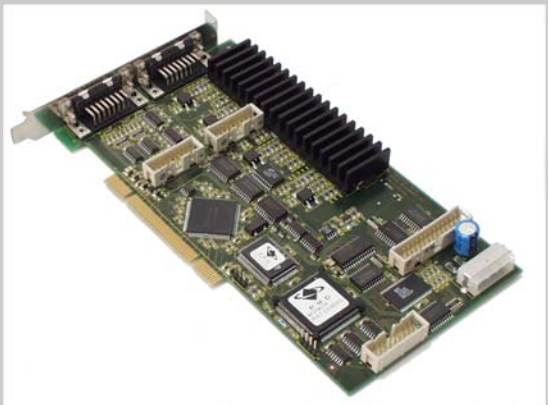
Piezo · Nano · Positioning

SM149E Software Manual

C-843 GCS Commands

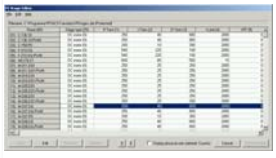
PI General Command Set

Release: 2.0.0 Date: 2009-11-17



This document describes software for use with the following products:

- C-843.21
  Motor Controller Card (PCI), 2 motor axes
- C-843.41
  Motor Controller Card (PCI), 4 motor axes



Auf der Römerstr. 1 · 76228 Karlsruhe, Germany
Tel. +49-721-4846-0 · Fax: +49-721-4846-299
info@pi.ws · www.pi.ws

First printing 2009-11-17
Document Number SM149E, Bro, Release 2.0.0
C-843_GCS_Commands_SM149E.doc

Contents

1 Introduction

The PI General Command Set (GCS) is supported by a wide range of PI systems. This command set is well-suited for positioning tasks with one or more axes. The command set itself is independent of the specific hardware (controller or attached stages).

GCS commands for the C-843 are provided by the C843_GCS_DLL which must be installed on the PC. You can type commands, for example, in the *Command Entry* window of PIMikroMove™, or in the PITerminal (to which the C-843 board must be connected via the C-843 GCS DLL).

Commands are used to set operating modes, initiate axis motion and to query system and motion values. Because of the variety of functions and parameters, a sequence of commands must often be transferred in order to achieve a desired system action.

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

The hardware and software installation procedures for the C-843 are described in the C-843 User Manual (MS77E). Software tools which might be mentioned in this document are described in their own manuals. All documents are available as PDF files on the C-843 CD. Updated releases are available for download at www.pi.ws or via e-mail: contact your Physik Instrumente Sales Engineer or write info@pi.ws.

List of manuals related to this document:

C-843_User_MS77E	User Manual for C-843 DC-Servo-Motor Controller board
C-843_GCS_DLL_MS112E	Manual for GCS Library (Windows and Linux Versions)
C843_GCS_LabVIEW_MS89E	Manual for GCS LabVIEW Drivers
PIMikroMoveUserManual_SM148E	Manual for PIMikroMove™ Host Software
PiStageEditor_SM144E	Manual for PIStageEditor Tool for Stage Database Handling
GCSData_User_SM146E	GCS Data Format Description
A000T0014_100_UserProfileModeSoftware	Technical Note for User Profile Mode and Profile Generator DLL

2 Operating Considerations

2.1 Initialization

The C843_GCS_DLL supports a mechanism which mirrors the hardware basics of the connected stage and the required closed-loop control settings in parameters. The parameter values have to be adjusted properly before initial operation of a stage. For that purpose, the CST (p. 18) and INI (p. 34) commands must be sent whenever you start working: CST loads stage parameters suitable for your hardware from a stage database, and INI writes the loaded values to the controller to initialize the motion control chip on the C-843 board. This is done on a per-axis basis. Note that PIMikroMove™ performs this procedure automatically upon stage selection. See “Parameter Databases” (p. 77) for the available stage databases.

2.2 Referencing

Because the encoder signals used for position feedback provide only relative motion information, the controller cannot know the absolute position of an axis upon startup. This is why a referencing procedure is required before absolute target positions can be commanded and reached.

For the implementation of the referencing functionality in the individual host software components, see the appropriate manuals.

2.2.1 Reference Mode

The current reference mode setting of the controller (ask with RON?, p. 44) determines how referencing can be performed. By default, a reference move must be performed (see Section 2.2.2), but it is also possible to set absolute positions manually (see Section 2.2.3). To switch between the two reference modes, use the RON command (p. 44).

2.2.2 Perform a Reference Move

When the reference mode is set to "1" (factory default), referencing is done by performing a reference move.

You can start reference moves with the following commands:

- FRF (p. 32), FNL (p. 30) or FPL (p. 31)— if multiple axes are given in one of these commands, they are moved synchronously.
- REF (p. 43), MPL (p. 39) or MNL (p. 37)—if multiple axes are given in one of these commands, one axis after another is moved.

NOTES:

Neither relative nor absolute targets can be commanded as long as referencing was not successfully performed.

FRF and REF require that the axis has a reference switch (ask with REF? (p. 43)), and FPL, FNL, MPL and MNL require that the axis has limit switches.

For best repeatability, always reference in the same way. The FRF and REF commands always approach the reference switch from the same side, no matter where the axis is when it is issued.

2.2.3 Set Absolute Position

When the reference mode is set to "0", referencing is done by entering an absolute position value using the POS command (p. 42).

NOTES

Only relative targets but no absolute targets can be commanded as long as referencing was not successfully performed.

If the controller is given an incorrect position with POS (p. 42), the axis can be run into a limit switch, where the motor voltage is switched off. To get back to working conditions, send an INI (p. 34) command for the axis.

2.3 Data Recording

For general information regarding the data recording you can send HDR? (p. 33) (lists available record options and trigger options, gives information about additional parameters and commands regarding data recording).

The data recorder configuration, i.e. the assignment of data sources and record options to the recorder tables, can be changed with DRC (p. 23), and the current configuration can be read with DRC? (p. 24). Data recorder tables with record option 0 are deactivated, i.e. nothing is recorded. If one data recorder table is deactivated, all subsequent tables are deactivated too.

Recording can be triggered in several ways. Ask with DRT? (p. 26) for the current trigger option and use DRT (p. 25) to change it. By default data recording is triggered when a step response measurement is made with STE (p. 49). Note that STE automatically changes the record option of the first data recorder table to "actual position" (2).

The C-843 has 4 data recorder tables (ask with TNR? (p. 54)). The points available for recording are in equal shares allocated to the tables with non-zero DRC record options (for the total number of points to allocate ask SPA? (p. 48) with parameter 0x16000200, maximum value is 32,256; see also NOTES below).

The record table rate can be set with the RTR command (p. 44). The power-on default of this value is 1 (the unit is servo cycles; ask with RTR? (p. 45)). You can cover longer periods by increasing the record table rate. Note that the cycle time of the C-843 is 410 µs.Note that STE automatically resets the data recorder sample period to 1.

Recording always takes place for all data recorder tables with non-zero record options.

Recording ends when the content of the data recorder tables has reached the maximum number of points.

The last recorded data can be read with DRR? (p. 25) or with STE? (p. 50) (reads only the content of the first table). While with DRR? the data are reported in GCS array format, you will get a list of position values with STE?. For details regarding the GCS array see the separate manual (SM146E) which is provided on the C-843 CD. Reading can take long depending on the number of points to be read! It is possible to read the data while recording is still in progress.

When the controller is powered down, the content of the data recorder tables and all data recorder configuration and trigger settings are lost. The configuration and trigger settings are reset to their defaults with INI (p. 34).

NOTES

The data recorder makes use of the volatile memory provided on the C-843 card (referred to as “external RAM” in the MotionProcessor Users Guide). It shares this memory (32,256 points) with the multi-axis motion profiles which can be created by the User Profile Mode commands (UPx commands beginning on p. 55, see separate Technical Note A000T0014_100_UserProfileModeSoftware for more information regarding the User Profile Mode). To free memory occupied by user-defined motion profiles, send UPC $ $ -1 -1 (deletes all Clusters and all Blocks of the profile).

With some older C-843 hardware models, usage of this volatile memory and usage of the digital output lines on the 26-pin IDC connector (J5) are mutually exclusive. This means that after the C-843 board was connected in the software, only the functionality called first is available. The selection is reset any time the C-843 board is reconnected. If, for example, the digital output lines have been set with DIO (p. 22) before any data-recorder-related command was sent, you can not use the data recorder until you reconnect the C-843.

2.4 External Triggering / Signaling

Using the digital IO lines of the C-843, it is possible to trigger external devices, and to receive signals from external devices. The lines can be brought out of the PC housing using an adapter bracket with sub-D 25f connector and sub-D 15m connector (included with C-843). Note that older revisions of the adapter bracket do not provide the sub-D 15m connector which gives access to the All-Axes 16-pin IDC connector on the C-843 board. If required, contact your PI sales engineer or write info@pi.ws to obtain a new version of the adapter bracket.

The following digital IO lines are available:

- Eight input and output lines are located on the 26-pin IDC connector (J5) of the C-843 board (connects to the sub-D 25f connector of the adapter bracket; see C-843 User Manual for component location). Note that these lines have no direct connection to the motion processor. In the software, they are each identified with A to H.
  With DIO? (p. 23), you can directly read the input lines A to H.
  Using DIO (p. 22), you can activate/deactivate the output lines A to H.
  Note that with some older C-843 hardware models, usage of the output lines A to H and usage of the volatile memory on the C-843 board (data recorder, User Profile mode) are mutually exclusive. This means that after the C-843 board was connected in the software, only the functionality called first is available. The selection is reset any time the C-843 board is reconnected. If, for example, the data recorder configuration was queried with DRC? (p. 24) before any DIO command was sent, you can not set the digital output lines until you reconnect the C-843.
- Two (with C-843.21) or four (with C-843.41) input and output lines (TTL, active high) are available on the All-Axes 16-pin IDC connector (J8) of the C-843 board (connects to the sub-D 15m connector of the adapter bracket; see C-843 User Manual for component location). These lines offer direct access to the motion processor. In the software, they are each identified with 1 to n (n = 2 or 4, depending on C-843 model).

The input lines 1 to n can be used to detect the state of an external signal during moves with FED (p. 28), e.g. to find a certain position. Using TRO (p. 54) and CTO (p. 19), you can program the output lines 1 to n to trigger other devices.

Using the TIO? command (p. 53), you can query the number of digital IO lines on the 26-pin IDC connector (J5). The IO lines on the 16-pin IDC connector (J8) are not contained in the TIO? response.

2.5 Joystick Control

For joystick control, connect the joystick device to the PC **before** you start the C-843 GCS DLL (which is called, for example, if you connect to the C-843 in PIMikroMove™ or in PITerminal). Otherwise the joystick will not be recognized by the software. Avoid removing and reconnecting the joystick at run time of the software since this can cause unpredictable results.

Before a joystick can be operated correctly, a calibration routine may need to be performed. Activating a joystick before calibration may cause the joystick controlled axis to start moving even though the joystick is in the neutral position.

When a joystick device is connected directly to the controller and enabled, it is the velocity of the motion axes that is determined by the displacement of the corresponding joystick axes. A linear lookup table defines the velocity response at a certain amplitude of the joystick axis. The values in the lookup table are factors which will during joystick control be applied to the velocity set with VEL (p. 63) for the controller axis.

In open-loop mode (servo off) no joystick operation is possible. When disabling a joystick, the target position is set to the current position for joystick-controlled axes.

JON (p. 35) enables or disables a specified joystick device for joystick operation. While a joystick is active on a controller axis, no move commands are accepted for that axis. With JON? (p. 36) you can query the current activation state of joystick devices.

JAX (p. 35) sets the controller axis which is to be controlled by a joystick axis. Each axis of a controller can only be controlled by one joystick axis. JAX? (p. 35) queries the current assignment of controller axes to joystick axes.

3 GCS Commands

All position or velocity parameter values in GCS commands are in physical units. Note that the physical units used for the different axes by the parameters in GCS commands can be scaled with the DFF command (p. 21). By default this scale factor is 1 for each axis, which means that the "basic" physical units mm or deg (or derived units such as mm/s or deg/s) are used, while a factor of 25.4, for example, converts mm to inches. Use the DFF? command (p. 21) to determine the current scale factor.

3.1 Format of Normal Commands

Commands consist of ASCII characters and have the following format:

<CMD> { [SP<argument>] } LF

where:

CMD	token (mnemonic) of the specific command
SP	one space (ASCII char #32),
<argument>	an identifier or a signed or unsigned value.
LF	LineFeed (ASCII char #10).

Items in braces { } may be repeated, items in brackets [ ] are optional

Use of the spaces SP shown is recommended, although some are not required in the current version definition.

Multiple spaces are not permitted.

Example:

Send: MOVSP1SP10 SP 2SP5SP 3SP20LF

Moves axis 1 to position 10, axis 2 to 5 and axis 3 to 20 (positions in mm)

Format of responses (reports):

Some commands deliver a response having the following format:

X=sVLF

where:

X	axis identifier
s	sign (positive values are transmitted without sign)
LF	LineFeed (ASCII char #10).

Example:

Send: POS? SP1SP2LF

Report: 1=10SP LF
2=5LF

There is one space (char #32) before the LineFeed character on all lines of the response *except* the last line.

In the command descriptions and examples in the rest of this manual, the spaces and line feed characters will not all be individually shown.

3.2 Format of Special Commands

A number of special commands consist of a single ASCII character. These commands and the responses to them are not followed by a terminaltion character.

3.3 Item Identifiers

The identifiers listed below are used to address the appropriate items with GCS commands:

- **Axes**:

  The identifiers are 1 and 2 (with C-843.21) or 1 to 4 (with C-843.41) by default.
  The default identifiers can be changed using SAI (p. 45). The new identifiers must then be used with all axis commands, but are lost when the C-843 board is disconnected in the software.

- **Digital output lines**:

  A to H for the lines concerned by DIO (p. 22). These output lines can be brought out of the PC housing using the sub-D 25f connector of the adapter bracket included with C-843. The corresponding ribbon cable connects to the 26-pin IDC connector (J5) on the C-843 (see C-843 User Manual for details).

  1 and 2 (with C-843.21) or 1 to 4 (with C-843.41) for the lines concerned by CTO (p. 19) and TRO (p. 54). These output lines can be brought out of the PC housing using the sub-D 15m connector of the adapter bracket included with C-843. The corresponding ribbon cable connects to the All-axes 16-pin IDC connector of the C-843 board (J8; see C-843 User Manual for details). Note that older revisions of the adapter bracket do not provide the sub-D 15m connector. If required, contact your PI sales engineer or write info@pi.ws to obtain a new version.

  The identifiers of the digital output lines can not be changed.

- **Digital input lines**:

  A to H for the lines concerned by DIO? (p. 23). These input lines are located on the same 26-pin IDC connector (J5) like the digital output lines A to H, see above for how to bring them out of the PC housing using a bracket.

  1 and 2 (with C-843.21) or 1 to 4 (with C-843.41) for the lines which can be used by FED (p. 28). These input lines are located on the same 16-pin IDC connector (J8) like the digital output lines 1 to 2 (or 4), see above for how to bring them out of the PC housing using a bracket.

  The identifiers of the digital input lines can not be changed.

- **Data recorder tables** (memory tables for recorded data):

  1 to 4; the identifiers can not be changed. See "Data Recording" (p. 6) for more information.

- **Joystick**:

  Each joystick connected to the host PC is identified by a joystick device ID, and each of its axes is identified by a joystick axis ID. Both joystick device IDs and joystick axis IDs start with 1 and can not be changed. See "Joystick Control" (p. 8) for more information.

- **Clusters and Blocks for User Profile Mode**:

  Using the User Profile Mode commands (U*xx*), you can perform motion by processing Datasets in the specified Cluster(s). A Cluster consists of space for a specified number of Datasets (a Dataset specifies a point on a 1-D trajectory; the values it contains are used for trajectory interpolation—time and position are always required, while velocity, acceleration and jerk are optionally required). Data can only be introduced into a Cluster using the UPA command (p. 55), which swaps Datasets into a Cluster from a Block, which in turn was filled using the UPD command (p. 60).

  Possible cluster IDs are A to G, the maximum number of Blocks that can be assigned to a Cluster is 32.

3.4 Rounding Considerations

When converting move commands in (scaled) physical units to the hardware-dependent units required by the motion control layers, rounding errors can occur. The GCS software is so designed that a relative move of x physical units will always result in a relative move of the same number of hardware units. Because of rounding errors, this means, for example, that 2 relative moves of x physical units may differ slightly from one relative move of 2x. When making large numbers of relative moves, especially when moving back and forth, either intersperse absolute moves, or make sure that each relative move in one direction is matched by a relative move of the same size in the other direction.

Examples

Assuming 5 hardware units = 33 x 10^{-6} physical units:

Relative moves smaller than 0.000003 physical units cause move of 0 hardware units.
Relative moves of 0.000004 to 0.000009 physical units cause move of 1 hardware unit.
Relative moves of 0.000010 to 0.000016 physical units cause move of 2 hardware units.
Relative moves of 0.000017 to 0.000023 physical units cause move of 3 hardware units.
Relative moves of 0.000024 to 0.000029 physical units cause move of 4 hardware units.

Hence:

2 moves of 10 x 10^{-6} physical units followed by 1 move of 20 x 10^{-6} in the other direction cause a net motion of 1 hardware unit forward.

100 moves of 22 x 10^{-6} followed by 200 of -11 x 10^{-6} result in a net motion of -100 hardware units.

5000 moves of 2 x 10-6 result in no motion.

3.5 Conversion of Units

The GCS system uses physical units of measure. Most controllers and GCS software have default conversion factors chosen to convert hardware-dependent units (e.g. encoder counts) into mm or degrees ("basic" physical units), as appropriate (SPA parameters 0xE and 0xF). These defaults are generally taken from a stage database. An additional scale factor can be applied to the basic physical unit, making a second physical unit (scaled physical unit) available (see the DFF command, p. 21, and “Motion Parameter” beginning on p. 67).

3.6 Command Overview (Alphabetical)

Note that the full range of GCS functionality is also available with the GCS DLL(s) (see DLL Manual) and, for most controllers, with a COM server.

*IDN? (p. 16)	Get Identify Number
ACC (p. 16)	Set Closed-Loop Acceleration
ACC? (p. 17)	Get Closed-Loop Acceleration
BRA (p. 17)	Switches brake for given axes on or off
BRA? (p. 17)	Returns list of axes with a brake
CLR (p. 18)	CLeaR axis status
CST (p. 18)	Assigns axes to stages and queries “ERR?”. With this command the stage assignment of the connected axes can be changed.
CST? (p. 19)	Returns the name of the connected stage for queried axes
CTO (p. 19)	Set Trigger Configuration
CTO? (p. 20)	Get Trigger Configuration
DEC (p. 20)	Set Closed-Loop Deceleration
DEC? (p. 21)	Get Closed-Loop Deceleration
DFF (p. 21)	Applies a scale factor to the "basic" physical unit of an axis (normally mm or deg, or derived units) to obtain a different physical unit (default factor is 1; a factor of 25.4, for example, converts mm to inches).
DFF? (p. 21)	Returns current scale factor for specified axes (e.g. 25.4 for inches)
DFH (p. 22)	Defines the current position as home position
DFH? (p. 22)	Returns home position in absolute coordinates
DIO (p. 22)	Switches digital outputs on or off
DIO? (p. 23)	Returns digital input values for queried digital inputs
DRC (p. 23)	Set Data Recorder Configuration
DRC? (p. 24)	Get Data Recorder Configuration
DRR? (p. 25)	Get Recorded Data Values
DRT (p. 25)	Set Data Recorder Trigger Source
DRT? (p. 26)	Get Data Recorder Trigger Source
EGE (p. 27)	Set Electronic Gearing State
EGE? (p. 27)	Get Electronic Gearing State
ERR? (p. 28)	get ERRor
FED (p. 28)	Find Edge
FED? (p. 29)	Get Find Edge Parameters
FES? (p. 30)	Get Find Edge State

VEL (p. 63)	Set Velocity
VEL? (p. 64)	Get Velocity
VER? (p. 64)	Get Version
VST? (p. 64)	Returns the names of available stages connected to the controller
#5 (p. 65)	Polls the motion status of the connected axes by sending the single ASCII character 5
#7 (p. 65)	Asks “Is controller ready?”
#9 (p. 66)	Response indicating which axes, if any, are performing User Profile Mode motion
#24 (p. 66)	Stops motion by sending the single ASCII character 24

3.7 Command Reference (alphabetical)

*IDN? (Get Identify Number)

Description:	Reports the device identity number.
Format:	*IDN?
Arguments:	none
Response:	One-line string terminated by line feed, e.g.: C-843 DLL V.100

ACC (Set Closed-Loop Acceleration)

Description:	Set acceleration of given axes. The ACC setting only takes effect when the given axis is in closed-loop operation (servo on). ACC can be changed while the axis is moving. ACC changes the value of the Current closed-loop acceleration parameter (ID 0xB), can also be changed with SPA (p. 47). The maximum value which can be set with the ACC command is given by the Maximum closed-loop acceleration parameter, ID 0x4A (can be changed with SPA (p. 47)). During vectorial moves started with MVE (p. 40), velocities, accelerations and decelerations will be calculated to ensure that all axes follow the path. The current settings for velocity, acceleration and deceleration define the maximum possible values, and the slowest axis determines the resulting velocities.
Format:	ACC {<AxisID> <Acceleration>}
Arguments:	<AxisID> is one axis of the controller <Acceleration> is the acceleration value in physical units/s^2, float

Response:	none
Troubleshooting:	Illegal axis identifier

ACC? (Get Closed-Loop Acceleration)

Description:	Get the current value of the closed-loop acceleration.
Format:	ACC? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	{<AxisID>"="<float> LF} where <float> is the current active closed-loop acceleration value in physical units / s2.

BRA (BRAke)

Description:	Sets brake on or off. CAUTION: Setting the brake with BRA does not affect the servo state of the axis. I.e. if you activate the brake, the servo remains on so that the motor may work against the brake which can cause overheating. In this case, it may be necessary to switch the servo off temporarily. Do not deactivate the brake when the servo is switched off! Otherwise unwanted motion can occur. Unwanted motion could cause irreparable damage to the stage and the application setup. Note that the brake is activated automatically when the servo is switched off with SVO (p. 51), and deactivated when the servo is switched on.
Format:	BRA {<AxisID> <BrakeState>}
Arguments:	<AxisID> is one axis of the controller <BrakeState> can have the following values: 0 = brake off 1 = brake on
Response:	none.
Troubleshooting:	axis has no brake

BRA? (get axes with BRAke)

Description:	Lists the axes with brakes.
Format:	BRA?
Arguments:	none
Response:	identifiers of axes with brakes e.g.: 13 if no axis has a brake, the answer is an empty line.

CLR (CLeaR axis status)

Description:	Clears the axis status. The following actions are done by CLR: Switches the servo on. Resets error to 0. If the stage has tripped a limit switch, CLR will move it away from the limit switch until the limit condition is no longer given, and the target position is set to the current position afterwards.
Format:	CLR [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	none

CST (Change STage)

Description:	Assigns axes to stages. This is done by loading stage parameters suitable for the connected hardware from a stage database. Afterwards, the loaded values must be written to the controller using the INI command (p. 34) to initialize the motion control chip on the C-843 board. Valid stage names can be listed with VST? (p. 64) which reports the content of the stage databases (PIStages2.dat, C843Userstages2.dat, M-xxx.dat files) used by the C843_GCS_DLL. If no stage is connected to the corresponding socket, or if motion of the axis is strictly forbidden, the stage name should be “NOSTAGE”. This deactivates the axis which means that this axis is not available for axis-related commands any more (e.g. motion commands, position queries). You can undo axis deactivation at any time by setting a valid stage name with CST. Note: To connect a stage, always use CST. Do not set the Stage Name parameter (ID 0x3C) with SPA. Otherwise the stage parameters will not be loaded properly from the stage database.
Format:	CST {<AxisID> <Stagename>}
Arguments:	<AxisID> is one axis of the controller <Stagename>: name of the stage to be connected to the axis
Response:	none
Troubleshooting:	Unknown stage

CST? (get stagename)

Description:	Returns the name of the connected stage for queried axes.
Format:	CST? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	{<AxisID>"="<string> LF} where <string> is the name of the stage assigned to the axis. The stage name is read from the Stage Name parameter (ID 0x3C).

CTO (Set Trigger Configuration)

Description:	Configures the trigger output conditions for the given digital output line. The trigger output will only become active when enabled with TRO (p. 54).
Format:	CTO {<TrigOutID> <CTOPam> <Value>}
Arguments:	<TrigOutID> is one digital output line located on the J8 (“All-axes”) connector on the C-843 board (digital output from the motion processor, TTL, max. 5 mA). with C-843.21: can be 1 and 2 with C-843.41: can be 1 to 4 The lines can be brought out of the PC housing using an adapter bracket with a sub-D 15m connector (included with C-843). <CTOPam> is the CTO parameter ID in decimal format, available IDs are 2 = Axis 3 = TriggerMode 7 = Polarity <Value> is the value to which the CTO parameter is set: for <CTOPam> = 2: the axis to connect to the trigger output line for <CTOPam> = 3: 2 = OnTarget 5 = Motion Error 6 = In Motion for <CTOPam> = 7: 0 = Active Low 1 = Active High
Response:	none
Example:	In this example, the trigger output line 1 shall be set from low to high when axis A starts to move. The following parameters must be set: TriggerLine = 1 Axis = A (axis identifier was changed with SAI)

TriggerMode = 6

Polarity = Active High

So you have to send:

CTO 1 2 A 1 3 6 1 7 1

CTO? (Get Trigger Configuration)

Description: Replies with the value set for specified trigger output lines and parameters

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

Arguments: <TrigOutID>: is one digital output line located on the J8 ("All-axes") connector on the C-843 board (digital output from the motion processor, TTL, max. 5 mA)
with C-843.21: can be 1 and 2
with C-843.41: can be 1 to 4
The lines can be brought out of the PC housing using an adapter bracket with a sub-D 15m connector (included with C-843).

<CTOPam> is the CTO parameter ID in decimal format, available IDs are

2 = Axis
3 = TriggerMode
7 = Polarity

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

For <Value> see CTO.

DEC (Set Closed-Loop Deceleration)

Description: Set deceleration of given axes.

The DEC setting only takes effect when the given axis is in closed-loop operation (servo on).

DEC can be changed while the axis is moving.

DEC changes the value of the Current closed-loop deceleration parameter (ID 0xC), can also be changed with SPA (p. 47).

The maximum value which can be set with the DEC command is given by the Maximum closed-loop deceleration parameter, ID 0x4B (can be changed with SPA (p. 47)).

During vectorial moves started with MVE (p. 40), velocities, accelerations and decelerations will be calculated to ensure that all axes follow the path. The current settings for velocity, acceleration and deceleration define the maximum possible values, and the slowest axis determines the resulting velocities.

Format: DEC {<AxisID> <Deceleration>}

Arguments: <AxisID> is one axis of the controller

	<Deceleration> is the deceleration value in physical units/s^2.
Response:	none
Troubleshooting:	Illegal axis identifier

DEC? (Get Closed-Loop Deceleration)

Description:	Get the current value of the closed-loop acceleration.
Format:	DEC? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	{<AxisID>"="<float> LF}
	where
	<float> is the current active closed-loop deceleration value in physical units / s2.

DFF (DeFine Factor)

Description:	Set the scaling factor for physical units. This factor is applied to the counts-per-physical-unit value (parameter 0xE / parameter 0xF). For example, a scaling factor of 25.4 sets the working units to inches. Changing the scale factor will change the numerical results of other commands.
	Note: To change the sacling factor for an axis, always use DFF. Do not set the Scaling Factor parameter (ID 0x12) with SPA. Otherwise the scaling factor will not be applied properly.
Format:	DFF {<AxisID> <ScaleFactor>}
Arguments:	<AxisID> is one axis of the controller
	<ScaleFactor> is the scaling factor to be applied
Response:	none
Troubleshooting:	Illegal axis identifier

DFF? (get factor)

Description:	Gets the scaling factor set by the DFF command (parameter 0x12).
Format:	DFF? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	{<AxisID>"="<float> LF}
	where
	<float> is the scale displacement value
Troubleshooting:	Illegal axis identifier

DFH (DeFine Home

Description:	Makes the current position the new home position (by setting the current position value to 0.00. Due to the change of the home position the limits of the travel range are changed accordingly (ask with TMN? (p. 53) and TMX? (p. 53)).
Format:	DFH [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	none
Troubleshooting:	Illegal axis identifier

DFH? (get home positions)

Description:	Gets actual position which was valid when the last DFH command was sent.
Format:	DFH? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	{<AxisID>"="<float> LF} where <float> is the actual position value of the axis which was valid when the last DFH command was sent
Troubleshooting:	Illegal axis identifier

DIO (set Digital I/O)

Description:	Switches the specified output line(s) to specified state(s). Can be used to trigger external devices. Note that with some older C-843 hardware models, usage of the output lines and usage of the volatile memory on the C-843 board (data recorder, User Profile mode) are mutually exclusive. This means that after the C-843 board was connected in the software, only the functionality called first is available. The selection is reset any time the C-843 board is reconnected. If, for example, the data recorder configuration was queried with DRC? (p. 24) before any DIO command was sent, you can not set the digital output lines until you reconnect the C-843.
Format:	DIO {<DIOID> <OutputOn>}
Arguments:	<DIOID> is one digital output line of the 26-pin IDC connector (J5) of the C-843 board. They can be brought out of the PC housing using an adapter bracket with a sub-D 25f connector (included with C-843) identified with A to H <OutputOn> is the state of the digital output line 0 for LOW , 1 for HIGH
Response:	none

DIO?

Description:	Lists the states of the specified input lines. Can be used to query externally generated signals.
Format:	DIO? [{<DIOID>}]
Arguments:	<DIOID> is one digital input line of the 26-pin IDC connector (J5) of the C-843 board. They can be brought out of the PC housing using an adapter bracket with a sub-D 25f connector (included with C-843) identified with A to H
Response:	{<DIOID>"="<InputOn> LF} where <InputOn> gives the state of the digital input line 1 if input line is high 0 if input line is low

DRC (Set Data Recorder Configuration)

Description:	Set data recorder configuration: determines the data source and the kind of data (RecOption) used for the given data recorder table. The record option set with DRC for data recorder table 1 is automatically changed to "actual position" when a step response measurement is made with STE (p. 49). If one data recorder table is deactivated by choosing record option 0 ("nothing is recorded"), all subsequent tables are deactivated too. The points available for recording are in equal shares allocated to the tables with non-zero record options (for the total number of points to allocate ask SPA? (p. 48) with parameter 0x16000200, for the maximum number of record tables ask TNR? (p. 54)). Note that the data recorder shares the 32,256 points of volatile memory provided on the C-843 card (referred to as “external RAM” in the MotionProcessor Users Guide) with the multi-axis motion profiles which can be created by the User Profile Mode commands (UPx commands beginning on p. 55). It may be necessary to free memory occupied by user-defined motion profiles using the UPC command (p. 58) to have enough memory for data recording. With HDR? (p. 33) you will obtain a list of available record options and trigger options and information about additional parameters and commands regarding data recording. Note that with some older C-843 hardware models, you can not use the data recorder if the digital output lines have been set with DIO (p. 22) before any data-recorder-related command was sent. To use the data recorder, reconnect the C-843 board in the software. For detailed information see "Data Recording" (p. 6).
Format:	DRC {<RecTableID> <Source> <RecOption>}

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

<Source>: is one axis of the controller

<RecOption>: is the kind of data to be recorded:

0 = nothing is recorded

1 = commanded position of axis

2 = actual position of axis

3 = position error of axis

70 = commanded velocity of axis

71 = commanded acceleration of axis

72 = actual velocity of axis

73 = motor output of axis

74 = chipset time

75 = capture register of axis

76 = integral of axis

77 = derivative of axis

78 = event status register of axis

79 = activity status register of axis

80 = signal status register of axis

82 = PID servo error of axis (input of PID servo filter)

Response: none

DRC? (Get Data Recorder Configuration)

Description: Returns settings made with DRC.

Note that with some older C-843 hardware models, you can not use the data recorder if the digital output lines have been set with DIO (p. 22) before any data-recorder-related command was sent. To use the data recorder, reconnect the C-843 board in the software.

Format: DRC? [{<RecTableID>}]

Arguments: <RecTableID>: is one data recorder table of the controller; if omitted settings for all tables are given.

Response: The current DRC settings:

{<RecTableID>"="<Source> <RecOption> LF}

where

<Source>: is one axis of the controller.

<RecOption>: is the kind of data to be recorded, see DRC for a list of the available record options.

DRR? (Get Recorded Data Values)

Description: Reading of the last recorded Data Set.

Reading can take long depending on the number of points to be read!

It is possible to read the data while recording is still in progress.

With <NumberOfPoints> = -1 all points of the last record are read.

If <RecTableID> is omitted, the data from all tables with non-zero record option (see DRC (p.23)) is read.

Step response measurements done with STE (p. 49) can also be read with STE? (p. 50).

With HDR? (p. 33) you will obtain a list of available record options and trigger options and information about additional parameters and commands regarding data recording.

Note that the data recorder shares the 32,256 points of volatile memory provided on the C-843 card (referred to as “external RAM” in the MotionProcessor Users Guide) with the multi-axis motion profiles which can be created by the User Profile Mode commands (UPx commands beginning on p. 55). It may be necessary to free memory occupied by user-defined motion profiles using the UPC command (p. 58) to have enough memory for data recording.

Note that with some older C-843 hardware models, you can not use the data recorder if the digital output lines have been set with DIO (p. 22) before any data-recorder-related command was sent. To use the data recorder, reconnect the C-843.

For detailed information see "Data Recording" (p. 6).

Format: DRR? [<StartPoint> <NumberOfPoints> [{<RecTableID>}]]

Arguments: <StartPoint>: is the start point in the data recorder table, starts with index 1

<NumberOfPoints>: is the number of points to be read per table

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

Response: The recorded data as GCS array, see the separate manual for the GCS array, SM 146E.

DRT (Set Data Recorder Trigger Source)

Description: Data Recorder Trigger source

Defines a trigger source.

By default data recording is triggered when a step response measurement is made with STE (p. 49).

A trigger option set with DRT will become valid for all data recorder tables with non-zero record option.

With HDR? (p. 33) you will obtain a list of available record options and trigger options and information about additional parameters and commands regarding data recording.

Note that with some older C-843 hardware models, you can not use the data recorder if the digital output lines have been set with DIO (p. 22) before any data-recorder-related command was sent. To use the data recorder, reconnect the C-843.

For detailed information see "Data Recording" (p. 6).

Format: DRT {<RecTableID> <TriggerSource> <Value>}

Arguments: <RecTableID>: is one data recorder table of the controller. At present, only 0 is valid, which means that the specified trigger source is set for all data recorder tables.

<TriggerSource>: ID of the trigger source (i.e. trigger option):

0 = default setting; data recording is triggered with STE

1 = any command changing position (e.g. MVR (p. 41), MOV (p. 38), SMO (p. 46))

2 = next command, resets trigger after execution

4 = start on InMotion, stop on AxisSettled

5 = start immediately, stop on AxisSettled

<Value>: must be a dummy (e.g. 0)

Response: none

DRT? (Get Data Recorder Trigger Source)

Description: Returns the Data Recorder Trigger source for the data recorder tables.

Note that with some older C-843 hardware models, you can not use the data recorder if the digital output lines have been set with DIO (p. 22) before any data-recorder-related command was sent. To use the data recorder, reconnect the C-843.

Format: DRT? [{<RecTableID>}]

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

Response: {<RecTableID>"="<TriggerSource> <Value> LF}

where

<TriggerSource> is the ID of the trigger source, see DRT for details

<Value> is a dummy

EGE (Set Electronic Gearing State)

Description: Enable or disable electronic gearing for given axis. Via electronic gearing a "master" and a "geared" (slave) axis are linked, so that motion of the master automatically entails proportional motion of the slave.

Enabling electronic gearing with EGE for an axis means that this axis will be linked as slave to the master axis selected with MAS (p. 37). The gear ratio to be applied can be set with SRA (p. 48). Master selection and ratio setting for an axis are only possible if electronic gearing is disabled for that axis. The ratio setting is checked automatically upon the activation of electronic gearing. If the slave axis is not able to follow the master axis, you have to adapt the ratio value.

Electronic gearing can only be enabled for axes which are referenced. Reference moves are not allowed for axes which are involved in electronic gearing (as master or slave), and their referencing mode can not be changed.

A slave axis can not be commanded directly by move commands. It is only moved when its master axis moves. When motion is commanded for the master axis, the available travel ranges for master and slave are checked.

Joystick operation is possible for master axes. Slave axes connected to a joystick-controlled master will move correspondingly. Slave axes can not be assigned to joystick axes, i.e. they can not be controlled directly by a joystick. If joystick control is enabled for an axis, electronic gearing can not be enabled for that axis.

Format: EGE {<AxisID> <ElecGearState>}

Arguments: <AxisID> is one axis of the controller

<ElecGearState> can have the following values:
1 = enable electronic gearing
0 = disable electronic gearing

Response: none

EGE? (Get Electronic Gearing State)

Description: Gets electronic gearing enable status for given axis.

Format: EGE? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<ElecGearState> LF}

where

<ElecGearState> is the current servo state of the axis:
1 = electronic gearing enabled
0 = electronic gearing disabled

ERR? (get ERRor)

Description: Get Error

Error messages refer to the previously transferred command. For a complete list of possible error codes see p. 79.

Some selected error codes:

0	No error
1	Parameter syntax error
2	Unknown command
5	Unallowable move attempted on unreferenced axis, or move attempted with servo off
7	Position out of limits
8	Velocity out of limits
10	Controller was stopped by command
15	Invalid axis identifier
17	Parameter out of range
22	Axis identifier specified more than once
23	Illegal axis
24	Incorrect number of parameters
26	Parameter missing

Format: ERR?

Arguments: none

Response: Error code

FED (Find Edge)

Description: Moves given axis to a given signal edge.

Servo must be enabled with SVO (p. 51) for the commanded axis prior to using this command (closed-loop operation).

In contrast to the referencing commands (MNL (p. 37), MPL (p. 39), REF (p. 43), FNL (p. 30), FPL (p. 31) and FRF (p. 32)), this command does not change the reference state of the axis and does not set a certain position value at the selected edge. It does move out of the limit condition, therefore the axis motion finishes at the same position as with the corresponding referencing commands.

If multiple axes are given in the command, they are moved synchronously.

The C-843 GCS DLL detects the presence or absence of reference switch and limit switches using parameters (ID 0x14 for reference switch; ID 0x32 for limit switches). According to the values of those parameters, the C-843 GCS DLL enables or disables FED motions to the appropriate signal edges. You can adapt the parameter values to your hardware using SPA (p. 47). See "Motion Parameter" (p. 67) for more information.

FED can be used to measure the physical travel range of a new mechanics and thus to identify the values for the corresponding parameters: the distance from negative to positive limit switch, the distance between the negative limit switch and the reference switch (DISTANCE_REF_TO_N_LIM, parameter ID 0x17), and the distance between reference switch and positive limit switch (DISTANCE_REF_TO_P_LIM, parameter ID 0x2F). See "Travel Range Adjustment" (p. 74) for more information.

Motion commands like FED are not allowed when the joystick is active for the axis.

If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is switched off automatically for that axis, motion of all other axes is stopped immediately, and error code -1024 is set.

Format: FED {<AxisID> <EdgeID> <Param>}

Arguments: <AxisID> is one axis of the controller

<EdgeID> is the type of edge the axis has to move to. See below for available edge types.

<Param> depends on the selected edge and qualifies it. See below for details.

The following edge types with their parameter settings are available:

1 = negative limit switch, <Param> is 0 when the default setting should be used (e.g. from Pistages2.dat), 1 when active high, -1 when active low

2 = positive limit switch, <Param> is 0 when the default setting should be used (e.g. from Pistages2.dat), 1 when active high, -1 when active low

3 = reference switch, <Param> is 0 when the default setting should be used (e.g. from Pistages2.dat), 1 when active high, -1 when active low

4 = autofind additional switch changing its state at a certain position (signal must be connected to the digital input line of the motion chip which corresponds to the axis given in the FED command (All-axes connector J8 on the C-843 board)), <Param> gives the signal state to the left of the edge (high = 1 or low = -1)

Response: none

Troubleshooting: Illegal axis identifier; limit switches and/or reference switch are disabled; servo is off

FED? (Get Find Edge Parameters)

Description: Returns the settings of the last FED motion.

Format: FED? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<EdgeID> <Param> LF}

where

<EdgeID> is the type of edge the axis has moved to. See FED for available edge types.

<Param> depends on the selected edge and qualifies it. See FED for details.

FES? (Get Find Edge State)

Description: Get "finde edge" status. Indicates whether the last FED command (p. 28) was successful or not.

Format: FES? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<FedStatus> LF}

where

<FedStatus> indicates whether an edge was found with the last FED command (= 1) or not (= 0)

FNL (Fast move to Negative Limit)

Description: Moves the given axes synchronously to their negative limit switches. Use FRF? to check whether this was successful or not. Use FNL instead of FRF to reference axes which have no reference sensor (check with REF?). Use MNL instead of FNL if you want move the axes one after another.

Servo must be enabled with SVO (p. 51) for the commanded axis prior to using this command (closed-loop operation).

The reference mode must be set to "1" (default) with the RON command (p. 44) if referencing is to be done by performing a reference move. See "Referencing" (p. 5) for further details.

With the C-843, the negative limit switch of the mechanics is used to determine the negative physical limit of the travel range. The difference of VALUE_AT_REF_POS (parameter ID 0x16) and DISTANCE_REF_TO_N_LIM (parameter ID 0x17) is set as the current position when the axis is at the negative limit switch (value can be negative).

If the soft limits (MAX_TRAVEL_RANGE_POS, parameter 0x15 and MAX_TRAVEL_RANGE_NEG, parameter 0x30) are used to reduce the travel range, the limit switches can not be used for reference moves. The FNL, MNL, FPL and MPL commands will provoke an error message, and only the reference switch can be used for a reference move (FRF or REF).

If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is

switched off automatically for that axis, motion of all other axes is stopped immediately, and error code -1024 is set.

Format: FNL [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller, if omitted, all axes are affected.

Response: none

Troubleshooting: Illegal axis identifier

FPL (Fast move to Positive Limit)

Description: Moves the given axes synchronously to their positive limit switches. Use FRF? to check whether this was successful or not. Use FPL instead of FRF to reference stages which have no reference sensor (check with REF?). Use MPL instead of FPL if you want move the axes one after another.

Servo must be enabled with SVO (p. 51) for the commanded axis prior to using this command (closed-loop operation).

The reference mode must be set to "1" (default) with the RON command (p. 44) if referencing is to be done by performing a reference move. See "Referencing" (p. 5) for further details.

With the C-843, the positive limit switch of the mechanics is used to determine the positive physical limit of the travel range. The sum of VALUE_AT_REF_POS (parameter ID 0x16) and DISTANCE_REF_TO_P_LIM (parameter ID 0x2F) is set as the current position when the axis is at the positive limit switch.

If the soft limits (MAX_TRAVEL_RANGE_POS, parameter 0x15 and MAX_TRAVEL_RANGE_NEG, parameter 0x30) are used to reduce the travel range, the limit switches can not be used for reference moves. The FNL, MNL, FPL and MPL commands will provoke an error message, and only the reference switch can be used for a reference move (FRF or REF).

If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is switched off automatically for that axis, motion of all other axes is stopped immediately, and error code -1024 is set.

Format: FPL [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller, if omitted, all axes are affected.

Response: none

Troubleshooting: Illegal axis identifier

FRF (Fast move to ReFerence position)

Description:	Moves the specified axes synchronously to their reference position. Use FRF? to check whether referencing was successful or not. If the axes have no reference sensors (check with REF?), use FPL, MPL, FNL or MNL instead. Servo must be enabled with SVO (p. 51) for the commanded axis prior to using this command (closed-loop operation). The reference mode must be set to "1" (default) with the RON command (p. 44) if referencing is to be done by performing a reference move. See "Referencing" (p. 5) for further details. The value of the VALUE_AT_REF_POS parameter (ID 0x16) is set as the current position when the axis is at the reference switch. If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is switched off automatically for that axis, motion of all other axes is stopped immediately, and error code -1024 is set.
Format:	FRF [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller, if omitted, all axes are affected.
Response:	none
Troubleshooting:	Illegal axis identifier

FRF?

Description:	Indicates whether the last FNL, MNL, FPL, MPL, FRF or REF command was successful or not.
Format:	FRF? [{<AxisID>}]
Arguments:	<AxisID>: is one axis of the controller
Response:	{<AxisID>"="<uint> LF} where <uint> indicates whether the axis was successfully referenced (=1) or not (=0).
Troubleshooting:	Illegal axis identifier

GOH (GO Home)

Description:	Move given axes to home position. GOH [{<AxisID>}] is the same as MOV {<AxisID> 0} Servo must be enabled for the commanded axis prior to using this command (closed-loop operation).

	If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is switched off automatically for that axis, motion of all other axes is stopped immediately, and error code -1024 is set.
Format:	GOH [{<AxisID>}]
Arguments:	<AxisID>: is one axis of the controller, if omitted, all axes are affected
Response:	none
Troubleshooting:	Illegal axis identifier

HDR? (Get All Data Recorder Options)

Description:	List a help string which contains all information available about data recording (record options and trigger options, information about additional parameters and commands concerned with data recording).
Format:	HDR?
Arguments:	none
Response:	#RecordOptions {<RecordOption>"="<DescriptionString>[of <Channel>]} #TriggerOptions [{<TriggerOption>"="<DescriptionString>}] #Parameters to be set with SPA [{<ParameterID>“=“<DescriptionString>}] #Additional information [{<Command description>"("<Command>")"}] end of help

HLP? (Get List of Available Commands)

Description:	Lists all available commands
Format:	HLP?
Arguments:	none
Response:	Command list

HLT (HaLT)

Description:	Halt the motion of given axes smoothly, with regard to system inertia. Error code 10 is set. After the axes are stopped, their target positions are set to their current positions. #24 (p. 66) and STP (p. 51) in contrast abort current motion as fast as possible for the controller without taking care of systems inertia or oscillations.
Format:	HLT [{<AxisID>}]

Arguments:	<AxisID>: is one axis of the controller, if omitted all axes are halted
Response:	none
Troubleshooting:	Illegal axis identifier

HPA? (Get List of Available Parameters)

Description:	Responds with a help string which contains all available parameters with short descriptions. See "Motion Parameter" (beginning on p. 67) for further details.
Format:	HPA?
Arguments:	none
Response:	{<PamID>"="<string> LF} where <PamID> is the ID of one parameter, hexadecimal format <string> is a string which describes the corresponding parameter. The string has the following format: `<CmdLevel>TAB<MaxItem>TAB<DataType>TAB<FunctionGroupDescription>TAB <ParameterDescription>[{TAB<PossibleValue>"="<ValueDescription>}]`

INI (INItialization)

Description:	Initializes motion control chip for the axis The following actions are done by INI: Writes the stage parameters which were loaded with CST (p. 18) from the stage database to the controller. Switches the servo on. Sets reference mode to 1, i.e. REF, FRF, MNL, FNL, MPL or FPL is required to reference the axis, usage of POS is not allowed. Sets reference state to "not referenced". If the stage has tripped a limit switch, INI will move it away from the limit switch until the limit condition is no longer given, and the target position is set to the current position afterwards. Sets trigger output mode to default configuration.
Format:	INI [{<AxisID>}]
Arguments:	<AxisID>: is one axis of the controller, if omitted all axes are initialized
Response:	none
Troubleshooting:	Illegal axis identifier

JAX (Set Axis Controlled By Joystick)

Description:	Set axis controlled by a joystick connected to the PC. Each axis of the controller can only be controlled by one joystick axis. For joystick control, connect the joystick device to the PC **before** you start the C-843 GCS DLL (which is called, for example, if you connect to the C-843 in PIMikroMove™ or in PITerminal). Otherwise the joystick will not be recognized by the software. Avoid removing and reconnecting the joystick at run time of the software since this can cause unpredictable results. Joystick control must be enabled with JON (p. 35). See “Joystick Control” on p. 8 for more information.
Format:	JAX <JoystickID> <JoystickAxis> [{<AxisID>}]
Arguments:	<JoystickID> is one joystick device connected to the PC, IDs start with 1 <JoystickAxis> is one of the axes of the joystick device, IDs start with 1 <AxisID> is one axis of the controller; if omitted, all controller axes are disconnected from the corresponding <JoystickAxis>
Response:	none

JAX? (Get Axis Controlled By Joystick)

Description:	Get axis controlled by a joystick which is connected to the PC.
Format:	JAX? [{<JoystickID> <JoystickAxis>}]
Arguments:	<JoystickID> is one joystick device connected to the PC <JoystickAxis> is one of the axes of the joystick device
Response:	{<JoystickID> <JoystickAxis>"="{<AxisID> }LF} where <AxisID> is one axis of the controller

JON (Set Joystick Activation Status)

Description:	Enable or disable a joystick device which is connected to the PC. For joystick control, connect the joystick device to the PC **before** you start the C-843 GCS DLL (which is called, for example, if you connect to the C-843 in PIMikroMove™ or in PITerminal). Otherwise the joystick will not be recognized by the software. Avoid removing and reconnecting the joystick at run time of the software since this can cause unpredictable results.

For joystick control of a controller axis, this axis must be assigned to a joystick axis with JAX (p. 35).

While a joystick connected to the C-843 is enabled with the JON command, this joystick controls the axis velocity. In open-loop mode (servo off) no joystick operation is possible. When disabling a joystick, the target position is set to the current position for joystick-controlled axes.

Motion commands like MOV (p. 38) are not allowed when a joystick is active on the axis.

See “Joystick Control” on p. 8 for more information.

Format: JON {<JoystickID> <JoystickOn>}

Arguments: <JoystickID> is one joystick device connected to the controller; IDs start with 1

<JoystickOn> gives the activation state of the joystick device:
1 enables joystick device
0 disables joystick device

Response: none

JON? (Get Joystick Activation Status)

Description: Get activation state of the given joystick device which is connected to the PC.

Format: JON? [{<JoystickID>}]

Arguments: <JoystickID> is one joystick device connected to the controller

Response: {<JoystickID>"="<uint>}

where

<uint> is the joystick activation state: 1 = joystick device enabled, 0 = joystick device disabled

LIM? Indicate Limit Switches

Description: Indicates whether axes have limit switches

The C843_GCS_DLL detects the presence or absence of limit switches using a parameter (ID 0x32). According to the value of this parameter, the C843_GCS_DLL enables or disables the stopping of the motion at the limit switches and reference moves using FNL, MNL, FPL or MPL.

Format: LIM? [{<AxisID>}]

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

Response: {<AxisID>"="<uint> LF}

where

<uint> indicates whether the axis has limit switches (=1) or not (=0).

Troubleshooting: Illegal axis identifier

MAS (Set Master For Electronic Gearing)

Description: Master selection for electronic gearing: the given axis will be linked as slave to the selected master axis.

The gear ratio to be applied can be set with SRA (p. 48). Electronic gearing can be enabled or disabled with EGE (p. 27). Master selection and ratio setting for an axis are only possible if electronic gearing is disabled for that axis.

Via electronic gearing you can only link axes which are located on the same controller board.

It is not possible to link axes so that they form a circle.

Format: MAS {<AxisID> <MasterAxisID>}

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

<MasterAxisID>: identifier of the axis which is to be the master for <AxisID>

Response: none

MAS? (Get Master For Electronic Gearing)

Description: Get the master axis to which the given axis is linked as slave for electronic gearing.

Format: MAS? [{<AxisID>}]

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

Response: {<AxisID>"="<MasterAxisID> LF}

where

<MasterAxisID> is the current valid master for electronic gearing.

MNL (Move to Negative Limit)

Description: Moves the axis to the negative limit switch and indicates whether this was successful or not. Use MNL instead of REF to reference axes which have no reference sensor (check with REF?). For fast (synchronous) moving of multiple axes use FNL instead.

Servo must be enabled with SVO (p. 51) for the commanded axis prior to using this command (closed-loop operation).

The reference mode must be set to "1" (default) with the RON command (p. 44) if referencing is to be done by performing a reference move. See "Referencing" (p. 5) for further details.

With the C-843, the negative limit switch of the mechanics is used to determine the negative physical limit of the travel range. The difference of VALUE_AT_REF_POS (parameter ID 0x16) and DISTANCE_REF_TO_N_LIM (parameter ID

0x17) is set as the current position when the axis is at the negative limit switch (value can be negative).

If the soft limits (MAX_TRAVEL_RANGE_POS, parameter 0x15 and MAX_TRAVEL_RANGE_NEG, parameter 0x30) are used to reduce the travel range, the limit switches can not be used for reference moves. The FNL, MNL, FPL and MPL commands will provoke an error message, and only the reference switch can be used for a reference move (FRF or REF).

If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is switched off automatically for that axis, motion of all other axes is stopped immediately, and error code -1024 is set.

Format: MNL [{<AxisID>}]

Arguments: <AxisID>: is one axis of the controller; if omitted, all axes are affected

Response: 1: if the negative limit switch is reached
0: if the negative limit switch was not reached, or the stage has no limit switches

Troubleshooting: Illegal axis identifier

MOV (MOVe absolute)

Description: Set new absolute target position for given axis. Axes will start moving to the new positions if ALL given targets are within the allowed range and ALL axes can move.

All axes start moving simultaneously.

This command can be interrupted by #24, STP and HLT.

Servo must be enabled for all commanded axes prior to using this command.

Use TMN? (p. 53) and TMX? (p. 53) to ask for the current valid travel range limits.

During a move, a new move command resets the target to a new value and the old one may never be reached.

Motion commands like MOV are not allowed when the joystick is active for the axis.

If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is switched off automatically for that axis, motion of all other axes is stopped immediately, and error code -1024 is set.

Format: MOV {<AxisID> <Position>}

<AxisID> is one axis of the controller

<Position> is the new absolute target position in physical units

Response:	none
Troubleshooting:	Parameter out of limits or Illegal axis identifier

MOV? (read target position)

Description:	Returns last valid commanded target position. The target position is changed by all commands that cause motion (MOV, MVR, REF, STE ...) and with a joystick connected to the PC. Note that MOV? gets the commanded positions. Use POS? to get the current positions.
Format:	MOV? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller
Response:	{<AxisID>"="<float> LF} where <float> is the last commanded target position in physical units
Troubleshooting:	Illegal axis identifier

MPL (Move to Positive Limit)

Description:	Moves the axis to the positive limit switch and indicates whether this was successful or not. Use MPL instead of REF to reference axes which have no reference sensor (check with REF?). For fast (synchronous) moving of multiple axes use FPL instead. Servo must be enabled with SVO (p. 51) for the commanded axis prior to using this command (closed-loop operation). The reference mode must be set to "1" (default) with the RON command (p. 44) if referencing is to be done by performing a reference move. See "Referencing" (p. 5) for further details. With the C-843, the positive limit switch of the mechanics is used to determine the positive physical limit of the travel range. The sum of VALUE_AT_REF_POS (parameter ID 0x16) and DISTANCE_REF_TO_P_LIM (parameter ID 0x2F) is set as the current position when the axis is at the positive limit switch. If the soft limits (MAX_TRAVEL_RANGE_POS, parameter 0x15 and MAX_TRAVEL_RANGE_NEG, parameter 0x30) are used to reduce the travel range, the limit switches can not be used for reference moves. The FNL, MNL, FPL and MPL commands will provoke an error message, and only the reference switch can be used for a reference move (FRF or REF). If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is

switched off automatically for that axis, motion of all other axes is stopped immediately, and error code -1024 is set.

Format: MPL [{<AxisID>}]

Arguments: <AxisID>: is one axis of the controller; if omitted, all axes are affected

Response: 1: if the positive limit switch is reached
0: if the positive limit switch was not reached, or the stage has no limit switches

Troubleshooting: Illegal axis identifier

MVE (Set Target Position for Vectorial Move)

Description: Set new absolute target positions for given axes. This command is intended for simultaneous "vectorial" motion of coupled axes resulting in a straight line.

Servo must be enabled for all commanded axes prior to using this command (closed-loop operation). If servo is switched off or a motion error occurs during motion, all axes are stopped.

If the affected axes are mounted in a way that they move perpendicular to each other, the combined motion of them will describe a linear path. This is achieved by appropriate calculation of accelerations, velocities and decelerations. The current settings for velocity, acceleration and deceleration define the maximum possible values, and the slowest axis determines the resulting velocities.

The target position must be inside the travel range limits. Use TMN? (p. 53) and TMX? (p. 53) to ask for the current valid travel range limits.

The motion can be interrupted by #24 (p. 66), STP (p. 51) and HLT (p. 33).

No other motion commands (e.g. MOV, MVR, MVE) are allowed during vectorial motion.

Motion commands like MVE are not allowed when a joystick is active on the axis.

Format: MVE {<AxisID> <Position>}

Arguments: <AxisID> is one axis of the controller

<Position> is the new absolute target position in physical units.

Response: none

MVR (MoVe relatiVe)

Description: Move given axes relative to the last commanded target position.

The new target position is calculated by adding the given value <Distance> to the last commanded target value.

Servo must be enabled for the commanded axis prior to using this command (closed-loop operation).

The target position must be inside the travel range limits. Use TMN? (p. 53) and TMX? (p. 53) to ask for the current valid travel range limits; and MOV? (p. 39) for the current target.

The motion can be interrupted by #24 (p. 66), STP (p. 51) and HLT (p. 33).

During a move, a new move command resets the target to a new value and the old one may never be reached.

Motion commands like MVR are not allowed when the joystick is active for the axis.

If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is switched off automatically for that axis, motion of all other axes is stopped immediately, and error code -1024 is set.

Format: MVR {<AxisID> <Distance>}

<AxisID> is one axis of the controller.

<Distance> gives the distance to move; the sum of the distance and the last commanded target position is set as new target position (in physical units).

Response: none

Troubleshooting: Parameter out of limits, illegal axis identifier

Example:

Send: MOV 1 0.5

Note: This is an absolute move.

Send: POS? 1

Receive: 1=0.500000

Send: MOV? 1

Receive: 1=0.500000

Send: MVR 1 2

Note: This is a relative move.

Send: POS? 1

Receive: 1=2.500000

Send: MVR 1 2000

Note: New target position of axis 1 would exceed motion range. Command is ignored, i.e. the target position remains unchanged, and the axis does not move.

Send: MOV? 1

Receive: 1=2.500000

Send: POS? 1

Receive: 1=2.500000

ONT? (axis ON Target)

Description: Reports whether the specified axis has reached the target position.

The on-target status becomes "true" when the current position stays in the settle window (parameter 0x36) for at least the settle time (parameters 0x3F and 0x38). If the settle time is set to 0, then the axis is on target when the trajectory has finished, irrespective of the current position.

Format: ONT? [{<AxisID>}]

<AxisID> is one axis of the controller.

Response: {<AxisID>"="<uint> LF}

where

<uint> = "1" when the specified axis is on-target, "0" otherwise.

Troubleshooting: Illegal axis identifier

POS (set real POSition)

Description: Sets the current position (does not cause motion)

Setting the current position with POS is only possible when the reference mode is set to "0", see RON (p. 44).

An axis is considered as "referenced" when the position was set with POS (for more information refer to "Referencing", p. 5).

The minimum and maximum commandable positions (TMN? (p. 53), TMX? (p. 53)) are not adapted when a position is set with POS. This may result in target positions which are allowed by the software and cannot be reached by the hardware. Also target positions are possible which can be reached by the hardware but are denied by the software. Furthermore, the home position can be outside of the physical travel range after using POS.

Format: POS { <AxisID> <Position>}

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

<Position> is the new current position in physical units.

Response: none

Troubleshooting: Reference mode ON,
Illegal axis identifier

POS? (read real POSition)

Description:	Reports the current position of the specified axes
Format:	POS? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller; if omitted, all axes are queried
Response:	{<AxisID>"="<float> LF} where <float> is the current axis position in physical units
Troubleshooting:	Illegal axis identifier

REF (move to REFerence position)

Description:	Moves the specified axes to the reference position and sets the reference state to "reference OK". If the stage has no reference sensor (check with REF?), use MPL or MNL instead. For fast (synchronous) referencing of multiple axes use FRF instead. Servo must be enabled with SVO (p. 51) for the commanded axis prior to using this command (closed-loop operation). The reference mode must be set to "1" (default) with the RON command (p. 44) if referencing is to be done by performing a reference move. See "Referencing" (p. 5) for further details. The value of the VALUE_AT_REF_POS parameter (ID 0x16) is set as the current position when the axis is at the reference switch. If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is switched off automatically for that axis, motion of all other axes is stopped immediately, and error code -1024 is set.
Format:	REF [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller; if omitted, all axes are moved
Response:	1: if the reference position is reached 0: if the reference position was not reached, or the stage has no reference sensor
Troubleshooting:	Illegal axis identifier

REF? (Indicate Reference Switch)

Description:	Lists axes which have a reference sensor
Format:	REF? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller; if omitted, all axes are queried
Response:	{<AxisID>"="<uint> LF}

where

<uint> indicates whether the axis has a reference switch (=1) or not (=0).

Troubleshooting: Illegal axis identifier

RON (Set Reference Mode)

Description: Set reference mode of given axes. For more information refer to "Referencing" (p. 5) and "Travel Range Adjustment" (p. 74).

Format: RON {<AxisID> <ReferenceOn>}

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

<ReferenceOn> can be 0 or 1:

0= referencing moves with FRF, REF, FNL, MNL, FPL and MPL are not possible, absolute position must be set with POS (p. 42) to reference the axis.

1= FRF, REF, FNL, MNL, FPL or MPL is required to reference the axis, usage of POS is not allowed.

1 is default.

Response: none

Troubleshooting: Illegal axis identifier

RON? (Get Reference Mode)

Description: Get reference mode of given axes.

Format: RON? [{ <AxisID>}]

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

Response: {<AxisID>"="<ReferenceOn> LF}

where

<ReferenceOn> is the current reference mode of the controller, see RON

Troubleshooting: Illegal axis identifier

RTR (Set Record Table Rate)

Description: Sets the record table rate, i.e. the number of servo-loop cycles to be used in data recording operations. Settings larger than 1 make it possible to cover longer time periods.

RTR changes the value of the Data Recorder Table Rate parameter (ID 0x16000000), can also be changed with SPA (p. 47).

The duration of the recording can be calculated as follows:

Rec. Duration = Servo Cycle Time * RTR value * Number of Points

where

Servo Cycle Time is 410 µs for the C-843

Number of Points is the length of data recorder table, max. 32,256 (if only the first table is configured for recording and User Profile Mode is not used)

For more information see "Data Recording" (p. 6).

Note that with some older C-843 hardware models, you can not use the data recorder if the digital output lines have been set with DIO (p. 22) before any data-recorder-related command was sent. To use the data recorder, reconnect the C-843 board in the software.

Format: RTR <RecordTableRate>

Arguments: <RecordTableRate> is the table rate to be used for recording operations (unit: number of servo-loop cycles), must be an integer value larger than zero

Response: None

RTR? (Get Record Table Rate)

Description: Gets the current record table rate, i.e. the number of servo-loop cycles used in data recording operations.

Format: RTR?

Arguments: None

Response: <RecordTableRate> is the table rate used for recording operations (unit: number of servo-loop cycles)

SAI (Set Axis Identifier)

Description: Sets the axis identifiers for the given axes.

After it was set with SAI, the new axis identifier must be used as <AxisID> in all axis-related commands.

An axis can be identified with one character. Use TVI? (p. 54) to ask for valid characters.

The axis identifier is reset to default (see p. 10) with any new connection.

Format: SAI {<AxisID> <NewAxisID>}

Arguments: <AxisID> is one axis of the controller

<NewAxisID> is the new identifier to use for the axis

Response: none

Troubleshooting: Illegal axis identifier or duplicate axis identifier

SAI? (get axis identifier)

Description:	Gets the axis identifiers. See also “Item Identifiers” on p. 10.
Format:	SAI? [ALL]
Arguments:	ALL is optional. Since the C-843 allows for axis deactivation (set axis to NOSTAGE using CST, p. 18), ALL ensures that the answer also includes the axes which are "deactivated".
Response:	{<AxisID>}LF <AxisID> is one axis of the controller.

SMO (Set Motor Output)

Description:	Set control value for the motor output directly to move the axis. Trajectory generator and servo filter are omitted. Servo must be switched off (open-loop operation) when using this command. CAUTION: Limit switches are deactivated so that the stage can run into the hard stop. This can cause damage to equipment.
Format:	SMO {<AxisID> <MotorOutput>}
Arguments:	<AxisID> is one axis of the controller <MotorOutput> is the control value for the motor output (dimensionless), range: -32767 to 32767. The sign of the value determines the direction of motion: -32767 corresponds to maximum motor command output in negative direction of motion, 32767 to the maximum motor command output in positive direction of motion.
Response:	none
Troubleshooting:	Illegal axis identifier, parameter out of range, servo ON

SMO? (Get Motor Output)

Description:	Gets the actual value of the motor output register. In servo-on mode (closed-loop operation) the actual value is set by the servo filter. In servo-off mode (open-loop operation) this value is set by the SMO command.
Format:	SMO? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller; if omitted, all axes are queried
Response:	{<AxisID>"="<float> LF} where <float> is the last valid control value for the motor output (dimensionless), see SMO
Troubleshooting:	Illegal axis identifier

SPA (Set Parameter)

Description: Sets parameters for the specified axes.

CAUTION

Wrong values of the parameters may lead to improper operation or damage of your hardware. Be careful when changing parameters.

Whenever you start working, first the CST (p. 18) and INI (p. 34) commands must be sent: CST loads stage parameters suitable for your hardware from a stage database, and INI writes the loaded values to the controller to initialize the motion control chip on the C-843 board. Afterwards, you can change parameters using SPA. See “Parameter List” on p. 68 for valid parameter IDs. Note that some parameters should normally not be changed (see marks in the parameter list).

Note that all parameter changes with SPA are temporarily (done in C843_GCS_DLL and in the motion chip of the C-843 board). To store parameter values, save them to the C843UserStages2.dat stage database (see PIMikroMove™ manual for more information).

With HPA? (p. 34) you can obtain a list of the available parameters and their IDs.

See "Motion Parameter" on p. 67.for more information.

Format: SPA {<ItemID> <PamID> <PamValue>}

Arguments: <ItemID> is the item for which a parameter is to be changed in volatile memory. An item can be an axis (see p. 10 for the default axis identifiers, can be changed with SAI, p. 45) or the whole system (identifier is 1).

<PamID> is the parameter ID, can be written in hexadecimal or decimal format.

<PamValue> is the value to which the given parameter of the given item is set

Response: none

Troubleshooting: Illegal item identifier

Example: SPA **1** 1 100 **1** 2 25 **1** 3 200 **2** 1 150 **2** 2 35 **2** 3 300LF

Sets P=100 for Axis **1**
I=25 for Axis **1**
D=200 for Axis **1**
P=150 for Axis **2**
I=35 for Axis **2**
D=300 for Axis **2**

(In this example bold characters marks the axis identifiers for better understanding; parameter IDs are given in decimal format.)

SPA? (Get Parameter)

Description: Get the value of a parameter of a given axis.

With HPA? (p. 34) you can obtain a list of the available parameters and their IDs.

See "Motion Parameter" on p. 67.for more information.

Format: SPA? [{<ItemID> <PamID>}]

Arguments: <ItemID> is the item for which a parameter is to be queried. An item can be an axis (see p. 10 for the default axis identifiers, can be changed with SAI, p. 45) or the whole system (identifier is 1).

<PamID> is the parameter ID, can be written in hexadecimal or decimal format.

Response: {<ItemID> <PamID>"="<PamValue> LF}

where

<PamValue> is the value of the given parameter for the given item

Troubleshooting: Illegal item identifier

Example: The PID values for axis 1 can be displayed with the following command:

SPA? **1** 1 **1** 2 **1** 3

Response:

1SP1=100 SP LF
1SP2=25 SP LF
1SP3=200 LF

(In this example bold characters marks the axis identifiers for better understanding; parameter IDs are given in decimal format.)

SRA (Set Gear Ratio For Electronic Gearing)

Description: Gear ratio setting for electronic gearing: the given ratio is applied when electronic gearing is enabled for the given axis which is then the slave:
Ratio = Travel of Master / Travel of Slave

The master to which the axis is to be linked as slave can be set with MAS (p. 37). Electronic gearing can be enabled or disabled with EGE (p. 27). Master selection and ratio setting for an axis are only possible if electronic gearing is disabled for that axis.

Format: SRA {<AxisID> <GearRatio>}

Arguments:	<AxisID> is one axis of the controller
	<GearRatio> is the gear ratio setting to be set
Response:	none

SRA? (Get Gear Ratio For Electronic Gearing)

Description:	Get the gear ratio setting of the given axis (ratio = Master/Slave).
Format:	SRA? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller; if omitted, all axes are queried
Response:	{<AxisID>"="<float> LF}
	where
	<float> is the current valid gear ratio setting for electronic gearing.

SRG? (Read register)

Description:	Read the values of the specified registers.
Format:	SRG? [{<AxisID> <RegisterID>}]
Arguments:	<AxisID>: one axis of the controller.
	<RegisterID>: the ID of the specified register, can be 1 = Event Status register 2 = Activity Status register 3 = Signal Status register 4 = Signal Sense mask For detailed descriptions of the registers see the Motion Processors User Guide on the C-843 CD. Note that the states of the digital input and output lines located on the J8 (“All-axes”) connector on the C-843 board are provided in the Signal Status register (AxisIn and AxisOut bits).
Response:	{ItemId RegisterId=Value}
Troubleshooting:	Illegal axis identifiers

STE (STEp function)

Description:	Starts performing a step and recording up to 32,640 position values as the axis response.
	A "step" is a motion pos. a → pos. b, performed relative to the current position.
	The recorded data can be read with STE? (p. 50) or with DRR? (p. 25).
	STE automatically changes the DRC (p. 23) record option of the first data recorder table to "actual position" (2).
	The DRC configuration of record tables 2 to 4 is not changed by STE. This makes it possible to record additional data, but these data can only be read with DRR?.

The number of points which are recorded with STE depends on the DRC settings: the points available for recording are in equal shares allocated to the tables with non-zero DRC record options (for the total number of points to allocate ask SPA? (p. 48) with parameter 0x16000200, maximum value is 32,256).

STE automatically resets the data recorder sample period to 1 (see also RTR (p. 44) or SPA (p. 47) parameter 0x16000000).

If you do not want to deal with the restrictions induced by STE, you can use MVR (p. 41) instead to make a relative step move and configure recording according to your requirements.

Note that the data recorder shares the 32,256 points of volatile memory provided on the C-843 card (referred to as “external RAM” in the MotionProcessor Users Guide) with the multi-axis motion profiles which can be created by the User Profile Mode commands (UPx commands beginning on p. 55). It may be necessary to free memory occupied by user-defined motion profiles using the UPC command (p. 58) to have enough memory for data recording.

Note that with some older C-843 hardware models, you can not use the data recorder if the digital output lines have been set with DIO (p. 22) before any data-recorder-related command was sent. To use the data recorder, reconnect the C-843.

Motion commands like STE are not allowed when the joystick is active for the axis.

Format: STE <AxisID> <Distance>

Arguments: <AxisID> is one axis of the controller

<Distance> is the height of the step in (scaled) physical units

Response: none

Troubleshooting: Illegal axis identifier,

The target caused by the given step height is out of limits: use TMN? (p. 53) and TMX? (p. 53) to ask for the current valid travel range limits.

STE? (Read Step Response)

Description: Lists the recorded step response (positions).

It is recommended to read the recorded position values with STE? immediately after STE was used.

Format: STE? <AxisID> <StartPoint> <NumberOfValues>

Arguments: <AxisID> is one axis of the controller

<StartPoint> is the start index (begins with 0)

<NumberOfValues> number of values to be read

Response:	{<AxisID>"="<float> LF} where <float> is the current position in physical units
Troubleshooting:	Illegal axis identifier, no step response values stored, too many values requested

STP (Stop Motion)

Description:	Stops all motion abruptly. STP stops all motion caused by move commands (e.g. MOV, MVR, MVE, GOH, STE) and referencing commands (FNL, MNL, FPL, MPL, FRF and REF). Sets error code to 10. After the axes are stopped, their target positions are set to their current positions. This command is identical in function to #24 (p. 66) which should be preferred when the controller is performing time-consuming tasks. HLT (p. 33) in contrast to STP stops motion with given system deceleration with regard to system inertia.
Format:	STP
Arguments:	None
Response:	None

SVO (set SerVO on or off)

Description:	Sets servo-control state ON or OFF (closed-loop/open-loop operation). When switching from open-loop to closed-loop mode, the target is set to the current position to avoid jumps of the mechanics. The current servo state affects the applicable move commands: servo-control on: use MOV (p. 38), MVR (p. 41), GOH (p. 32), MVE (p. 40) or joystick control (p. 8) servo-control off: only direct motor output (velocity-related) is possible using SMO (p. 46); note that the limit switches are disabled. Servo must be switched on before reference moves can be started with FRF (p. 32), REF (p. 43), FNL (p. 30), MNL (p. 37), FPL (p. 31) or MPL (p. 39). When servo is switched off while the axis is moving, the axis stops. If the position error of an axis falls out of the window formed by the Maximum position error parameter, the servo is switched off automatically for that axis,

motion of all other axes is stopped immediately, and error code -1024 is set.

Stages with brake: The brake is activated automatically when the servo is switched off with SVO, and deactivated when the servo is switched on.

CAUTION

Before setting servo-control off make sure that the stage can not perform unwanted motion in servo-off mode. Unwanted motion could cause irreparable damage to the stage and the application setup.

Setting the brake with BRA (p. 17) does not affect the servo state of the axis. I.e. if you activate the brake, the servo remains on so that the motor may work against the brake which can cause overheating. In this case, it may be necessary to switch the servo off temporarily. Do not deactivate the brake when the servo is switched off! Otherwise unwanted motion can occur.

Format: SVO {<AxisID> <ServoState>}

Arguments: <AxisID> is one axis of the controller

<ServoState> can have the following values:
0 = servo off (open-loop operation)
1 = servo on (closed-loop operation)

Response: none

Troubleshooting: Illegal axis identifier

SVO? (get servo status)

Description: Gets servo-control state of given axes.

Format: SVO? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller; if omitted; all axes are queried

Response: {<AxisID>"="<ServoState> LF}

where

<ServoState> is the current servo state of the axis:
0 = servo off (open-loop operation)
1 = servo on (closed-loop operation)

Troubleshooting: Illegal axis identifier

TIO? (Tell Digital I/O Lines)

Description:	Tell number of installed digital I/O lines TIO? queries the number of digital IO lines on the 26-pin IDC connector (J5). The IO lines on the 16-pin IDC connector (J8) are not contained in the TIO? response. See “External Triggering / Signaling” on p. 7 for more information.
Format:	TIO?
Arguments:	none
Response:	I=<uint1> O=<uint2> where <uint1> is the number of digital input lines. <uint2> is the number of digital output lines.

TMN? (Tell Minimum Travel Value)

Description:	Get the minimum commandable position in physical units. The minimum commandable position is defined by the MAX_TRAVEL_RANGE_NEG parameter, ID 0x30.
Format:	TMN? [{ <AxisID>}]
Arguments:	<AxisID> is one axis of the controller; if omitted; all axes are queried
Response	{<AxisID>"="<float> LF} where <float> is the minimum commandable position in physical units

TMX? (Tell Maximum Travel Value)

Description:	Get the maximum commandable position in physical units. The maximum commandable position is defined by the MAX_TRAVEL_RANGE_POS parameter, ID 0x15.
Format:	TMX? [{ <AxisID>}]
Arguments:	<AxisID> is one axis of the controller; if omitted; all axes are queried
Response	{<AxisID>"="<float> LF} where <float> is the maximum commandable position in physical units

TNR? (Tell Number Of Record Tables)

Description:	Tell number of data recorder tables currently available on the controller.
Format:	TNR?
Arguments:	none
Response:	<uint> is the number of data recorder tables which are currently available

TRO (Set Trigger Output State)

Description:	Enables or disables the TRrigger Output mode which was set with CTO (p. 19) for the given digital output line.
Format:	TRO {<TrigOutID> <TrigMode>}
Arguments:	<TrigOutID> is one digital output line of the controller; see CTO (p. 19) <TrigMode> can have the following values: 0 = trigger output disabled 1 = trigger output enabled
Response:	none

TRO? (Get Trigger Output State)

Description:	Gets trigger output mode enable status for given digital output line.
Format:	TRO? [{<TrigOutID>}]
Arguments:	<TrigOutID> is one digital output line of the controller; see CTO (p. 19) for details; if omitted response includes all output lines
Response:	{<TrigOutID>"="<TrigMode> LF} where <TrigMode> is the current enable state of the trigger output line: 0 = trigger output disabled 1 = trigger output enabled

TVI? (Tell Valid axis Identifiers)

Description:	Gets a string with characters which can be used for axis identifiers. Use SAI (p. 45) to change the axis identifiers and SAI? (p. 46) to ask for the current valid axis identifiers.
Format:	TVI?
Arguments:	none
Response:	<string> is a list of characters

UPA (User Profile Activate)

Description:	For User Profile Mode: Activates the specified Data Block of the specified Cluster, i.e. the Datasets of the Block replace Datasets of the Cluster, starting at the defined Offset of the Block plus 1). If the axis assigned to Cluster is moving, waits until no Datasets in the section of the Cluster which the Block will occupy are participating in the current interpolation. UPC, UPB and UPD must have been used to define the Cluster, the Block and to fill the Block. See separate Technical Note A000T0014_100_UserProfileModeSoftware for more information. Note that with some older C-843 hardware models, you can not use the User Profile Mode if the digital output lines have been set with DIO (p. 22) before any User-Profile-Mode-related command was sent. To use the User Profile Mode, reconnect the C-843.
Format:	UPA {<Cluster> <Block>}
Arguments:	<Cluster>: the designator of one User Profile Cluster, can be A-G <Block>: is a number which identifies a Data Block of the given Cluster; that Block is activated (swapped into its place in the Cluster)
Response:	uint: indicates whether activation was successful (= 1) or not (= 0)
Example 1:	Send: UPA A 0 Note: Swap Block 0 of Cluster A into Cluster A
Example 2:	UPA A 1 B 1 Note: Swap Block 1 of Cluster A into Cluster A and Block 1 of Cluster B into Cluster B

UPA? (User Profile Get Block Number)

Description:	For User Profile Mode: Reads the number of the Block from which the specified Dataset originated. UPC must have been used to define the Cluster See separate Technical Note A000T0014_100_UserProfileModeSoftware for more information. Note that with some older C-843 hardware models, you can not use the User Profile Mode if the digital output lines have been set with DIO (p. 22) before any User-Profile-Mode-related command was sent. To use the User Profile Mode, reconnect the C-843.
Format:	UPA? {<Cluster> <DataSetOffset>}

Arguments: <Cluster>: the designator of one User Profile Cluster, can be A-G

<DataSetOffset> is the offset of a Dataset in the given Cluster; for that Dataset, the number of the most recently activated block, if any, will be determined

Response: <Cluster> <DataSetOffset>"="<Value>

One line per cluster/dataset; each line terminated by LF; all but the last line with a preceding SP; if <Value> = -1 then no block has been activated for the corresponding dataset

Example 1: Send: UPA? A 0

Response: A0=6

Note: Dataset at offset 0 of Cluster A originated in Block 6 of Cluster A.

Example 2: Send: UPA? A 1 B 2

Response: A1=6SP LF
B2=-1LF

Note: Dataset at offset 1 of Cluster A originated in Block 6 of Cluster A. No Block has been activated with a Dataset for offset 2 of Cluster B

UPB (User Profile Block Create or Modify)

Description: For User Profile Mode: Either creates or modifies a Data Block for given Cluster, setting given parameters

or

Deletes all Blocks of the given Cluster

UPC must have been previously issued for this Cluster.

See separate Technical Note A000T0014_100_UserProfileModeSoftware for more information.

Note that with some older C-843 hardware models, you can not use the User Profile Mode if the digital output lines have been set with DIO (p. 22) before any User-Profile-Mode-related command was sent. To use the User Profile Mode, reconnect the C-843.

Format: UPB {<Cluster> <Block> <ParamID> <Value>}

or

UPB <Cluster> -1 -1 -1 (deletes all Blocks of the Cluster)

Arguments: <Cluster>: one Cluster for User Profile Mode

<Block>: the Data Block number; if not already defined, Block will be created.

<ParamID>: code for the parameter to be set to <Value>, see below; parameters whose IDs are omitted keep previous values, if any, else are set to 0.

<Value>: the value to be set, as explained for each <ParamID> below

<ParamID>: can be as follows:

1 Length; then <Value> gives the maximum number of Datasets for the Data Block. Length plus Offset (param. 3) must not exceed the length of the Cluster which is given with UPC. Changing the length of a Block after placing data in it with UPD may cause loss of data and parameter values!

2 Type; then <Value> will give the type of the block:
0 = complete motion block (start & end velocity = 0)
1 = start block (start velocity = 0)
2 = constant travel block
3 = stop block (end velocity = 0); Note: when the HLT command is issued for an axis moving in User Profile Mode, the offsets of the Stop Blocks of the associated Cluster are compared with the index of the Datasets currently being interpolated. The Stop Block which would “come soonest” is then activated.

3 Offset; then <Value> is the offset of the position the Block will take in the Cluster when it is swapped in (activated). If <Value> is 0, when the Block is activated, the first Dataset of the Block will replace the first Dataset of the Cluster, the second will replace the second, etc. up to the end of the block (see UPD).

Response: uint: indicates whether creation/modification was successful (= 1) or not (= 0)

Example: Send: UPB A 0 1 25

Note: Defines block 0 for Cluster A with room for 25 Datasets. With no further definition the other parameters remain at 0, i.e. it will be a Complete-Motion Block (type = 0), and, when activated, will be swapped into the Cluster with offset = 0

UPB? (User Profile Read Block Configuration)

Description: For User Profile Mode: Reads Data Block configuration of given Block of given Cluster.

See separate Technical Note A000T0014_100_UserProfileModeSoftware for more information.

Note that with some older C-843 hardware models, you can not use the User Profile Mode if the digital output lines have been set with DIO (p. 22) before any User-Profile-Mode-related command was sent. To use the User Profile Mode, reconnect the C-843.

Format: UPB? {<Cluster> <Block> <ParamID>}

One line per value queried; each line terminated by LF; all but the last line with a preceding SP

Arguments: <Cluster>: one Cluster for User Profile Mode

<Block>: the Data Block number; if not already defined, Block will be created.

<ParamID>: code for the parameter to be queried.

Response: <Cluster> <Block> <ParamID>"="<Value>

<ParamID>: can be as follows:

1 Length; then <Value> gives the maximum number of Datasets for the Data Block.

2 Type; then <Value> will give the type of the block:
0 = complete motion block (start & end velocity = 0)
1 = start block (start velocity = 0)
2 = constant travel block
3 = stop block (end velocity = 0); Note: when the HLT command is issued for an axis moving in User Profile Mode, the offsets of the Stop Blocks of the associated Cluster are compared with the index of the Datasets currently being interpolated. The Stop Block which would “come soonest” is then activated.

3 Offset; then <Value> is the offset of the position the Block will take in the Cluster when it is swapped in (activated).

Example: Send: UPB? A 0 1

Receive: A 0 1 = 25

Note: Queries parameter 1 (Length) of block 0 of Cluster A

UPC (User Profile Cluster Configuration)

Description: For User Profile Mode: Creates a Cluster or deletes all Clusters and all Blocks.

A Cluster consists of space for a specified number of Datasets (a Dataset specifies a point on a 1-D trajectory; the values it contains are used for trajectory interpolation—time and position are always required, while velocity, acceleration and jerk are optionally required).

Upon creation, each Cluster is assigned to one axis. Multiple Clusters can be assigned to the same axis.

Data can only be introduced into a Cluster using the UPA command, which swaps Datasets into a Cluster from a Block, which in turn was filled using the UPD command.

The Buffer Run command (UPR) specifies which Cluster is used for User Profile Mode motion.

The maximum number of Blocks that can be assigned to a Cluster is 32.

See separate Technical Note A000T0014_100_UserProfileModeSoftware for more information.

Note that with some older C-843 hardware models, you can not use the User Profile Mode if the digital output lines have been set with DIO (p. 22) before any User-Profile-Mode-

related command was sent. To use the User Profile Mode, reconnect the C-843.

Note that the data recorder (p. 6) shares the 32,256 points of volatile memory provided on the C-843 card (referred to as “external RAM” in the MotionProcessor Users Guide) with the multi-axis motion profiles which can be created by the User Profile Mode commands.

Format: UPC {<AxisID> <Cluster> <NumberOfDataSets> <LengthOfDataSets>}

or

UPC $ $ -1 -1 (deletes all Clusters and all Blocks, if any)

Arguments: <AxisID>: axis designator of one axis of a controller which supports User Profile Mode

<Cluster>: one Cluster for User Profile Mode, it must be A-G.

<NumberOfDataSets>: gives the maximum number of Datasets in the Cluster, i.e. the length of the Cluster

<LengthOfDataSets>: gives the number of values in each Dataset of the Cluster.
Valid numbers are:
2: time, position
3: time, position, velocity
4: time, position, velocity, acceleration
5: time, position, velocity, acceleration, jerk (derivative of acceleration)

Response: <uint>: indicates whether the setup was successful (= 1) or not (= 0)

Example 1: Send: UPC 1 A 25 4 2 B 25 4

Note: Sets up Cluster A assigned to axis 1 holding 25 Datasets which each hold 4 values, and Cluster B, assigned to axis 2, will have the same size.

Example 2: Send: UPC 1 A 25 4 1 B 25 4

Note: Sets up Cluster A assigned to axis 1 holding 25 Datasets which each hold 4 values. Cluster B, assigned to axis 1 as well, will have the same size.

UPC? (User Profile Read Cluster Configuration)

Description: For User Profile Mode: Reads configuration of given Cluster or of all defined Clusters

See separate Technical Note A000T0014_100_UserProfileModeSoftware for more information.

Note that with some older C-843 hardware models, you can not use the User Profile Mode if the digital output lines have been set with DIO (p. 22) before any User-Profile-Mode-related command was sent. To use the User Profile Mode, reconnect the C-843.

Format:	UPC? [{<Cluster>}]
Arguments:	<Cluster>: one Cluster for User Profile Mode, it must be A-G.
Response:	{<Cluster>"="<AxisID> <NumberOfDataSets> <LengthOfDataSets>LF} If no error, one line per cluster queried; each line terminated by LF; all but the last line with a preceding SP <AxisID>: axis designator of one axis of the controller <NumberOfDataSets>: gives the maximum number of Datasets in the Cluster, i.e. the length of the Cluster <LengthOfDataSets>: gives the number of values in each Dataset of the Cluster. Valid numbers are: 2: time, position 3: time, position, velocity 4: time, position, velocity, acceleration 5: time, position, velocity, acceleration, jerk (derivative of acceleration) If no Clusters are defined, or if one or more specifed Clusters are not defined error -1073 is set and a blank response is given (LF)
Example:	Send: UPC? A Receive: A=1 10 4 Note: Cluster A is assigned to axis 1 and has 10 Datasets containing 4 values each.

UPD (User Profile Data)

Description:	For User Profile Mode: Writes one Dataset to the given Data Block of the given Cluster. A Dataset contains the data for one fixed point to be used in trajectory interpolation—time and target position are always required elements in a Dataset, while velocity, acceleration and jerk are only required if so specified in the UPC (Cluster configuration) command for the given Cluster. If the number of arguments given with UPD is smaller than the number required by the UPC settings, or if the Dataset index given with UPD does not exist, the write procedure will not be successful. If the number of arguments given with UPD is greater than the number required by the UPC settings, the values which are not part of the UPC configuration will be ignored while the other values will be accepted. Only one Dataset can be written at a time. UPC and UPB must have been issued for this Cluster and Block before issuing UPD. See separate Technical Note A000T0014_100_UserProfileModeSoftware for more information.

Note that with some older C-843 hardware models, you can not use the User Profile Mode if the digital output lines have been set with DIO (p. 22) before any User-Profile-Mode-related command was sent. To use the User Profile Mode, reconnect the C-843.

Format: UPD <Cluster> <Block> <DataSetOffset> <Time> <Pos> [<Vel> [<Acc> [<Jerk>]]]

Arguments: <Cluster>: is one Cluster for User Profile Mode

<Block>: is the Data Block number

<DataSetOffset>: is the index of the Dataset in the block

<Time>: is the amount of time that will be available during User Profile Mode motion to achieve the position and other values specified in the next Dataset. The value, given in seconds, must be a multiple of the controller's cycle time. The value 0 ends User Profile Mode motion (required for the last Dataset if its Dataset index is smaller than the number of Datasets specified for the Cluster with UPC, or if the wrap-around feature is not to be used).

<Pos>: is the target position

<Vel>: is the velocity

<Acc>: is the acceleration

<Jerk>: is the jerk (rate of change of acceleration)

Response: <uint>: indicates whether the write procedure was successful (= 1) or not (= 0)

Example 1: Send: UPD A 0 0 10 50 2 0.5 0.5

Note: Write the following data to Block 0 of Cluster A: Dataset 0: time: 10 s, position 50, velocity 2, acceleration 0.5 and jerk 0.5. If Cluster A is set up to hold Datasets of length less than 5, one or more of the latter values (from right to left) will be ignored.

Example 2: Send: UPD A 0 33 10 60

Note: Write data to Block 0 of Cluster A, block 0, Dataset 33: time: 10 s, position 60. Because only time and position are included, Cluster A must be set up for Dataset length 2, and with room for at least 33 Datasets.

UPD? (User Profile Data Read)

Description: For User Profile Mode: Reads given Dataset of given block of given Cluster.

See separate Technical Note A000T0014_100_UserProfileModeSoftware for more information.

Note that with some older C-843 hardware models, you can not use the User Profile Mode if the digital output lines have been set with DIO (p. 22) before any User-Profile-Mode-related command was sent. To use the User Profile Mode, reconnect the C-843.

Format: UPD? {<Cluster> <Block> <DataSetOffset>}

Arguments: <Cluster>: is one Cluster for User Profile Mode

<Block>: is the Data Block number

<DataSetOffset>: is the index of the Dataset in the block

Response: {<Cluster> <Block> <DataSetOffset>"="<Time> <Pos> {<Vel> {<Acc> {<Jerk>}}}LF}

One line per dataset queried; each line terminated by LF; all but the last line with a preceding SP

<Time>: is Dataset validity time

<Pos>: is the target position

<Vel>: is the velocity

<Acc>: is the acceleration

<Jerk>: is the jerk (rate of change of acceleration)

Example: Send: UPD? A 1 25

Receive: A 1 25=10 55 23

Note: Dataset 25 of block 1 in Cluster A holds 10 as time, 55 as the position and 23 as velocity

UPR (User Profile Run)

Description: For User Profile Mode: Starts User Profile Mode motion, by processing Datasets in the specified Cluster(s); all axes specified start synchronously.

Motion can be stopped with HLT (if there is at least one Stop Block associated with the Cluster), STP or #24.

Before issuing UPR, each specified Cluster must have been created by UPC, and Datasets swapped into it by UPA from at least one Block, which must have been defined by UPB and filled with UPD.

See separate Technical Note A000T0014_100_UserProfileModeSoftware for more information.

Note that with some older C-843 hardware models, you can not use the User Profile Mode if the digital output lines have been set with DIO (p. 22) before any User-Profile-Mode-related command was sent. To use the User Profile Mode, reconnect the C-843.

Format: UPR {<AxisID> <Cluster> <StartOffset>}

Arguments: <AxisID>: axis designator of one axis of a controller which supports User Profile Mode

<Cluster>: a Cluster for User Profile Mode. It must have been previously assigned to <AxisID> (using UPC).

<StartOffset>: the Dataset index in <Cluster> with which processing is to start.

	The axis-cluster-startindex group can be repeated to specify synchronous motion of multiple axes (up to 3 in the current version). Of course, no axis or cluster may be specified more than once.
Response:	uint: indicates whether the examination of the Datasets in the Cluster(s) was successful (= 1) or not (= 0)
Example:	Send: UPR 1 A 20 2 B 20
	Note: Starts axis 1 with Cluster A and axis 2 with Cluster B simultaneously, both running from Dataset with index 20 simultaneously.

VEL (Set Velocity)

Description:	Set velocity of given axes.
	VEL can be changed while the axis is moving.
	The VEL setting only takes effect when the given axis is in closed-loop operation (servo on).
	The lowest possible value for <Velocity> is 0.
	VEL changes the value of the Current closed-loop velocity parameter (ID 0x49) in volatile memory (can also be changed with SPA, p. 47).
	The maximum value which can be set with the VEL command is given by the Maximum closed-loop velocity parameter, ID 0xA (can be changed with SPA).
	During vectorial moves started with MVE (p. 40), velocities, accelerations and decelerations will be calculated to ensure that all axes follow the path. The current settings for velocity, acceleration and deceleration define the maximum possible values, and the slowest axis determines the resulting velocities.
Format:	VEL {<AxisID> <Velocity>}
Arguments:	<AxisID> is one axis of the controller
	<Velocity> is the velocity value in physical units/s.
Response:	none
Problem Solver:	Illegal axis identifiers

VEL? (Get Velocity)

Description:	Get the current velocity value. If all arguments are omitted, gets current value of all axes.
Format:	VEL? [{<AxisID>}]
Arguments:	<AxisID> is one axis of the controller; if omitted, all axes are queried
Response:	{<AxisID>"="<float> LF} where <float> is the current active velocity value in physical units / s.

VER? (Get Version)

Description:	Returns the VERsion of the firmware and the underlying drivers and DLLs.
Format:	VER?
Arguments:	none
Response:	{<string1>":" <string2> [<string3>]LF} where <string1> is the name of the component <string2> is the version information of the component <string1> <string3> is an optional note

Example:	The response for C-843 is something like:
	`C843_GCS_DLL.dll: V1.5.1.109`
	`PMD: 21200023`

VST? (Get available Stages)

Description:	Lists the names of stages selectable by the CST command (p. 18). The list comprises the content of the stage databases (PIStages2.dat, C843Userstages2.dat, M-xxx.dat files) used by the C843_GCS_DLL. See “Parameter Databases” on p. 77 for more information.
Format:	VST?
Arguments:	none
Response:	{<string> LF} where <string> is a stage name.

#5 (Poll Motion Status)

Description:	Reports the current motion status.
Format:	#5 (single ASCII character number 5, no LF)
Arguments:	none
Response:	The answer <uint> is bit-coded and returned as the hexadecimal sum (without leading “0x”) of the following codes: 1 = first axis is moving 2 = second axis is moving 4 = third axis is moving 8 = fourth axis is moving Note that when no stage is connected to an axis (NOSTAGE is returned by CST?, p. 19), this axis is not included in the bit-coded answer (see example 3 below)!
Example 1:	0 indicates motion of all axes complete
Example 2:	7 indicates axes 1, 2 and 3 are moving
Example 3:	Note: Presumed that axes A, B, D are connected, while axis C is not connected (NOSTAGE is returned by CST?), then the bit-coded answer is as shown below Send: MOV A 10 Note: Only axis A is moving Send: #5 Receive: 1 Send: MOV B 10 Note: Only axis B is moving Send: #5 Receive: 2 Send: MOV D 10 Note: Only axis D is moving Send: #5 Receive: 4

#7 (Controller Ready?)

Description:	Test if controller is ready to perform a new command. Use #5 instead of #7 to verify if motion has finished.
Format:	#7 (single ASCII character number 7, no LF)
Arguments:	none
Response:	Controller is ready: <uint> = ASCII 177, Hex B1, "±" Controller is not ready: <uint> = ASCII 176, Hex B0, "°" (e.g. performing a referencing command)

#9 (User Profile Mode Motion in Progress?)

Description:	Test if User Profile (UP) mode motion is in progress
Format:	#9 (single ASCII character number 9, no LF)
Arguments:	none
Response:	The answer <uint> is bit-coded and returned as the hexadecimal sum (without leading “0x”) of the following codes: 1 = first axis is moving in UP mode 2 = second axis is moving in UP mode 4 = third axis is moving in UP mode 8 = fourth axis is moving in UP mode Note that when no stage is connected to an axis (NOSTAGE is returned by CST?, p. 19), this axis is not included in the bit-coded answer. If, for example, the third axis was set to NOSTAGE, then the #9 response would show “4” if the fourth axis would move in UP mode.
Example:	Response: 5 (single ASCII character 53): axes 1 and 3 are moving in User Profile mode

#24 (Stop all axes)

Description:	Stop all axes abruptly. #24 stops all motion caused by move commands (e.g. MOV, MVR, MVE, GOH, STE) and referencing commands (FNL, MNL, FPL, MPL, FRF and REF). Sets error code to 10. After the axes are stopped, their target positions are set to their current positions. This command is identical in function to STP (p. 51). HLT (p. 33) in contrast to #24 and STP stops motion with given system deceleration with regard to system inertia.
Format:	#24 (single ASCII character number 24, no LF)
Arguments:	none
Response:	none

4 Motion Parameters

4.1 Parameter Handling

The C843_GCS_DLL supports a mechanism which mirrors the hardware basics of the connected stage and the required closed-loop control settings in parameters. The parameter values have to be adjusted properly before initial operation of a stage. For that purpose, the CST (p. 18) and INI (p. 34) commands must be sent whenever you start working: CST loads stage parameters suitable for your hardware from a stage database, and INI writes the loaded values to the controller to initialize the motion control chip on the C-843 board. This is done on a per-axis basis. Note that PIMikroMove™ performs this procedure automatically upon stage selection. See “Parameter Databases” (p. 77) for the available stage databases.

With HPA? (p. 34) you can obtain a list of all available parameters with information about each (e.g. short descriptions). The current valid parameter values can be read with the SPA? command (p. 49). Using the VST? command (p. 64) you can list the names of stages selectable by the CST command (i.e. the stages for which parameter sets are available in the stage databases).

Using the "general" modification command SPA (p. 47), parameters can be changed. In addition to this general modification command, there are some commands which change certain specific parameters (see table in “Parameter List” below). Note that all that parameter changes are temporarily (done in C843_GCS_DLL and in the motion chip of the C-843 board). To store parameter values, save them to the C843UserStages2.dat stage database (see PIMikroMove™ manual for more information).

CAUTION

Wrong values of the parameters may lead to improper operation or damage of your hardware. Be careful when changing parameters.

The interrelation of the hardware-dependent parameters 0x15, 0x16, 0x17, 0x2F and 0x30 is described in "Travel Range Adjustment” (p. 74).

4.2 Parameter List

For additional information regarding most of the parameters listed below, see the User's Guide and the Programmer's Reference for the PMD Navigator MC2140CP Motion Processor which are on the C-843 CD.

Parameter ID (hexadecimal / decimal)	Data Type	Parameter Description	Changing with SPA not recommended	Possible Values/Notes
0x1 / 1	FLOAT	P-term for position control		0 to 32767 Gives the P-term of the proportional-integral-derivative (PID) algorithm applied by the servo filter. See MotionProcessors User Guide for more information.
0x2 / 2	FLOAT	I-term for position control		0 to 32767 Gives the I-term of the proportional-integral-derivative (PID) algorithm applied by the servo filter. See MotionProcessors User Guide for more information.
0x3 / 3	FLOAT	D-term for position control		0 to 32767 Gives the D-term of the proportional-integral-derivative (PID) algorithm applied by the servo filter. See MotionProcessors User Guide for more information.
0x4 / 4	FLOAT	I-limit for position control		0 to 32767 Gives the integration limit for the accumulated error processed by the servo filter. See MotionProcessors User Guide for more information.
0x5 / 5	FLOAT	Velocity feed forward		0 to 32767 Gives the velocity feed forward term of the servo filter. See MotionProcessors User Guide for more information.
0x6 / 6	FLOAT	Output scaling factor		0 to 65536 Gives an output scale factor which is applied by the servo filter to produce the final motor output. See MotionProcessors User Guide for more information.
0x7 / 7	FLOAT	Output bias value		0 to 32767 Gives a bias value which is added by the servo filter to produce the final motor output. When an axis is subject to a net external force in one direction (such as a vertical axis pulled downward by gravity), the servo filter can compensate for it by adding a constant DC bias to the filter output. See MotionProcessors User Guide for more information.

Parameter ID (hexadecimal / decimal)	Data Type	Parameter Description	Changing with SPA not recommended	Possible Values/Notes
0x8 / 8	FLOAT	Maximum position error (user unit)		0 to 32767 Used for stall detection. If the position error (i.e. the absolute value of the difference between current position and commanded position) in closed-loop operation exceeds the given maximum, the C843_GCS DLL sets error code -1024 (“Motion error”), the servo will be switched off automatically for the axis concerned, and motion of all axes is stopped immediately.
0x9 /9	FLOAT	Motor output limit	X	Gives a limiting value for the output of the servo filter. The motor output limit prevents the filter output from exceeding a boundary magnitude in either direction. If the filter produces a value greater than the limit, the motor command takes the limiting value. The motor limit applies only in closed-loop operation (servo on). It does not affect the motor command value set with SMO (p. 46) in open-loop operation. See MotionProcessors User Guide for more information.
0xA / 10	FLOAT	Maximum closed-loop velocity (user unit/s)	X	> 0 Gives the maximum value for parameter 0x49.
0xB / 11	FLOAT	Current closed-loop acceleration (user unit/s^2) also changed by ACC command (p. 16)		Gives the current acceleration, limited by parameter 0x4A
0xC / 12	FLOAT	Current closed-loop deceleration (user unit/s^2) also changed by DEC command (p. 20)		Gives the current deceleration, limited by parameter 0x4B
0xD / 13	FLOAT	Maximum allowed jerk (user unit/s^3)	X	Gives the maximum allowed jerk. Limits the slope of the acceleration. In S-curve profile mode, reducing the jerk will smoothen the motion.
0xE / 14	FLOAT	Numerator of the counts-per-physical-unit factor	X	1 to 2147483647 for each parameter. The counts-per-physical-unit factor determines the “user” unit for closed-loop motion commands. When you change this factor, all other parameters

Parameter ID (hexadecimal / decimal)	Data Type	Parameter Description	Changing with SPA not recommended	Possible Values/Notes
0xF / 15	FLOAT	Denominator of the counts-per-physical-unit factor	X	whose unit is based on the “user” unit are adapted automatically, e.g. closed-loop velocity and parameters regarding the travel range. Note: To customize your physical unit use DFF command instead (see parameter 0x12).
0x10 / 16	FLOAT	Output mode	X	0 = Analog 1 = PWM
0x11 / 17	FLOAT	Invert direction	X	-1 = invert direction 1 = do not invert
0x12 / 18	FLOAT	Scaling factor, also changed by DFF command (p. 21)		This factor can be used to change the physical unit of the stage, e.g. a factor of 25.4 converts a physical unit of mm to inches. It is recommended to use DFF to change this factor.
0x13 / 19	FLOAT	Rotary stage	X	1 = rotary stage 0 = no rotary stage
0x14 / 20	FLOAT	Stage has a reference switch	X	1 = stage has a reference switch 0 = stage has no reference switch
0x15 / 21	FLOAT	MAX_TRAVEL_RANGE_POS The maximum travel in positive direction (user unit)	X	"Soft limit", based on the home (zero) position. If the soft limit is smaller than the position value for the positive limit switch (which is given by the sum of the parameters 0x16 and 0x2F), the positive limit switch can not be used for referencing. Can be negative.
0x16 / 22	FLOAT	VALUE_AT_REF_POS The position value at the reference switch position (user unit)	X	The position value which is to be set when the mechanics performs a reference move to the reference switch. Must be set even if no reference switch is present in the mechanics because it is used to to calculate the position values to be set after reference moves to the limit switches.
0x17 / 23	FLOAT	DISTANCE_REF_TO_N_LIM The distance between reference switch and negative limit switch (user unit)	X	Represents the physical distance between the reference switch and the negative limit switch integrated in the mechanics. Must be set even if no reference switch is present in the mechanics because the position is set to the difference of VALUE_AT_REF_POS and DISTANCE_REF_TO_N_LIM when the mechanics performs a reference move to the negative limit switch.

Parameter ID (hexadecimal / decimal)	Data Type	Parameter Description	Changing with SPA not recommended	Possible Values/Notes
0x18 / 24	FLOAT	Limit switch polarity	X	0 = positive limit switch active high (pos-HI), negative limit switch active high (neg-HI) 1 = positive limit switch active low (pos-LO), neg-HI 2 = pos-HI, neg-LO 3 = pos-LO, neg-LO
0x19 / 25	FLOAT	Stage type	X	0 = DC motor 2 = Voice coil
0x1A / 26	FLOAT	Stage has brakes	X	0 = Stage has no brakes 1 = Stage has brakes
0x1B / 27	FLOAT	Current profile mode	X	0 = Trapezoidal point-to-point 1 = Velocity contouring (use only with rotary stages) 2 = S-curve point-to-point 4 = User profile mode (read only) Note that S-curve profile mode does not support changes to any of the profile parameters while the axis is in motion. See MotionProcessors User Guide for more information.
0x2F / 47	FLOAT	DISTANCE_REF_TO_P_LIM The distance between reference switch and positive limit switch (user unit)	X	Represents the physical distance between the reference switch and the positive limit switch integrated in the mechanics. Must be set even if no reference switch is present in the mechanics because the position is set to the sum of VALUE_AT_REF_POS and DISTANCE_REF_TO_P_LIM when the mechanics performs a reference move to the positive limit switch.
0x30 / 48	FLOAT	MAX_TRAVEL_RANGE_NEG The maximum travel in negative direction (user unit)	X	"Soft limit", based on the home (zero) position. If the soft limit is larger than the position value for the negative limit switch (which is given by the difference of the parameters 0x16 and 0x17), the negative limit switch can not be used for referencing. Can be negative.
0x31 / 49	FLOAT	Invert reference switch signal	X	1 = invert reference switch signal 0 = do not invert
0x32 / 50	FLOAT	Stage has limit switches; enables / disables the stopping of the motion at the limit switches	X	0 = Stage has limit switches 1 = Stage has no limit switches

Parameter ID (hexadecimal / decimal)	Data Type	Parameter Description	Changing with SPA not recommended	Possible Values/Notes
0x36 / 54	FLOAT	Settle window (counts)		0 to 2^{31} The settle window is centered around the target position. The on-target status becomes "true" when the current position stays in this window for at least the settle time (parameters 0x3F / 0x38).
0x38 / 56	FLOAT	Settle time (cycles)		0 to 32767 Used for on-target detection: The on-target status becomes "true" when the current position stays in the settle window (parameter 0x36) for at least the settle time. If the settle time is set to 0, then the axis is on target when the trajectory has finished, irrespective of the current position. Parameter 0x38 has the same value as parameter 0x3F, but given in number of cycles. If parameter 0x3F is changed, parameter 0x38 is adapted automatically and vice versa.
0x3C / 60	CHAR	Stage name	X	Maximum 31 characters Can be queried with CST?. Note: To connect a stage, always use CST. Do not set the stage name parameter with SPA. Otherwise the stage parameters will not be loaded properly from the stage database.
0x3F / 63	FLOAT	Settle time (s)		The same value as parameter 0x38, but given in seconds. If parameter 0x38 is changed, parameter 0x3F is adapted automatically and vice versa.
0x49 / 73	FLOAT	Current closed-loop velocity (user unit/s) also changed by VEL command (p. 63)		Gives the current velocity, limited by parameter 0xA
0x4A / 74	FLOAT	Maximum closed-loop acceleration (user unit/s^2)	X	Gives the maximum value for parameter 0xB
0x4B / 75	FLOAT	Maximum closed-loop deceleration (user unit/s^2)	X	Gives the maximum value for parameter 0xC
0x50 / 80	FLOAT	Velocity for reference moves and find-edge moves (user unit/s)		Gives the maximum velocity to be used for reference moves with REF, FRF, MPL, FPL, MNL, FNL and for find-edge moves with FED. If set to 0, reference moves or find-edge moves are not possible.

Parameter ID (hexadecimal / decimal)	Data Type	Parameter Description	Changing with SPA not recommended	Possible Values/Notes
0x59 / 89	FLOAT	Acceleration feed forward		0 to 32767 Gives the acceleration feed forward term of the servo filter. See MotionProcessors User Guide for more information.
0x16000000 / 369098752	INT	Data Recorder Table rate (cycles) also changed by RTR command (p. 44)		Gives the data recorder sampling period (default value is one servo cycle). You can cover longer periods by increasing this value.
0x16000200 / 369099264	INT	Data Recorder Maximum record points	X	Gives the total number of points available for data recording (max. 32,256). The points available are in equal shares allocated to the tables with non-zero DRC record options. Note that the data recorder shares the 32,256 points of volatile memory provided on the C-843 card (referred to as “external RAM” in the MotionProcessor Users Guide) with the multi-axis motion profiles which can be created by the User Profile Mode commands (UPx commands beginning on p. 55).

4.3 Transmission Ratio and Scaling Factor

The physical unit used for the stages (i.e. for the axes of the controller) results from the following interrelation of some stage parameters:

$$PU = \left(Cnt / \frac{CpuN}{CpuD} \right) \times SF$$

$$Cnt = \left(PU / SF \right) \times \frac{CpuN}{CpuD}$$

Name	Number*	Description
PU	-	Physical Unit
Cnt	-	Counts
CpuN	0xE	Numerator of the counts per physical unit factor
CpuD	0xF	Denominator of the counts per physical unit factor
SF	0x12	Scaling factor**

*Number is the parameter ID for SPA and SPA?, see also parameter list beginning on p. 68.

**See DFF command.

The "Counts per physical unit factor" which results from parameter 0xE divided by parameter 0xF includes the physical transmission ratio and the resolution of the stage.

CAUTION

To customize the physical unit of a stage do not change parameter 0xE and parameter 0xF but use DFF (p. 21) instead. Although DFF has the same effect as changing parameter 0x12 with SPA, you should only use DFF and not SPA to modify the scaling factor.

Example: If you set with DFF a value of 25.4 for an axis, the physical unit for this axis is converted from mm to inches.

4.4 Travel Range Adjustment

The figures below give a universal hardware scheme of a positioning stage with incremental sensor, reference and limit switches. To work with such a stage, the stage parameters must be adjusted properly (see "Parameter Handling" on p. 67 for how to modify parameter values).

In the example shown in the first figure, the travel range, i.e. the distance from negative to positive limit switch is 20 mm, the distance between the negative limit switch and the reference switch is 8 mm, and the distance between reference switch and positive limit switch is 12 mm. These hardware properties are represented by the following parameters:

DISTANCE_REF_TO_N_LIM (parameter ID 0x17) = 8

DISTANCE_REF_TO_P_LIM (parameter ID 0x2F) = 12

To allow for flexible localization of the home position (0), a special parameter is provided. It gives the offset between reference switch and home position which is to be valid for the stage after a reference move (see below). In the example, the home position is to be located at the negative limit switch after a reference move, and hence the offset between reference switch and home position is 8 mm.

VALUE_AT_REF_POS (parameter ID 0x16) = 8

To allow for absolute moves, either an absolute "initial" position can be set with the POS command, or the stage can perform a reference move to a known position where a defined position value will be set as the current position (see "Referencing" on p. 5 for further details). By default, a reference move is required. In the example, known positions for reference moves are given by the reference switch and the limit switches. Depending on the switch used for the reference move, a certain combination of the above-mentioned parameters is used to calculate the position to be set at the end of the move:

- Reference switch (REF or FRF command): the stage is moved to the reference switch, and the value of VALUE_AT_REF_POS is set as the current position.
- Negative limit switch (MNL or FNL command): the stage is moved to the negative limit switch and the difference of VALUE_AT_REF_POS and DISTANCE_REF_TO_N_LIM is set as the current position (can be negative).
- Positive limit switch (MPL or FPL command): the stage is moved to the positive limit switch and the sum of VALUE_AT_REF_POS and DISTANCE_REF_TO_P_LIM is set as the current position.

It is furthermore possible to set "soft limits" which establish a "safety distance" which the stage will not enter on both ends of the travel range. Those soft limits always refer to the current home position (0; in the example located at the negative limit switch after a reference move). The soft limits are to be

deactivated in the example so that the corresponding parameters must be as follows:

MAX_TRAVEL_RANGE_POS (parameter ID 0x15) = 20 mm

MAX_TRAVEL_RANGE_NEG (parameter ID 0x30) = 0 mm

(This means that the stage can move 20 mm in positive direction, starting from the home position, and 0 mm in negative direction, starting from the home position.)

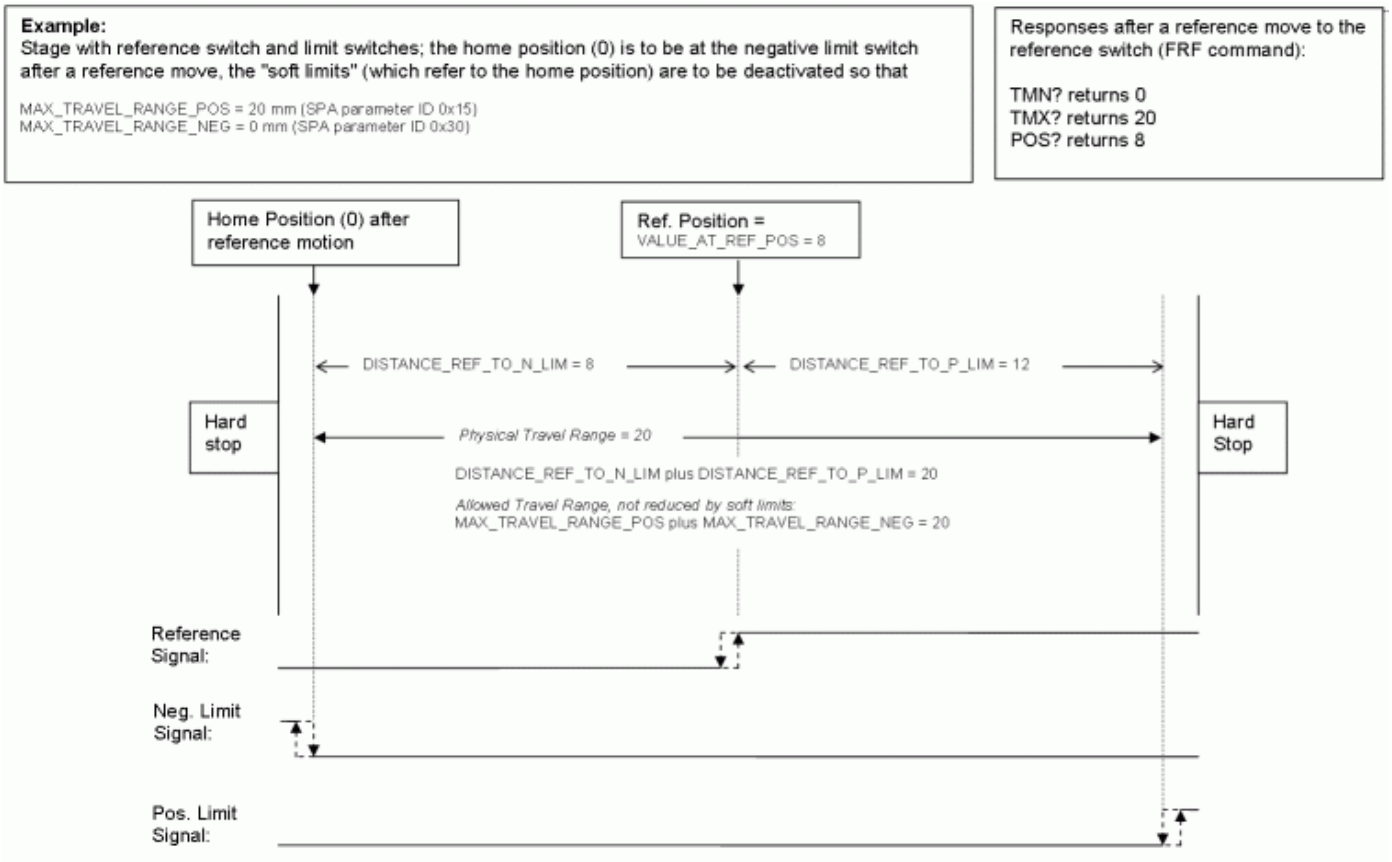


Figure 1: Positioning stage and corresponding controller parameters

Now in the same example, a "safety distance" is to be established on both ends of the travel range by setting soft limits, and the home position is to be located at about 1/3 of the distance between the new negative end of the travel range and the reference switch. The limit switches can not be used for reference moves anymore.

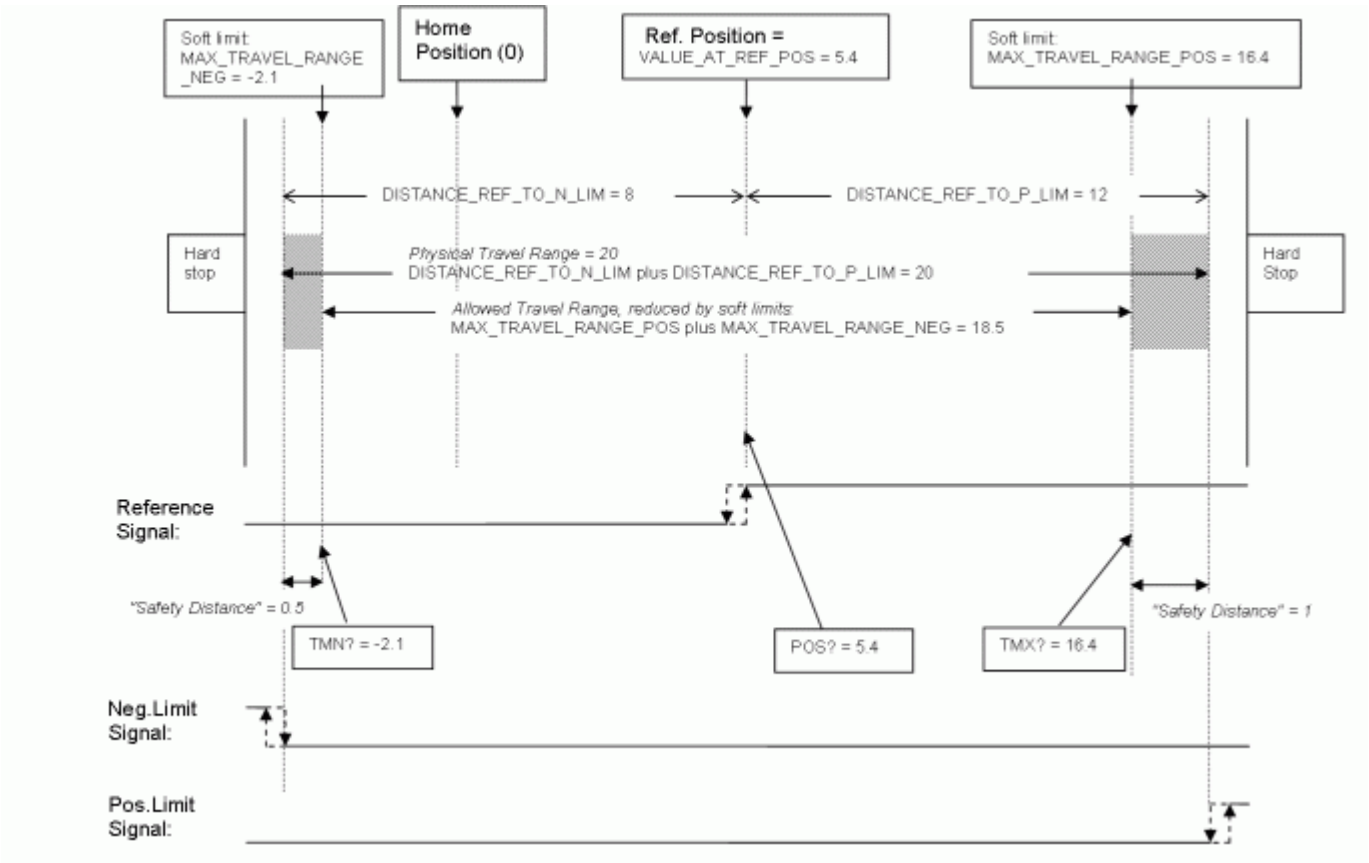


Figure 2: Positioning stage, soft limits set in the controller to reduce the travel range

After the stage was referenced again by moving it to the reference switch (with REF or FRF), the following responses will be given:

TMN? returns -2.1

TMX? returns 16.4

POS? returns 5.4

CAUTION

If the soft limits (MAX_TRAVEL_RANGE_POS and MAX_TRAVEL_RANGE_NEG) are used to reduce the travel range, the limit switches can not be used for reference moves. MNL, FNL, MPL and FPL will provoke an error message, and only the reference switch can be used for a reference move (REF or FRF).

Be careful when setting the values for VALUE_AT_REF_POS, MAX_TRAVEL_RANGE_POS and MAX_TRAVEL_RANGE_NEG because there is no plausibility check.

The soft limits may not be outside of the physical travel range:
MAX_TRAVEL_RANGE_POS ≤ DISTANCE_REF_TO_P_LIM + VALUE_AT_REF_POS
MAX_TRAVEL_RANGE_NEG ≥ VALUE_AT_REF_POS - DISTANCE_REF_TO_N_LIM
Otherwise, reference moves to the limit switches would have incorrect results because the values of the soft limits would be set at the end of the referencing procedure.

Be careful when referencing the stage by setting an initial absolute position with POS since the values for MAX_TRAVEL_RANGE_POS and MAX_TRAVEL_RANGE_NEG are not adapted. In the worst case, the soft limits will now be outside of the physical travel range, and the stage will no longer be able to move since the move commands check the soft limit settings.

4.5 Parameter Databases

The C843_GCS_DLL and the GCS-based host software from PI use multiple databases for stage parameters:

- **PIStages2.dat** contains parameter sets for all standard stages from PI and is automatically installed on the host PC with the setup. It cannot be edited; should changes in the file become necessary, you must obtain a new version from PI and install it on your host PC (see “Updating PIStages2.dat”, p. 78).
- **C843UserStages2.dat** allows you to create and save your own stages (see “Editing User DAT Files” in the PIMikroMove™ User Manual). This database is created the first time you connect stages in the host software (i.e. the first time the C843_qVST() or C843_CST() functions of the C843_GCS library are used which is the case, for example, when VST? or CST are sent in PITerminal or the *Select connected stages* startup step is performed in PIMikroMove™).
- **M-xxx.dat** files contain parameter sets for custom stages delivered by PI. Those files are provided by CDs which come with the stages and have to be copied to the host PC according to the accompanying instructions. M-xxx.dat files can not be edited; should changes become necessary, you must obtain a new version from PI.

The PIStages2.dat, C843UserStages2.dat and M-xxx.dat databases are located in the ...\PI\GcsTranslator directory on the host PC. The location of the PI directory is that specified upon installation, usually in C:\Documents and Settings\All Users\Application Data (Windows XP) or C:\ProgramData (Windows Vista). If this directory does not exist, the program that needs the stage databases will look in its own directory. In PIMikroMove™, you can use the *Version Info* item in the controller menu or the *Search for controller software* item in the *Connections* menu to identify the GcsTranslator path.

Notes for users which have already installed older versions of C843_GCS_DLL, PIMikroMove™ and PIStageEditor:

- The format of the stage parameter (DAT) files has changed (more parameters provided), realized by a file version change from 1 to 2. Note that PIStages and C843Userstages DAT files with version 2 contain a "2" in their file name, e.g. PIStages2.dat (instead of PIStages.dat for version 1).
- Existing C843Userstages DAT files of version 1 are automatically converted to version 2 files the first time you connect stages in the host software, i.e. the first time the C843_qVST() or C843_CST() functions of the C843_GCS library are used which is the case, for example, when VST? or CST are sent in PITerminal or the *Select connected stages* startup step is performed in PIMikroMove™. The *Edit user stages data…* item in the controller menu of PIMikroMove™ opens the PIStageEditor tool with the version 2 file (C843UserStages2.dat). Parameters which were not present in version 1 are set to default values during conversion.

- Version 4 and newer of the PIStageEditor supports stage parameter files of version 2 (in PIMikroMove™, you can check the version of the PIStageEditor with *Help → About PiStageEditor*). If it is necessary to update the PIStageEditor, run either the setup from the latest revision of the CD for your controller, or download the latest revision of the PIStageEditor from the PI website. It can be found there in the same directory like the PIStages2.dat stage database. See “Updating PIStages2.dat” below for download instructions and make sure to copy the PiStageEditor.dll to the ...\PI\GcsTranslator directory.

4.6 Updating PIStages2.dat

To install the latest version of PIStages2.dat from the PI Website proceed as follows:

1. On the www.pi.ws front page, move the cursor to *Manuals, Software, ISO Statements* in the *Service* section on the left.
2. Select *Link to Software Server* from the list that pops up.
3. On the *PI Support Site* page, click on the *General Software* category (no login or password is required).
4. Click on *PI Stages.*
5. Click on *pistages2.*
6. In the download window, switch to the ...\PI\GcsTranslator directory. The location of the PI directory is that specified upon installation, usually in C:\Documents and Settings\All Users\Application Data (Windows XP) or C:\ProgramData (Windows Vista) (may differ in other-language Windows versions).

   Note that in PIMikroMove™, you can use the *Version Info* entry in the controller menu or the *Search for controller software* entry in the *Connections* menu to identify the GcsTranslator path.
7. If desired, rename the existing PIStages2.dat (if present) so as to preserve a copy for safety reasons.
8. Download the file from the server as PIStages2.dat.

5 Error Codes

The error codes listed here are those of the PI General Command Set. As such, some may be not relevant to your controller and will simply never occur.

Controller Errors

0	PI_CNTR_NO_ERROR	No error
1	PI_CNTR_PARAM_SYNTAX	Parameter syntax error
2	PI_CNTR_UNKNOWN_COMMAND	Unknown command
3	PI_CNTR_COMMAND_TOO_LONG	Command length out of limits or command buffer overrun
4	PI_CNTR_SCAN_ERROR	Error while scanning
5	PI_CNTR_MOVE_WITHOUT_REF_OR_NO_SERVO	Unallowable move attempted on unreferenced axis, or move attempted with servo off
6	PI_CNTR_INVALID_SGA_PARAM	Parameter for SGA not valid
7	PI_CNTR_POS_OUT_OF_LIMITS	Position out of limits
8	PI_CNTR_VEL_OUT_OF_LIMITS	Velocity out of limits
9	PI_CNTR_SET_PIVOT_NOT_POSSIBLE	Attempt to set pivot point while U,V and W not all 0
10	PI_CNTR_STOP	Controller was stopped by command
11	PI_CNTR_SST_OR_SCAN_RANGE	Parameter for SST or for one of the embedded scan algorithms out of range
12	PI_CNTR_INVALID_SCAN_AXES	Invalid axis combination for fast scan
13	PI_CNTR_INVALID_NAV_PARAM	Parameter for NAV out of range
14	PI_CNTR_INVALID_ANALOG_INPUT	Invalid analog channel
15	PI_CNTR_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
16	PI_CNTR_INVALID_STAGE_NAME	Unknown stage name
17	PI_CNTR_PARAM_OUT_OF_RANGE	Parameter out of range
18	PI_CNTR_INVALID_MACRO_NAME	Invalid macro name
19	PI_CNTR_MACRO_RECORD	Error while recording macro

20	PI_CNTR_MACRO_NOT_FOUND	Macro not found
21	PI_CNTR_AXIS_HAS_NO_BRAKE	Axis has no brake
22	PI_CNTR_DOUBLE_AXIS	Axis identifier specified more than once
23	PI_CNTR_ILLEGAL_AXIS	Illegal axis
24	PI_CNTR_PARAM_NR	Incorrect number of parameters
25	PI_CNTR_INVALID_REAL_NR	Invalid floating point number
26	PI_CNTR_MISSING_PARAM	Parameter missing
27	PI_CNTR_SOFT_LIMIT_OUT_OF_RANGE	Soft limit out of range
28	PI_CNTR_NO_MANUAL_PAD	No manual pad found
29	PI_CNTR_NO_JUMP	No more step-response values
30	PI_CNTR_INVALID_JUMP	No step-response values recorded
31	PI_CNTR_AXIS_HAS_NO_REFERENCE	Axis has no reference sensor
32	PI_CNTR_STAGE_HAS_NO_LIM_SWITCH	Axis has no limit switch
33	PI_CNTR_NO_RELAY_CARD	No relay card installed
34	PI_CNTR_CMD_NOT_ALLOWED_FOR_STAGE	Command not allowed for selected stage(s)
35	PI_CNTR_NO_DIGITAL_INPUT	No digital input installed
36	PI_CNTR_NO_DIGITAL_OUTPUT	No digital output configured
37	PI_CNTR_NO_MCM	No more MCM responses
38	PI_CNTR_INVALID_MCM	No MCM values recorded
39	PI_CNTR_INVALID_CNTR_NUMBER	Controller number invalid
40	PI_CNTR_NO_JOYSTICK_CONNECTED	No joystick configured
41	PI_CNTR_INVALID_EGE_AXIS	Invalid axis for electronic gearing, axis can not be slave

42	PI_CNTR_SLAVE_POSITION_OUT_OF_RANGE	Position of slave axis is out of range
43	PI_CNTR_COMMAND_EGE_SLAVE	Slave axis cannot be commanded directly when electronic gearing is enabled
44	PI_CNTR_JOYSTICK_CALIBRATION_FAILED	Calibration of joystick failed
45	PI_CNTR_REFERENCING_FAILED	Referencing failed
46	PI_CNTR_OPM_MISSING	OPM (Optical Power Meter) missing
47	PI_CNTR_OPM_NOT_INITIALIZED	OPM (Optical Power Meter) not initialized or cannot be initialized
48	PI_CNTR_OPM_COM_ERROR	OPM (Optical Power Meter) Communication Error
49	PI_CNTR_MOVE_TO_LIMIT_SWITCH_FAILED	Move to limit switch failed
50	PI_CNTR_REF_WITH_REF_DISABLED	Attempt to reference axis with referencing disabled
51	PI_CNTR_AXIS_UNDER_JOYSTICK_CONTROL	Selected axis is controlled by joystick
52	PI_CNTR_COMMUNICATION_ERROR	Controller detected communication error
53	PI_CNTR_DYNAMIC_MOVE_IN_PROCESS	MOV! motion still in progress
54	PI_CNTR_UNKNOWN_PARAMETER	Unknown parameter
55	PI_CNTR_NO_REP_RECORDED	No commands were recorded with REP
56	PI_CNTR_INVALID_PASSWORD	Password invalid
57	PI_CNTR_INVALID_RECORDER_CHAN	Data Record Table does not exist
58	PI_CNTR_INVALID_RECORDER_SRC_OPT	Source does not exist; number too low or too high
59	PI_CNTR_INVALID_RECORDER_SRC_CHAN	Source Record Table number too low or too high
60	PI_CNTR_PARAM_PROTECTION	Protected Param: current Command Level (CCL) too low
61	PI_CNTR_AUTOZERO_RUNNING	Command execution not possible while Autozero is running
62	PI_CNTR_NO_LINEAR_AXIS	Autozero requires at least one linear axis

63	PI_CNTR_INIT_RUNNING	Initialization still in progress
64	PI_CNTR_READ_ONLY_PARAMETER	Parameter is read-only
65	PI_CNTR_PAM_NOT_FOUND	Parameter not found in non-volatile memory
66	PI_CNTR_VOL_OUT_OF_LIMITS	Voltage out of limits
67	PI_CNTR_WAVE_TOO_LARGE	Not enough memory available for requested wave curve
68	PI_CNTR_NOT_ENOUGH_DDL_MEMORY	Not enough memory available for DDL table; DDL can not be started
69	PI_CNTR_DDL_TIME_DELAY_TOO_LARGE	Time delay larger than DDL table; DDL can not be started
70	PI_CNTR_DIFFERENT_ARRAY_LENGTH	The requested arrays have different lengths; query them separately
71	PI_CNTR_GEN_SINGLE_MODE_RESTART	Attempt to restart the generator while it is running in single step mode
72	PI_CNTR_ANALOG_TARGET_ACTIVE	Motion commands and wave generator activation are not allowed when analog target is active
73	PI_CNTR_WAVE_GENERATOR_ACTIVE	Motion commands are not allowed when wave generator is active
74	PI_CNTR_AUTOZERO_DISABLED	No sensor channel or no piezo channel connected to selected axis (sensor and piezo matrix)
75	PI_CNTR_NO_WAVE_SELECTED	Generator started (WGO) without having selected a wave table (WSL).
76	PI_CNTR_IF_BUFFER_OVERRUN	Interface buffer did overrun and command couldn't be received correctly
77	PI_CNTR_NOT_ENOUGH_RECORDED_DATA	Data Record Table does not hold enough recorded data
78	PI_CNTR_TABLE_DEACTIVATED	Data Record Table is not configured for recording
79	PI_CNTR_OPENLOOP_VALUE_SET_WHEN_SERVO_ON	Open-loop commands (SVA, SVR) are not allowed when servo is on
80	PI_CNTR_RAM_ERROR	Hardware error affecting RAM

81	PI_CNTR_MACRO_UNKNOWN_COMMAND	Not macro command
82	PI_CNTR_MACRO_PC_ERROR	Macro counter out of range
83	PI_CNTR_JOYSTICK_ACTIVE	Joystick is active
84	PI_CNTR_MOTOR_IS_OFF	Motor is off
85	PI_CNTR_ONLY_IN_MACRO	Macro-only command
86	PI_CNTR_JOYSTICK_UNKNOWN_AXIS	Invalid joystick axis
87	PI_CNTR_JOYSTICK_UNKNOWN_ID	Joystick unknown
88	PI_CNTR_REF_MODE_IS_ON	Move without referenced stage
89	PI_CNTR_NOT_ALLOWED_IN_CURRENT_MOTION_MODE	Command not allowed in current motion mode
90	PI_CNTR_DIO_AND_TRACING_NOT_POSSIBLE	No tracing possible while digital IOs are used on this HW revision. Reconnect to switch operation mode.
91	PI_CNTR_COLLISION	Move not possible, would cause collision
100	PI_LABVIEW_ERROR	PI LabVIEW driver reports error. See source control for details.
200	PI_CNTR_NO_AXIS	No stage connected to axis
201	PI_CNTR_NO_AXIS_PARAM_FILE	File with axis parameters not found
202	PI_CNTR_INVALID_AXIS_PARAM_FILE	Invalid axis parameter file
203	PI_CNTR_NO_AXIS_PARAM_BACKUP	Backup file with axis parameters not found
204	PI_CNTR_RESERVED_204	PI internal error code 204
205	PI_CNTR_SMO_WITH_SERVO_ON	SMO with servo on
206	PI_CNTR_UUDECODE_INCOMPLETE_HEADER	uudecode: incomplete header
207	PI_CNTR_UUDECODE_NOTHING_TO_DECODE	uudecode: nothing to decode
208	PI_CNTR_UUDECODE_ILLEGAL_FORMAT	uudecode: illegal UUE format
209	PI_CNTR_CRC32_ERROR	CRC32 error

210	PI_CNTR_ILLEGAL_FILENAME	Illegal file name (must be 8-0 format)
211	PI_CNTR_FILE_NOT_FOUND	File not found on controller
212	PI_CNTR_FILE_WRITE_ERROR	Error writing file on controller
213	PI_CNTR_DTR_HINDERS_VELOCITY_CHANGE	VEL command not allowed in DTR Command Mode
214	PI_CNTR_POSITION_UNKNOWN	Position calculations failed
215	PI_CNTR_CONN_POSSIBLY_BROKEN	The connection between controller and stage may be broken
216	PI_CNTR_ON_LIMIT_SWITCH	The connected stage has driven into a limit switch, some controllers need CLR to resume operation
217	PI_CNTR_UNEXPECTED_STRUT_STOP	Strut test command failed because of an unexpected strut stop
218	PI_CNTR_POSITION_BASED_ON_ESTIMATION	While MOV! is running position can only be estimated!
219	PI_CNTR_POSITION_BASED_ON_INTERPOLATION	Position was calculated during MOV motion
230	PI_CNTR_INVALID_HANDLE	Invalid handle
231	PI_CNTR_NO_BIOS_FOUND	No bios found
232	PI_CNTR_SAVE_SYS_CFG_FAILED	Save system configuration failed
233	PI_CNTR_LOAD_SYS_CFG_FAILED	Load system configuration failed
301	PI_CNTR_SEND_BUFFER_OVERFLOW	Send buffer overflow
302	PI_CNTR_VOLTAGE_OUT_OF_LIMITS	Voltage out of limits
303	PI_CNTR_OPEN_LOOP_MOTION_SET_WHEN_SERVO_ON	Open-loop motion attempted when servo ON
304	PI_CNTR_RECEIVING_BUFFER_OVERFLOW	Received command is too long
305	PI_CNTR_EEPROM_ERROR	Error while reading/writing EEPROM
306	PI_CNTR_I2C_ERROR	Error on I2C bus

307	PI_CNTR_RECEIVING_TIMEOUT	Timeout while receiving command
308	PI_CNTR_TIMEOUT	A lengthy operation has not finished in the expected time
309	PI_CNTR_MACRO_OUT_OF_SPACE	Insufficient space to store macro
310	PI_CNTR_EUI_OLDVERSION_CFGDATA	Configuration data has old version number
311	PI_CNTR_EUI_INVALID_CFGDATA	Invalid configuration data
333	PI_CNTR_HARDWARE_ERROR	Internal hardware error
400	PI_CNTR_WAV_INDEX_ERROR	Wave generator index error
401	PI_CNTR_WAV_NOT_DEFINED	Wave table not defined
402	PI_CNTR_WAV_TYPE_NOT_SUPPORTED	Wave type not supported
403	PI_CNTR_WAV_LENGTH_EXCEEDS_LIMIT	Wave length exceeds limit
404	PI_CNTR_WAV_PARAMETER_NR	Wave parameter number error
405	PI_CNTR_WAV_PARAMETER_OUT_OF_LIMIT	Wave parameter out of range
406	PI_CNTR_WGO_BIT_NOT_SUPPORTED	WGO command bit not supported
502	PI_CNTR_REDUNDANCY_LIMIT_EXCEEDED	Position consistency check failed
503	PI_CNTR_COLLISION_SWITCH_ACTIVATED	Hardware collision sensor(s) are activated
504	PI_CNTR_FOLLOWING_ERROR	Strut following error occurred, e.g. caused by overload or encoder failure
555	PI_CNTR_UNKNOWN_ERROR	BasMac: unknown controller error
601	PI_CNTR_NOT_ENOUGH_MEMORY	not enough memory
602	PI_CNTR_HW_VOLTAGE_ERROR	hardware voltage error
603	PI_CNTR_HW_TEMPERATURE_ERROR	hardware temperature out of range
1000	PI_CNTR_TOO_MANY_NESTED_MACROS	Too many nested macros
1001	PI_CNTR_MACRO_ALREADY_DEFINED	Macro already defined

1002	PI_CNTR_NO_MACRO_RECORDING	Macro recording not activated
1003	PI_CNTR_INVALID_MAC_PARAM	Invalid parameter for MAC
1004	PI_CNTR_RESERVED_1004	PI internal error code 1004
1005	PI_CNTR_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
2000	PI_CNTR_ALREADY_HAS_SERIAL_NUMBER	Controller already has a serial number
4000	PI_CNTR_SECTOR_ERASE_FAILED	Sector erase failed
4001	PI_CNTR_FLASH_PROGRAM_FAILED	Flash program failed
4002	PI_CNTR_FLASH_READ_FAILED	Flash read failed
4003	PI_CNTR_HW_MATCHCODE_ERROR	HW match code missing/invalid
4004	PI_CNTR_FW_MATCHCODE_ERROR	FW match code missing/invalid
4005	PI_CNTR_HW_VERSION_ERROR	HW version missing/invalid
4006	PI_CNTR_FW_VERSION_ERROR	FW version missing/invalid
4007	PI_CNTR_FW_UPDATE_ERROR	FW update failed
5200	PI_CNTR_AXIS_NOT_CONFIGURED	Axis must be configured for this action

Interface Errors

0	COM_NO_ERROR	No error occurred during function call
-1	COM_ERROR	Error during com operation (could not be specified)
-2	SEND_ERROR	Error while sending data
-3	REC_ERROR	Error while receiving data
-4	NOT_CONNECTED_ERROR	Not connected (no port with given ID open)
-5	COM_BUFFER_OVERFLOW	Buffer overflow

-6	CONNECTION_FAILED	Error while opening port
-7	COM_TIMEOUT	Timeout error
-8	COM_MULTILINE_RESPONSE	There are more lines waiting in buffer
-9	COM_INVALID_ID	There is no interface or DLL handle with the given ID
-10	COM_NOTIFY_EVENT_ERROR	Event/message for notification could not be opened
-11	COM_NOT_IMPLEMENTED	Function not supported by this interface type
-12	COM_ECHO_ERROR	Error while sending "echoed" data
-13	COM_GPIB_EDVR	IEEE488: System error
-14	COM_GPIB_ECIC	IEEE488: Function requires GPIB board to be CIC
-15	COM_GPIB_ENOL	IEEE488: Write function detected no listeners
-16	COM_GPIB_EADR	IEEE488: Interface board not addressed correctly
-17	COM_GPIB_EARG	IEEE488: Invalid argument to function call
-18	COM_GPIB_ESAC	IEEE488: Function requires GPIB board to be SAC
-19	COM_GPIB_EABO	IEEE488: I/O operation aborted
-20	COM_GPIB_ENEB	IEEE488: Interface board not found
-21	COM_GPIB_EDMA	IEEE488: Error performing DMA
-22	COM_GPIB_EOIP	IEEE488: I/O operation started before previous operation completed
-23	COM_GPIB_ECAP	IEEE488: No capability for intended operation
-24	COM_GPIB_EFSO	IEEE488: File system operation error
-25	COM_GPIB_EBUS	IEEE488: Command error during device call
-26	COM_GPIB_ESTB	IEEE488: Serial poll-status byte lost

-27	COM_GPIB_ESRQ	IEEE488: SRQ remains asserted
-28	COM_GPIB_ETAB	IEEE488: Return buffer full
-29	COM_GPIB_ELCK	IEEE488: Address or board locked
-30	COM_RS_INVALID_DATA_BITS	RS-232: 5 data bits with 2 stop bits is an invalid combination, as is 6, 7, or 8 data bits with 1.5 stop bits
-31	COM_ERROR_RS_SETTINGS	RS-232: Error configuring the COM port
-32	COM_INTERNAL_RESOURCES_ERROR	Error dealing with internal system resources (events, threads, ...)
-33	COM_DLL_FUNC_ERROR	A DLL or one of the required functions could not be loaded
-34	COM_FTDIUSB_INVALID_HANDLE	FTDIUSB: invalid handle
-35	COM_FTDIUSB_DEVICE_NOT_FOUND	FTDIUSB: device not found
-36	COM_FTDIUSB_DEVICE_NOT_OPENED	FTDIUSB: device not opened
-37	COM_FTDIUSB_IO_ERROR	FTDIUSB: IO error
-38	COM_FTDIUSB_INSUFFICIENT_RESOURCES	FTDIUSB: insufficient resources
-39	COM_FTDIUSB_INVALID_PARAMETER	FTDIUSB: invalid parameter
-40	COM_FTDIUSB_INVALID_BAUD_RATE	FTDIUSB: invalid baud rate
-41	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_ERASE	FTDIUSB: device not opened for erase
-42	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_WRITE	FTDIUSB: device not opened for write
-43	COM_FTDIUSB_FAILED_TO_WRITE_DEVICE	FTDIUSB: failed to write device
-44	COM_FTDIUSB_EEPROM_READ_FAILED	FTDIUSB: EEPROM read failed
-45	COM_FTDIUSB_EEPROM_WRITE_FAILED	FTDIUSB: EEPROM write failed
-46	COM_FTDIUSB_EEPROM_ERASE_FAILED	FTDIUSB: EEPROM erase failed
-47	COM_FTDIUSB_EEPROM_NOT_PRESENT	FTDIUSB: EEPROM not present

-48	COM_FTDIUSB_EEPROM_NOT_PROGRAMMED	FTDIUSB: EEPROM not programmed
-49	COM_FTDIUSB_INVALID_ARGS	FTDIUSB: invalid arguments
-50	COM_FTDIUSB_NOT_SUPPORTED	FTDIUSB: not supported
-51	COM_FTDIUSB_OTHER_ERROR	FTDIUSB: other error
-52	COM_PORT_ALREADY_OPEN	Error while opening the COM port: was already open
-53	COM_PORT_CHECKSUM_ERROR	Checksum error in received data from COM port
-54	COM_SOCKET_NOT_READY	Socket not ready, you should call the function again
-55	COM_SOCKET_PORT_IN_USE	Port is used by another socket
-56	COM_SOCKET_NOT_CONNECTED	Socket not connected (or not valid)
-57	COM_SOCKET_TERMINATED	Connection terminated (by peer)
-58	COM_SOCKET_NO_RESPONSE	Can't connect to peer
-59	COM_SOCKET_INTERRUPTED	Operation was interrupted by a nonblocked signal
-60	COM_PCI_INVALID_ID	No device with this ID is present
-61	COM_PCI_ACCESS_DENIED	Driver could not be opened (on Vista: run as administrator!)

DLL Errors

-1001	PI_UNKNOWN_AXIS_IDENTIFIER	Unknown axis identifier
-1002	PI_NR_NAV_OUT_OF_RANGE	Number for NAV out of range--must be in [1,10000]
-1003	PI_INVALID_SGA	Invalid value for SGA--must be one of 1, 10, 100, 1000
-1004	PI_UNEXPECTED_RESPONSE	Controller sent unexpected response
-1005	PI_NO_MANUAL_PAD	No manual control pad installed, calls to SMA and related commands are not allowed

-1006	PI_INVALID_MANUAL_PAD_KNOB	Invalid number for manual control pad knob
-1007	PI_INVALID_MANUAL_PAD_AXIS	Axis not currently controlled by a manual control pad
-1008	PI_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
-1009	PI_THREAD_ERROR	Internal error--could not start thread
-1010	PI_IN_MACRO_MODE	Controller is (already) in macro mode--command not valid in macro mode
-1011	PI_NOT_IN_MACRO_MODE	Controller not in macro mode--command not valid unless macro mode active
-1012	PI_MACRO_FILE_ERROR	Could not open file to write or read macro
-1013	PI_NO_MACRO_OR_EMPTY	No macro with given name on controller, or macro is empty
-1014	PI_MACRO_EDITOR_ERROR	Internal error in macro editor
-1015	PI_INVALID_ARGUMENT	One or more arguments given to function is invalid (empty string, index out of range, ...)
-1016	PI_AXIS_ALREADY_EXISTS	Axis identifier is already in use by a connected stage
-1017	PI_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
-1018	PI_COM_ARRAY_ERROR	Could not access array data in COM server
-1019	PI_COM_ARRAY_RANGE_ERROR	Range of array does not fit the number of parameters
-1020	PI_INVALID_SPA_CMD_ID	Invalid parameter ID given to SPA or SPA?
-1021	PI_NR_AVG_OUT_OF_RANGE	Number for AVG out of range--must be >0
-1022	PI_WAV_SAMPLES_OUT_OF_RANGE	Incorrect number of samples given to WAV
-1023	PI_WAV_FAILED	Generation of wave failed
-1024	PI_MOTION_ERROR	Motion error while axis in motion, call CLR to resume operation

-1025	PI_RUNNING_MACRO	Controller is (already) running a macro
-1026	PI_PZT_CONFIG_FAILED	Configuration of PZT stage or amplifier failed
-1027	PI_PZT_CONFIG_INVALID_PARAMS	Current settings are not valid for desired configuration
-1028	PI_UNKNOWN_CHANNEL_IDENTIFIER	Unknown channel identifier
-1029	PI_WAVE_PARAM_FILE_ERROR	Error while reading/writing wave generator parameter file
-1030	PI_UNKNOWN_WAVE_SET	Could not find description of wave form. Maybe WG.INI is missing?
-1031	PI_WAVE_EDITOR_FUNC_NOT_LOADED	The WGWaveEditor DLL function was not found at startup
-1032	PI_USER_CANCELLED	The user cancelled a dialog
-1033	PI_C844_ERROR	Error from C-844 Controller
-1034	PI_DLL_NOT_LOADED	DLL necessary to call function not loaded, or function not found in DLL
-1035	PI_PARAMETER_FILE_PROTECTED	The open parameter file is protected and cannot be edited
-1036	PI_NO_PARAMETER_FILE_OPENED	There is no parameter file open
-1037	PI_STAGE_DOES_NOT_EXIST	Selected stage does not exist
-1038	PI_PARAMETER_FILE_ALREADY_OPENED	There is already a parameter file open. Close it before opening a new file
-1039	PI_PARAMETER_FILE_OPEN_ERROR	Could not open parameter file
-1040	PI_INVALID_CONTROLLER_VERSION	The version of the connected controller is invalid
-1041	PI_PARAM_SET_ERROR	Parameter could not be set with SPA--parameter not defined for this controller!
-1042	PI_NUMBER_OF_POSSIBLE_WAVES_EXCEEDED	The maximum number of wave definitions has been exceeded
-1043	PI_NUMBER_OF_POSSIBLE_GENERATORS_EXCEEDED	The maximum number of wave generators has been exceeded
-1044	PI_NO_WAVE_FOR_AXIS_DEFINED	No wave defined for specified axis

-1045	PI_CANT_STOP_OR_START_WAV	Wave output to axis already stopped/started
-1046	PI_REFERENCE_ERROR	Not all axes could be referenced
-1047	PI_REQUIRED_WAVE_NOT_FOUND	Could not find parameter set required by frequency relation
-1048	PI_INVALID_SPP_CMD_ID	Command ID given to SPP or SPP? is not valid
-1049	PI_STAGE_NAME_ISNT_UNIQUE	A stage name given to CST is not unique
-1050	PI_FILE_TRANSFER_BEGIN_MISSING	A uuencoded file transferred did not start with "begin" followed by the proper filename
-1051	PI_FILE_TRANSFER_ERROR_TEMP_FILE	Could not create/read file on host PC
-1052	PI_FILE_TRANSFER_CRC_ERROR	Checksum error when transferring a file to/from the controller
-1053	PI_COULDNT_FIND_PISTAGES_DAT	The PiStages2.dat database could not be found. This file is required to connect a stage with the CST command
-1054	PI_NO_WAVE_RUNNING	No wave being output to specified axis
-1055	PI_INVALID_PASSWORD	Invalid password
-1056	PI_OPM_COM_ERROR	Error during communication with OPM (Optical Power Meter), maybe no OPM connected
-1057	PI_WAVE_EDITOR_WRONG_PARAMNUM	WaveEditor: Error during wave creation, incorrect number of parameters
-1058	PI_WAVE_EDITOR_FREQUENCY_OUT_OF_RANGE	WaveEditor: Frequency out of range
-1059	PI_WAVE_EDITOR_WRONG_IP_VALUE	WaveEditor: Error during wave creation, incorrect index for integer parameter
-1060	PI_WAVE_EDITOR_WRONG_DP_VALUE	WaveEditor: Error during wave creation, incorrect index for floating point parameter
-1061	PI_WAVE_EDITOR_WRONG_ITEM_VALUE	WaveEditor: Error during wave creation, could not calculate value
-1062	PI_WAVE_EDITOR_MISSING_GRAPH_COMPONENT	WaveEditor: Graph display component not installed

-1063	PI_EXT_PROFILE_UNALLOWED_CMD	User Profile Mode: Command is not allowed, check for required preparatory commands
-1064	PI_EXT_PROFILE_EXPECTING_MOTION_ERROR	User Profile Mode: First target position in User Profile is too far from current position
-1065	PI_EXT_PROFILE_ACTIVE	Controller is (already) in User Profile Mode
-1066	PI_EXT_PROFILE_INDEX_OUT_OF_RANGE	User Profile Mode: Block or Data Set index out of allowed range
-1067	PI_PROFILE_GENERATOR_NO_PROFILE	ProfileGenerator: No profile has been created yet
-1068	PI_PROFILE_GENERATOR_OUT_OF_LIMITS	ProfileGenerator: Generated profile exceeds limits of one or both axes
-1069	PI_PROFILE_GENERATOR_UNKNOWN_PARAMETER	ProfileGenerator: Unknown parameter ID in Set/Get Parameter command
-1070	PI_PROFILE_GENERATOR_PAR_OUT_OF_RANGE	ProfileGenerator: Parameter out of allowed range
-1071	PI_EXT_PROFILE_OUT_OF_MEMORY	User Profile Mode: Out of memory
-1072	PI_EXT_PROFILE_WRONG_CLUSTER	User Profile Mode: Cluster is not assigned to this axis
-1073	PI_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
-1074	PI_INVALID_DEVICE_DRIVER_VERSION	The installed device driver doesn't match the required version. Please see the documentation to determine the required device driver version.
-1075	PI_INVALID_LIBRARY_VERSION	The library used doesn't match the required version. Please see the documentation to determine the required library version.
-1076	PI_INTERFACE_LOCKED	The interface is currently locked by another function. Please try again later.
-1077	PI_PARAM_DAT_FILE_INVALID_VERSION	Version of parameter DAT file does not match the required version. Current files are available at www.pi.ws.
-1078	PI_CANNOT_WRITE_TO_PARAM_DAT_FILE	Cannot write to parameter DAT file to store user defined stage type.

-1079	PI_CANNOT_CREATE_PARAM_DAT_FILE	Cannot create parameter DAT file to store user defined stage type.
-1080	PI_PARAM_DAT_FILE_INVALID_REVISION	Parameter DAT file does not have correct revision.
-1081	PI_USERSTAGES_DAT_FILE_INVALID_REVISION	User stages DAT file does not have correct revision.