined yet or that the last adjustment procedure has failed.

Troubleshooting: Illegal axis identifier

Notes: FPH? is only suitable for stages with 3-phase motors (for further details on motor type selection, see p. 12).

HDI? (Get Help For Interpretation Of DIA? Response)

Description: Lists descriptions and physical units for the measurands that can be queried with the DIA? command (p. 72).

Format: HDI?

Arguments: None

Response: {<MeasureID>"="<Description>TAB<PhysUnit> LF}

where

<MeasureID> is the identifier of the measurand.
<Description> is the name of the measurand.
<PhysUnit> is the physical unit of the measurand.

Notes: With C-891, the response to HDI? is as follows:

```
HDI?
1 = BUS Voltage     VOLTS
2 = Servo Load      PERCENT
3 = Coil Temperature       DEG_C
4 = I2T Value PHase I      PERCENT
5 = I2T Value PHase II     PERCENT
6 = I2T Value PHase III    PERCENT
7 = I2T Value PHase IV     PERCENT
8 = Bootloader Version
9 = Hardware Version
10 = TW8 Interpolation Factor
11 = Amplifier Temperature DEG_C
12 = EtherCAT Status
13 = EhterCAT DSM Information
end of help
```

HPV? (Get Parameter Value Description)

Description: Responds with a help string. Use HPA? instead to get a help string, which contains all available parameters with short descriptions.

Format:	HPV?
Arguments:	none
Response:	<string> has the following format: "#HPA_Category enabled" The string is used by the PC software for display purposes.

MAN? (Get Help String For Command)

Description:	Shows a detailed help text for individual commands.
Format:	MAN? <CMD>
Arguments:	<CMD> is the command mnemonic of the command for which the help text is to be displayed (see below).
Response:	A string that describes the command.
Notes:	A detailed help text can be displayed with MAN? for the following commands: CTO, WAV, WGO, WTR Further possibilities to get help: ▪ HLP? lists a help string which contains all commands available. ▪ HPA? responds with a help string which contains all available parameters with short descriptions. ▪ To get help regarding the data recorder commands and parameters, send the HDR? command.

Example:	Send: `MAN? WAV`
	Receive:
	`man? wav`
	`WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters> Set Waveform Definition`
	`#AvailableAppendModes`
	`<AppendWave> <AppendWave_Description>`
	`X clears the wave table and starts writing with the first point in the table`
	`& appends the defined segment to the already existing wave table contents`
	`#AvailableWaveTypeParameters`
	`<WaveType> <WaveTypeParameters>`
	`SIN_P <SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <CurveCenterPoint>`
	`RAMP <SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <SpeedUpDown> <CurveCenterPoint>`

```
LIN <SegLength> <Amp> <Offset> <WaveLength>
<StartPoint> <SpeedUpDown>
end of help
```

MAT (Calculate And Save To Variable)

Description: Carries out a mathematical operation or bit operation and saves the result as a variable.
The variable is present in volatile memory (RAM) only.

Format: MAT <Variable> "=" <FLOAT1> <OP> <FLOAT2>

Arguments: <Variable> is the name of the variable to which the result is to be saved.

<FLOAT1> and <FLOAT2> are the values from which the result is to be calculated. They can be given directly or via the value of a variable.

<OP> is the operator to be used: The following operators are possible:

<OP>	Operation	Type
+	Addition	Mathematical operation
-	Subtraction	Mathematical operation
*	Multiplication	Mathematical operation
AND	AND	Bit operation
OR	OR	Bit operation
XOR	XOR	Bit operation

Important: There must be a blank space before and after each "=" and the operator!

Response: None

Notes: Using MAT to set local variables is only possible in macros.

Example 1: Send: `MAT TARGET = ${POS} * 2.0`
The `TARGET` variable contains 2.0 times the value of the `POS` variable.

Example 2: Send: `MAT TARGET = 2 * 0x10`
Send: `VAR? TARGET`
Receive: `TARGET=32`
NOTICE: The values from which the result is to be calculated can be written in hexadecimal or decimal format. The result is always output in decimal format.

Example 3: Send: `MAT INVERT = 0x45 XOR 0xFF`
Send: `VAR? INVERT`
Receive: `INVERT=186`
NOTICE: The bit operation XOR with the value 0xFF corresponds to an inversion of the value 0x45. The result is output in decimal format.

OSM (Open-Loop Step Moving)

Description: Starts open-loop step motion of the given axis.

Format: OSM {<AxisID> <Steps>}

Arguments: <AxisID> is one axis of the controller.

<Steps> is the number of steps to perform; in float format.

Response: None

Troubleshooting: Incorrect axis ID; incorrect motor type selected; servo mode is switched on for the axis; brake is activated for the axis

Notes: OSM can only be used for stages with stepper motors (for further details on motor type selection, see p. 12).

Fractions of a full step can be commanded with OSM.

Velocity (in physical units/s), acceleration and deceleration (in physical units/s^2) can be set with the VEL, ACC and DEC commands and queried with VEL?, ACC? and DEC?. See also the descriptions of parameters 0x49, 0xA, 0xB, 0x4A, 0xC and 0x4B in “Adapting Settings” (p. 95).

For stages with brake: The brake must be deactivated with BRA before motion is possible with OSM. Secure the stage against unintentional motions before you deactivate the brake with BRA!

With each new OSM command, the number of steps to perform is counted starting from zero (there is no adding up of steps).

The motion can be stopped with #24, STP and HLT.

OSM? (Get Commanded Steps for Open-Loop Step Moving)

Description: Reads the number of steps specified in the last OSM command.

Format:	OSM? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response:	<AxisID>"="<float> LF where <float> gives the last commanded number of steps for open-loop step moving.
Notes:	OSM? is only suitable for stages with stepper motors (for further details on motor type selection, see p. 12).

OSN? (Get Number Of Remaining Open-Loop Steps)

Description:	Reads the number of steps still to be performed after the last OSM command.
Format:	OSN? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response:	<AxisID>"="<float> LF where <float> gives number of steps which are still to be performed.
Notes:	OSN? is only suitable for stages with stepper motors (for further details on motor type selection, see p. 12).

TCV? (Get Commanded Closed-Loop Velocity)

Description:	Gets the current value of the velocity for closed-loop operation (value calculated by the target generator of the C-891).
Format:	TCV? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller.
Response:	{<AxisID>"="<float> LF} where <float> is the velocity value in physical units / s.

TRO (Set Trigger Output State)

Description:	Enables or disables the trigger output conditions which were set with CTO (p. 69) for the given digital output line.
Format:	TRO {<TrigOutID> <TrigMode>}
Arguments:	<TrigOutID> is a digital output line of the controller; see below for further details. <TrigMode> can have the following values: 0 = Trigger output disabled 1 = Trigger output enabled
Response:	None
Troubleshooting:	Illegal identifier of the digital output line
Notes:	<TrigOutID> corresponds to the digital output lines Output 1 to Output 4, IDs = 1 to 4; for further information, see "Digital I/O" (p. 119). Do not use DIO on digital output lines for which the trigger output is enabled with TRO.

TRO? (Get Trigger Output State)

Description:	Returns if the trigger output configuration made with CTO (p. 69) is enabled or disabled for the given digital output line. If all arguments are omitted, gets state of all digital output lines.
Format:	TRO? [{<TrigOutID>}]
Arguments:	<TrigOutID> is one digital output line of the controller, see TRO (p. 83) for more details.
Response:	{<TrigOutID>"="<TrigMode> LF} where <TrigMode> is the current state of the digital output line: 0 = Trigger output disabled 1 = Trigger output enabled
Troubleshooting:	Illegal identifier of the digital output line

TWG? (Get Number of Wave Generators)

Description:	Gets the number of wave generators available in the controller.
Format:	TWG?
Arguments:	None
Response	<uint> is the number of wave generators which are available

WAV (Set Waveform Definition)

Description:	Defines a waveform of given type for given wave table. To allow a flexible definition, a waveform (wave table contents) can be built up by stringing together "segments". Each segment is defined with a separate WAV command. A segment can be added to the existing wave table contents with the <AppendWave> argument (see below). To change individual segments or to modify their order, the complete waveform must be recreated segment-by-segment. A segment is based on predefined "curve" shapes (see the <WaveType> argument below). Waveforms cannot be changed while they are being output by a wave generator. Before a waveform is modified with WAV, the wave generator output from the associated wave table must be stopped first. The waveform values are absolute target values. The duration of one output cycle for the waveform can be calculated as follows: Output Duration = Servo Cycle Time * WTR Value * Number of Points where Servo Cycle Time for the C-891 is given by the parameter 0x0E000200 (in seconds) WTR (wave table rate) value gives the number of servo cycles the output of a waveform point lasts, default is 1 Number of Points corresponds to the wave table length (sum of the lengths of all segments in this table) For more information, see "Wave Generator" (p. 46).
Format:	WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>
Arguments:	<WaveTableID> is the wave table identifier.

<AppendWave> can be "X" or "&":
"X" clears the wave table and starts writing at the first point in the table.
"&" attaches the defined segment to the existing wave table contents in order to extend the waveform.

<WaveType> The type of curve used to define the segment. This can be one of
"PNT" (user-defined curve)
"SIN_P"(inverted cosine curve)
"RAMP" (ramp curve)
"LIN" (single scan line curve)

<WaveTypeParameters> stands for the parameters of the curve:

For "PNT":

<WaveStartPoint> <WaveLength> {<WavePoint>}

<WaveStartPoint>: The index of the starting point. Must be ≥ 1.
<WaveLength>: The number of points to be written in the wave table (= segment length).
<WavePoint>: The value of one single point in the wave table.

For "SIN_P":

<SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <CurveCenterPoint>

<SegLength>: The length of the wave table segment in points. If the <SegLength> value is larger than the <WaveLength> value, the missing points in the segment are filled with the endpoint value of the curve.

<Amp>: The amplitude of the sine curve.

<Offset>: The offset of the sine curve.

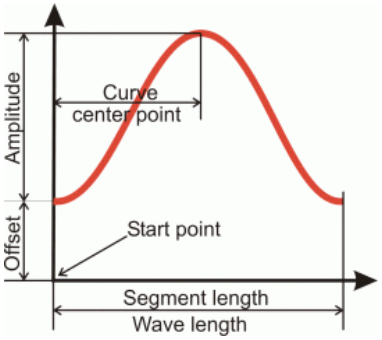
<WaveLength>: The length of the sine curve in points.

<StartPoint>: The index of the starting point of the sine curve in the segment. Gives the phase shift. Lowest possible value is 0.

<CurveCenterPoint>: The index of the center point of the

sine curve. Determines if the curve is symmetrical or not. Lowest possible value is 0.

Example (for further examples, see "Defining the Waveform" (p. 48)):



For "RAMP":

<SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <SpeedUpDown> <CurveCenterPoint>

<SegLength>: The length of the wave table segment in points. If the <SegLength> value is larger than the <WaveLength> value, the missing points in the segment are filled with the endpoint value of the curve.

<Amp>: The amplitude of the ramp curve.

<Offset>: The offset of the ramp curve.

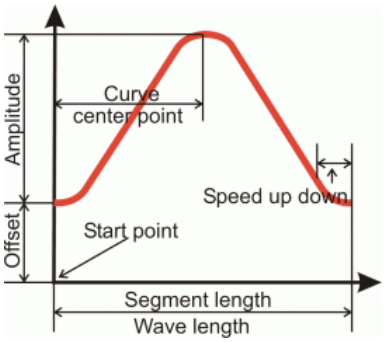
<WaveLength>: The length of the ramp curve in points.

<StartPoint>: The index of the starting point of the ramp curve in the segment. Gives the phase shift. Lowest possible value is 0.

<SpeedUpDown>: The number of points for speed-up and slow-down.

<CurveCenterPoint>: The index of the center point of the ramp curve. Determines if the curve is symmetrical or not. Lowest possible value is 0.

Example (for further examples, see "Defining the Waveform" (p. 48)):



For "LIN":

<SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <SpeedUpDown>

<SegLength>: The length of the wave table segment in points. If the <SegLength> value is larger than the <WaveLength> value, the missing points in the segment are filled with the endpoint value of the curve.

<Amp>: The amplitude of the scan line.

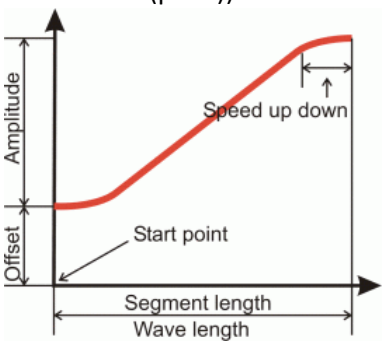
<Offset>: The offset of the scan line.

<WaveLength>: The length of the single scan line curve in points.

<StartPoint>: The index of the starting point of the scan line in the segment. Lowest possible value is 0.

<SpeedUpDown>: The number of points for speed-up and slow-down.

Example (for further examples, see "Defining the Waveform" (p. 48)):



Response:	None
Troubleshooting:	Invalid wave table identifier
Notes:	The waveform points give absolute target positions for closed-loop operation. For that reason, the waveform must be defined so that the resulting target values do **not** exceed the travel range limits (can be queried with `TMN?` and `TMX?`). The frequency of the wave generator output depends, among other factors, on the length of the waveform. The waveform must be selected so that the frequency of the wave generator output is smaller than the maximum permissible operating frequency of the connected mechanical system (see specifications of the stage). When the frequency is too high, the motor driver in the C-891 can also overheat (see also “Safety Features of the C-891”, p. 35).

WAV? (Get Waveform Definition)

Description:	Gets the value of a wave parameter for a given wave table. For more information, see "Wave Generator" (p. 46).
Format:	WAV? [{<WaveTableID> <WaveParameterID>}]
Arguments:	<WaveTableID> is the wave table identifier. <WaveParameterID> is the wave parameter ID: 1 = Current wave table length as a number of points
Response:	{<WaveTableID> <WaveParameterID>"="<float> LF} where <float> depends on the <WaveParameterID>; gives the current number of waveform points in the wave table for <WaveParameterID> = 1
Troubleshooting:	Invalid wave table identifier

WGC (Set Number Of Wave Generator Cycles)

Description:	Sets the number of output cycles for the given wave generator (the output itself is started with WGO (p. 90)). For more information, see "Wave Generator" (p. 46).
Format:	WGC {<WaveGenID> <Cycles>}
Arguments:	<WaveGenID> is the wave generator identifier <Cycles> is the number of wave generator output cycles. If cycles = 0 then the waveform is output without limitation until it is stopped by WGO or #24 or STP.
Response:	None

WGC? (Get Number Of Wave Generator Cycles)

Description:	Gets the number of output cycles set for the given wave generator. For more information, see "Wave Generator" (p. 46).
Format:	WGC? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier
Response:	{<WaveGenID>"="<Cycles> LF} where <Cycles> is the number of wave generator output cycles set with WGC (p. 89).

WGO (Set Wave Generator Start/Stop Mode)

Description:	Starts and stops the specified wave generator. The number of output cycles can be limited by WGC (p. 89). With the WTR command (p. 93), you can lengthen the individual output cycles of the waveform. The wave generator output is also continued when the PC software with which it was started is exited. The #9 command can be used to query the current activation state of the wave generators. For more information, see "Wave Generator" (p. 46).
Format:	WGO {<WaveGenID> <StartMode>}
Arguments:	<WaveGenID> is the wave generator identifier <StartMode> is the start mode for the specified wave generator: 0 = wave generator output is stopped. You can also stop the wave generator output with #24, STP or HLT. 1 = start wave generator output immediately, synchronized by servo cycle. In addition, one data recording cycle is started
Response:	None
Troubleshooting:	Invalid wave generator identifier There is no wave table connected to the wave generator. Connect the wave table with WSL (p. 92). Wave generator output and motion commands: When the wave generator output is active, motion commands such as MOV are not allowed for the associated axis.
Notes:	The wave table contains a waveform definition, but the individual points of the waveform are created only when the waveform is output by the wave generator or when it is queried with the GWD? command. The wave generator output can only be started in closed-loop operation. When the wave generator is running for the axis, switching off the servo mode or switching off the motor stops the output.

Depending on the current position of the axis at wave generator start, the axis may perform a jerky movement to reach the start point of the waveform. The jerky movement can cause a motion error of the axis.

- If a motion error occurred, re-establish readiness for operation, see “Safety Features of the C-891” (p. 35) for details.
- Avoid jerky movement: Before you start the wave generator output, move the axis to the position of the waveform start point with an appropriate MOV command.

WGO? (Get Wave Generator Start/Stop Mode)

Description:	Gets the start/stop mode of the given wave generator. The #9 command can be used to query the current activation state of the wave generators. For more information, see "Wave Generator" (p. 46).
Format:	WGO? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier
Response:	{<WaveGenID>"="<StartMode> LF} where <StartMode> is the last commanded start mode of the wave generator.
Notes:	#24, STP and HLT stop the wave generator output and set the start mode value to zero.

WGR (Starts Recording In Sync With Wave Generator)

Description:	Starts the data recording when the wave generator is active. The data recorder can be configured with DRC. The recorded data can be read with DRR?. For more information, see "Wave Generator" (p. 46) and "Data Recorder" (MS205E User Manual of the C-863.11 Mercury DC Motor Controller).
Format:	WGR
Arguments:	None
Response:	None
Notes:	Starting the wave generator output with WGO starts an initial data recording cycle at the same time. For further trigger options for starting the data recording, see DRT.

WSL (Set Connection Of Wave Table To Wave Generator)

Description:	Wave table selection: connects a wave table to a wave generator or disconnects the selected generator from any wave table. As long as a wave generator is running, it is not possible to change its wave table connection. For more information, see "Wave Generator" (p. 46).
Format:	WSL {<WaveGenID> <WaveTableID>}
Arguments:	<WaveGenID> is the wave generator identifier <WaveTableID> is the wave table identifier. If <WaveTableID> = 0, the selected generator is disconnected from any wave table.
Response:	None

WSL? (Get Connection Of Wave Table To Wave Generator)

Description:	Gets current wave table connection settings for the specified wave generator. For more information, see "Wave Generator" (p. 46).
Format:	WSL? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier
Response:	{<WaveGenID>"="<WaveTableID> LF} where <WaveTableID> is the wave table identifier. If <WaveTableID> = 0, no wave table is connected to the wave generator.

WTR (Set Wave Generator Table Rate)

Description:	Sets wave generator table rate and interpolation type.
Format:	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}
Arguments:	<WaveGenID> is the wave generator identifier. See below for details. <WaveTableRate> is the wave generator table rate (unit: number of servo cycles); must be an integer value that is greater than zero <InterpolationType> is the interpolation type. When a wave generator table rate higher than 1 is set, this option can be used to apply interpolation to the wave generator output between wave table points. The following interpolation types can be selected: 0 = no interpolation 1 = straight line (default)
Response:	None

Notes:	With the WTR command, the individual output cycles of the waveform can be lengthened. The duration of an output cycle for the waveform can be calculated as follows: Output Duration = Servo Cycle Time * WTR Value * Number of Points where the servo cycle time for the C-891 is given by the parameter 0x0E000200 (in seconds) WTR value gives the number of servo cycles the output of a waveform point lasts, default is 1 Number of Points is the length of the waveform (i.e. the length of the wave table) Since the individual target points of a waveform are only created when the waveform is output by the wave generator, a wave table rate of 1 is recommended. For more information, see "Wave Generator" (p. 46).

WTR? (Get Wave Generator Table Rate)

Description:	Gets the current wave generator table rate and the used interpolation type. For more information, see "Wave Generator" (p. 46).
Format:	WTR? [{<WaveGenID>}]
Arguments:	<WaveGenID> is the wave generator identifier
Response:	{<WaveGenID>"="<WaveTableRate> <InterpolationType> LF} where <WaveTableRate> is the wave generator table rate (unit: number of servo cycles) <InterpolationType> is the interpolation type applied to outputs between wave table points when the output rate is higher than 1. For available interpolation types, see WTR (p. 93).

Adapting Settings

Parameters reflect the properties of the connected stage (e.g. travel range) and specify the behavior of the C-891 (e.g. settings for the safety features).

Detailed information can be found in the "Adapting Settings" chapter of the MS205E User Manual of the C-863.11 Mercury DC Motor Controller.

Saving Parameter Values in a Text File

INFORMATION

The C-891 is configured via parameters, e.g., for adaptation to the mechanics connected. Changing parameter values can cause undesirable results.

- Create a backup copy on the PC before changing the parameter settings of the C-891. You can then restore the original settings at any time.
- Create an additional backup copy with a new file name each time after optimizing the parameter values or adapting the C-891 to a particular stage.

INFORMATION

Parameter values saved in a text file on the PC can be loaded back to the nonvolatile memory of the C-891 in PIMikroMove or PITerminal. The ***Send file...*** button is available for this purpose in the send command window. Before loading into the C-891, the individual lines of the text files must be converted into command lines that contain the corresponding SEP commands.

1. Establish communication with PIMikroMove or PITerminal between the C-891 and the PC.
2. If you use PIMikroMove, open the window for sending commands:
   - In the main window select the ***Tools > Command entry*** menu item or press the F4 key on the keyboard.

   In PITerminal the main window from which commands can be sent is opened automatically after establishing communication.
3. Get the parameter values from the nonvolatile memory of the C-891:
   - Send the `SEP?` command.
4. Click on the ***Save...*** button.

   The ***Save content of terminal as textfile*** window opens.
5. In the ***Save content of terminal as textfile*** window, save the queried parameter values in a text file on your PC.

Changing Parameters

NOTICE

Improper parameter settings!
The values in the nonvolatile memory are loaded to the volatile memory as default values when the C-891 is switched on or rebooted and take effect immediately. Improper parameter settings can cause damage to the connected stage.

- Only change parameter values after careful consideration.
- Save the current parameter values to the PC before you perform changes in the nonvolatile memory, see “Saving Parameter Values in a Text File” (p. 95).

INFORMATION

The write access for the parameters of the C-891 is defined by command levels (“CCL”). After the controller is switched on or rebooted, the active command level is always level 0. For particular parameters, write access is only allowed on command level 1. On command levels > 1, write access is only available to PI service personnel.

- If necessary, send the `CCL 1 advanced` command or enter the password `advanced` to change to command level 1.
- Contact the customer service department if there seem to be problems with parameters of command level 2 or higher (p. 121).

INFORMATION

In the PC software from PI, you can load a suitable parameter set from a stage database to the nonvolatile memory of the C-891 (p. 27). Note that the stage database does **not** contain values for system parameters (Item Type = System in the parameter list on p. 99).

- Load a parameter set only if the C-891 and the stage are **not** supplied as a pre-configured system, or if you have replaced one of the system components.
- If you are not sure if your system is pre-configured, save the parameter values from the nonvolatile memory on the PC before you load a parameter set. See “Saving Parameter Values in a Text File” (p. 95).
- After you have loaded a parameter set, perform a commutation angle adjustment using the FPH command (p. 75).

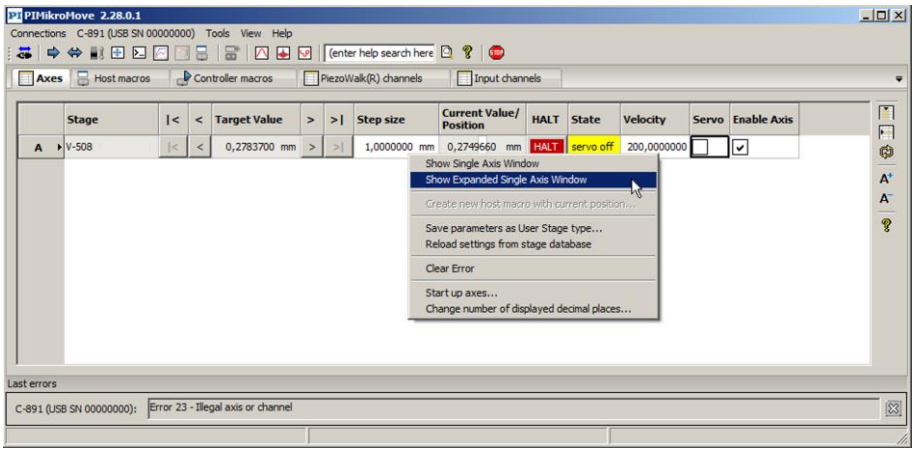
To check and change parameter values, you can use PIMikroMove.

Note that in the parameter display of PIMikroMove, the designation ***Active Value*** is used for the parameter values in the volatile memory and ***Startup Value*** is used for the parameter values in the nonvolatile memory. If a parameter value in the ***Active Value*** column is different from the parameter value in the ***Startup Value*** column, the line in the parameter list is highlighted in color.

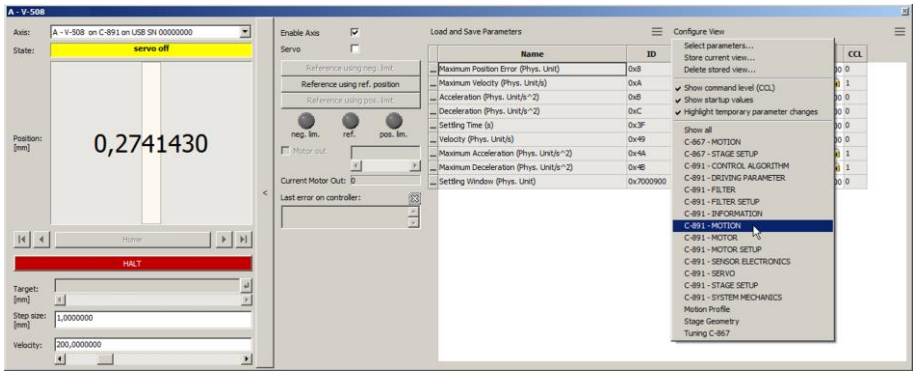
1. Display the parameter list in PIMikroMove.

   If you want to change the axis-related parameter of the C-891:

   a) In the main window of PIMikroMove, open the expanded single axis window for the connected stage by clicking the right mouse button on the corresponding line of the ***Axes*** tab and selecting ***Show Expanded Single Axis Window*** in the context menu.

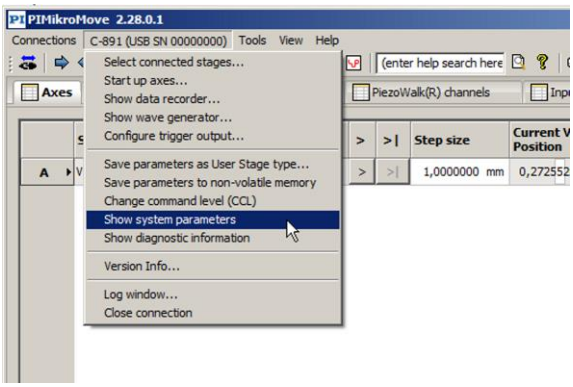


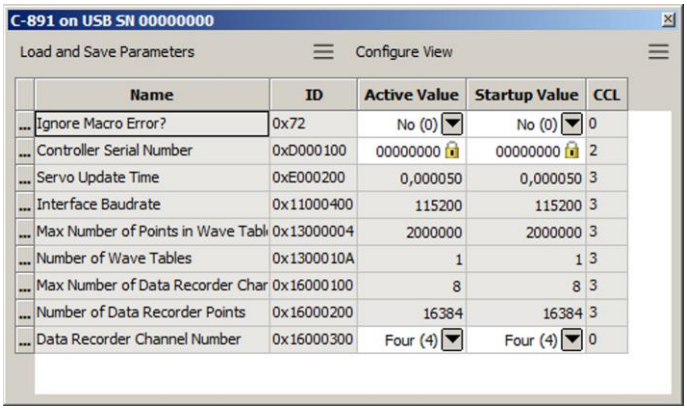
   b) If the parameter to be modified is not included in the list on the right side of the window, click on ***Configure View > Select parameters...*** and add it to the list. You can also display certain groups of parameters or all axis-related parameters.



If you want to change the system-related parameters of the C-891:

- In the main window of PIMikroMove, open the window for the system-related parameters of the C-891 by selecting ***C-891> Show system parameters*** in the menu.





2. In the corresponding parameter list, change the desired parameter values in the volatile or nonvolatile memory of the C-891.

   If you want to change parameter values in the volatile memory, you have the following options:

   - Enter values: Type the new parameter value into the corresponding input field in the ***Active Value*** column of the list. Press the Enter key on the PC keyboard or click with the mouse outside the input field to transfer the parameter value to the volatile memory of the C-891.
   - Load values from nonvolatile memory: Click on ***Load and Save Parameters -> Load all startup parameters of the axis / system from controller*** in order to load the values of all axis-related / system-related parameters from the nonvolatile memory of the C-891.
   - Load values from stage database: In the extended single-axis window, click ***Load and Save Parameters > Load parameters from stage database…*** to load a selected parameter set from the stage database for the axis. You can use ***Load and Save Parameters > Reload parameters from stage database…*** to reload the currently loaded parameter set.

If you want to change parameter values in the nonvolatile memory, you have the following options:

- Enter values: Type the new parameter value into the appropriate input field in the ***Startup Value*** column of the list. Press the Enter key on the PC keyboard or click with the mouse outside the input field to transfer the parameter value to the nonvolatile memory of the C-891.
- Save values from volatile memory to nonvolatile memory: Click on ***Load and Save Parameters -> Save all currently active axis / system parameters as startup parameters to controller*** to write the values of all axis-related / system-related parameters from the volatile to the nonvolatile memory of the C-891. You can skip parameters for which there is no write access on the current command level.

Parameter Descriptions

The table below gives short descriptions of the parameters of the C-891.

For the parameters highlighted in grey, the parameter value can be set in nonvolatile memory only (with SEP). Changes become active only after a reboot of the C-891. The current active value can be read with the SPA? command.

The parameter group assignment is used in the PC software, e.g. in PIMikroMove, for a clearer display. See „Changing Parameters“ (p. 96) for examples.

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x3000	0	Axis	FLOAT	Control Algorithm	P Term (Kp)	See diagrams of control algorithm, p. 15
0x3001	0	Axis	FLOAT	Control Algorithm	I Term (Tn)	
0x3002	0	Axis	FLOAT	Control Algorithm	D Term (Tv)	
0x3003	0	Axis	FLOAT	Control Algorithm	PT1 Position Control (Tp)	
0x3004	0	Axis	FLOAT	Control Algorithm	Limit Control Out	Gives range limits for the output of the position control
0x3005	0	Axis	FLOAT	Control Algorithm	Position Velocity Gain	See diagrams of control algorithm, p. 15
0x3010	0	Axis	FLOAT	Control Algorithm	P Velocity Control (Kp)	See diagrams of control algorithm, p. 15
0x3011	0	Axis	FLOAT	Control Algorithm	I Velocity Control (Tn)	
0x3012	0	Axis	FLOAT	Control Algorithm	D Velocity Control (Tv)	
0x3013	0	Axis	FLOAT	Control Algorithm	PT1 Velocity Control (Tp)	

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x3014	0	Axis	FLOAT	Control Algorithm	Limit Velocity Control	Gives range limits for the output of the velocity control
0x301A	0	Axis	FLOAT	Control Algorithm	Feed-Forward Position	See diagrams of control algorithm, p. 15
0x301B	0	Axis	FLOAT	Control Algorithm	Feed-Forward Velocity	
0x301C	0	Axis	FLOAT	Control Algorithm	Feed-Forward Acceleration	
0x301D	0	Axis	INT	Control Algorithm	Controller Structure	Structure of the control algorithm, diagrams see p. 15 0 = POS/VEL I = parallel structure of position and velocity control, with feed forward (default) 1 = POS-VEL II = serial structure of position and velocity control, with feed forward 2 = POS only 3 = VEL only
0x30B0	0	Axis	INT	Control Algorithm	Target Control Window	Automatic adaptation of the servo-control parameters if the current position is in a certain window around the target position 0 = Disabled 1 = Enabled The window limits are specified by parameter 0x30B1.
0x30B1	0	Axis	FLOAT	Control Algorithm	Target Control Window (Phys. Unit)	Specifies the target control window limits The parameter value gives half the window width. The window is only used if parameter 0x30B0 has the value 1.
0x30B2	0	Axis	FLOAT	Control Algorithm	P Term (Kp1) Position Target Control-Window	Servo-control parameters at target position. The parameters are only relevant if the target control window is used (i.e., if parameter 0x30B0 has the value 1). When the current position is inside the target control window, the C-891 adapts the servo-control parameters linearly between the values of the general parameters (0x3000 to 0x3002, 0x3010 to 0x3012) and the values of parameters 0x30B2 to 0x30B7. See p. 19 for further information.
0x30B3	0	Axis	FLOAT	Control Algorithm	I Term (Tn1) Position Target Control-Window	
0x30B4	0	Axis	FLOAT	Control Algorithm	D Term (Tv1) Position Target Control-Window	

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x30B5	0	Axis	FLOAT	Control Algorithm	P Term (Kp1) Velocity Target Control-Window	
0x30B6	0	Axis	FLOAT	Control Algorithm	I Term (Tn1) Velocity Target Control-Window	
0x30B7	0	Axis	FLOAT	Control Algorithm	D Term (Tv1) Velocity Target Control-Window	
0x3020	0	Axis	FLOAT	Filter Setup	Velocity Cut off Frequency (3dB) (Hz)	Cutoff frequency of the velocity filter (3 dB limit)
0x3021	0	Axis	INT	Filter Setup	Velocity Filter Type	Structure of the velocity filter: 0 = OFF 1 = IIR 2. Ord 2 = IIR 4. Ord 3 = IIR fast 4 = Notch
0x3024	0	Axis	FLOAT	Filter Setup	Position Cut off Frequency (3dB) (Hz)	Cutoff frequency of the position filter (3 dB limit)
0x3025	0	Axis	INT	Filter Setup	Position Filter Type	Structure of the position filter: 0 = OFF 1 = IIR 2. Ord 2 = IIR 4. Ord 3 = IIR fast 4 = Notch
0x3028	0	Axis	FLOAT	Filter Setup	Controller Cut off Frequency (3dB) (Hz)	Cutoff frequency of the control value filter (3 dB limit)
0x3029	0	Axis	INT	Filter Setup	Controller Filter Type	Structure of the filter for the control value: 0 = OFF 1 = IIR 2. Ord 2 = IIR 4. Ord 3 = IIR fast 4 = Notch
0xE	3	Axis	INT	Information	Numerator	Numerator and denominator of the factor for counts per physical length unit. Read only, must be set via parameters 0x3033 and 0x3034.
0xF	3	Axis	INT	Information	Denominator	
0x3F	0	Axis	FLOAT	Motion	Settling Time (s)	Delay time for setting the on-target state

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x49	0	Axis	FLOAT	Motion	Velocity (Phys. Unit/s)	Velocity in closed-loop and open-loop operation. Can be set with the VEL command in volatile memory. Is limited by parameter 0xA.
0x4A	1	Axis	FLOAT	Motion	Maximum Acceleration (Phys. Unit/s^2)	Maximum acceleration in closed-loop and open-loop operation Specifies the maximum value for parameter 0xB.
0x4B	1	Axis	FLOAT	Motion	Maximum Deceleration (Phys. Unit/s^2)	Maximum deceleration in closed-loop and open-loop operation Specifies the maximum value for parameter 0xC.
0x7000900	0	Axis	FLOAT	Motion	Settling Window (Phys. Unit)	Settling window around the target position Specifies the window limits. If the current position exits the settling window, the target position is no longer considered as reached. The parameter value corresponds to half the width of the window. Note that as long as the current position is inside the settling window and stays there at least for the duration of the delay time (parameter 0x3F), the velocity control is deactivated, see “Generation of Target Values for Closed-Loop Operation” (p. 15) for details.
0x8	0	Axis	FLOAT	Motion	Maximum Position Error (Phys. Unit)	Maximum position error in closed-loop operation When the position error for the axis exceeds this value, the C-891 switches off the servo mode and the motor for the axis (SVO? and EAX? both will return 1 = 0) Safety feature, for details see „Motion Error“ (p. 36).
0xA	1	Axis	FLOAT	Motion	Maximum Velocity (Phys. Unit/s)	Maximum velocity in closed-loop and open-loop operation Specifies the maximum value for parameter 0x49.

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0xB	0	Axis	FLOAT	Motion	Acceleration (Phys. Unit/s^2)	Acceleration in closed-loop and open-loop operation. Can be set with the ACC command in volatile memory. Is limited by parameter 0x4A.
0xC	0	Axis	FLOAT	Motion	Deceleration (Phys. Unit/s^2)	Deceleration in closed-loop and open-loop operation. Can be set with the DEC command in volatile memory. Is limited by parameter 0x4B.
0x1A	1	Axis	INT	Motor Setup	Has Brake?	Does the stage have a brake? 0 = No brake present 1 = Yes, brake present. In this case, switching on/off the servo mode and activating/deactivating the brake are intercoupled; see p. 14 for further details.
0x3040	1	Axis	FLOAT	Motor Setup	P Current Control 1 (Kp)	P term of the subordinate current control (Id)
0x3041	1	Axis	FLOAT	Motor Setup	I Current Control 1 (Tn)	I term of the subordinate current control (Id)
0x3043	1	Axis	FLOAT	Motor Setup	Limit Current Control 1	Limit of the subordinate current control (Id)
0x3044	1	Axis	FLOAT	Motor Setup	P Current Control 2 (Kp)	With 3-phase motor (0x3062 Motor Type = 0) only: P term of the subordinate current control (Iq)
0x3045	1	Axis	FLOAT	Motor Setup	I Current Control 2 (Tn)	With 3-phase motor (0x3062 Motor Type = 0) only: I term of the subordinate current control (Iq)
0x3047	1	Axis	FLOAT	Motor Setup	Limit Current Control 2	With 3-phase motor (0x3062 Motor Type = 0) only: Limit of the subordinate current control (Iq)
0x3048	1	Axis	INT	Motor Setup	Enable Current Control	Enable current control. With DC-motors (0x3062 Motor Type = 2) only. 0 = disabled 1 = enabled

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x3049	1	Axis	FLOAT	Motor Setup	Current Filter Cutoff Frequency (3dB) (Hz)	Cutoff frequency for current filter. Can be disabled by a value of 10000 Hz. Minimum value is 50 Hz.
0x304E	1	Axis	FLOAT	Motor Setup	Steps Per Revolution	With stepper motors (0x3062 Motor Type = 1) only: Number of (full) steps required for a full motor revolution
0x304F	1	Axis	FLOAT	Motor Setup	Distance Per Revolution [Phys. Unit]	With stepper motors (0x3062 Motor Type = 1) only: The distance the axis moves with a full motor revolution
0x3050	1	Axis	FLOAT	Motor Setup	Electrical Pitch of the Motor Magnets	With 3-phase motor (0x3062 Motor Type = 0) only: Travel caused by one electrical cycle of the motor
0x3051	1	Axis	FLOAT	Motor Setup	Phase Correction Offset (Phys. Unit)	With 3-phase motor (0x3062 Motor Type = 0) only: Offset for minimization of the phase difference between the sensor position and electrical motor position. The parameter value is set by the commutation angle adjustment procedure (FPH, p. 75) in volatile memory.
0x3052	3	Axis	FLOAT	Motor Setup	BUS Voltage (V)	Actual value of supply voltage Read only. The voltage selection is set via parameter 0x3056.
0x3053	1	Axis	INT	Motor Setup	Phase Finding (PMSM)	Determines how the axis moves during the commutation angle adjustment procedure (p. 75) 0 = Type A (default): The axis moves one full electrical motor cycle (see parameter 0x3050). 1 =Type B: The motion can extend over the entire travel range. 2 = Type C: Only small motion, i.e., the axis moves by a maximum of a few millimeters.

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x3054	1	Axis	INT	Motor Setup	Angle Reset	With 3-phase motor (0x3062 Motor Type = 0) only: Enable electrical angle reset. May be required for rotating motors: To avoid errors coming from the sensor-to-angle calculation, the electrical angle can be set to zero each time a defined distance (set by 0x3055) has been covered. 0 = disable 1 = enable
0x3055	1	Axis	FLOAT	Motor Setup	Angle reset intervall [0 = OFF] (Phys. Unit)	With 3-phase motor (0x3062 Motor Type = 0) only: Distance for angle reset
0x3056	1	Axis	INT	Motor Setup	BUS Voltage Setting	Supply voltage of the C-891 0 = automatic detection (24 to 48 V) 1 = 24 V fix (default) 2 = 48 V fix With automatic detection, read the actual supply voltage via parameter 0x3052. The supply voltage determines the intermediate circuit voltage.
0x3060	1	Axis	FLOAT	Motor Setup	Motion Motor Voltage (V)	Amplitude of the phase voltage during reference move and during motion started with FPH (commutation angle adjustment)
0x3061	1	Axis	INT	Motor Setup	Motor Stop Mode	Motor voltage profile when the servo mode is switched off: 0 = Real Zero (default): All phases at GND, PWM deactivated→ **No self-locking of the axis, if no brake is present!** 1 = Holding Level: Keep the last valid voltage, i.e. motor holds the position in open-loop operation
0x3062	1	Axis	INT	Motor Setup	Motor Type	Motor type: 0 = 3-phase motor 1 = Stepper motor 2 = DC motor (for future use) For further details, see “Motor Types” (p. 12).
0x3064	1	Axis	FLOAT	Motor Setup	Phase Resistor (Ohm)	Phase resistance in ohm

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x3065	1	Axis	FLOAT	Motor Setup	Phase Inductor (mH)	Phase inductance in mH
0x3066	1	Axis	INT	Motor Setup	Invert Motor Direction (Rotating Field Direction)?	Direction of motion inverted? 0 = No 1 = Yes
0x3070	1	Axis	FLOAT	Motor Setup	Maximum Motor Current At Standstill [A]	With stepper motors (0x3062 Motor Type = 1) only: Maximum permissible motor current at standstill.
0x3071	1	Axis	FLOAT	Motor Setup	Maximum Motor Current At Constant Speed [A]	With stepper motors (0x3062 Motor Type = 1) only: Maximum permissible motor current during motion with constant velocity.
0x3072	1	Axis	FLOAT	Motor Setup	Maximum Motor Current During Acceleration [A]	With stepper motors (0x3062 Motor Type = 1) only: Maximum permissible motor current during acceleration.
0x3073	1	Axis	FLOAT	Motor Setup	Current Max Trip Zone (A)	Maximum current per phase, in A. Safety feature: The limitation of the output current protects the motor driver from overload.
0x3074	1	Axis	FLOAT	Motor Setup	Continuous Phase Current (A)	Parameters for calculation of I^2t overcurrent threshold Safety feature, for details see “Overcurrent Protection of the System” (p. 38).
0x3075	1	Axis	FLOAT	Motor Setup	Peak Phase Current (A)	
0x3076	1	Axis	FLOAT	Motor Setup	Peak Phase Current Duration (s)	
0x3077	1	Axis	INT	Motor Setup	Temperature Fuse	Activation state of motor overtemp protection: 0 = None: Disabled 1 = Version A: Enabled (PT100) 2 = Version B: For future use Safety feature, for details see „Overtemp Protection of the Motor“ (p. 38).
0x3079	1	Axis	FLOAT	Motor Setup	Limitation of Continuous Current for Phase Finding (%)	Limits the continuous current (0x3074) for the commutation angle adjustment procedure (p. 75).

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x3080	1	Axis	INT	Motor Setup	Motor Power At Startup	Automatic switch-on of the motor after the C-891 is switched on or rebooted 0 = OFF: Motor is not automatically switched on (default; EAX? returns 1 = 0) 1 = ON: Motor is automatically switched on (EAX? returns 1 = 1) Safety feature, for details see „Motor Deactivation“ (p. 35).
0x3081	1	Axis	INT	Motor Setup	Safety Pin Action	Configuration of safety pin action: 0 = OFF: Safety pin deactivated (default) 1 = Fast stop with maximum deceleration (active low) 2 = Fast stop with maximum deceleration (active high) 3 = Stop with deceleration (active low) 4 = Stop with deceleration (active high) Safety feature, for details see „Stop Axis by External Hardware Switch“ (p. 37).
0x3082	1	Axis	INT	Motor Setup	Safety Stop	Configuration of axis behavior after safety pin action: 0 = Runner Free : Servo mode and motor are switched off (default; SVO? and EAX? both return 1 = 0) → **No self-locking of the axis, if no brake is present!** 1 = Runner Clamped: The servo mode is switched off, and the motor remains on (SVO? returns 1 = 0, and EAX? returns 1 = 1) Safety feature, for details see „Stop Axis by External Hardware Switch“ (p. 37).
0x3095	1	Axis	FLOAT	Motor Setup	Brake Voltage (V)	Supply voltage levels for an electromagnetic brake For details, see p. 14.
0x3096	1	Axis	FLOAT	Motor Setup	Brake Continious Voltage (V)	

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x30D4	1	Axis	FLOAT	Motor Setup	Second Encoder Phase Correction Offset (Phys. Unit)	With 3-phase motor only (parameter 0x3062 has the value 0): Offset for minimization of the phase difference between the position of the second encoder and the electrical motor position. The parameter value is set by the commutation angle adjustment procedure (FPH, p. 75) in volatile memory.
0x3003301	1	Axis	INT	Sensor Electronics	Sensor Hysteresis (Deg)	Configuration of the interpolation applied to analog SIN/COS signals
0x3003302	1	Axis	INT	Sensor Electronics	Sensor Digital Gain	
0x3003303	1	Axis	INT	Sensor Electronics	Sensor Digital Offset SIN (V)	
0x3003304	1	Axis	INT	Sensor Electronics	Sensor Digital Offset COS (V)	
0x3003305	1	Axis	INT	Sensor Electronics	Sensor Digital Phase (Deg)	
0x3003306	1	Axis	INT	Sensor Electronics	Sensor Analog Gain (dB)	
0x3003307	1	Axis	INT	Sensor Electronics	Sensor Analog Offset SIN (V)	
0x3003308	1	Axis	INT	Sensor Electronics	Sensor Analog Offset COS (V)	
0x3003309	1	Axis	INT	Sensor Electronics	Sensor Autoadaption	For future use
0x300330A	1	Axis	INT	Sensor Electronics	Sensor Period Error Correction	For future use
0x3030	1	Axis	INT	Sensor Electronics	Interpolation Factor	Interpolation factor for stages with incremental sensor. Recommended values: 16384, 32768 or 65536.

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x3031	3	Axis	FLOAT	Sensor Electronics	Sensor Pitch (Phys. Unit)	Grating period of the incremental sensor.
0x3032	1	Axis	INT	Sensor Electronics	Encoder Type:	Type of sensor: 0 = Analog SIN/COS signals, interpolation in the C-891 1 = Digital BiSS interface 2 = A/B (TTL) signals via Sensor connector 3 = A/B (TTL) signals via Motor connector 4 = No sensor connected to the C-891 5 = Step counter (with stepper motors only; motor steps are counted; if a sensor is present, it is ignored) For further details, see “Sensor Types” (p. 13).
0x3033	1	Axis	INT	Sensor Electronics	Set new Numerator	Input parameters for the numerator (0xE) and denominator (0xF) of the factor for counts per physical length unit. The values of parameters 0xE and 0xF are set to the values of 0x3033 and 0x3034 during the next start or reboot of the C-891. Default value of denominator is 1; must be smaller than the numerator value.
0x3034	1	Axis	INT	Sensor Electronics	Set new Denominator	
0x3035	1	Axis	INT	Sensor Electronics	Invert Encoder Counting Direction?	Sensor counting direction 0 = not inverted 1 = inverted
0x3036	1	Axis	INT	Sensor Electronics	Sensor Correction LUT	Reserved for future use.
0x3037	1	Axis	FLOAT	Sensor Electronics	Sensor Correction LUT Start	
0x3038	1	Axis	FLOAT	Sensor Electronics	Sensor Correction LUT Interval Length	
0x303A	1	Axis	INT	Sensor Electronics	BISS Data Length (bits)	Data length of the BiSS protocol
0x303B	1	Axis	INT	Sensor Electronics	BISS Polynom	Value for calculation of the BiSS polynomial

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x303C	1	Axis	INT	Sensor Electronics	BISS CRC Inversion	Is the CRC inverted? 0 = False 1 = True
0x303D	1	Axis	INT	Sensor Electronics	BISS Number of Error bits	Number of error bits in the BiSS protocol
0x303E	1	Axis	INT	Sensor Electronics	BISS Sensor Value Divider	Divides position value of BiSS encoder 0 = 1 1 = 2 2 = 4 3 = 8 4 = 16 5 = 32 6 = 64 7 = 128
0x30D0	1	Axis	INT	Sensor Electronics	Second Encoder Signal Processing	Determines which signal processing is used for the second sensor: 0 = NONE 1 = Velocity 2 = Angle 3 = Angle (index reset)
0x30D1	1	Axis	INT	Sensor Electronics	Second Encoder Numerator (INC/Phys. Unit/s)	Parameters for numerator and denominator of the second encoder's factor for counts per physical length unit. The parameter values must be set with SEP and become active during the next start or reboot of the C-891. Default value of denominator is 1; must be smaller than the numerator value.
0x30D2	1	Axis	INT	Sensor Electronics	Second Encoder Denominator (default 1; smaller than Numerator)	
0x30D3	1	Axis	INT	Sensor Electronics	Second Encoder Invert Encoder Counting Direction?	Invert counting direction of the second encoder? 0 = No 1 = Yes
0x30D5	1	Axis	INT	Sensor Electronics	Commutation Angle Encoder 0 = Main Encoder and 1 = Second Encoder	Determines the encoder to be used for angle commutation (and hence the response to FPH?, p. 77). 0 = Main encoder 1 = Second encoder

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x3003900	1	Axis	INT	Sensor Electronics	Quadrature Encoder Filter: 0 = Off 1 = Fast 2 = Medium 3 = Slow	Input filter to reduce noise when using the A/B motor encoder. 0 = Off 1 = Fast 2 = Medium 3 = Slow
0x13	1	Axis	INT	Stage Setup	Is Rotary Stage?	Is this a rotary stage? 0 = No 1 = Yes No evaluation by the C-891, but only by the PC software: PIMikroMove determines on the basis of this value which motions are permissible.
0x14	1	Axis	INT	Stage Setup	Has Reference?	Does the stage have a reference point switch? 0 = No 1 = Yes This parameter enables or disables reference moves to the reference point switch installed.
0x15	1	Axis	FLOAT	Stage Setup	Maximum Travel In Positive Direction (Phys. Unit)	Soft limits of the travel range. The limits establish the permissible travel range of the axis. Motion commands are executed only if the commanded position is within these soft limits. For further details, see “Travel Range and Soft Limits” (p. 14).
0x30	1	Axis	FLOAT	Stage Setup	Maximum Travel In Negative Direction (Phys. Unit)	
0x18	1	Axis	INT	Stage Setup	Limit Mode, P- / N-Limit Aktive	Signal logic of the limit switches: 0 = Low / Low 1 = Low / High 2 = High / Low 3 = High / High
0x3078	1	Axis	FLOAT	Stage Setup	Coil Temperature Maximum (C)	Safety feature, for details see „Overtemp Protection of the Motor“ (p. 38).
0x3090	1	Axis	INT	Stage Setup	Reference Input Selection	Selects the connector to be used for the input of the reference signal (the selection is irrelevant for the limit signal input): 0 = Motor connector 1 = Sensor connector For further details, see “Reference Moves” (p. 13).

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x30A0	1	Axis	FLOAT	Stage Setup	Analog Input Amplification	Settings for the analog input. For further details, see p. 45 and p. 120.
0x30A1	1	Axis	FLOAT	Stage Setup	Analog Input Offset	
0x31	1	Axis	INT	Stage Setup	Reference Active	Signal logic of the reference signal: 0 = Low 1 = High
0x32	1	Axis	INT	Stage Setup	Has No Limit Switches?	Does the stage have limit switches? 0 = No 1 = Yes This parameter enables or disables a stop of the motion at the limit switches installed.
0x3C	0	Axis	CHAR	Stage Setup	Stage Name max. 20 characters	Information on the connected stage
0x48	1	Axis	FLOAT	Stage Setup	Position Offset	Position offset: Gives an offset that is added to the sensor position. Can be positive or negative. Using the position offset, the zero position of the axis can be shifted. For further details, see “Reference Moves” (p. 13) and “Travel Range and Soft Limits” (p. 14).
0x50	1	Axis	FLOAT	Stage Setup	Velocity For Reference Moves (Phys. Unit/s)	Velocity for reference move and for motion started with FPH (commutation angle adjustment). This value should at a maximum be as great as the value of parameter 0x49. If the value of parameter 0x50 is set to 0, reference moves and moves for commutation angle adjustment (FPH) are not possible.
0x70	1	Axis	INT	Stage Setup	Reference Type	Reference signal type: 0 = Homing track with one signal transition low/high or high/low 1 = Impulse 2 = Impulse via negative limit 3 = Impulse via positive limit 4 = No reference signal (e.g. with absolute measuring sensors) 5 = Negative limit 6 = Positive limit For further details, see “Reference Moves” (p. 13).

Parameter ID (hexadecimal)	Command Level for Write Access	Item Type	Data Type	Parameter Group	Parameter Name	Description
0x3057	1	Axis	FLOAT	Stage Setup	Velocity For Phase Finding (Phys. Unit/s)	Determines the velocity for the commutation angle adjustment procedure (p. 75).
0x7000601	1	Axis	CHAR	Stage Setup	Axis Unit	Unit symbol The value is not evaluated by the C-891 but is only used by the PC software for display purposes.
0xF000000	1	Axis	INT	Stage Setup	Power Up Read ID-Chip Enable	Determines whether the C-891 reads parameters from ID-chip. 0 = disable 1 = enable
0xF000200	1	Axis	CHAR	Stage Setup	Stage Serial Number	Information on the connected stage
0xF000300	1	Axis	CHAR	Stage Setup	Stage Assembly Date	Information on the connected stage
0xF000400	1	Axis	INT	Stage Setup	Stage HW Version	Information on the connected stage
0x11000400	3	System	INT	System	Interface Baudrate	Baud rate for RS-232 communication: 115,200 baud
0x13000004	3	System	INT	System	Max Number of Points in Wave Table	Information on the wave table. Details see „Commandable Items“ (p. 11).
0x1300010A	3	System	INT	System	Number of Wave Tables	
0x16000100	3	System	INT	System	Max Number of Data Recorder Channels	Configuration of data recorder tables. Details see „Commandable Items“ (p. 11).
0x16000200	3	System	INT	System	Number of Data Recorder Points	
0x16000300	0	System	INT	System	Data Recorder Channel Number	
0x72	0	System	INT	System	Ignore Macro Error?	Determines whether the controller macro is stopped when an error occurs while it is being executed. 0 = Stop macro when error occurs (default) 1 = Ignore error
0xD000100	2	System	CHAR	System	Controller Serial Number	Serial number of the C-891
0xE000200	3	System	FLOAT	System	Servo Update Time	Servo cycle time of the C-891 (= sensor sampling time)

Maintenance

Updating Firmware

INFORMATION

The *IDN? command reads the version number of the firmware among other things.
Example of a response of the C-891:

```
(c)2015 Physik Instrumente (PI) GmbH & Co. KG, C-891.120200,
115057758, 01.015
```

- 115057758: Serial number of the C-891
- 01.015: Firmware version

INFORMATION

The LEDs on the front panel indicate the firmware update process:

- **PWR** LED flashing with 2 Hz: C-891 is in firmware update mode
- **PWR** LED flashing with 1 Hz: New firmware has been successfully written to the C-891
- **ERR** and **PWR** LED flashing alternately: No or wrong firmware has been written to the C-891.

➢ See „Product View“ (p. 8) for details on the behavior of the other LEDs.

Prerequisite

✓ You have connected the C-891 to the PC **via the RS-232 interface**.
✓ The PI Firmware Updater program is installed on the PC (p. 23).
✓ You have received a new firmware file from our customer service department (p. 121) and copied it to a directory on the PC.
✓ You have read and understood the documentation which you received from our customer service department together with the new firmware file.

Updating firmware of the C-891

1. Switch on the C-891 by connecting the power cord of the power supply to the power socket.
2. Start the PI Firmware Updater program on the PC.
3. Select the controller: C-891.120200.
4. Select the COM port of the PC to which you have connected the C-891.
5. Select the firmware type: DSP
6. Select the new firmware file (in a separate file selection window).
7. Click ***Start Update***.

   The firmware update starts.

   When the firmware update has finished, it takes another period of 5 to 10 s until the C-891 reboots. After the reboot, the C-891 is in firmware update mode again (**PWR** LED flashing with 2 Hz; this is useful in the case of a failed update which could now be repeated).

When the C-891 has been in firmware update mode for about 10 s without repeating the update:

- Firmware update was successful: C-891 switches to normal operation. The PWR LED is continuously lit.
- No or wrong firmware was written to the C-891: **ERR** and **PWR** LED flash alternately. Repeat the update with an appropriate firmware file.

Cleaning the C-891

NOTICE

Short circuits or flashovers!
The C-891 contains electrostatic sensitive devices that can be damaged by short circuits or flashovers when cleaning fluids enter the case.

- ➢ Before cleaning, remove the C-891 from the power source by pulling the power plug.
- ➢ Prevent cleaning fluid from entering the case.

1. Disconnect the C-891 from the power supply.
2. Wait a minute to be sure that any residual voltage has dissipated.
3. Clean the C-891 case surface with a cloth lightly dampened with a mild cleanser or disinfectant.

Technical Data

Specifications

Preliminary Data	**C-891.130300**
Function	PIMag® motion controller for 3-phase motors, sine-commuted, field-oriented current control, and for stepper motors
Motor channels	1
Sensor channels	1
Motion and control	
Servo characteristics	PID controller for position and velocity, parameter change on-the-fly
Servo cycle time	50 µs
Profile generator	Trapezoidal velocity profile, setting of maximum velocity and acceleration
Encoder input	Incremental sensors: analog (sin/cos; 1 Vpp) or digital signals (A/B differential; TTL) Absolute measuring sensors: 32-bit BiSS interface
Reference point switch and limit switches	TTL
Electrical properties	
Max. output voltage	48 / 24 V depending on operating voltage
Max. average output current, amplitude of sine	5 A
Peak output current, amplitude of sine	10 A
Max. average output current, RMS	3.6 A
Peak output current, RMS	7.2 A
Interfaces and operation	
Communication interfaces	USB, RS-232
Motor connector	HD D-sub 26 (f)
Sensor connection	D-sub 15 (m)
I/O port	HD D-sub 15 (f): 4 x digital input 4 x digital output LEMO EPG.00.302.NLN: Analog input, -10 V to 10 V differential
Command set	PI General Command Set (GCS)
User software	PIMikroMove
Software drivers	API for C / C++ / C# / VB.NET / MATLAB / Python, drivers for NI LabVIEW
Supported functions	Point-to-point motion, data recorder, macros, wave generator, commutation angle adjustment for 3-phase motors
Safety features	Axis stop by hardware switch, overload protection of motor driver, overtemp protection of motor, overcurrent protection of the system

Miscellaneous	
Operating voltage	48 / 24 V DC Power adapter 24 V DC (120 W/5 A) in the scope of delivery.
Max. current consumption	10 A (load-dependent)
Operating temperature range	5 to 40 °C
Max. mass	1.0 kg
Dimensions	190 × 83 × 110 mm (206 × 83 × 112 mm including rubber feet and supply voltage connection)

Maximum Ratings

The C-891 is designed for the following operating data:

Input on:	Maximum Operating Voltage	Operating Frequency	Maximum Current Consumption
M12	48 V	⎓	10 A

Ambient Conditions and Classifications

The following ambient conditions and classifications must be observed for the C-891:

Area of application	For indoor use only
Maximum altitude	2000 m
Relative humidity	Highest relative humidity 80% for temperatures up to 31°C Decreasing linearly to 50% relative humidity at 40°C
Storage temperature	0°C to 70°C
Transport temperature	–25°C to +85°C
Overvoltage category	II
Protection class	I
Degree of pollution	2
Degree of protection according to IEC 60529	IP20

Pin Assignment

Motor

HD Sub-D 26 (f)

Pin	Signal	Function
1	OUT0a	Phase I
2	OUT0b	Phase I
3	OUT1a	Phase II
4	OUT1b	Phase II
5	OUT2a	Phase III
6	OUT2b	Phase III
7	OUT3a	Phase IV
8	OUT3b	Phase IV
9	Brake Out	Motor Brake
10	Ref	Reference Signal (Reference Type parameter (0x70) = 0)
11	NLIM	Negative Limit
12	PLIM	Positive Limit
13	HALL0	reserved for Hall Sensor Input I
14	HALL1	reserved for Hall Sensor Input II
15	HALL2	reserved for Hall Sensor Input III
16	24/48 VDC	Output of power supply voltage (24/48 V DC)
17	reserved	reserved
18	VCC_ENC	VCC Encoder
19	ENCA+	A+ (TTL)*
20	ENCA-	A- (TTL)*
21	ENCB+	B+ (TTL)*
22	ENCB-	B- (TTL)*
23	INDEX+	Index pulse + (Reference Type parameter (0x70) = 1, 2, or 3)*
24	INDEX-	Index pulse - (Reference Type parameter (0x70) = 1, 2, or 3)*
25	GND	GND
26	VCC_ENC	VCC Encoder

* If the “main” position sensor is connected to the **Sensor** connector of the C-891, a second encoder can be connected to those pins of the **Motor** connector. Refer to p. 17 for more information.

Sensor

Sub-D 15 (m)

Pin	Signal	Function
1	VCC 5V	VCC Sensor
2	AGND	GND Sensor
3	Sensor-A +	Sin (1 Vpp) / A+ (TTL) / BISS MA1
4	Sensor-A -	Sin (1 Vpp) / A- (TTL) / BISS MA1
5	GND TEMP	
6	Sensor-B +	Cos (1 Vpp) / B+ (TTL) / BISS SL1
7	Sensor-B -	Cos (1 Vpp) / B- (TTL) / BISS SL1
8	N-Limit	Negative Limit
9	PT100	4 mA current source with ADC input
10	REF +	Reference Signal + (Reference Type parameter (0x70) = 0, 1, 2, or 3)
11	Thermistor	For future use (4 mA current source)
12	Ref -	Reference Signal - (Reference Type parameter (0x70) = 0, 1, 2, or 3)
13	reserved	reserved
14	GND	GND
15	P-Limit	Positive Limit

Digital I/O

HD Sub-D 15 (f)

Pin	Signal	Function
1	IN 1 D	Digital (0/5 V)
2	IN 2 D	Digital (0/5 V)
3	IN 3 D	Digital (0/5 V)
4	IN 4 D	Digital (0/5 V)
5	OUT 1	Digital (0/5 V)
6	OUT 2	Digital (0/5 V)
7	OUT 3	Digital (0/5 V)
8	OUT 4	Digital (0/5 V)
9	SAFETY	Terminate with >30 ohm (4 mA current source)
10	GND	GND
11	reserved	reserved
12	reserved	reserved
13	reserved	reserved
14	reserved	reserved
15	reserved	reserved

Analog In

LEMO EPG.00.302.NLN

Position of the coding

Pin	Function
1	Analog $_{in-}$
2	Analog $_{in+}$

|Analog $_{in+}$ - Analog $_{in-}$| ≤ 10 V in the range of -10 V to +10 V

Resolution: 16 bit

Bandwidth: 400 Hz

Max. impedance of the signal source: >40 kOhm

RS-232

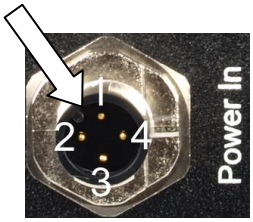
Sub-D 9 (m)

Pin	Function
1	nc
2	RXD receive data
3	TXD send data
4	nc
5	DGND ground
6	nc
7	nc
8	nc
9	nc

Power In

M12 panel plug, 4-pin, male

Position of the coding pin



Pin	Function
1	GND (power)
2	GND (power)
3	Input: 24/48 V DC*
4	Input: 24/48 V DC*

* The BUS Voltage Setting parameter (0x3056) must be set accordingly. For details, see “Parameter Descriptions” (p. 96).

Customer Service

For inquiries and orders, contact your PI sales engineer or send us an e-mail (info@pi.ws).

- 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 present)
  - Version of the driver or the software (if present)
  - Operating system on the PC (if present)
- If possible: Take photographs or make videos of your system that can be sent to our customer service department if requested.

The latest versions of the user manuals are available for download (p. 6) on our website.

EC Declaration of Conformity

For the C-891, an EC Declaration of Conformity has been issued in accordance with the following European directives:

EMC Directive

RoHS Directive

The applied standards certifying the conformity are listed below.

EMC:	EN 61326-1
Safety:	EN 61010-1
RoHS:	EN 50581