ESRF ISG

INSTRUMENT SUPPORT GROUP

MoCo

User Manual

Monochromator Controller

Description

The Monochromator Controller (MoCo) is an electronic module designed to regulate the position of an optical component in a synchrotron radiation beamline. The controller corrects the position of the component, typically a mirror or a monochromator, by monitoring the outgoing beam and actively compensating low frequency drifts due to thermal load changes or mechanical instability.

Rev. 2.0 May / 2008

Document Versions

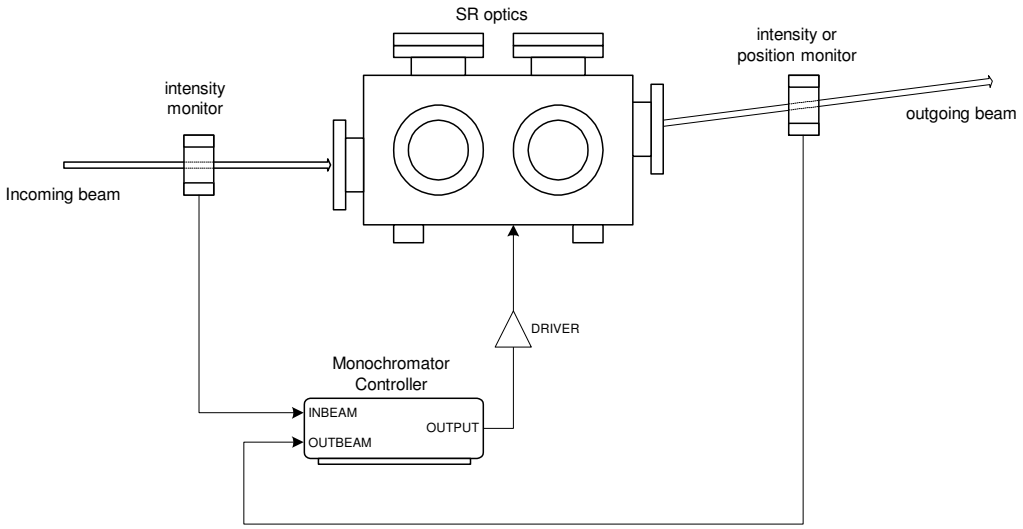
Date	Version	Comments
15/02/2003	1.2a	First complete version. Corresponds to firmware version 01.02
04/03/2005	1.2b	Minor corrections for Appendix A
31/05/2008	2.0	Firmware 2.0, includes oscillation mode and associated commands

Contents

1. FUNCTIONAL DESCRIPTION

The Monochromator Controller Unit (MoCo) is an electronic module designed to regulate the position of an optical component in a synchrotron radiation beamline, typically a mirror or a monochromator, by actively measuring and correcting any drift due to thermal load changes or mechanical instability.

The controller takes as input signals information from two beam monitors and generates an output voltage that drives the optics by means of a piezoelectric actuator or a similar device. The following figure depicts the general configuration.



The first monitor (*INBEAM*) measures the intensity of the radiation beam coming into the optical element and it is used for normalisation and to detect the presence of the beam.
The second monitor (*OUTBEAM*) measures either the intensity or the position of the beam after the optics. The type of monitor that is required for a particular application depends on the response of the optical system.
The regulation algorithm drives the optical element, usually by tilting a crystal or mirror, in order to cancel the difference between the measured value (POSITION or INTENSITY) and a user predefined setpoint value. When using an intensity monitor the previous scheme does not allow to regulate the beam at the maximum of intensity, for instance at the top of the reflectivity curve of a crystal monochromator. To overcome this limitation, the controller may be also configured in an third regulation scheme (OSCILLATION) that adds a small oscillatory component to the optics drive and applies synchronous detection techniques to measure the derivative of the system response.

Among the features of the controller there are the following:

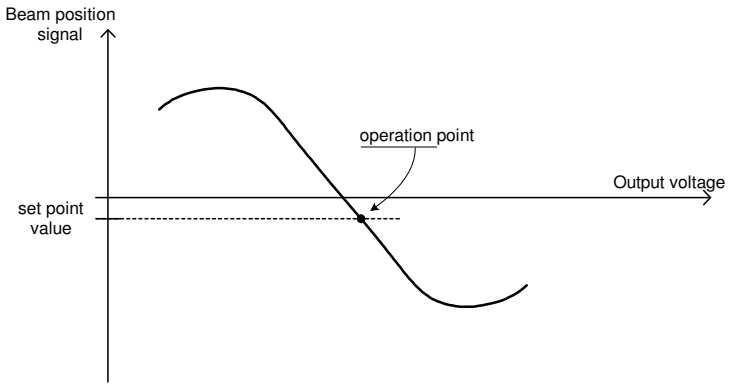
- Low-current or voltage inputs with built-in gain control
- Regulation time constant from 1 ms to 1 minute
- Detection of beam presence
- Automatic tuning and autoscaling
- Full control by serial line
- Software INBEAM monitor
- Hardware handshake with other devices
- Gain control of external preamplifiers

1.1. Operation Modes

As mentioned above, the controller has to be operated in different close-loop modes depending on the type information provided by the *OUTBEAM* monitor and the regulation scheme. When the regulation is not active, the controller can be used to drive the optics in open-loop from a host computer.

1.1.1. Position mode

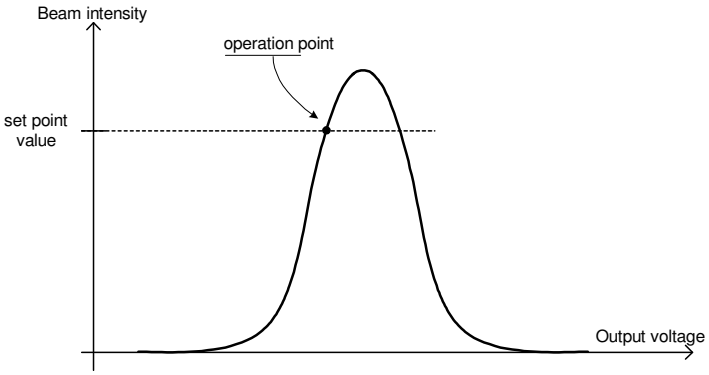
In this mode the OUTBEAM signal is proportional to the position of the beam. The controller then assumes a dependency of the signal with the output voltage that varies monotonically, at least around the central range, as it is represented in the figure. This is the signal shape expected from a beam position monitor after a mirror or monochromator.



In this mode the controller regulates the position of the optical element in order to keep the beam position constant.
For proper operation in this mode, in addition to the set point value, the controller requires information about the slope of the response function. This value can be either provided externally or estimated by the controller itself by means of a scanning procedure (see Tuning below).

1.1.2. Intensity mode

In this case the OUTBEAM signal is proportional to the intensity of the beam. The controller expects a dependency of the signal with the output voltage that presents a maximum at a certain position of the optics as it is represented in the figure. Such a peak-shaped response function is usually produced by a double-crystal monochromator or by a mirror followed by a fixed slit.



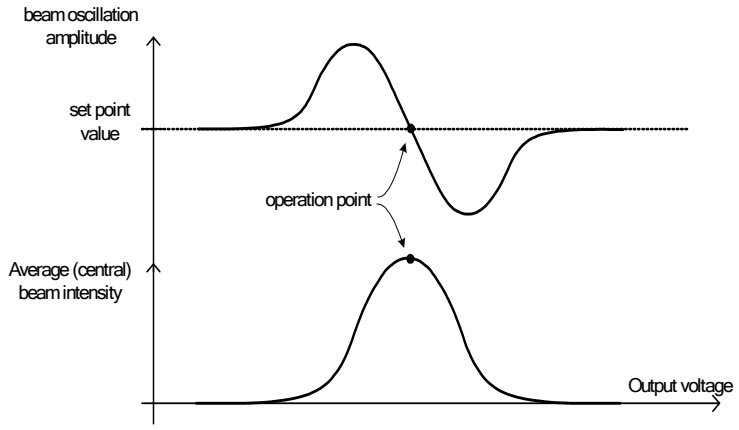
In this mode the controller regulates the position of the optical element in order to keep constant the operation at a relative position of the peak value. Intensity normalisation by the *INBEAM* signal is usually required to cope with the intensity decay found in a synchrotron radiation beamline.

For a certain setpoint value, the operation point can be selected either at the left or at the right side or the peak.

For proper operation in this mode, in addition to the relative set point value, the controller needs the absolute height of the peak and the peak width. As in the position mode, these additional information can be either provided externally or obtained by the controller itself by means of a tuning procedure (see Tuning below).

1.1.3. Oscillation mode

This mode implements a special regulation scheme to be used with intensity monitors. The intensity mode described above does not allow to regulate the beam at the maximum of beam intensity. In the special oscillation mode a small AC component of fixed amplitude and frequency is summed to the output voltage that drives the optics. Such AC component results in a oscillation of the beam intensity that can be measured by the beam monitor. The controller extracts the amplitude of the intensity modulation at the oscillation frequency by applying a synchronous detection technique analogous to that used by lock-in amplifiers. The measured oscillation amplitude is proportional to the derivative of the beam intensity with respect to the driving voltage as it is depicted in the following figure.



In this mode the controller regulates the position of the optical element in order to keep the amplitude of the measured oscillation equal to the setpoint value. If the setpoint is set to zero, the most usual case, the controller regulates the optical component so that it sits at the point of maximum transmitted intensity, where the intensity derivative is also zero.

In addition to the set point value, the controller requires information about the slope of the beam oscillation amplitude with respect to the driving voltage.

1.1.4. Open loop

When closed-loop regulation is switched off, the controller can be used to move the optics by driving the output voltage to fixed values. In this case the voltage is not changed instantaneously but ramped at a constant programmable speed towards the desired value.

1.2. Main Features

1.2.1. Regulation

The monochromator controller implements regulation based on a simple integral correction algorithm. There are no proportional or differential parts as in full-featured PID controllers. Therefore the regulation settings are reduced to a single time constant value *tau*. It is possible to program the controller with time constant as low as 1 millisecond but in order to achieve proper regulation, *tau* must be set longer than the response time of the whole optical system, including the driving actuator, the mechanics and the outgoing beam monitor.

Most of the piezoelectric drives used in synchrotron radiation instrumentation have response times that are between 5 and 100 milliseconds. Therefore even in the case of very light mechanics (low inertia) and fast enough beam monitors, the time constant of the controller should not be set lower than 100 milliseconds to 1 second.

A controller that uses only integral part guaranties the stability of the closed-loop, avoids risks of induced oscillations and is well adapted to correct slow drifts due to thermal effects. However it cannot compensate for faster externally generated vibrations or oscillations that a full PID controller might damp.

In order to accept tau values in time units (see TAU command), the controller requires the low frequency gain of the whole optical system. This is obtained from the slope of the response function. Such slope is a required parameter when the controller operates in position or oscillation mode and in the intensity mode it is estimated from height and width of the peak.

1.2.2. Tuning and autoscaling

The tuning feature is a scanning procedure implemented internally in the controller aimed to measure the response function parameters required for regulation (peak height and width in intensity mode, or slope in position or oscillation mode). Once this procedure is started it does not require any further intervention by the user. If this procedure is not executed, the user must explicitly set these parameters in the controller (see PEAK and SLOPE commands).

If autoscaling is selected for any of the beam monitor inputs (see INBEAM and OUTBEAM commands), the gain of the corresponding channel is automatically adjusted during the tuning procedure. Autoscaling only has effect during tuning, the gain of the input channels being never changed when the controller is in regulation mode. The autoscaling feature is different from the autorange option (see SET AUTORANGE) that will only be effective when the controller is not regulating (idle state).

The tuning procedure can be started by the user (see TUNE command) or can be optionally triggered automatically on input channel overload conditions or after certain well defined events like the loss of the photon beam or the inhibition of the controller operation forced by a external signal (see AUTOTUNE and AUTOPEAK commands).

By default, once the scanning sequence is successfully completed the controller switches into regulation mode. In intensity mode it is possible to make the controller to move to the peak position instead of switching into regulation mode (see TUNE PEAK command). In this way it is possible to position the optics automatically at to top of the transmission curve of the optics (a rocking curve for instance). This mode requires optics with good enough position repeatability, i.e. a negligible hysteresis or backlash.

1.2.3. Normalisation

The *OUTBEAM* monitor signal can be optionally normalised by the incoming beam intensity *INBEAM.* This is generally needed when the controller is operated in intensity mode in synchrotron radiation beamlines to cope with the intensity decay of the source. In other cases normalisation may be necessary or not depending on the particular instrumental setup and the type of beam monitor used (see SET NORMALISE command).

1.2.4. Beam detection

The controller may optionally analyse the incoming beam intensity (*INBEAM*) to take decisions on the presence or not of the photon beam. When the beam detection feature is activated (see SET BEAMCHECK command) and the internal algorithm considers that the beam has been lost, operation is stopped and the controller goes into “wait beam” mode. When the beam is back, the controller may wait an additional programmable time or resume operation immediately (see BEAMCHECK command).

1.2.5. Soft INBEAM and filtering

It is possible to replace the *INBEAM* monitor with a value that is periodically set in the controller through the serial line. In what concerns the functionality of the controller this software value is equivalent to a hardware signal physically connected to the *INBEAM* channel. This “soft” INBEAM signal can be used both for normalisation and for beam detection purposes.

Usually the soft INBEAM value is in updated in the controller at a slow rate, typically once every few seconds, the controller internally applies a digital low-pass filter to the software values to smooth out the signal steps that, even if small, may perturb the regulation when normalisation is active.

1.2.6. Autorun

If the autorun feature is selected the controller memorises its operating state when it is switched off and resumes operation when it is switched on again (see SET AUTORUN command). In this way and if autorun is used in combination with the autotune feature, the controller can operate unattended even in the case of an electrical shutdown.

1.2.7. Pause and inhibit

Under certain conditions it may be convenient to temporarily stop the controller while other instruments are operated, for instance when a shutter is closed downstream or when a different optical element is realigned. The controller can be forced into this “pause” mode either by an external inhibit input signal or through the serial line (see INHIBIT and PAUSE commands).
Once the pause/inhibit condition is removed, the controller resumes operation in whatever mode was active before the pause event.

1.2.8. Interlock

If the interlock option is selected (see SET INTERLOCK command), whenever a special input is active the controller goes into alarm mode and drives the piezo output to a predefined value, usually zero volts. This interlock feature is intended to prevent high-voltage sparkling in

in-vaccum piezoelectric actuators when the system pressure reaches values at which electrical breakdown voltages are low.

1.2.9. Input signal measurement

The current values of the monitor signal can be read at any time from the controller. In this way the unit can be operated as a measuring device even if the regulation function is not used. Two values are available per channel: a non-filtered value that corresponds to the actual signal at the corresponding input, and a low-pass filtered value calculated by an internal digital filter (see ?BEAM and ?FBEAM requests).

Every time the gain of an input channel is changed, the low-pass digital filter is reset to the instantaneous value. If the autorange option is selected, this feature may produce an apparent distortion (slope discontinuity) of the signal when the corresponding filtered value of the controller is used to record the shape of fast time varying signals.

In the current version of the controller firmware, the bandwidth of the digital filter can not be set independently and is derived from the time constant used by the beam detection algorithm (see BEAMCHK command).

In most of the cases photodiodes and ion chambers can be directly connected to the internal electrometers of the controller. In these cases the controller provides the values of the input signals in physical units (amps) and takes care of gain control. If external electrometers with compatible gain control are used, it is possible to set the controller to manage the range changes and the scale factors of the whole signal chains.

1.2.10. Oscillation measurements and regulation

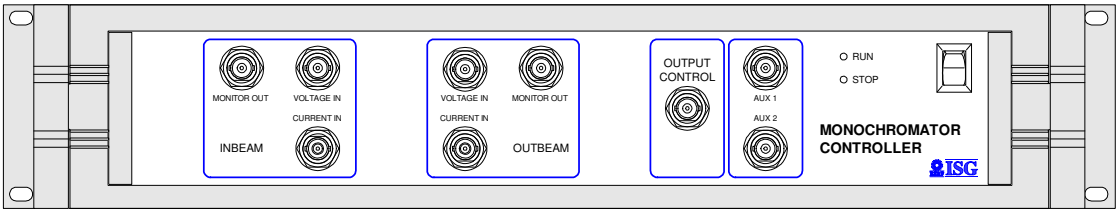
In the special oscillation mode and when the oscillation is on (see OSCIL command), the measured amplitudes of the intensity oscillation can be read from the controller by the ?OSCBEAM request. As the measurement is synchronous with the excitation, the oscillation amplitude is phase dependent. The oscillation can therefore be represented by two orthogonal components (main and quadrature) with a phase difference of 90 degrees. The ?OSCBEAM request returns two amplitude values that correspond to the two orthogonal components.

The phase reference used to evaluate the two oscillation components can be changed arbitrarily (see PHASE command). As the beam regulation uses only the first component (main), it is convenient to adjust the phase reference in such a way that the first component (main) is maximised and the second (quadrature) is null.

2. HARDWARE DESCRIPTION

2.1. Front Panel

The front panel of the MoCo unit is depicted in the following figure:



2.1.1. INBEAM, OUTBEAM

There are three connectors assigned to each of the input channels *INBEAM* and *OUTBEAM*:

- CURRENT IN
  - Socket: two-lug triaxial. Matches the following plug series: Trompeter PL74, Keithley CS-141, Pomona 3600 and 5056
  - Input range: from 1.25 nA to 1 mA full scale in 24 ranges
  - Impedance: virtual ground

- VOLTAGE IN
  - Socket: standard BNC
  - Input range: 1.25, 2.5, 5 and 10 V full scale
  - Impedance: 10 KΩ to ground

- MONITOR OUT
  - Socket: standard BNC
  - Output range: 10 volts full scale corresponds to input full scale value.
  - Max. current: 4 mA sink or source
  - Bandwith: 80 KHz (voltage input)
    800 Hz (1.25 nA to 10 µA full scale ranges)
    3.6 KHz (12.5 µA to 1 mA full scale ranges)

Software configuration allows selecting between current and voltage inputs, inverting the polarity of the signals and selecting unipolar or bipolar modes in the internal ADC's.

The monitor output provides the image of the corresponding input signal at the output of the internal preamplifiers.

2.1.2. CONTROL OUTPUT

This connector provides the output voltage to steer the optical component, usually by driving a piezoelectric actuator through a high-voltage amplifier or a dedicated controller.

- Socket: standard BNC
- Output ranges: 0 to 5V, 0 to 10V, -5 to +5V and –10 to +10V
- Output impedance: 50 Ω
- Max. current: 25 mA sink or source

2.1.3. AUX1, AUX2

These standard BNC connectors are foreseen for additional hardware options that are not currently available.

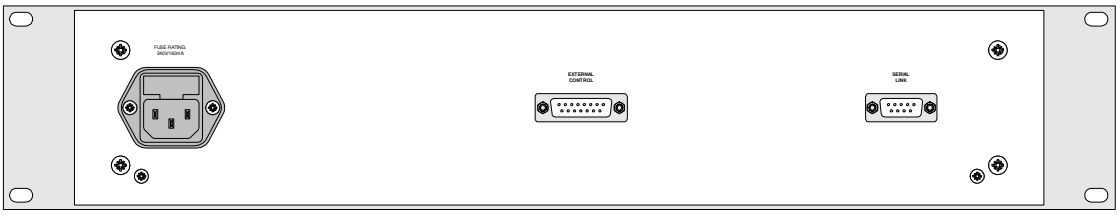
2.1.4. RUN, STOP

The RUN (green) and STOP (red) front panel LED's provide a visual indication of the state of the controller according to the following table:

RUN	STOP	Controller state
off	off	IDLE
blink	off	SCAN or SEARCH
on	off	RUN
off	blink	WAITBEAM or WAIT
off	on	PAUSED
blink	blink	OVERLOAD or ALARM

See the ?STATE request for a description of the different states of the controller.

2.2. Back Panel



2.2.1. Mains socket and fuse holder

The mains socket is a standard IEC (CEE22) power inlet with built-in fuse holder

- Nominal voltage: 230 VAC
- Fuse rate: 240V/160mA

2.2.2. EXTERNAL CONTROL

This connector provides a number of digital inputs and outputs to synchronise the operation of the controller with other devices. It also provides three control bits per input channel to drive the gain control of external preamplifiers, usually electrometers.

- Socket: DB15 female
- Inputs: TTL-level compatible, 47KΩ to ground
- Outputs: TTL-level compatible, max. output current 10mA
- Connector layout and signals:

Pin	Direction	Signal Name	Function
6	OUT	GINB0	3-bit gain control for INBEAM external preamplifier
5	OUT	GINB1	
4	OUT	GINB2	
1	OUT	GOUTB0	3-bit gain control for OUTBEAM external preamplifier
2	OUT	GOUTB1	
3	OUT	GOUTB2	
7	OUT	READY	Ready output. High in RUN state
9	OUT	NREADY	Not ready output. Low in RUN state
8	IN	INHIBIT	Inhibit input. Configurable polarity.
11	IN	INTERLOCK	Interlock input. Active low.
12	IN	-	Not used
13	-	VCC	5 volt power supply
15	-	GND	Logic ground

2.2.3. SERIAL LINK

This connector includes two serial line ports, RS232 and RS422. Both ports have the same function and any of them can be used for communication purposes.

- Socket: DB9 female
- Connector layout and signals:

Pin	Port	Signal	Direction
2	RS232	TXD	OUT
3		RXD	IN
5	common	GND	-
8	RS422	TXD+	OUT
9		TXD-	OUT
6		RXD+	IN
7		RXD-	IN

3. OPERATION INSTRUCTIONS

3.1. Installation and configuration

The installation of MoCo requires the connection of the beam monitors to the controller as well as the connection of the control output to the electronics that drives the optics. A number of configurable options have also to be set through the serial line by means of software commands. There is no intervention required inside the controller such as setting switches or jumpers. The configuration parameters are stored in an internal non-volatile memory and are retained after power off/on cycles.

3.1.1. Serial line

Access to the internal functions of the controller and configuration parameters can only be achieved through the serial line. Therefore the first step required is to connect the controller to a host computer or to an ASCII character terminal.
Any of the two available serial ports (RS232 or RS422) can be used for this purpose; the controller will respond by the same port it receives the commands.

The settings of both serial ports are: 9600 baud
No parity
1 stop bit

All the commands and requests sent to the controller are sequences of printable ASCII characters terminated by a carriage return (ASCII 0x0D character). The answer of the controller depends on the communication mode selected that can be changed with the commands ECHO and NOECHO.

Echo mode (terminal mode)

This mode should be used when the controller is connected to a dumb terminal. In this case the user types commands on the keyboard and reads the answers and error messages on the terminal screen without computer intervention. This mode is not active by default and the user has to send the ECHO command every time the controller is powered on.

In echo mode all the characters sent to the controller are echoed back to the terminal. The controller also sends human-readable messages to be printed on the terminal screen whenever an error is detected in commands or requests.

Noecho mode (host computer)

This is the default. In this mode no characters are echoed and no error messages are returned unless they are explicitly requested by means of the ?ERR request. This mode is intended to be used when a program running in a host computer communicates with the controller, sending commands and analysing the answers.

Commands are keywords starting by an alphabetical (“A” to “Z”) character followed by optional parameters. Commands do not produce any answer from the controller.

Requests are keywords starting by a question mark (“?”) character. The controller returns an answer to every request it receives even in the case of errors. An answer consists on one or more lines terminated by a carriage return + line feed character sequence (ASCII 0x0D 0x0A). When the answer consists of more that one line, the first and last lines contain a single “$” character (plus the two line terminator characters always present).
If wrong or not valid request is sent to the controller, its answer is the “ERROR” keyword.

A more detailed description of the protocol, including some advanced communication features not mentioned here like command acknowledgement, daisy-chaining of controllers or addressing, is presented in the Appendix A.

3.1.2. Control output

The output voltage range has to be set to match the input range of the electronics used to steer the optical instrument, usually a driver for piezoelectric actuators. The operation range is selected by setting the min and max voltage values by means of the OPRANGE command. The min and max voltage values must be comprised between –10 and +10 volts.

The current range can be read back from the controller with the ?OPRANGE request.

The operation of the voltage output can be tested by forcing the output to several voltage values within its operation range with the PIEZO command and verifying the output signal with an external meter.

3.1.3. Input channels

The input channels INBEAM and OUTBEAM must be connected to the corresponding beam monitors. In most of the cases where the beam detection is based on photodiodes, ion chambers or other devices that provide a low-current signal, the monitors can be directly connected to the triaxial current inputs of the controller. If a beam monitor generates a voltage signal or if an external preamplifier is used, the monitor signal should be connected to the voltage input of the corresponding channel.

The only exceptions to the previous paragraph are those cases in which the INBEAM channel is either not used (no normalisation, no beam presence detection required) or it is replaced by a software value. In the first case the configuration of the INBEAM channel is not required. In the second case the controller must be set to expect software values instead of a hardware signal (see INBEAM SOFT and SOFTBEAM commands).

In cases where physical monitors are required, each input signal should fall in one of the following three categories:

Current signals

The current input has to be selected with the INBEAM CURR or OUTBEAM CURR commands and the signal must be connected to the corresponding CURRENT IN input with a compatible triaxial plug.

It is preferable to use a triaxial cable to carry the signal from the monitor to the controller. If the monitor is floating, what in general is the best case, the current source (a floating photodiode for instance) should be connected between the inner conductor and the inner shield of the cable, the external shield being connected to the mechanical ground and used for overall shielding.

If the monitor is grounded, the current source should be connected to the inner conductor and the external cable shield used for signal return. In this case is still advisable to use triaxial cable and leave the inner shield unconnected at the monitor side. The inner shield will be still effective as signal guard to reduce low frequency current injection (50 Hz) in the signal path along the cable.

Voltage signals

The voltage input has to be selected with the INBEAM VOLT or OUTBEAM VOLT commands and the signal must be connected to the corresponding VOLTAGE IN input with a BNC terminated coaxial cable.

The full scale value of the voltage inputs can be internally selected to 4 ranges from 1.25 volts to 10 volts. Therefore if the monitor signal were in the millivolt range, it would be convenient to add external amplification. If the external amplifiers provide gain control that is compatible with MoCo, it is advisable to setup the controller to handle the gain as it is explained in the next case.

Use of external amplifiers

The EXTERNAL CONTROL connector at the back panel provides three TTL output bits per channel that may be used to select the gain of an external voltage or current amplifier. Any combination of these bits can be used, providing the possibility of setting up to eight different gain values.

In order to use this feature, the output from the amplifiers should be connected to the corresponding VOLTAGE IN input with a BNC terminated coaxial cable.
External gain control must be selected with the INBEAM EXT or OUTBEAM EXT commands.
The GAIN command must then be used to set the actual external amplifiers gains in the configuration memory of the controller.

Regardless of the type of input signal used, the polarity of each input channel can be inverted. The associated analog-to-digital converted can also be set either in unipolar or bipolar mode. All the input channel configuration options are selectable by means of the INBEAM and OUTBEAM commands.

The adequacy of the settings of an input channel can be verified by applying a known signal to the channel and reading the corresponding value with the ?BEAM of ?FBEAM requests.
In order to ease this procedure it may be convenient to enable the automatic range option with the SET AUTORANGE command (see Usage Notes). If this option is not selected and the signal value is too low or the channel saturates, the proper gain can be selected either by setting the full scale value (again by using the INBEAM or OUTBEAM commands), or issuing the AUTOBEAM command that tells the controller to choose the most appropriate gain for the current signal level.

Intensity signal should be read as positive values. If there are doubts about the polarity of a unipolar signal, as it is often the case with intensity monitors, it is preferable to set the channel in bipolar mode initially. Once the sign of the signal has been identified and the channel polarity has been inverted if needed to get a positive readout, the channel can then be switched to unipolar mode.

3.1.4. Inhibit and interlock options

The INHIBIT command can be used to enable or disable the inhibit function as well as to select the polarity of the external signal used. The inhibit signal must be able to drive the TTL INHIBIT input at the back panel control connector.

The interlock function is also triggered by an external TTL signal applied to the corresponding input at the control connector. The feature is enabled by the SET INTERLOCK command. The interlock input should be driven high for normal operation. When it is driven low or the cable is disconnected, the controller goes into alarm mode and the output voltage is driven to

a “safe” value. The safe voltage value is usually zero volts, but it can be set to any other value within the operation voltage range by means of the OPRANGE command.

3.2. Regulation setup

3.2.1. Regulation parameters

In order to be able to access the regulation features, first the operation mode has to be selected with the MODE command. The selected mode, either position or intensity, must correspond to the type of OUTBEAM monitor. The signal normalisation option must also be enabled or disabled as required by means of the SET/CLEAR NORMALISE commands.

The regulation time constant has also to be set with the TAU command and the operating point with the SETPOINT command.

When operating in intensity mode, the controller needs also to know the height of the peak response function of the optics. This value can be entered explicitly in the controller with the PEAK command, but in general it is simpler and more convenient to let the controller measure the peak height by itself during a tuning procedure as explained in 1.2.2.
The same is applicable for the slope of the response function. However, even if not recommended in general, it is possible to explicitly enter a slope value in the controller. In the case of intensity regulation, the slope of the response function is indirectly estimated from the peak width set through the PEAK command. In position mode the SLOPE command can be used for this purpose, paying attention to the sign of the programmed value.

The SPEED command can be used to set two velocity values that the controller applies when it drives the optics at constant speed. One value is used for open loop operation and the other during scanning phases for tuning or beam search purposes. The scanning range can be reduced with respect to the full operation range by means of the SRANGE command.

The beam detection features must be enabled or disabled as required by means of the SET/CLEAR BEAMCHECK commands.

3.2.2. Module operation

The internal state of the controller is displayed by the front panel LED’s (see 2.1.4) and can be read from the controller at any time as a character string. See the ?STATE request for a list of possible states and their meaning.

The PIEZO command can be used to drive the output voltage to any value within the operation range. The current output value can be read at any time from the controller with the ?PIEZO request.

The controller can be put into regulation mode by means of the GO or TUNE commands. GO starts regulation immediately with the current settings including the current peak or slope values. In general it is preferable to use the TUNE command instead, that performs a full tuning procedure as explained in 1.2.2 before switching the controller into regulation mode. If the tuning procedure fails, no peak is found in intensity mode for instance, the controller goes into idle state and does not start regulation.

The STOP command interrupts regulation and switches the controller to idle state. The output remains stable at the voltage value present at the output connector when the command was received.

When the controller is in regulation mode and receives a PIEZO or any other command intended to set one of the internal parameters, it first stops regulation and afterwards executes the received command. Therefore every time a controller parameter is changed, regulation must be explicitly restarted with a GO or TUNE command.

The TUNE PEAK is a special version of the TUNE command, only valid in intensity mode, that works somewhat differently than explained above: once the tuning procedure is successfully completed the controller drives the output voltage to the value at which the peak maximum was found and then switches to idle state.

Normally the controller starts a tuning procedure when it is explicitly requested by means of the TUNE or TUNE PEAK commands. The AUTOTUNE and AUTOPEAK commands may be used to instruct the controller to automatically start tuning sequences after certain events.

3.2.3. Setting the oscillation mode

Setting up the oscillation mode requires, in addition to the basic configuration, the selection of the parameters of the oscillation excitation. These parameters are the frequency, amplitude and phase of the oscillatory component that is summed to the controller output (see FREQUENCY, AMPLITUDE and PHASE commands).
This parameters are fixed and do not change during regulation.
Phase adjustment is tricky ...
Possibility of saturation (?OSCSAT)

3.3. Usage tips

- The ?INFO request provides a summary of the configuration parameters in the controller. It is a convenient and fast way of checking the current configuration.

- In general it is only advisable to select the autorange mode (SET AUTORANGE) when the controller is used only as a measurement instrument to monitor the beam. When it is used for regulation it is normally better to switch auitorange off (CLEAR AUTORANGE).

- If, in intensity mode, a TUNE command does not succeed, the controller remains in *IDLE* state instead of going into regulation mode and switching to *RUN* state. Among the possible causes for this behaviour are the following: there is no beam, the peak is not centered and not completely contained in the scanning range or there is a shift in the beam intensity due to a radiation background in the monitor.

- The default parameters for the beam detection feature (BEAMCHECK 0 …) should work properly in nearly all practical cases. It is not advisable to change the default values unless the rather cumbersome algorithm and the influence of the different parameters are well understood.

4. COMMAND SET

4.1. Command summary

Command		Description	Page
MODE	?MODE	Set/query operation mode	42
OPRANGE	?OPRANGE	Set/query output voltage range	46
SRANGE	?SRANGE	Set/query scanning voltage range	61
SPEED	?SPEED	Set/query scanning speed values	60
OUTBEAM	?OUTBEAM	Set/query *OUTBEAM* configuration	50
INBEAM	?INBEAM	Set/query *INBEAM* configuration	38
GAIN	?GAIN	Set/query preamplifier gain table	35
OFFSET	?OFFSET	Set/query electrometer input offset correction	45
BEAMCHECK	?BEAMCHECK	Set/query beam detection parameters	28
INHIBIT	?INHIBIT	Set/query hardware inhibition settings	41
SET	?SET	Set/query operation flags	56
CLEAR	?CLEAR	Clear/query operation flags	30
AUTOTUNE	?AUTOTUNE	Set/query autotune options	26
AUTOPEAK	?AUTOPEAK	Set/query autopeak options	25
AUTOBEAM		Adjust input channel ranges	24
AMPLITUDE	?AMPLITUDE	Set/query amplitude of oscillation	23
FREQUENCY	?FREQUENCY	Set/query oscillation frequency	34
PHASE	?PHASE	Set/query oscillation phase	53
TUNE		Start tuning sequence	65
GO		Start regulation	36
STOP		Stop operation	63
PAUSE	?PAUSE	Set/query pause mode	51
PIEZO	?PIEZO	Set/query output voltage	54
PEAK	?PEAK	Set/query peak parameters	52
SLOPE	?SLOPE	Set/query response function slope	59
SETPOINT	?SETPOINT	Set/query setpoint value	57
TAU	?TAU	Set/query regulation time constant	64
OSCIL	?OSCIL	Start/query oscillation state	48
	?STATE	Query controller state	62
	?BEAM	Query input values	27
	?FBEAM	Query filtered input values	33
SOFTBEAM	?SOFTBEAM	Set/query software *INBEAM* values	58
	?OSCBEAM	Query main and quadrature signal amplitudes	47
	?OSCSAT	Query ...	49
RESET		Reset controller	55
ECHO		Select echo mode	31
NOECHO		Cancel echo mode	44
	?ERR	Query last error	32
ADDR	?ADDR	Set/query serial line address	22
	?CHAIN	Query secondary serial port status	29
NAME	?NAME	Set/query module name	43
	?VER	Query firmware version	66
	?HELP	Query list of available commands	37

4.2. Command reference

ADDR / ?ADDR

Set/query serial line address

Syntax:

ADDR <slAddress>

Description:

Sets the serial line address to the <slAddress>. The address may be any alphanumerical character string. The maximum length is 9 characters. Any leading zeroes in the address are discarded.

When different modules are daisy-chained for communication, a particular module can be accessed by preceding any command with a prefix formed by the module address followed by a colon character (:). If the first character of the address is non-numeric, a leading zero must be added.

Syntax:

?ADDR

Answer:

<slAddress>

Description:

Returns the serial line address of the module.

Examples:

Command:	?ADDR
Answer:	<empty>
Command:	ADDR 3
Command:	?ADDR
Answer:	3
Command:	ADDR M2
Command:	?ADDR
Answer:	M2

AMPLITUDE / ?AMPLITUDE

Set/query amplitude of oscillation

Syntax:

AMPLITUDE <oscAmplitude>

Description:

Sets the amplitude of the oscillatory component of the output signal to <oscAmplitude> in volts. This value only has an effect in OSCILLATION mode.

In oscillation mode, the output voltage is the sum of the value returned by the ?PIEZO query and an oscillatory component of fixed amplitude, frequency and phase. Note however that controller only generates the oscillatory component when the regulation is active or if the oscillation is explicitly started by the OSCIL command. Otherwise there is no oscillatory component at the output connector.

Syntax:

?AMPLITUDE

Answer:

<oscAmplitude>

Description:

Returns the amplitude of the oscillatory component.

Examples:

Command:	`AMPLITUDE 0.1`
Command:	`?AMPLITUDE`
Answer:	`0.1`

AUTOBEAM

Adjust input channel ranges

Syntax:

AUTOBEAM

Description:

Sets the gains of the input channels to the appropriate ranges for the current signal levels.

This command is only effective when the controller is in IDLE state. It is intended as a simple way of readjusting the input channels when the AUTORANGE option is not set.

Example:

Command: `AUTOBEAM`

AUTOPEAK / ?AUTOPEAK

Set/query autopeak options

Syntax:

AUTOPEAK <flag> [<flag> […]]

AUTOPEAK OFF

Description:

Enables the autopeak modes selected by the list of flags entered as parameters. The parameter OFF clears all the autopeak flags.

Every autopeak mode has an associated trigger condition. If a particular mode is enabled and the trigger condition happens when the controller not regulating, a TUNE PEAK command will automatically be executed internally as soon as the trigger condition is removed.

The available flags and conditions are the same than for the AUTOTUNE command.

Syntax:

?AUTOPEAK

Answer:

<flag> <flag> …
or
OFF

Description:

Returns a list with the autupeak flags currently set. If no autopeak flag is set, it returns the string OFF.

Examples:

Command:	`AUTOPEAK OFF`
Command:	`?AUTOPEAK`
Answer:	`OFF`
Command:	`AUTOPEAK BEAMLOSS`
Command:	`AUTOPEAK INHIBIT`
Command:	`?AUTOPEAK`
Answer:	`BEAMLOSS INHIBIT`
Command:	`AUTOPEAK OFF OVERLOAD`
Command:	`?AUTOPEAK`
Answer:	`OVERLOAD`

AUTOTUNE / ?AUTOTUNE

Set/query autotuning options

Syntax:

AUTOTUNE <flag> [<flag> […]]

AUTOTUNE OFF

Description:

Enables the autotuning modes selected by the list of flags entered as parameters. The parameter OFF disables all autotuning modes.

Every autotuning mode has an associated trigger condition. If a mode is enabled and the trigger condition happens when the controller is in regulation mode, a TUNE command will automatically be executed internally as soon as the trigger condition is removed.

The available flags and conditions are the following:

Flags		Trigger condition
BEAMLOSS	-	The photon beam is back after a beam loss and the controller leaves a waiting state. Beam detection (see SET BEAMCHECK) must be enabled.
OVERLOAD	-	An overload in any of the active input channels was detected.
INHIBIT	-	The controller was inhibited by an external control signal. The inhibit feature (see INHIBIT command) must be enabled.

Syntax:

?AUTOTUNE [OFF]

Answer:

<flag> <flag> …
or
OFF

Description:

Returns a list with the autuning flags currently set. If no autotuning flag is set, it returns the string OFF.

With the parameter “OFF” returns the list of autuning flags that are not currently set.

Examples:

Command:	AUTOTUNE OFF BEAMLOSS INHIBIT
Command:	?AUTOTUNE
Answer:	BEAMLOSS INHIBIT
Command:	AUTOTUNE OFF
Command:	?AUTOTUNE
Answer:	OFF

?BEAM

Query input values

Syntax:

?BEAM

Answer:

<inbeamValue> <outbeamValue>

Description:

Returns the current values of the input channels in the units that correspond to the particular settings of each channel.

If a channel is set to “VOLT” or “CURR” mode, the channel value is expressed in volts or amps respectively. If the channel is set to “EXT”, the value is expressed in the corresponding units at the input of the external preamplifier.

If the *INBEAM* channel is set to “SOFT” mode, <inbeamValue> is expressed in whatever unit were used in the SOFTBEAM command.

Example:

Command: `?BEAM`

Answer: `4.256e-09 1.033e-07`

BEAMCHECK / ?BEAMCHECK

Set/query beam detection parameters

Syntax:

BEAMCHECK <absThresh> <relThresh> [[<inbTau>] [<settTime>]

Description:

Sets the parameters of the internal beam detection algorithm. The relative threshold <relThresh> is the value used to detect the “beam is lost” and “beam is back” conditions. The time constant <inbTau> determines the time scale applied in seconds.

When controller starts operation, an absolute beam detection threshold is set to predefined values. If the *INBEAM* input channel is set in current or voltage mode, the threshold is set to 2% of the full scale range. In software mode the threshold is set to the <softThresh> parameter (see INBEAM command).

Once regulation is active, the absolute threshold is continuously recalculated by multiplying the low-pass filtered *INBEAM* intensity by the relative threshold value <relThresh>. For instance if the relative threshold is 0.2, the beam is considered lost when the *INBEAM* intensity drops below 20% of its current value in a time interval shorter than <inbTau>.

If beam detection is active and the beam is considered lost, the controller goes into WAITBEAM state and the beam intensity absolute threshold is stored. When the low-pass filtered *INBEAM* intensity stabilises at a value higher than the stored absolute threshold, the controller considers that the beam is back and switches to WAIT state. The controller remains in WAIT state for <settTime> seconds before resuming normal operation.

The beam detection feature can be activated and cancelled by the SET and CLEAR commands.

The <inbTau> parameter sets also the time constant of the internal digital filters that produce the filtered values returned by the ?FBEAM query.

Syntax:

?BEAMCHECK

Answer:

<absThresh> <relThresh> <inbTau> <settTime>

Description:

Returns the parameters set by the BEAMCHECK command.

Examples:

Command: `BEAMCHECK 0 0.3`

Command: `?BEAMCHECK`

Answer: `0 0.333333 1.024 0`

?CHAIN

Query secondary serial port status

Syntax:

?CHAIN

Answer:

{YES | **NO} {RS232** | **RS422** | **NONE}**

Description:

Returns whether or not (“YES” or “NO”) there is another instrument connected to the secondary serial port as well as its electrical specification (“RS232” or “RS422”).

The secondary serial port is the port that is not used for communications with the host computer and that can be eventually used for daisy-chaining of modules.

Example:

Command: `?CHAIN`

Answer: `NO RS422`

CLEAR / ?CLEAR

Clear operation flags

Syntax:

CLEAR <flag> [<flag> […]]

Description:

Clears any combination of the operation flags. See the SET command for a list of available flags and their meaning.

Syntax:

?CLEAR

Answer:

<flag> <flag> …

Description:

Returns a list of the operation flags that are currently NOT set.

Examples:

Command:	`?SET`
Answer:	`RIGHT BEAMCHECK NORMALISE AUTORANGE`
Command:	`CLEAR AUTORANGE NORMALISE`
Command:	`?SET`
Answer:	`RIGHT BEAMCHECK`

ECHO

Switches the echo mode on

Syntax:

ECHO

Description:

This mode is intended to be used when the instrument is connected to a dumb character terminal. In this mode the characters received by the unit through the serial lined are sent back to the terminal and error messages are produced as soon as the corresponding errors are detected.

If the instrument is controlled by a program running in host computer the echo mode should be switched off (see NOECHO command). In this case the error messages can be requested by means of the ?ERR query.

Example:

Command: `ECHO`

?ERR

Query last error

Syntax:

?ERR

Answer:

{OK | <errorMessage>}

Description:

Returns the string “OK” if the execution of the last command was successful or an error message describing the error in the last command.

Example:

Command:	`?VER`
Answer:	`MOCO 01.02`
Command:	`?ERR`
Answer:	`OK`
Command:	`?VERSION`
Answer:	`ERROR`
Command:	`?ERR`
Answer:	`Command not recognised.`

?FBEAM

Query filtered input values

Syntax:

?FBEAM

Answer:

<inbeamFilteredValue> <outbeamFilteredValue>

Description:

Returns the values of the input channels after the digital low pass filter in the same units than non filtered values returned the ?BEAM query.

The filter value of the *INBEAM* channel is used only by the beam detection algorithm. The regulation algorithm makes use of the non filtered values.

Both filters are set to the same time constant that can be selected with the BEAMCHECK command.

Example:

Command:	`?FBEAM`
Answer:	`4.25612e-08 1.456751`

FREQUENCY / ?FREQUENCY

Set/query oscillation frequency

Syntax:

FREQUENCY <oscFrequency>

Description:

Sets the frequency of the oscillatory component of the output signal to <oscFrequency> in Herzs. This value only has an effect in OSCILLATION mode.

Only frequency values greater or equal than 500 Hz and that correspond to an oscillation period of an integer number of milliseconds can be set in the controller.

In oscillation mode, the output voltage is the sum of the value returned by the ?PIEZO query and an oscillatory component of fixed amplitude, frequency and phase. Note however that controller only generates the oscillatory component when the regulation is active or if the oscillation is explicitly started by the OSCIL command. Otherwise there is no oscillatory component at the output connector.

Syntax:

?FREQUENCY

Answer:

<oscFrequency>

Description:

Returns the frequency of the oscillatory component.

Examples:

Command:	`FREQUENCY 160`
Command:	`?FREQUENCY`
Answer:	`166.6`

GAIN / ?GAIN

Set/query preamplifier gain table

Syntax:

GAIN {INBEAM | OUTBEAM} <g0> [<g1> [<g2> [...[<g7>]]]]]

GAIN {INBEAM | OUTBEAM} DEFAULT

Description:

Sets in the controller the gain values of the preamplifier of an input channel (INBEAM or OUTBEAM).

This command is primarily intended to load the available gain ranges in the controller memory when external preamplifiers with compatible gain control are used. In this case the input channel has to be set to “EXT” mode.

MoCo outputs three control bits per channel to drive the preamplifier gain. Bit patterns from 0 (binary 000) to 7 (binary 111) may be used to select the up to 8 different amplification ranges. The values g0 to g7 are the gain values of the corresponding ranges. The gain values for non-valid bit patterns have to be set to zero.

All the gain values have to be expressed in volts per input unit. It is convenient to set the input unit as the physical unit of the signal at the preamplifier input, usually volts or amps.

This command can also be used when the input channel is in “CURR” mode but the gain of the built-in electrometers has been changed with respect to the default values by modifying internal components. In that case the “DEFAULT” parameter can be used to restore the default gain values.

This command cannot be issued when the input channel is in “VOLT” mode.

Syntax:

?GAIN {INBEAM | OUTBEAM}

Answer:

<g0> <g1> <g2> <g3> <g4> <g5> <g6> <g7>

or

DEFAULT

Description:

Returns the gain table of the specified input channel. This command cannot be issued when the input channel is in “VOLT” mode.

Examples:

Command: `GAIN OUTBEAM 0 1e6 0 1e7 1e8`

Command: `?GAIN OUTBEAM`

Answer: `0 1e+06 0 1e+07 1e+08 0 0 0`

GO

Start regulation

Syntax:

GO [{ <sPoint> | # }]

Description:

Puts the unit into regulation mode. The controller drives the actuator in order to stabilise either the intensity or the position of the beam depending on the current mode selected (see MODE command).

With no parameters the operation point is defined by the current values of the operation settings. In intensity mode the operation settings can be selected by the SETPOINT and PEAK commands and the RIGHT or LEFT flags (see SET command). In position mode the operation settings can be selected by the SETPOINT and SLOPE commands.

With the <sPoint> or "#" parameters, before regulation starts, the setpoint value is changed either to the given value or to the current operation point (see SETPOINT command for details). The new setpoint value can be checked with the ?SETPOINT query. The other operation parameters (peak or slope, depending on the mode) must be properly set beforehand.

In all cases, setting of the operation parameters may be simplified by using the TUNE command.

Examples:

Command: `GO`

Command: `GO 0.85`

Command: `GO #`

?HELP

Query list of available commands

Syntax:

?HELP

Description:

Returns the list of available commands and queries.

Example:

Command: `?HELP`

Answer:

```
$
     RESET
        GO
      STOP
      TUNE
  AUTOTUNE  ?AUTOTUNE
  AUTOPEAK  ?AUTOPEAK
     PIEZO  ?PIEZO
            ?STATE
            ?BEAM
            ?FBEAM
  AUTOBEAM
  SOFTBEAM  ?SOFTBEAM
   OPRANGE  ?OPRANGE
    SRANGE  ?SRANGE
       SET  ?SET
     CLEAR  ?CLEAR
     PAUSE  ?PAUSE
   INHIBIT  ?INHIBIT
       TAU  ?TAU
  SETPOINT  ?SETPOINT
      MODE  ?MODE
    INBEAM  ?INBEAM
   OUTBEAM  ?OUTBEAM
      GAIN  ?GAIN
      ZERO  ?ZERO
    OFFSET  ?OFFSET
  ADCCALIB  ?ADCCALIB
 BEAMCHECK  ?BEAMCHECK
     SPEED  ?SPEED
      PEAK  ?PEAK
     SLOPE  ?SLOPE
            ?INFO
   ISGTEST  ?ISGTEST
      ECHO
    NOECHO
            ?ERR
      ADDR  ?ADDR
            ?CHAIN
      NAME  ?NAME
            ?VER
            ?HELP
$
```

INBEAM / ?INBEAM

Set/query INBEAM configuration

Syntax:

INBEAM [{CURR | VOLT | EXT}] [{NORM | INV}] [{BIP | UNIP}] [<fS>] [{AUTO | NOAUTO}]

INBEAM [SOFT] [<softThresh>]

Description:

Sets the configuration of the *INBEAM* channel. The signal source can be set to “CURR”, “VOLT”, “EXT” or “SOFT”.

In current (“CURR”) or voltage (“VOLT”) mode the *INBEAM* value is the measure of the electrical signals applied to the corresponding front panel connector. When using external preamplifiers with several digitally controlled gain ranges, the *INBEAM* channel can be set to “EXT”. In this case the output signal of the preamplifier must be applied at the voltage input of the controller. If the gain values of the preamplifier are properly set with the GAIN command, the signal value will be expressed in units at the preamplifier input.

It is possible to select if the signal polarity is inverted (“INV”) or not (“NORM”), change the gain by specifying the full scale value (<fS>) and set or clear the autoscaling (“AUTO” or “NOAUTO”).

The channel can also be configured in unipolar (“UNIP”) or bipolar (“BIP”) mode. In bipolar mode the valid input signal range is from -<fS> to +<fS>. In unipolar mode the range goes from 0 to +<fS>. Unipolar negative signals can be measured by inverting the signal polarity.

<fS> must be specified in units that correspond to the signal source (amps, volts or units at the inputs of the external preamplifier). When a full scale value is passed as parameter, the internal gain is adjusted to get the closest full scale value higher than <fS>.

When the *INBEAM* vale is used for normalisation (see SET NORMALISE), the channel cannot be set in bipolar mode.

In software mode (“SOFT”) the *INBEAM* value is set by the host computer by issuing the SOFTBEAM command.

The parameter <softThresh> is the initial detection threshold used for beam detection purposes when regulation starts. It must be expressed in the same units than <softBeamVal> in the SOFTBEAM command.

Syntax:

?INBEAM

Answer:

{CURR | VOLT | EXT} {NORM | INV} {BIP | UNIP} <fS> {AUTO | NOAUTO}

or

SOFT <softThreshold>

Description:

Returns the configuration parameters of the *INBEAM* channel.

Examples:

Command: `?INBEAM`

Answer: `CURR NORM UNIP 1.25e-09 AUTO`

Command: `INBEAM INV 9e-8`

Command: `?INBEAM`

Answer: `CURR INV UNIP 1e-07 AUTO`

Command: `INBEAM SOFT`

Command: `?INBEAM`

Answer: `SOFT 1`

?INFO

Query module configuration

Syntax:

?INFO [ALL]

Description:

Returns a multiline answer with the current configuration.

The information is presented as a list of valid commands with the appropriate parameters that can be sent back to the controller in the case that reconfiguration is needed.

With the parameter “ALL” returns additional calibration parameters that should not normally modified by the average user.

Example:

Command: `?INFO`

Answer:

```
$
MOCO 01.02  -  Current settings:
  NAME "no name"
  ADDR ""

  OPRANGE 0 10 0
  INBEAM SOFT 1
  GAIN INBEAM DEFAULT
  OUTBEAM EXT NORM UNIP 1.25e-08 AUTO
  GAIN OUTBEAM  0 1e+06 0 1e+07 1e+08 0 0 0

  MODE INTENSITY
  PEAK 1 0.1 0
  SETPOINT 0.8
  TAU 1
  SET RIGHT BEAMCHECK NORMALISE
  CLEAR AUTORUN AUTORANGE INTERLOCK
  AUTOTUNE OFF
  AUTOPEAK OFF
  BEAMCHECK 0 0.333333 1.024 0
  SRANGE 0 10
  SPEED 2 50
  INHIBIT OFF LOW
$
```

INHIBIT / ?INHIBIT

Se/query hardware inhibition settings

Syntax:

INHIBIT [{ ON | OFF }] [{ HIGH | LOW }]

Description:

Enables or disables the hardware inhibition feature (“ON” or “OFF”) and selects the active level of the inhibit signal (“HIGH” or “LOW”). When this feature is enabled and the external logic signal *inhibit* switches to its active level, the controller goes into PAUSE state and regulation stops. When the *inhibit* signal switches to its non-active logic level regulation resumes.

The controller behaviour is analogous to the obtained with the PAUSE command but in this case it is hardware driven.

Returns the current settings hardware inhibition feature.

Syntax:

?INHIBIT

Answer:

{ ON | OFF } { HIGH | LOW }

Description:

Returns the current settings of the hardware inhibition feature.

Examples:

Command:	`INHIBIT ON`
Command:	`?INHIBIT`
Answer:	`ON HIGH`

MODE / ?MODE

Set/query operation mode

Syntax:

MODE {POSITION | INTENSITY | OSCILLATION}

Description:

Sets the operation mode to either intensity, position or oscillation. The modes position and intensity must correspond to the type of beam monitor connected to *OUTBEAM.* The mode oscillation requires an intensity monitor.

Syntax:

?MODE

Answer:

{POSITION | INTENSITY | OSCILLATION}

Description:

Returns the current operation mode.

Examples:

Command:	`?MODE`
Answer:	`POSITION`
Command:	`MODE INTENSITY`
Command:	`?MODE`
Answer:	`INTENSITY`

NAME / ?NAME

Set/query module name

Syntax:

NAME <appName>

Description:

Sets the internal module name to the ASCII string <appName>. This name is only used for identification purposes and user convenience.

The maximum length is 20 characters.

Syntax:

?NAME

Answer:

<appName>

Description:

Returns the application name field.

Examples:

Command:	`NAME DEV01`
Command:	`?NAME`
Answer:	`DEV01`
Command:	`NAME "Main Monochromator"`
Command:	`?NAME`
Answer:	`Main Monochromator`

NOECHO

Cancel echo mode

Syntax:

NOECHO

Description:

Switches the echo mode off. See the ECHO command for more details.

Example:

Command: `NOECHO`

OFFSET / ?OFFSET

Set/query input offset correction

Syntax:

OFFSET {INBEAM | OUTBEAM} [<offsCorr>]

Description:

Sets the magnitude in mV of the correction used to cancel the input offset voltage of the internal current amplifier of the selected channel. When using beam monitors with low output resistance current sources, like large area photodiodes in photovoltaic mode, the input offset voltage appears as a burden voltage that induces an extra current component that adds up to the beam monitor signal.

WARNING: These parameters does not correct for signal offsets in the input sisgnals. This setting is part of a calibration procedure and should normally not be changed by the user.

Syntax:

?OFFSET

Answer:

<inbeamOffsCorr> <outbeamOffsCorr>

Description:

Returns the input offset voltage correction of the *INBEAM* and *OUTBEAM channels* in mV.

Examples:

Command:	`OFFSET INBEAM 0.153`
Command:	`OFFSET OUTBEAM -0.023`
Command:	`?OFFSET`
Answer:	`0.153 -0.023`

OPRANGE / ?OPRANGE

Set/query output voltage range

Syntax:

OPRANGE <Vmin> <Vmax> [<Vsafe>]

Description:

Sets the operation range of the output voltage expressed in volts. The output signal is always limited to the range between <Vmin> and <Vmax> that should always between –10V and +10V.

The <Vsafe> is usually zero and corresponds to the voltage that the controller will output in case it goes into ALARM state after an interlock event.

Syntax:

?OPRANGE

Answer:

<Vmin> <Vmax> <Vsafe>

Description:

Returns the operation range and the “safe” output voltage in volts.

Examples:

Command:	`OPRANGE -5 5`
Command:	`?OPRANGE`
Answer:	`-5 5 0`

?OSCBEAM

Query main and quadrature signal amplitudes

Syntax:

?OSCBEAM

Answer:

<mainSignal> <quadSignal>

Description:

In oscillation mode returns the amplitudes of the OUTBEAM signal at the frequency of the oscillatory component. As the detection is synchronous, this query returns two values: one value, <mainSIgnal>, that corresponds to the component in phase with the oscillation and another one, <quadSignal>,that corresponds to the component in quadrature at 90 degress of phase shift.

Examples:

Command:	`?OSCBEAM`
Answer:	`1.345 -0.03625`
Command:	`OSCIL OFF`
Command:	`?OSCBEAM`
Answer:	`0 0`

OSCIL / ?OSCIL

Start/query oscillation state

Syntax:

OSCIL [{ ON | OFF }]

Description:

In oscillation mode, this command starts/stops the oscillatory component in the voltage output. When the controller is not regulating, the oscillation is by default switched off. This command is required to start the oscillation for instance during the setup phase or for diagnostics.

Syntax:

?OSCIL

Answer:

{ ON | OFF }

Description:

Returns the oscillation state of the voltage output.

Examples:

Command:	`OSCIL ON`
Command:	`?OSCIL`
Answer:	`ON`

?OSCSAT

Query saturation level

Syntax:

?OSCSAT

Answer:

<satLevel>

Description:

In oscillation mode, this query returns ...

Examples:

Command:	`?OSCSAT`
Answer:	`0.12`

OUTBEAM / ?OUTBEAM

Set/query OUTBEAM configuration

Syntax:

OUTBEAM [{CURR | VOLT | EXT}] [{NORM | INV}] [{BIP | UNIP}] [<fS>] [{AUTO | NOAUTO}]

Description:

Sets the configuration of the *OUTBEAM* channel. The signal source can be set to current ("CURR") or voltage ("VOLT"). Different input connectors must be used for current or voltage signals. When using external preamplifiers with several digitally controlled gain ranges, the *OUTBEAM* channel can be set to "EXT". In this case the output signal of the preamplifier must be applied at the voltage input of the controller. If the gain values of the preamplifier are properly set with the GAIN command, the signal value will be expressed in units at the preamplifier input.

It is possible to select if the signal polarity is inverted ("INV") or not ("NORM"), change the gain by specifying the full scale value (<fS>) and set or clear the autoscaling ("AUTO" or "NOAUTO").

The channel can also be configured in unipolar ("UNIP") or bipolar ("BIP") mode. In bipolar mode the valid input signal range is from -<fS> to +<fS>. In unipolar mode the range goes from 0 to +<fS>. Unipolar negative signals can be measured by inverting the signal polarity.

<fS> must be specified in units that correspond to the signal source (amps, volts or units at the inputs of the external preamplifier). When a full scale value is passed as parameter, the internal gain is adjusted to get the closest full scale value higher than <fS>.

Syntax:

?OUTBEAM

Answer:

{CURR | VOLT | EXT} {NORM | INV} {BIP | UNIP} <fS> {AUTO | NOAUTO}

Description:

Returns the configuration parameters of the *OUTBEAM* channel.

Examples:

Command:	`?OUTBEAM`
Answer:	`CURR NORM UNIP 1.25e-09 AUTO`
Command:	`OUTBEAM VOLT NOAUTO`
Command:	`?OUTBEAM`
Answer:	`VOLT NORM UNIP 1.25 NOAUTO`

PAUSE / ?PAUSE

Set/clear/query pause mode

Syntax:

PAUSE [{ON | OFF }]

Description:

Switches the pause mode on or off. When pause mode is switched on the controller goes into PAUSE state, if the regulation was active it stops and the output voltage is kept constant. When the mode is switch off the controller resumes its previous mode of operation.

A similar effect can be obtained by a change of polarity of a external logic signal (see INHIBIT command).

Syntax:

?PAUSE

Answer:

{ON | OFF}

Description:

Returns the current state of the pause mode.

Examples:

Command:	PAUSE
Command:	?PAUSE
Answer:	ON
Command:	PAUSE OFF
Command:	?PAUSE
Answer:	OFF

PEAK / ?PEAK

Set/query peak parameters

Syntax:

PEAK <pkHeight> <pkWidth> [<pkPos>]

Description:

Sets the peak height and width parameters that the controller uses when operates in intensity mode.

<pkHeight> is the peak value of the response function of the system. If normalisation is not set, the peak height must be expressed in *OUTBEAM* units. If normalisation is set, the peak height must be also normalised by the *INBEAM* signal and expressed in the appropriate units.

<pkWidth> is the full width at half maximum (FWHM) of the peak expressed in volts at the output signal.

<pkPos> is the peak position expressed in volts. This value is not actually used by the controller but it is stored for information purposes.

In general, instead of using the PEAK command, it is recommended to use the TUNE command to let the controller to estimate these values.

Syntax:

?PEAK

Answer:

<pkHeight> <pkWidth> <pkPos>

Description:

Returns the settings for peak height and width as well as the peak position. These values are only used when the controller operates in intensity mode.

Examples:

Command: `PEAK 1.03E3 0.17`

Command: `?PEAK`

Answer: `1.03e+03 0.17 0`

PHASE / ?PHASE

Set/query oscillation phase

Syntax:

PHASE <oscPhase>

Description:

Sets the phase of the oscillatory component of the output signal to <oscPhase> in degrees. This value only has an effect in OSCILLATION mode.

In oscillation mode, the output voltage is the sum of the value returned by the ?PIEZO query and an oscillatory component of fixed amplitude, frequency and phase. Note however that controller only generates the oscillatory component when the regulation is active or if the oscillation is explicitly started by the OSCIL command. Otherwise there is no oscillatory component at the output connector.

Syntax:

?PHASE

Answer:

<oscPhase>

Description:

Returns the phase of the oscillatory component.

Examples:

Command:	`PHASE 90`
Command:	`?PHASE`
Answer:	`90`

PIEZO / ?PIEZO

Set/query output voltage

Syntax:

PIEZO <outVoltage>

Description:

Ramps the output voltage to <outVoltage>. The voltage changes at the rate <moveSpeed> programmed with the SPEED command. <outVoltage> is expressed in volts and must be within the range defined by the OPRANGE command.

The controller first goes into MOVING state while the ramp is in progress and once the final voltage is reached switches to IDLE state. Any regulation, peak search or scanning procedure before this command is issued is aborted and not resumed.

Syntax:

?PIEZO

Answer:

<outVoltage>

Description:

Returns the current value of the output voltage.

Examples:

Command:	`PIEZO 3.45`
Command:	`?PIEZO`
Answer:	`3.45`

RESET

Reset controller

Syntax:

RESET [DEFAULT]

Description:

With no parameters this command reinitialises the controller keeping the current configuration. With the DEFAULT parameter it also resets the internal configuration to default values.

Examples:

Command: `RESET`

SET / ?SET

Set/query operation flags

Syntax:

SET <flag> [<flag> […]]

Description:

Sets the operation flags in the parameter list. The available flags are the following:

General flags		Description
NORMALISE	-	Enables normalisation mode.
BEAMCHECK	-	Enables the beam detection feature.
AUTORUN	-	Enables the autorun feature.
AUTORANGE	-	Enables the autorange feature.
INTERLOCK	-	Enables the interlock feature.

Intensity mode only:

LEFT	-	Operation point at the low voltage side of the response function.
RIGHT	-	Operation point at the high voltage side of the response function.

The command CLEAR can be used to reset the operation flags.

Syntax:

?SET

Answer:

<flag> <flag> …

Description:

Returns a list of the operation flags that are currently set.

Examples:

Command:	?SET
Answer:	NORMALISE
Command:	?CLEAR
Answer:	BEAMCHECK AUTORUN AUTORANGE INTERLOCK
Command:	SET BEAMCHECK INTERLOCK
Command:	?SET
Answer:	NORMALISE BEAMCHECK INTERLOCK

SETPOINT / ?SETPOINT

Set/query setpoint value

Syntax:

SETPOINT {<sPoint> | # }

Description:

Sets the operation setpoint value. The meaning of <sPoint> differs depending on the operation mode:

In intensity mode <sPoint> is the relative intensity in the response function. The actual operation point is obtained by multiplying <sPoint> by the peak height returned by ?PEAK. For instance, a value of 0.8 means that the controller will regulate the optical system to operate at 80% of the peak intensity.

In position mode <sPoint> is the absolute beam position at which the controller will make the system operate. If normalisation is not set, as it would be the usual case in this mode, the beam position must be expressed in *OUTBEAM* units. If normalisation is set, the beam position must be also normalised by the *INBEAM* signal and expressed in the appropriate units.

With the "#" parameter the setpoint value is changed to the current operation point. The new setpoint value can be checked with the ?SETPOINT query.

Syntax:

?SETPOINT

Answer:

<sPoint>

Description:

Returns the current operation set point value.

Examples:

Command:	`SETPOINT 0.8`
Command:	`?SETPOINT`
Answer:	`0.8`
Command:	`SETPOINT #`

SOFTBEAM / ?SOFTBEAM

Set/query software INBEAM values

Syntax:

SOFTBEAM <softBeamVal>

Description:

When the *INBEAM* channel is set to software mode, this command is used to set the internal *INBEAM* value.

A low-pass digital filter algorithm is applied to the <softBeamVal> values received from the host computer. The *INBEAM* value is set to the result of the filtering procedure.

Syntax:

?SOFTBEAM

Answer:

<softBeamVal>

Description:

Returns the current software *INBEAM* value.

Examples:

Command:	`SOFTBEAM 180`
Command:	`?SOFTBEAM`
Answer:	`180`

SLOPE / ?SLOPE

Set/query response function slope

Syntax:

SLOPE <slopeVal>

Description:

When the controller operates in position regulation mode this command sets the slope of the response function. If normalisation is not active the slope must be expressed in *OUTBEAM* signal units per output volt.

If normalisation is active the slope value has also to be normalised to respect to the *INBEAM* value.

Syntax:

?SLOPE

Answer:

<slopeVal>

Description:

Returns the slope parameter.

Examples:

Command:	`SLOPE 0.0362`
Command:	`?SLOPE`
Answer:	`0.0362`

SPEED / ?SPEED

Set/query scanning speed values

Syntax:

SPEED <scanSpeed> [<moveSpeed>]

Description:

Sets the speeds in volts/sec applied when ramping the output voltage.

<scanSpeed> is the value used when the controller is scanning or searching the peak during a TUNE command.

<moveSpeed> is the value used when the optical instrument is moved to a in open-loop by the PIEZO command.

During closed-loop operation the previous values are ignored and the output voltage change rate is related the regulation time constant set by the TAU command.

Syntax:

?SPEED

Answer:

<scanSpeed> [<moveSpeed>]

Description:

Returns the speeds in volts/sec applied when ramping the output voltage.

Examples:

Command:	`SPEED 1 10`
Command:	`?SPEED`
Answer:	`1 10`
Command:	`SPEED 0.5`
Command:	`?SPEED`
Answer:	`0.5 10`

SRANGE / ?SRANGE

Set/query output voltage range

Syntax:

SRANGE <Vmin> <Vmax>

Description:

Sets the scanning range of the output voltage. <Vmin> and <Vmax> are expressed in volts. The controller limits the output voltage excursion to this range during scanning procedures.

Syntax:

?SRANGE

Answer:

<Vmin> <Vmax>

Description:

Returns the scaning range of the output voltage in volts.

Examples:

Command:	`OPRANGE -10 10`
Command:	`SRANGE -2 8`
Command:	`?SRANGE`
Answer:	`-2 8`
Command:	`OPRANGE 0 10`
Command:	`?SRANGE`
Answer:	`0 8`

?STATE

Query controller state

Syntax:

?STATE

Answer:

Answer: [PAUSED] <currState>

Description:

Returns the current state of the controller as one of the following strings:

State	Meaning
IDLE	- The controller is idle
MOVE	- The output voltage is being ramped in open loop.
SCAN	- The controller is scanning the output voltage during the tuning procedure.
SEARCH	- Regulation is on but the controller is searching the operation point.
RUN	- Regulation is on and the system is locked at the operation point.
WAITBEAM	- The controller waits for the beam is back after a beam loss.
WAIT	- The controller waits for the beam stabilisation after a beam loss.
OVERLOAD	- Any of the active input channels is saturated.
ALARM	- The interlock input has tripped the controller.

If the controller is in pause mode the state is preceded of the keyword PAUSED

Examples:

Command:	?STATE
Answer:	RUN
Command:	STOP
Command:	?STATE
Answer:	IDLE

STOP

Stop operation

Syntax:

STOP

Description:

Stops regulation or any scanning activity in progress and sets the unit in IDLE state.

When any configuration parameter is changed in the controller, it always executes internally a STOP command.

Example:

Command: `STOP`

TAU / ?TAU

Set/query regulation time constant

Syntax:

TAU <regTau>

Description:

Sets the value of the regulation time constant in seconds.

Syntax:

?TAU

Answer:

<regTau>

Description:

Returns the value of the regulation time constant.

Example:

Command:	`TAU 1`
Command:	`?TAU`
Answer:	`1`

TUNE

Start tuning sequence

Syntax:

TUNE [{<sPoint> | #}]

Description:

Starts the tuning procedure and at the end if tuning was successfully completed, the controller goes into regulation mode. Otherwise it goes into IDLE state.

The parameters <sPoint> and "#" allow to change the setpoint before starting the tuning sequence. The meaning of the parameters is the same than for the SPOINT command.

Example:

Command: `TUNE`

Command: `TUNE 0.8`

Command: `TUNE #`

TUNE PEAK

Start peak tuning sequence

Syntax:

TUNE PEAK

Description:

In intensity mode the controller starts a tuning procedure. If a peak was found and tuning is successfully completed, the controller moves the optics to the peak position and goes into IDLE state.

Example:

Command: `TUNE PEAK`

?VER

Query firmware version

Syntax:

?VER

Answer:

MOCO XX.YY

Description:

Returns the version number XX.YY of the firmware.

Example:

Command: `?VER`

Command: `MOCO 01.02`

Appendix A. COMMUNICATION PROTOCOL

This section covers the communication protocol implemented in a number of instruments developed in the Instrument Support Group at the ESRF. An instrument that adheres to this protocol and conventions is referred as an *isgdevice*. The information is presented here in a generic way and most of the information applies to any *isgdevice*. Some of the options, like GPIB communication or binary data transfer, may not be implemented in a particular instrument and therefore the related information should be ignored.

A.1. Communication port

An *isgdevice* is equipped with one or more communication ports. A communication port may be an asynchronous serial line or a GPIB interface. The type and number of ports as well as the electrical interface (RS232 or RS422) depend on the particular instrument.
In a general case an *isgdevice* is equipped with a RS232 port, a RS422 port and a GPIB interface.

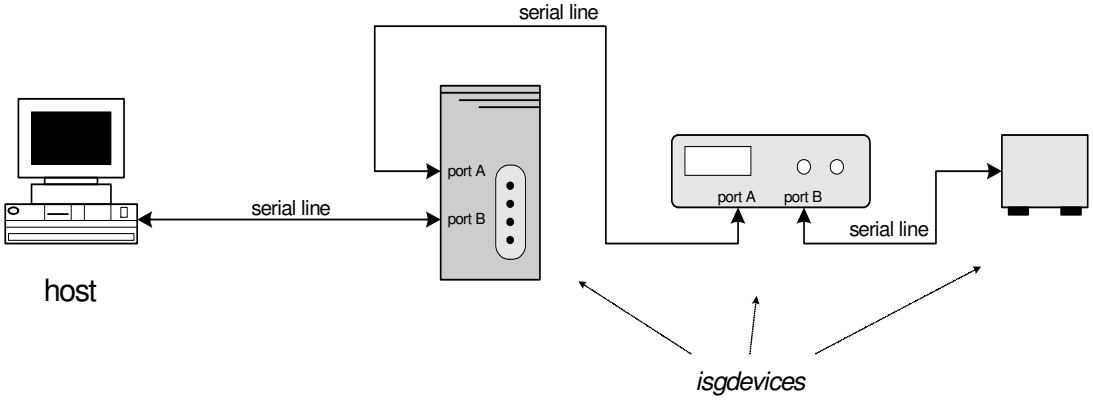
If an *isgdevice* is equipped with both serial ports and GPIB, the GPIB interface only becomes active if no cables are connected to the serial line ports. Whenever a connection is made to any of the serial ports, the GPIB interface is disabled.

A.1.1. Serial line ports

Serial line ports are asynchronous communication ports electrically compatible with either the RS232 or RS422 specification. In most of the *isgdevices* there are two serial ports available, one of each type.

The primary port is the one through which the *isgdevice* receives commands and requests. There is no special configuration procedure to select the primary port. After power-up the device listens to all the available serial ports and the first one it receives valid data through is selected as primary. If there is another one, it is treated as secondary.

A secondary serial line port in an *isgdevice* can be used to connect another *isgdevice*. In this way an arbitrary number of *isgdevices* can be connected in a daisychain configuration and controlled through a single serial line connection. The following figure shows an example of a host computer communicating to three different devices using this feature.



The serial ports operate at the following settings:

Baudrate	9600
Parity	NO
Stop bits	1

In normal mode, command and response messages are transferred as lines of printable ASCII characters. The only exception is binary data transfer, a special feature described in A.5 and only supported by a few *isgdevices*.
Commands messages sent to an *isgdevice* by a serial port must be formatted as sequences of printable characters terminated by a "*carriage return*" (ASCII 0x0D). Additional control characters, like "*line feed*", are ignored.
Response messages produced by the device consist on lines terminated by a "*carriage return*" + "*line feed*" character sequence (ASCII 0x0D 0x0A).

A.1.2. GPIB interface

Some *isgdevices* include a communication interface compliant with the GPIB (IEEE-488.1) standard. Every *isgdevice* with this type of interface has a default GPIB address. The procedure for changing the address as well as the default value depends on the type of device.

Command messages sent to the device must be always terminated by an EOI bus signal or a terminator character. Both characters "*carriage return*" (ASCII 0x0D) and "*line feed*" (ASCII 0x0A) may be used as message terminators.

Response messages produced by the device when it is addressed as talker consist on lines terminated by a "*carriage return*" + "*line feed*" character sequence (ASCII 0x0D 0x0A) as in the case of serial lines. The EOI bus signal is activated with the last character of every message line.

As well as serial lines and in addition to the normal ASCII based commands, GPIB communication provides some support for binary data transfer. This only applies to special cases as described in A.5.

A.2. Syntax conventions

In the most usual case remote control is implemented by an application program running in a host computer that sends commands and requests to the *isgdevice* as sequences of ASCII characters. The syntax rules are described below. See A.6 for practical examples.

Some devices support binary commands or requests. Binary commands and requests are are also issued as ASCII command lines, but they are followed by a binary data block as described in A.5.

A.2.1. Commands and requests

- Command lines consist of a command keyword optionally followed by parameters.
  - The number and type of parameters depend on the particular command.
  - The way a command line is terminated depends on the type of communication port (see A.1).
- Command keywords are not case sensitive.
  - The device converts internally all the characters to uppercase before any syntax checking.
  - Parameters are also converted to uppercase unless they are enclosed between double quotes (″″, ASCII 0x22).
- Commands may be optionally preceded by the acknowledge character.
  - The acknowledge character is a hash symbol (#, ASCII 0x23) that must appear in the command line immediately before the first character of the command keyword.
- Normal (non query) commands never produce response messages unless the acknowledge character is used.
  - Non query command keywords always start by an alphabetical character (A to Z).
  - If the acknowledge character is used, the device produces the response string OK if the command execution was successful.
  - If the acknowledge character is used and the command does not executes successfully, the device produces either the string ERROR or a string containing a human readable error message. The behaviour depends on the current setting of the *echo* mode (see A.4).
  - Binary commands are special cases where the command keyword includes an extra asterisk symbol (*, ASCII 0x2A) immediately before the first alphabetical character. Binary commands may include any number of ASCII formatted parameters but the command line must be followed by a binary data block as described in A.5.
- Requests are query commands that produce response messages from the device.
  - Requests keywords always start by a question mark character (?, ASCII 0x3F).
  - If the request is successful the content of the response message depends on the particular request.
  - If request fails the device produces either the string ERROR or a string containing and human readable error message. The behaviour depends on the current setting of the *echo* mode (see A.4).
  - The acknowledge character has no effect when used with requests.

- Binary requests are a special case where the request keyword includes an extra asterisk symbol (*, ASCII 0x2A) immediately after the question mark character. In binary requests, instead of an ASCII response message, the device sends to the host a binary data block as described in A.5.

- Response messages consist of one or more ASCII character lines.
  - The way every line in a response message is terminated depends on the type of communication port (see A.1).
  - A response message may contain either the output of a request, an acknowledgement keyword (`OK` or `ERROR`) or a human readable error message.
  - When a response message consists of more than one line, the first and last lines contain a single dollar character (`$`, ASCII 0x3F).

A.2.2. Addressing (serial ports)

- The addressing features allow to dialogue with more than one *isgdevice* connected in a serial line daisychain configuration (see A.1).
  - If the command line does not include any addressing character, the command is accepted and executed by the first device in the communication chain. Binary commands and requests do not accept any addressing characters (see A.5) and can only be sent to the first device in the chain.
  - Addressing does not apply in the case of GPIB communication; commands and queries should not include any extra addressing characters. GPIB addressing is managed according to the GPIB standard (IEEE 488.1) and is not related to the addressing features described in this section.

- Commands and requests may be preceded by any number of skip characters and/or an addressing prefix.

- The skip character is the “greater than” symbol (`>`, ASCII 0x3E) and can be used to address an *isgdevice* placed at a given physical position in the communication chain.
  - When a device founds a skip character as the first character of a command line, it ignores the content of the line and forwards the remaining characters to the next device in the communication chain.
  - If a host computer connected to a chain of *isgdevices*, sends a command line starting with *n* skip characters, the *n* first devices in the chain will ignore the command. The command line with the skip characters removed will be forwarded to the *n*+1 device in the chain.

- An addressing prefix consists of an optional alphanumerical address string followed by a colon character (`:`, ASCII 0x3A).
  - The address string must start by a numeric character (0-9) and consists of any combination of numbers and alphabetical (A-Z) characters.
  - A command line with an address string will be interpreted and executed only by the first device in the communication chain whose internal address matches the one in the address string.
  - Once one of the devices recognises the address string as its own, the command line will not be further forwarded to the remaining devices in the chain.
  - Leading zeros in the address string are ignored for identification purposes but can be used to access *isgdevices* whose internal address begins with a non-numeric character.

- An addressing prefix consisting of only the colon character (`:`) with no address string is interpreted as a broadcast command. In that case the command line is forwarded to all the devices in the communication chain.

A.2.3. Binary data transfer

- Commands or requests that include an asterisk symbol (*, ASCII 0x2A) as the first keyword character are followed by a binary data block.
  - The command line may include any number of ASCII formatted parameters, the binary transfer starts immediately after the end of the command line that must not include any addressing character. The command is accepted and executed by the first device in the communication chain.
  - Addressing does not apply in the case of GPIB communication; commands and queries should not include any extra addressing characters. GPIB addressing is managed according to the GPIB standard (IEEE 488.1) and is not related to the addressing features described in this section.

- Commands and requests may be preceded by any number of skip characters and/or an addressing prefix.

A.3. Common commands

The following commands are implemented in all *isgdevices* regardless of their specific functionality:

Command	Description
ECHO	Activates the *echo* mode. Must be used only in interactive mode with dumb terminals (see A.4).
NOECHO	Cancels the *echo* mode. Echo mode must not be active when the device is controlled by a computer program (see A.4).
?ERR	Returns the error message corresponding to the last command or request issued to the device. Returns the keyword OK if no error happened during the last command.
ADDR <devaddr>	Sets the module address to the string <devaddr>. Any leading zeroes are removed from <devaddr>. It can be formed by any combination of up to 9 alphanumerical characters.
?ADDR	Returns the current address of the device. If no address has been set, returns an empty string.
?CHAIN	Returns the status and type of the secondary communication port. The status indicates if there is another *isgdevice* connected to the secondary port by the strings YES or NO. The port type is one the strings: NONE, RS232, RS422, BOTH
NAME <devname>	Sets the module private name to the string <devname>. <devname> may include any combination of up to 20 printable characters, including whitespaces.
?NAME	Returns the current module private name.
?VER	Returns the type of isgdevice and its firmware version as a string in the form: <type> XX.YY
?HELP	Returns a list of the available commands

A.4. Terminal mode

When an *isgdevice* is accessed through a serial port (primary port), two possible communication modes are available that can be selected with the commands `ECHO` and `NOECHO`. The differences between these two modes are described below. If the primary port is not a serial port (GPIB interface), the device accepts the `ECHO` and `NOECHO` commands but it always responds as if it were set in the noecho mode.

Echo mode (terminal mode)

This mode should be used when an *isgdevice* is connected to a dumb terminal. In this case the user types commands on the keyboard and reads the answers and error messages on the terminal screen without computer intervention. This mode is usually not active by default and the user has to send the `ECHO` command every time the device is powered on.

In echo mode all the characters sent to the device are echoed back to the terminal. The device also sends human-readable messages to be printed on the terminal screen whenever an error is detected in commands or requests.
Case conversion takes place before the characters are sent back to the terminal, therefore characters are echoed back as uppercase even if they are typed and sent to the device as lowercase.
The backspace character (ASCII 0x08) has the effect of deleting the last character received by the device. In this way a minimum editing functionality is provided.

Noecho mode (host computer)

This is usually the default. In this mode no characters are echoed and no error messages are returned unless they are explicitly requested by means of the `?ERR` request. This mode is intended to be used when a program running in a host computer communicates with the controller, sending commands and analysing the answers.

A.5. Binary transfer

Binary transfer is a special mode that extends the standard protocol allowing faster data transfer. Most of the isdevices do not implement binary transfer and therefore this section is not relevant to them.

[to be developed]

A.6. Examples

In the following examples it is assumed that the device is in “noecho” mode.

Commands and requests

#1	Command: Answer:	NOECHO *…no answer…*	Sets the device in “noecho” mode
#2	Command: Answer:	?VER MOCO 01.02	Queries the type and version number of the device.
#3	Command: Answer:	NAME “My Device“ *…no answer…*	Sets the private name of the device. There is no response message.
#4	Command: Answer:	#NAME “My Device“ OK	If the acknowledge character is used, the device produces an “OK” response message

Errors

#5	Command: Answer:	?ERR OK	If no error happened during the last command or request.
#6	Command: Answer:	? VER ERROR	Errors in requests (extra white space in this case) always produce a response message.
#7	Command: Answer:	?ERR Command not recognised.	If an error happened, returns the corresponding error message.
#8	Command: Answer:	NAME *…no answer…*	Errors in commands (missing parameter in this case) do not produce response messages.
#9	Command: Answer:	#NAME ERROR	If the acknowledge character is used, the device produces a response message.
#10	Command: Answer:	?ERR Wrong Number of Parameter(s).	If an error happened, returns the corresponding error message.

Addressing (assumes daisychaining of at least three *isgdevices*)

#11	Command: Answer:	:NOECHO *…no answer…*	Sends the NOECHO command to all the devices in the chain (broadcast).
#12	Command: Answer:	?ADDR 12	Queries the address of the first device in the communication chain.
#13	Command: Answer:	>>?ADDR LFT3	Queries the address of the third device in the chain (skips the first two devices in the chain).
#14	Command: Answer:	12:?VER MOCO 01.02	Queries the version of the device with address “12”.
#15	Command: Answer:	0LFT3:?VER OPIOM 01.00	Queries the version of the device with address “LFT3”.
#16	Command: Answer:	>>?VER OPIOM 01.00	Queries the version of the third device in the chain.