Digital Current Source DCS-6K

Software Manual

Man-DCS6K-S-011
ISSUE 1

13th May 2019

STL Systems AG

Hauptstrasse 137E

8274 Taegerwilen

Schweiz

TABLE OF CONTENTS

1. INTRODUCTION

1.1. OVERVIEW

The digital current source DCS-6k is a high performance system with following highlights

- Digital communication by USB or TCP/IP
- Extreme high accuracy
- Large intrinsic bandwidth and slewrate of 1 kHz
- Direct analogue and digital controller
- Programmable with user defined waveforms
- Accuracy-controlled sequence running
- Feedback with internal shunt or external sensor
- Automatic load control and controller PID settings adaption
- Extendable to a multi-channel system

The current source is available in 2 types:

- Power limit of 50 Ampere and 120 Volt, 3 phase 360V / 50 Hz
- Power limit of 20 Ampere and 100 Volt, 1 phase 240V / 50 Hz

General system information is obtained with the command *IDN? This manual is applicable to the current source of type 530B and 531B.

1.2. Functionality

Fig. 1 shows a functional overview of the complete DCS-6k digital current source. The system generates currents with the highest level of accuracy. The operation principle is as follows and shown in Fig. 1 .

1) **Input**: The nominal value can be applied as analogue or digital value (“Analogue in” / “Digital in” on the left in Fig. 1), where the analogue value can be re-normalised with an offset and scale to fit the physical entity, see §4.2. The nominal value can be applied in any arbitrary units (a.u.) see also step 4).
2) **Sequence**: The digital input can be set to any direct input value, or by a pre-defined sequence in the current source memory.
3) **Operation Mode**: The input source (“analogue” or “digital”) is set by the internal variable “Operation mode”
4) **Field To Current**: The nominal value in arbitrary units is converted to current by a slope / offset conversion.
5) **PID types**: The PID controller coefficients are set to any of the 4 pre-configured coefficient sets.
6) **Controller mode**: the controller reference can be internal (high accuracy shunt) or is externally formed by a sensor (for example a magnetic field sensor in a Coil which is driven by the current source). This functionality is only available in digital operation mode.
7) **Reference Sensor**: The Reference Sensor Parameters are used to convert the analogue input signal to the arbitrary units in use by a slope / offset conversion.
8) **Output state**: The output is set off or active with the output state.

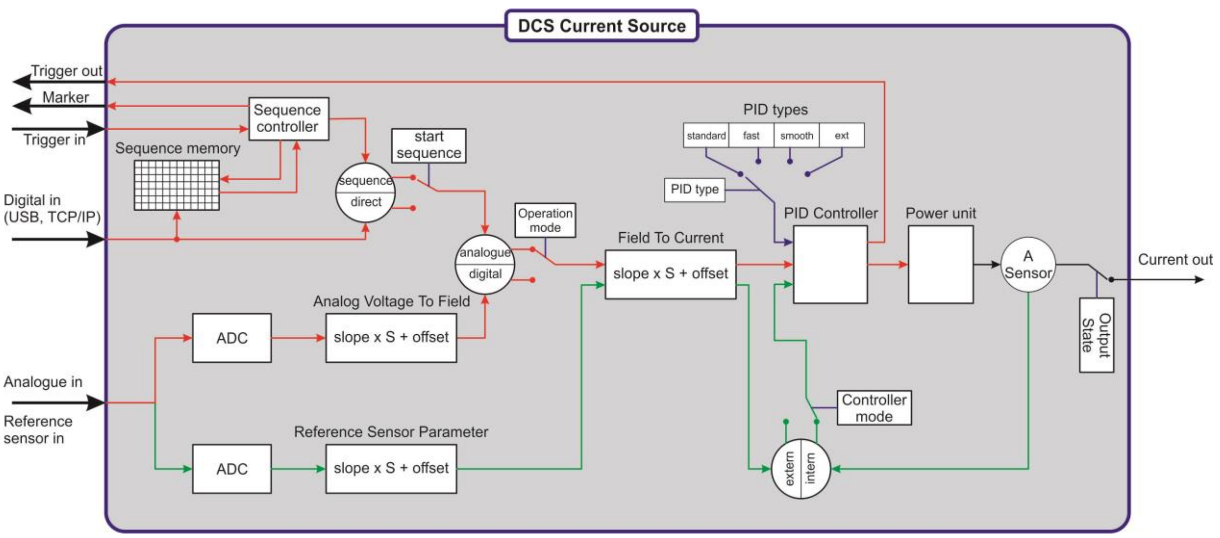

Fig. 1. Functional overview of the DCS-6k current source.

2. Software Installation

2.1. Hardware and Software Requirements

The DCS-6K Application Example and system driver can be installed on computers with at least 8 GB of memory running Microsoft Windows 7 or Windows 10. At least 500 MB of free disk space are required for installation and at least 100 MB for the work with the software.

The communication between the DCS-6k and PC/Laptop is based on USB 2.0 or TCP/IP-Interface.

2.2. Driver Installation for USB

When the USB is connected to the PC, a COM interface is automatically established. If this does not appear, please install the latest FTDI driver from the FTDI website:

https://www.ftdichip.com/Drivers/VCP.htm

2.3. Driver Installation for TCP/IP

The default settings for the TCP/IP communication are listed in Table 1. The settings can be changed with the command: `SetEthernetConfig(IPaddress,submask,gateway).`

Table 1. Default settings for the TCP/IP

Command	Value
IPaddress	192.168.1.1
submask	255.255.0.0
gateway	127.0.0.1

To verify the communication between the DCS-6k and the computer proceed as follows:

Start up "CMD.exe", a DOS-box starts up

in the DOS-box type in: "ping 192.168.1.1" and "Enter"

If the connection is OK, you will see the following result shown in Fig. 2.

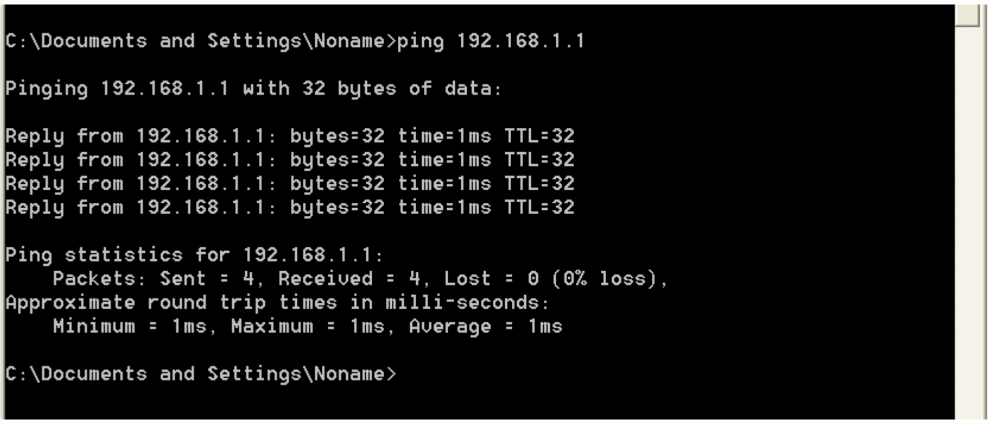

Fig. 2. The result of verifying the TCP/IP communication.

In the case that the software cannot communicate with the current source, please check the firewall settings.

2.4. Application Example Software Installation

The software is installed automatically from CD. If for any reason the automatic start of the CD fails, please select the CD-ROM from the file manager and run the file "setup.exe" in the root directory.

3. Application Example

3.1. Interface Definitions

- USB : Interface to current source
- TCP/IP : Interface to current source

3.2. User Interface

All commands which are available are described in the section §4.13, the application example shows the various commands including the parameter settings and the return values.

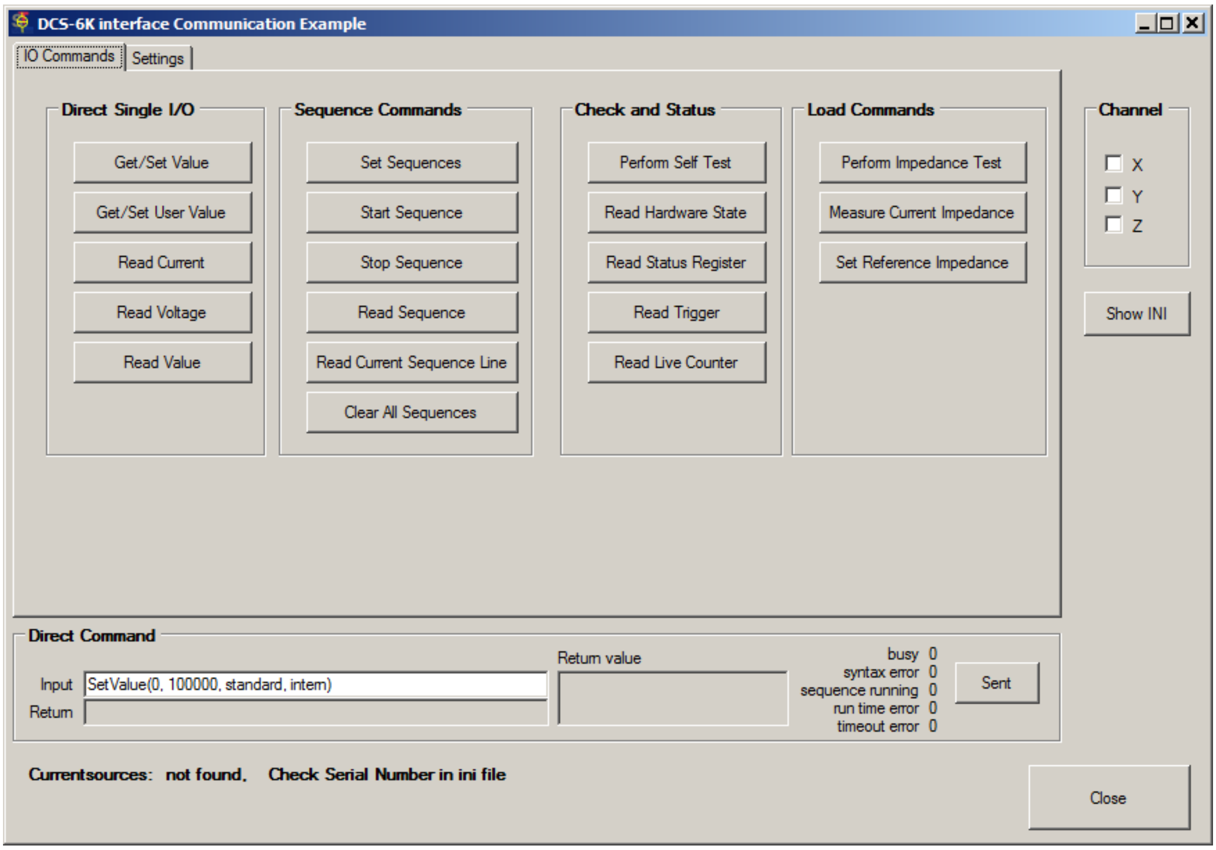

Fig. 3. Overview of the user interface showing the user commands to run the current source.

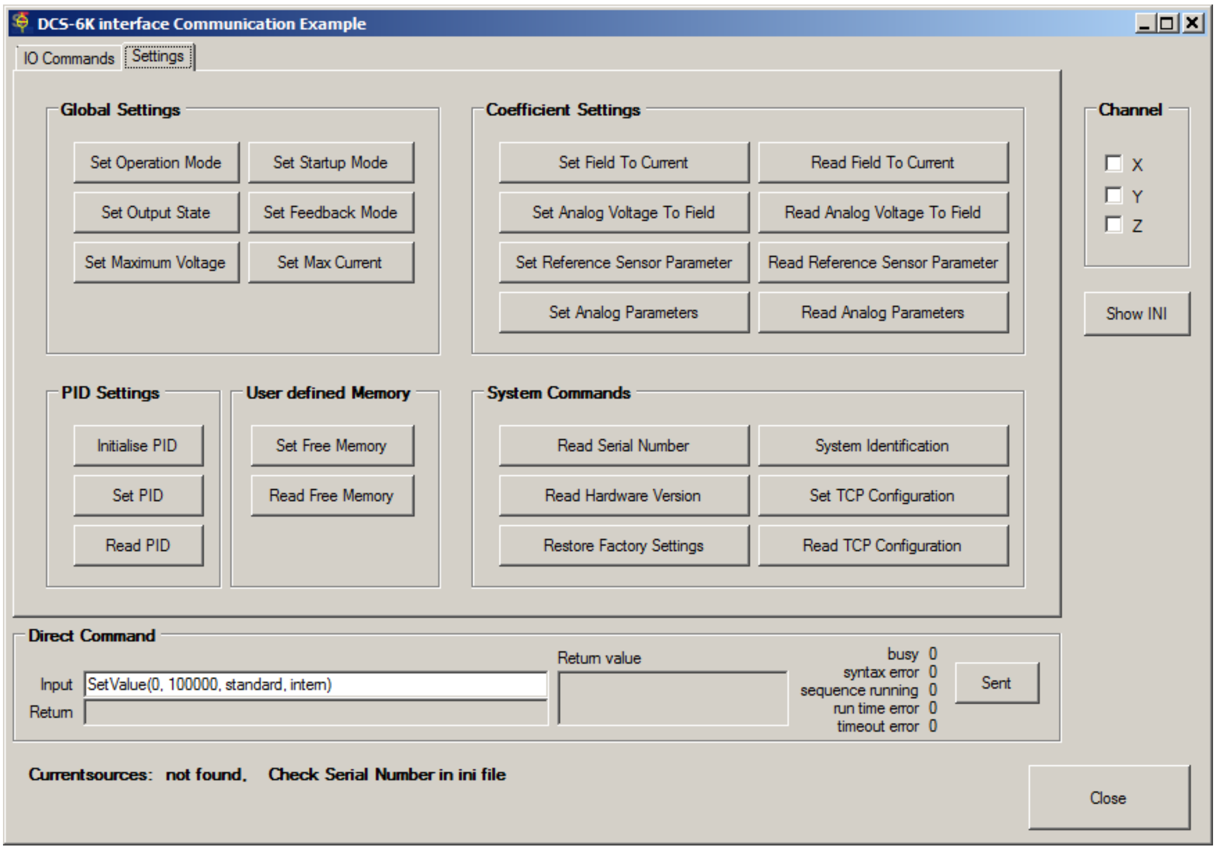

Fig. 4. Overview of the user interface showing the user commands to program the current source.

4. DLL DESCRIPTION

4.1. INTRODUCTION

The current source DCS-6k allows a digital interface, direct current programming and waveform programming. The interface is either USB or TCP/IP.

The current source allows for various applications and can be programmed to be used for many tasks. In the following sections different aspects of the current source are described:

In §4.2 the interface initialisation is described.

In §4.3 the system accuracy settings and functions are described.

The nominal input values can be in units of Ampere or in any units suitable for the application. The unit conversion options are described in §4.4.

In §4.5 the use of an external reference sensor is described.

During communication with the current source, each command is replied by an echo and a 5 bit command status, followed by the return value if applicable. The description of the command handling and the command status is described in §4.6

In §4.7 the error handling is described.

In the sections 4.8, 4.9 and 4.10 some examples of typical applications are given, followed by the complete list of commands and the description for each of the commands.

4.2. Interface Initialisation

4.2.1. USB Interface

The COM-Port settings are listed in Table 2.

Table 2. Default TCP/IP settings.

Command	Value
Baudrate	3000000
Databits	8
Parity	None
Stopbits	1

4.2.2. TCP/IP Initialisation

The default TCP/IP settings are listed in Table 3. The TCP/IP settings can be changed with the command "`SetEthernetConfig`" as shown in Table 4.

The TCP/IP interface takes about 5 seconds to initialise. If the interface is not in use, the TCP/IP interface can be disabled by setting the IP address to 000.000.0.0.

Table 3. Default TCP/IP settings.

Command	Value
IPaddress	192.168.1.1
submask	255.255.0.0
gateway	127.0.0.1

Table 4. TCP/IP Interface settings.

Command	Description
`SetEthernetConfig(IPaddress,submask,gateway)`	Set the IPAdress, submask and gateway.

4.3. Output Current Accuracy

The current output is controlled by a PID controller which minimises the difference between the nominal and actual current value. The trigger output signal reflects the accuracy state of the output current.

A typical output current after nominal value change is shown in Fig. 5. First the output current is outside the tolerance range and the output trigger = 0. After a remain in the tolerance range for a period of “tolerance time”, the output trigger = 1, indicating that the accuracy is reached.

The tolerance time is set for analogue application with the command `SetAnalogParameters`. In the digital mode the tolerance time is defined for sequencing with the command `SetSequenceLine`.

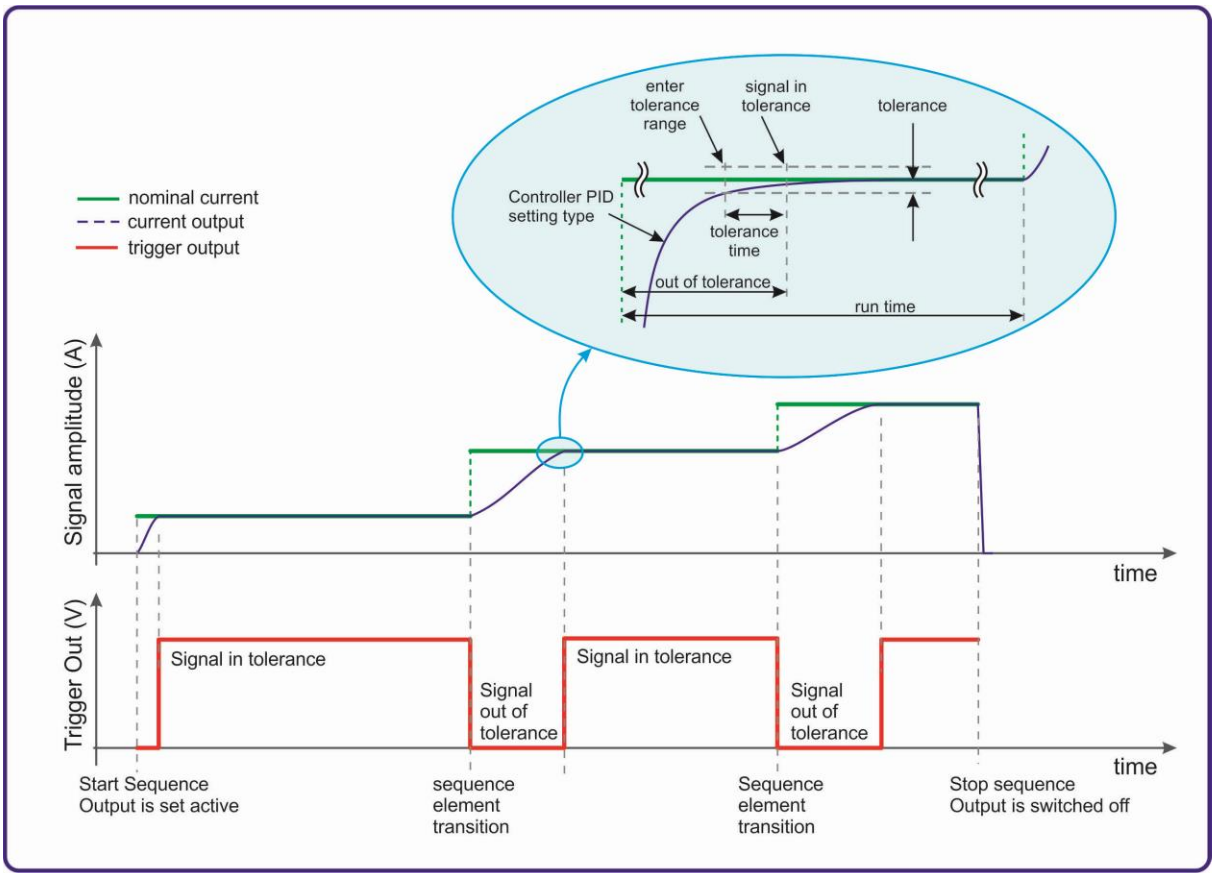

Fig. 5. Accuracy definition at the trigger output.

4.4. Value Scaling

4.4.1. Field To Current

The digital nominal value definition can be performed in units of ampere, or in any other unit. This can be useful if the current source is part of an application, for example in a coil system in which a magnetic field is generated with the current in the loop.

In this case the nominal magnetic field value can be entered as input in the current source. With the command `SetFieldToCurrent` the slope and offset conversion from magnetic field to current is programmed where the slope corresponds with the inverse coil coefficient (in e.g. ampere/tesla). With this conversion, the entered value for the field is interpreted such that the correct current is sent to the coil to generate the requested field value.

4.4.2. Analogue Voltage to Field

The analogue nominal value input range is ±10 Volt. From the factory settings, this range is mapped to the maximum current range of the current source (20 or 50 A). The `SetAnalogVoltageToField` command programmes the slope and offset which allows the conversion of the input voltage to any other unit. This can be useful if the current source is part of an application, e.g. in a coil system in which a magnetic field is generated with the current.

After the analogue input voltage unit is defined, the command `SetFieldToCurrent` must be used to fit the alternative units.

4.4.3. Reference sensor parameter

The external reference sensor can be used alternatively to the internal reference shunt to control the output current.

The analogue external sensor can be used if the current source is part of an application, for example in a coil system in which a magnetic field is generated with the current. In this case, the external sensor is a magnetic field sensor, which feedback the magnetic field to the current source. The reference sensor parameter is used to convert the sensor output (in voltage) to the magnetic field.

4.5. Reference Sensor

In Fig. 1 and Fig. 5 the principle of the internal and external feedback sensors is schematically shown. The feedback sensor is used as input to the PID controller to set the output current. The internal feedback sensor consists of a current sensor at the current source output. This sensor is default used by the PID controller.

In the case that the application uses another physical entity which results from the current (for example the magnetic field from a coil, generated by the output current through the coil loop) an external sensor (for this example a magnetic field sensor) can be used as feedback sensor.

The external sensor only can be used in the digital mode. With the command `SetReferenceSensorParameters` the sensitivity and bias of the external sensor are defined. The external sensor can be activated with the commands `SetUserValue` and `StartSequenc` with the parameter `<contr>`.

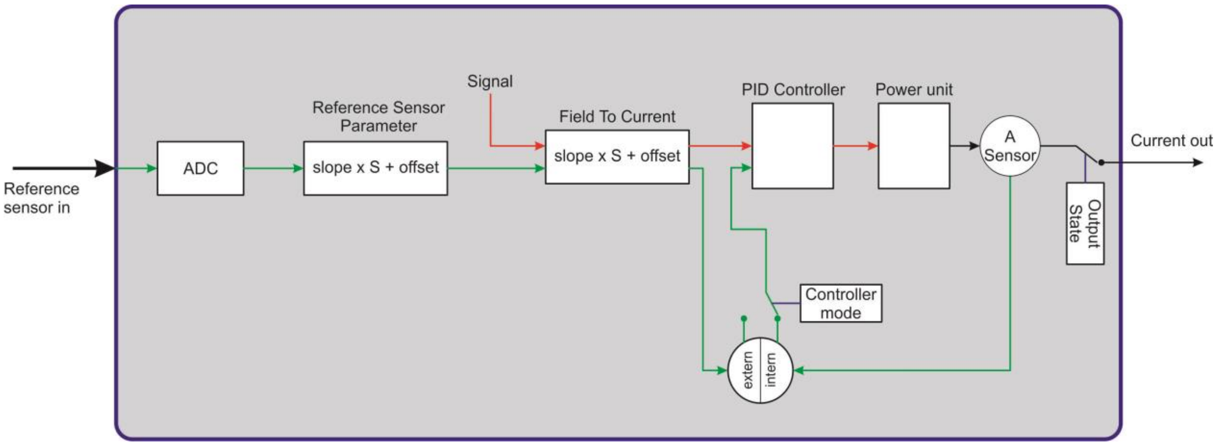

Fig. 6. Internal and external feedback sensor.

4.6. Command Status

The current source can be used in an analogue or digital mode. The factory setting is digital mode. In this mode the current source starts up with the output disabled.

The current source operation mode, nominal value input or any other action is performed by sending a command over USB or TCP/IP. After the command is set, a feedback is given in the form

Command name xxxx,return values

with xxxxx the status. For example, if the current source output is set to active, the command

SetOutputState(active)

is sent to the current source. After the command is processed, following will be returned:

SetOutputState 00000

The output 00000 is called the status. Each of the 5 entries is a status bit, which is explained in Fig. 7

Bit nr	Value	Meaning
1	0	System is ready to accept a command
	1	System is executing a command
2	0	Command is error free
	1	Command shows a syntax error or parameter is out of range
3	0	System is ready to execute a command
	1	System is running an automatic sequence
4	0	Command execution is error free
	1	Command shows an execution error
5	0	Command execution is error free
	1	Command shows a time out error

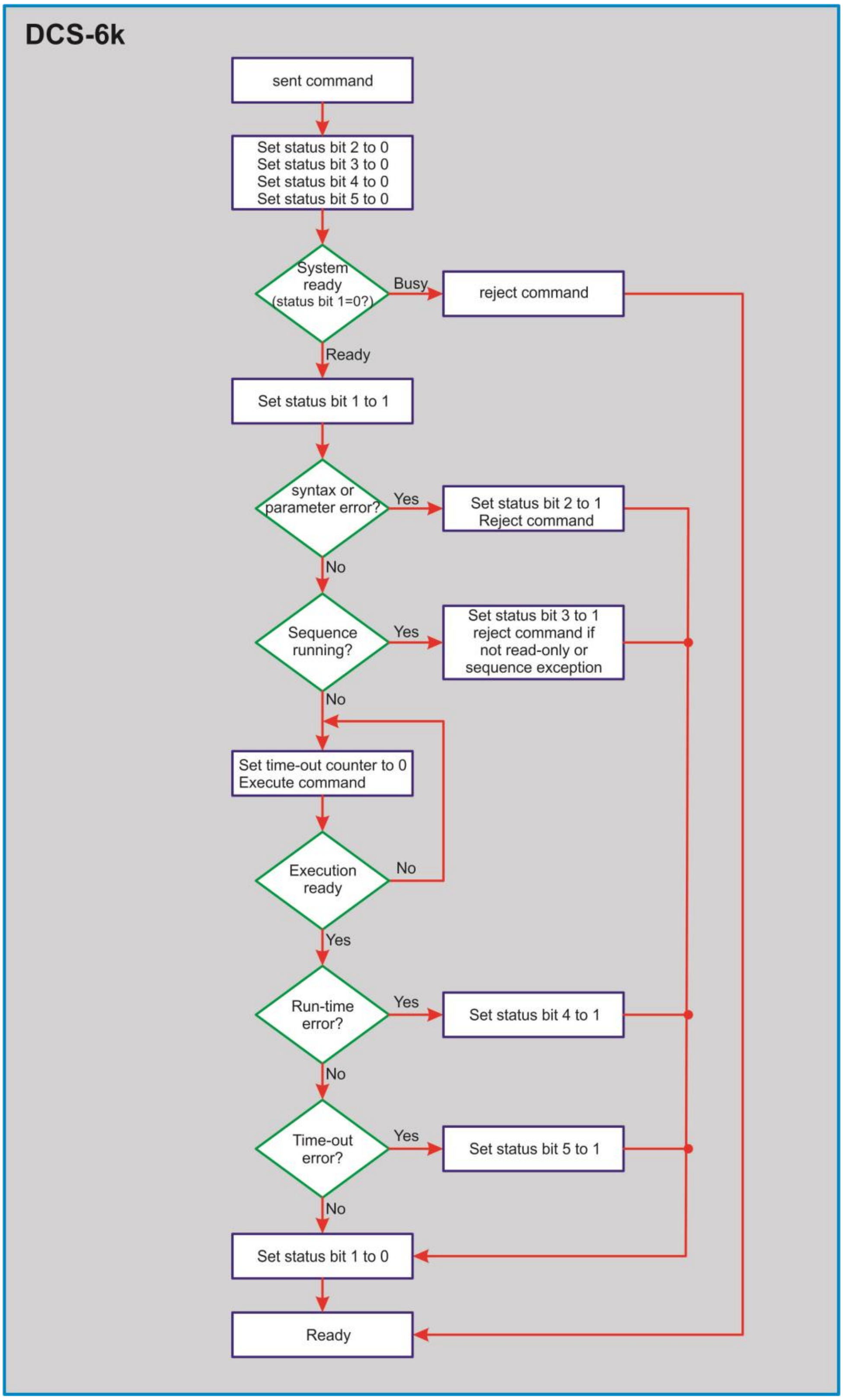

Fig. 7. Functional overview of the command processing and status register definition.

4.7. Error Handling

During the current source hardware start-up, the hardware is tested, and the attached load impedance is measured. In order to make sure that the controller settings are adapted to the load, the measured impedance is compared with the stored impedance values. If the hardware shows no error, and the stored impedance values equals the measured impedance values, the front panel On/Off LED turns green, and the internal hardware state, as read with the command `ReadHardwareState` returns the value 0.

The current source distinguishes between three types of error:

- Hardware error
- Load error
- Accuracy error

In the following sections the meaning of these errors is described.

4.7.1. Hardware Error

If the hardware error occurs, the electronics system cannot start up correctly. When this error occurs, the front panel On/Off LED turns red, and the internal hardware state, as read with the command `ReadHardwareState` returns the value 2. The error can be caused by a hardware defect, an over temperature, a power supply error or a leakage current.

With the command `PerformSelfTest` the hardware start-up including al tests is repeated.

4.7.2. Load error

During current source start up, automatically the connected load impendence is measured. The load error indicates that the measured load impedance differs from the internal stored load impedance values. This may indicate that the stored controller PID settings must be adapted to the connected load to avoid uncontrolled oscillations which causes malfunctioning and major current accuracy errors. When this error occurs, the front panel On/Off LED turns orange, and the internal hardware state, as read with the command `ReadHardwareState` returns the value 1. This result is equal to the result of the command `PerformImpedanceTest` which tests the difference between the stored and measured load impedance values.

When the controller parameters have to be adapted to a new load, the commands `InitializPID` and `SetPID` are available. The command `InitializPID` first measures the impedance of the load, stores the values in the impedance reference, and adapts the PID settings to default values. With these initial settings, the current source can be run with the load and the On/Off LED turns green. The command `SetPID` can be used to optimise the controller setting.

With the command `MeasureCurrentImpedance` the actual load impedance is measured, and with the command `SetReferenceImpedance` the reference impedance values is programmed. If the reference impedance is set to the measured impedance, the command `PerformImpedanceTest()` will returns a 0 (OK) after the adaption is made. However, make sure that the controller parameters are set such that it fits the load.

4.7.3. Accuracy error

The accuracy error indicates that the output current has not reached the tolerance definition. When this error occurs, the front panel On/Off LED stays green, but the front panel Current Error LED turns red during the tolerance deviation time. The analogue output trigger signal corresponds to the Current Error LED signal.

If the accuracy error occurs frequently or continuously, switch off the current source output and inspect the controller behaviour and adapt the controller PID settings when necessary as this can indicate that the controller oscillates.

4.8. Example: System Initialisation

The current source can be used in an analogue or digital mode. The factory setting is digital mode. In this mode the current source starts up with the output disabled.

4.8.1. The Analogue Mode

If the current source is to be used in the analogue mode, the commands as listed in Table 5 must be executed to change the operation mode to analogue; to make sure that the current source will start up in the analogue mode and to set the analogue default parameters.

Table 5. Operation mode commands.

Mode Command	Description
SetOperationMode(analog)	Run the current source in analogue mode
SetStartupMode(analog)	Start up the current source in analogue mode
SetAnalogParameters(1,standard,1,0.05)	

The commands as listed in Table 6 are used to adapt the controller parameters (P, I and Slewrate) to the attached load. With the command "InitializePID" the impedance of the attached load is measured, and the P, I and Slewrate parameters are adapt to make sure that the controller will run stable. The "SetPID" command allows optimising the controller parameters manually. With the "SetOutputState", the output is set active.

Table 6. Load adaption commands

Load Command	Description
InitializePID()	Obtain a default PID setting for the attached load. The result will be stored in PID type "standard"
SetPID(standard,P,I,Slewrate)	With this command the controller PID parameters can be optimized. P=Proportional factor, I=integral factor
SetOutputState(active)	Activate the output

The commands as listed in Table 7 are used to adapt the controller to the application. For example if a coil system is attached as load, a magnetic field is generated. The listed commands enable the input to be interpreted in units of the generated magnetic field, for example Gauss or mT. With the command "SetAnalogVoltageToField" the conversion of the input value (volt) to the requested units is performed. With the command "SetFieldToCurrent" the conversion of the field value (for example Gauss or mT) to the requested current is performed.

Table 7. Application adaption commands

Application Command	Description
SetAnalogVoltageToField(1,0)	Default, the input voltage range of ⟨-10, 10⟩ volt covers the output current range ⟨-20, 20⟩ or ⟨-50, 50⟩*⁾
SetFieldToCurrent(slope,offset)	Conversion of input data units to current (Ampere)

*⁾ Dependent on hardware type

4.8.2. The Digital Mode

In the factory setting, the current source starts up in the digital mode, with the output disabled. If the operation mode is analogue, the digital mode is set according the commands as listed in Table 8.

Table 8. Operation mode commands.

Mode Command	Description
SetOperationMode(digital)	Run the current source in digital mode.
SetStartupMode(digital)	Start up the current source in digital mode.

The commands as listed in Table 9 are used to adapt the controller parameters (P, I and Slewrate) to the attached load. With the command "InitializePID" the impedance of the attached load is measured, and the P, I and Slewrate parameters are adapt to make sure that the controller will run stable. The "SetPID" command allows optimising the controller parameters manually. With the "SetOutputState", the output is set active.

Table 9. Load adaption commands

Load Command	Description
InitializePID()	Obtain a default PID setting for the attached load. The result will be stored in PID type "standard"
SetPID(standard,P,I,Slewrate)	With this command the controller PID parameters can be optimized. P=Proportional factor, I=integral factor
SetOutputState(active)	Activate the output

The commands as listed in Table 10 are used to adapt the controller to the application. For example if a coil system is attached as load, a magnetic field is generated. The listed commands enable the input to be interpreted in units of the generated magnetic field, for example Gauss or mT. With the command "SetFieldToCurrent" the conversion of the field value (for example Gauss or mT) to the requested current is performed.

Table 10. Application adaption commands

Application Command	Description
SetFieldToCurrent(slope,offset)	Conversion of input data units to current (Ampere)

4.9. Example: Apply a Current

In the digital mode, a current can be applied when the system is initialised according to §4.8.2.

After the system is initialised, and the controller is adapted to the load, the output is enabled.

Table 11. Apply a Current

Application Command	Description
`SetOutputState(active)`	Enable the output
`SetValue(value)`	Set a current

4.10. Example: Apply a Sequence

4.10.1. Introduction

A sequence of values can be programmed such that these values are applied to the current source output with a constant time interval, the so-called random waveform generator. In this case the criterion for the transition from one sequence element to the next is controlled by a constant time. There are also other criteria for the transition. In this current source following criteria are implemented:

Pre-defined sequence time	: after a defined run time the transition to the next sequence takes place.
External trigger	: when the accuracy is reached and the external trigger is high the transition to the next sequence takes place.

The sequence progress procedure with external trigger is shown Fig. 8. At sequence start, the output state will turn from “off” to “active” and the first sequence element is activated. When the next element is started, the next sequence element is read from memory and activated. The transition to the next element is initiated either by time or by the input trigger, see § 4.10.4.

The duration of the complete sequence procedure is determined by the number of sequence elements, defined by the number of addresses selected, the duration of the sequence element transitions, the duration of each sequence element and the number of repetitions.

During the sequence is running, the third bit of the status is 1, and only a limited number of commands are accepted:

- Commands which only read from memory (for example read any coefficient)
- “StopSequence”, which stops the sequence.
- “StartSequence”, which will re-start the sequence from the start memory position.
- “SetOutputState”. If set to “off”, the sequence will be stopped.

The complete sequence procedure is ready when

- The last sequence is ready or
- The command “StopSequence” is entered or
- The command “SetOutputState(off)” is entered or
- The time-out time is elapsed or
- A run time error occurs

After the sequence procedure is ready, the output state will turn from “active” to “off”. The status register can be read with “ReadStatus” showing the status of the sequence exit:

- Status = 00000: The sequence is run successful
- Status = 00010: The sequence is stopped with a run time error
- Status = 00001: The sequence is stopped with a time-out error

Please note that after entering any other command than “ReadStatus”, the status register shown the status of the last entered command.

In the next sections the sequence programming, the progress type and the start / stop commands are described.

4.10.2. PROGRAM THE SEQUENCES

In Table 12 the all commands related to the sequence programming are listed. In order to clear the sequence memory, all sequence elements are deleted with the command `ClearSequences`.

The sequences are programmed to a memory address `<address>` with the command `SetSequenceLine` under definition of the signal amplitude `amp`, the tolerance `tolerance`, the tolerance time `toltime`, the PID controller definition `<type>` and activation / de-activation of the analogue output marker `<marker>`. The tolerance `tolerance` the tolerance time `toltime` and the output marker `<marker>` parameter are explained in the section §4.10.4. With the start command, further parameters are defined like the number of periods etc.

The programmed values are read from memory with the command `ReadSequenceLine`.

Table 12. Sequence programming commands

Sequence programming commands	Remark
`ClearSequences()`	
`SetSequenceLine(<address>,amp,tolerance,toltime,<type>,<marker>)`	
`ReadSequenceLine(<address>)`	

4.10.3. START / STOP THE SEQUENCE

After the sequence is programmed, the sequence can be started with the start command `StartSequence` containing the parameters: start address `<start>`, end address `<stop>`, number of repetitions `<rep>`, transition type `<proc>`, sequence time `runtime` , transition control type `<contr>` and maximum time before time-out error `time-out`.

The start and the stop address define the beginning and end of a sequence period. The sequence time defines the time per sequence element.

The transition type `<proc>` and the transition control type `<contr>` are explained in the next section § 0

Table 13. Start sequence command

Load Command	Remark
`StartSequence(<start>,<stop>,<rep>,<proc>,runtime,<contr>,time-out)`	
`StopSequence()`	

4.10.4. SEQUENCE PROGRESS

The transition type `<proc>` (with values “time” or “trigger”) defines whether the transition is caused by the runtime or by a trigger. If the transition is controlled by time, the parameters tolerance `tolerance`, and tolerance time `toltime` are meaningless.

If the transition is caused by a trigger, the trigger must be applied externally.

In §4.3 the meaning of the parameters tolerance `tolerance`, tolerance time `toltime` are explained. The output marker is on or off by the parameter `<marker>` setting to 0 or 1.

The different possible sequence progress procedures described below.

Time controlled sequence progress

The parameter `<proc>` is set to “time”. The sequence time is independent on any further parameter but the `runtime`. After the sequence transition has taken place, the sequence is kept for a duration of runtime, after which the next sequence is activated.

External trigger controlled sequence progress

The parameter `<proc>` is set to “trigger”: After the sequence is started, the output signal and the nominal value will deviate more than the tolerance, and the error signal will turn from 1 to 0. The error signal is on the analogue output signal “trigger output”. The PID controller will cause the current to enter the tolerance range. When the current is in the tolerance range for a duration of `toltime`, the external analogue signal “trigger output” will turn from 0 to 1. The next element of the sequence will start automatically when the trigger output signal is high and the trigger input signal is high.

This “trigger output” signal can be used to start an external sequence element timer to achieve a defined signal application time.

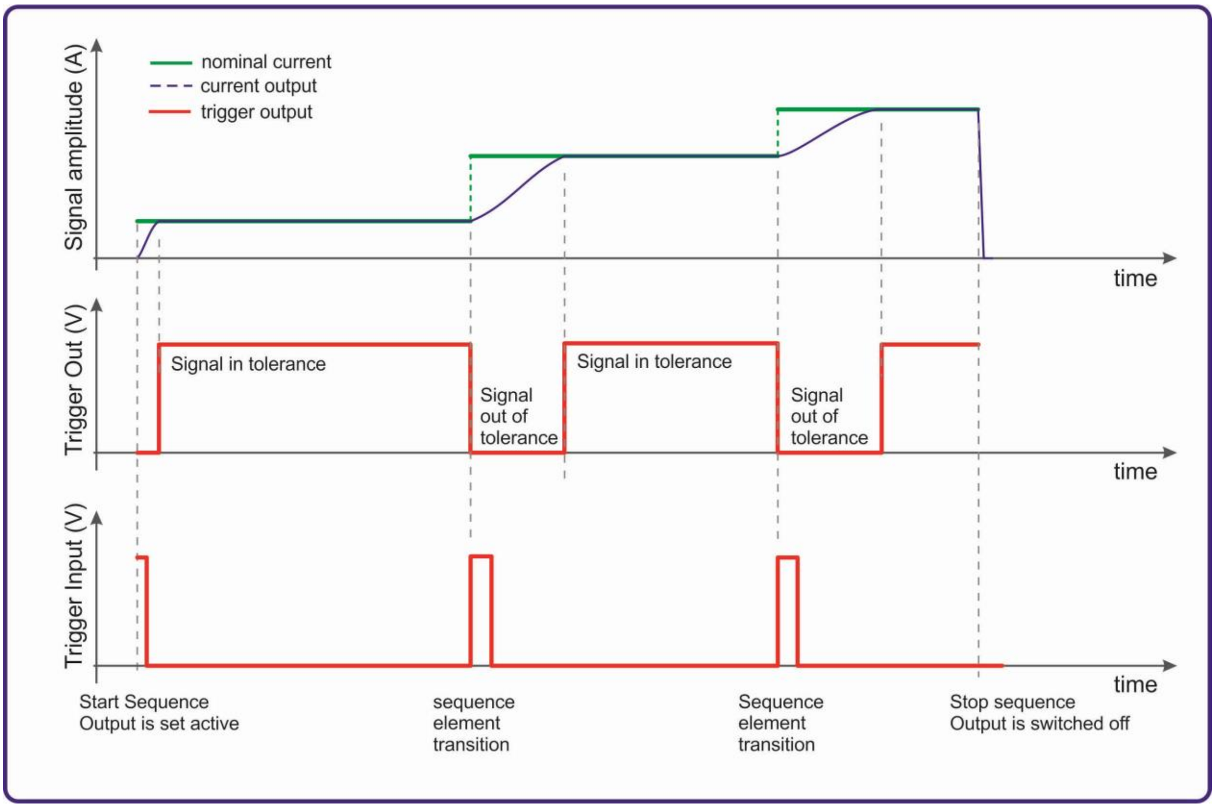

Fig. 8. Sequence progress with external trigger.

4.11. List of Available Commands

Command	Description
*IDN?	Return the global Identification of the hardware.
ClearSequences	Clears the full sequence memory.
InitializePID	Measures the resistance (R) and inductance (L) of the attached load, store the R and L values as reference values, calculates valid PID parameter and store the PID values as standard. If "all" is given as parameter, all locations will be overwritten.
MeasureCurrentImpedance	Measures the resistance and inductance of the attached load.
PerformImpedanceTest	Compares the result from MeasureCurrentImpedance and SetReferenceImpedance according to: (R_{ref} - deltaR_{ref} ≤ R ≤ R_{ref} + deltaR_{ref}) && (L_{ref} - deltaL_{ref} ≤ L ≤ L_{ref} + deltaL_{ref}).
PerformSelfTest	Performs an internal functionality test and output the result.
ReadAnalogParameters	Parameter setting for the analogue modus. If the nominal value is reached within the tolerance over a defined period of time, an analogue output trigger can be set.
ReadAnalogVoltageToField	In analogue mode, an analogue input voltage sets the nominal value for the output (in a.u.) by: OutputCurrent (a.u.) = slope x InputVoltage + offset.
ReadCurrent	Reads the actual current output.
ReadCurrentSequenceLine	Output the actual sequence parameters during a sequence run.
ReadEthernetConfig()	Reads the IP, submask and gateway of the Ethernet interface.
ReadFeedbackMode	Reads the format of the returns.
ReadFieldToCurrent	Reads the conversion factors "slope" and "offset" from Ampere to other units (a.u.).
ReadFirmware	Reads the firmware and software versions in the current source.
ReadFreeMemory	Read a byte from an address in free memory.
ReadHardwareState	Read the status of the current source.
ReadMaxCurrent	Reads the maximum output current.
ReadMaxVoltage	Reads the maximum output voltage.
ReadOperationMode	Read the current source operation mode: digital or analogue.
ReadOutputState	The current source output can be set active or can be

	switched off.
ReadPID	Reads PID values from one of the four PID types
ReadReferenceImpedance	Reads impedance and maximum deviations values for the inspection of an attached load.
ReadReferenceSensorParameters	An external sensor can be used as feedback signal for the current source output in (a.u.) by: SensorOutput (V) = sens x CurrentOutput (a.u.) + offset.
ReadSequenceLine	The current source can be operated in a digital or analogue mode. The operation mode at start up can be set.
ReadSerialNumber	Read the current source serial number.
ReadStartupMode	The current source can be operated in a digital or analogue mode. The operation mode at start up can be set.
ReadStatus	Read the 5 bit status register of the current source command.
ReadTimer	Read the elapsed time since system start.
ReadTrigger	Read the status of the trigger output, trigger input and the analogue marker.
ReadUserValue	Reads the actual value at the current output in a.u.
ReadValue	Reads the actual value at the current output in ampere.
ReadVoltage	Reads the actual voltage at the current output.
RestoreFactoryDefaults	Reset all setting to the factory setting. All customer settings and information will be lost.
SetAnalogParameters	Parameter setting for the analogue modus. If the nominal value is reached within the tolerance over a defined period of time, an analogue output trigger can be set.
SetAnalogVoltageToField	In analogue mode, an analogue input voltage sets the nominal value for the output (in a.u.) by: OutputCurrent (a.u.) = slope x InputVoltage + offset.
SetEthernetConfig()	Set the IP, submask and gateway of the Ethernet interface.
SetFeedbackMode	The echo-output+status as first element of the return value is activated or de-activated.
SetFreeMemor	Write any byte to free memory.
SetFieldToCurrent	Sets the conversion factor “slope” and “offset” from Ampere to other units (a.u.).
SetMaxCurrent	Sets the maximum output current.
SetMaxVoltage	Sets the maximum output voltage.
SetOperationMode	The current source is operated in a digital or analogue mode.
SetOutputState	The current source output can be set active or can be switched off.

SetPID	Stores PID values in one of the four PID type.
SetReferenceImpedance	Set the impedance and maximum impedance deviations values for the inspection of an attached load.
SetReferenceSensorParameters	An external sensor can be used as feedback signal for the current source output in (a.u.) by: SensorOutput (V) = sens × CurrentOutput (a.u.) + offset.
SetSequenceLine	Program a sequence of values to an internal address. The nominal output value is applied for “runtime” seconds as defined by **StartSequence**.
SetStartupMode	The current source can be operated in a digital or analogue mode. The operation mode at start up can be set.
SetUserValue	Set a field (a.u.) at the current source output. The output state is set to “active”.
SetValue	Set a current at the current source output. The output state is set to “active”. The PID setting “standard is used, the Slewrate is set to maximum.
StartSequence	Starts or restarts the sequence of programmed lines from address <start> to address <stop>. The progress mode is defined by <proc> (time or trigger).
StopSequence	Stops a running sequence process and switch the output state to “off”.

4.12. DEPENDENCIES

The dll uses following additional STL dll's:

- BasicMath.dll Version >= 1.0.1.0 or larger
- STL.Calc.dll Version >= 1.0.6.2 or larger
- STL.GeneralProcs.dll Version >= 1.1.5.15 or larger

4.13. DLL INTERFACE

Namespace : STL.DCS_6K

Class : StlCurrentSource

Version : "2.0.8.0"

Members in alphabetic order:

***IDN?**

Only available in the TCP/IP interface mode. Return the global Identification of the hardware

Command	*IDN?
Return value	STL DCS-6K <type> Response time: 0s
	<type> = Hardware type (first 4 alphanumerical elements of the serial number)

ClearSequences

Clears the full sequence memory.

Command	ClearSequences()
Return value	ClearSequences xxxxx Response time: 7s

InitializePID

Measures the resistance (R) and inductance (L) of the attached load, store the R and L values as reference values, calculates valid PID parameter and store the PID values as standard

Command	InitializePID()
Return value	InitializePID xxxxx Response time: ≤ 2s

InitializePID			
Measures the resistance (R) and inductance (L) of the attached load, store the R and L values as reference values, calculates valid PID parameter and store the PID values in the indicated location. If “all” is given as parameter, all locations will be overwritten.			
Command	InitializePID(<type>)		
		list	default
	<type>	standard,fast,slow,ext,all	standard
Return value	InitializePID xxxxx		
	Response time: ≤ 2s		

MeasureCurrentImpedance			
Measures the resistance and inductance of the attached load			
Command	MeasureCurrentImpedance()		
Return value	MeasureCurrentImpedance xxxxx,R,L		
	Response time: ≤ 10s	minimum	maximum
	R = resistive load (Ohm)	0	1000
	L = inductive load (Hy)	0	1000

PerformImpedanceTest		
Compares the result from MeasureCurrentImpedance and SetReferenceImpedance according to: $(R_{ref} - deltaR_{ref} \leq R \leq R_{ref} + deltaR_{ref})$ && $(L_{ref} - deltaL_{ref} \leq L \leq L_{ref} + deltaL_{ref})$		
Command	PerformImpedanceTest()	
Return value	PerformImpedanceTest xxxxx,compareResult	
	Response time: ≤ 10s	list
	compareResult Meaning: 0 : Comparison is true 1 : Comparison is false	0, 1

PerformSelfTest

Performs an internal functionality test and output the result

Command	PerformSelfTest()	
Return value	PerformSelfTest xxxxx,TestResult	
	Response time: 5s	list
	TestResult Meaning: 0 : OK 1 : Load Impedance error 2 : Current source hardware error	0,1,2

ReadAnalogParameters

Parameter setting for the analogue modus. If the nominal value is reached within the tolerance over a defined period of time, an analogue output trigger can be set.

Command	ReadAnalogParameters()		
Return value	ReadAnalogParameters xxxxx,tolerance,<type>,trigger,toltime		
	Response time: 0s	minimum	maximum
	tolerance = maximum signal deviation (a.u.)	0	2000
	<type> = PID type	standard,fast,smooth,ext	
	trigger = analogue trigger output Meaning: 0: trigger output off 1: trigger output active	0,1	
	toltime = stability criterion (s)	0	1000

ReadAnalogVoltageToField

In analogue mode, an analogue input voltage sets the nominal value for the output (in a.u.) by: OutputCurrent (a.u.) = slope x InputVoltage + offset.

Command	ReadAnalogVoltageToField()		
Return value	ReadAnalogVoltageToField xxxxx,slope,offset		
	Response time: 0s	minimum	maximum
	slope = scaling factor (a.u./Volt)	-125	125
	offset = bias setting (a.u.)	-2000	2000

ReadCurrent

Reads the actual current output

Command	ReadCurrent()		
Return value	ReadCurrent xxxxx,amp		
	Response time: 0s	minimum	maximum
	amp = actual output current (amp)	-20	20

ReadCurrentSequenceLine

Output the actual sequence parameters during a sequence run.

Command	ReadCurrentSequenceLine()		
Return value	ReadCurrentSequenceLine xxxxx,<address>,amp,tolerance,toltime,<type>, repetition,time left		
	Response time: 0s	minimum	maximum
	<address> = internal memory address	0	64k
	amp = scaled output (a.u.)	-2000	2000
	tolerance = maximum output deviation(a.u.)	0	2000
	toltime = time of applied value in tolerance (s.)	0	1000
	<type> = PID type	standard,fast,smooth,ext	
	repetition = current repetition	0	1024
	time left = remaining time until time-out	0	10000

ReadEthernetConfig()

Set the IP, submask and port of the Ethernet interface.

Command	ReadEthernetConfig()		
Return value	ReadEthernetConfig xxxxx,IPAdresse,submask,gateway		
	Response time: 0s	minimum	maximum
	IPadresse : IP address of the hardware	--	--
	submask : submask of the hardware	--	--
	gateway : Communication gateway	--	--

ReadFeedbackMode Reads the format of the returns		
Command	ReadFeedbackMode()	
Return value	ReadFeedbackMode xxxxx,<mode>	
	Response time: 0s	list
	mode = echo-output Meaning: norm: return starts with command echo none: echo + status is suppressed	norm,none

ReadFieldToCurrent reads the conversion factors “slope” and “offset” from Ampere to other units (a.u.) with OutputCurrent (A) = slope x Signal(a.u.) + offset(A).			
Command	ReadFieldToCurrent()		
Return value	ReadFieldToCurrent xxxxx,slope,offset		
	Response time: 0s	minimum	maximum
	slope = scaling factor (Ampere/a.u.)	(-125 to -0.008) ∪ (0.008 to 125)	
	offset = bias current (A)	-2000	2000

ReadFreeMemory Write any byte to free memory				
Command	ReadFreeMemory(<address>)			
		minimum	maximum	default
	<address>: memory allocation	0	64k	0
Return value	ReadFreeMemory xxxxx,value			
	Response time: 0s	minimum	maximum	
	value = value at address	0	255	

ReadHardwareState Read the status of the current source		
Command	ReadHardwareState()	
Return value	ReadHardwareState xxxxx,status	
	Response time: 0s	list
	status Meaning: 0 : OK 1 : difference between external Load impedance and reference impedance 2 : Error in hardware	0,1,2

ReadHardwareVersion reads the firmware and software versions in the current source			
Command	ReadHardwareVersion()		
Return value	ReadHardwareVersion xxxxx, software version, firmware version		
	Response time: 0s	minimum	maximum
	software version = actual software version	A000	A999
	firmware version = actual firmware version	v000	v999

ReadMaxCurrent Reads the maximum output current			
Command	ReadMaxCurrent()		
Return value	ReadMaxCurrent xxxxx,maxI		
	Response time: 0s	minimum	maximum
	maxI = Maximum output current (A)	0.1	20 / 50*)

*) Dependent on hardware type

ReadMaxVoltage Reads the maximum output voltage			
Command	ReadMaxVoltage()		
Return value	ReadMaxVoltage xxxxx,maxV		
	Response time: 0s	minimum	maximum
	maxV = Maximum output voltage (Volt)	0	200

ReadOperationMode The current source is operated in a digital or analogue mode.		
Command	ReadOperationMode()	
Return value	ReadOperationMode xxxxx,<mode>	
	Response time: 0s	list
	<mode> = The actual operation mode	analog,digital

ReadOutputState The current source output can be set active or can be switched off.		
Command	ReadOutputState()	
Return value	ReadOutputState xxxxx,<mode>	
	Response time: 0s	list
	<mode> = The actual output mode	active,off

ReadPID Reads PID values from one of the four PID types			
Command	ReadPID(<type>)		
		list	
	type = PID type	standard,fast,smooth,ext	
Return value	ReadPID xxxxx,p1,p2,p3		
	Response time: 0s	minimum	maximum
	p1 = proportional factor (V/A)	0	800
	p2 = integral factor (V/As)	0	4E6
	p3 = Slewrate (V/s)	1	5E5

ReadReferenceImpedance Reads impedance and maximum deviations values for the inspection of an attached load			
Command	ReadReferenceImpedance()		
Return value	ReadReferenceImpedance xxxxx,R_{ref},L_{ref},deltaR$_{ref}$,deltaL$_{ref}$		
	Response time: 0s	minimum	maximum
	R_{ref} = Resistive load reference (Ohm)	0	1000
	L_{ref} = Inductive load reference (Hy)	0	1000
	deltaR$_{ref}$ = Max. resistance deviation(Ohm)	0	1000
	deltaL$_{ref}$ = Max. inductance deviation (Hy)	0	1000

ReadReferenceSensorParameters An external sensor can be used as feedback signal for the current source output in (a.u.) by: SensorOutput (V) = sens x CurrentOutput (a.u.) + offset.			
Command	ReadReferenceSensorParameters()		
Return value	ReadReferenceSensorParameters xxxxx,sens,offset		
	Response time: 0s	minimum	maximum
	sens = sensor sensitivity (Volt/a.u.)	(-125 to -0.008) ∪ (0.008 to 125)	
	offset = sensor offset (Volt)	-2000	2000

ReadSequenceLine				
The current source can be operated in a digital or analogue mode. The operation mode at start up can be set				
Command	ReadSequenceLine(<address>)			
		minimum	maximum	default
	<address> = internal memory address	1	64k	1
Return value	ReadSequenceLine xxxxx,amp,tol,toltime,type,marker			
	Response time: 0s	minimum	maximum	
	amp = scaled output (a.u.)	-2000	2000	
	tolerance = maximum output deviation(a.u.)	0	2000	
	toltime = time of applied value in tolerance (s.)	0	1000	
	<type> = PID type	standard,fast,smooth,ext		
	Marker = analogue marker output Meaning: 0: Output off 1: Output active	0,1		

ReadSerialNumber			
Read the current source serial number			
Command	ReadSerialNumber()		
Return value	ReadSerialNumber xxxxx,serialnumber		
	Response time: 0s	minimum	maximum
	serialnumber = Serial number of the current source	--	--

ReadStartupMode		
The current source can be operated in a digital or analogue mode. The operation mode at start up can be set		
Command	ReadStartupMode()	
Return value	ReadStartupMode xxxxx,<mode>	
	Response time: 0s	list
	<mode> = The operation mode at start up	analog,digital

ReadStatus

Read the 5 bit status register of the current source command

Command	ReadStatus()	
Return value	ReadStatus xxxxx,status	
	Response time: 0s	list
	status = 5 bit status of the latest command Meaning: bit 1: current source state. 0=ready 1=busy bit 2: syntax, parameter range. 0=OK 1=error bit 3: sequence status. 0=ready 1=running bit 4: run time error. 0=no error 1=error bit 5: exit type. 0=exit OK 1=time out	xxxxx

ReadTimer

Read the elapsed time since system start. Can be used as life counter.

Command	ReadTimer()		
Return value	ReadTimer xxxxx,time		
	Response time: 0s	minimum	maximum
	time = elapsed time in ms	0	2^{31}

ReadTrigger

Read the status of the trigger output, trigger input and the analogue marker

Command	ReadTrigger()	
Return value	ReadTrigger xxxxx,triggerOut,triggerIn,marker	
	Response time: 0s	list
	triggerOut : 0=off, 1=active	0,1
	triggerIn : 0=off, 1=active	0,1
	marker : 0=off, 1=active	0,1

ReadUserValue Reads the actual value at the current output in a.u.			
Command	ReadUserValue(<contr>)		
		list	
	<contr> = internal or external feedback	intern,extern	
Return value	ReadUserValue xxxxx,value		
	Response time: 0s	minimum	maximum
	value = actual value at current output (a.u.)	-2000	2000

ReadValue Reads the actual value at the current output in ampere			
Command	ReadValue()		
Return value	ReadValue xxxxx,value		
	Response time: 0s	minimum	maximum
	value = actual value at current output (A)	-20/-50*)	20/50*)

*) Dependent on hardware type

ReadVoltage Reads the actual voltage at the current output			
Command	ReadVoltage()		
Return value	ReadVoltage xxxxx,volt		
	Response time: 0s	minimum	maximum
	volt = actual voltage at current output (volt)	-100/-120*)	100/120*)

*) Dependent on hardware type

RestoreFactoryDefaults Reset all setting to the factory setting. All customer settings and information will be lost			
Command	RestoreFactoryDefaults()		
Return value	RestoreFactoryDefaults xxxxx		
	Response time: 12s	minimum	maximum

SetAnalogParameters Parameter setting for the analogue modus. If the nominal value is reached within the tolerance over a defined period of time, an analogue output trigger can be set.				
Command	SetAnalogParameters(tolerance,<type>,trigger,toltime)			
		minimum	maximum	default
	tolerance = maximum signal deviation (a.u.)	0	2000	1
	<type> = PID type	standard,fast,smooth,ext		standard
	trigger = analogue trigger output Meaning: 0: off 1: active	0,1		1
	toltime = stability criterion (s)	0	1000	0.05
Return value	SetAnalogParameters xxxxx			
	Response time: 0s	minimum	maximum	

SetAnalogVoltageToField In analogue mode, an analogue input voltage sets the nominal value for the output (in a.u.) by: OutputCurrent (a.u.) = slope x InputVoltage + offset.				
Command	SetAnalogVoltageToField(slope,offset)			
		minimum	maximum	default
	slope = scaling factor (a.u./volt)	-125	125	1
	Offset = bias field (a.u.)	-2000	2000	0
Return value	SetAnalogVoltageToField xxxxx			
	Response time: 0s	minimum	maximum	

SetEthernetConfig()

Set the IP, submask and gateway of the Ethernet interface.

Command	SetEthernetConfig(IPaddress,submask,gateway)			
		minimum	maximum	default
	IPadresse : IP address of the hardware	--	--	192.168.1.1
	submask : submask of the hardware	--	--	255.255.0.0
	gateway : Communication gateway	--	--	127.0.0.1
Return value	SetEthernetConfig xxxxx			
	Response time: 0s	minimum	maximum	

SetFeedbackMode

The echo-output+status as first element of the return value is activated or de-activated

Command	SetFeedbackMode(<mode>)		
		list	default
	<mode> = echo-output Meaning: norm: return starts with command echo none: echo + status is suppressed	norm,none	norm
Return value	SetFeedbackMode xxxxx		
	Response time: 0s	minimum	maximum

SetFreeMemory

Write any byte to free memory

Command	SetFreeMemory(<address>,value)			
		minimum	maximum	default
	<address>: memory allocation	0	32 k	0
	value: 8 bit value	0	255	0
Return value	SetFreeMemory xxxxx			
	Response time: 0s	minimum	maximum	

SetFieldToCurrent				
Sets the conversion factor "slope" and "offset" from Ampere to other units (a.u.) by OutputCurrent (A) = slope x Signal(a.u.) + offset(A).				
Command	SetFieldToCurrent(slope,offset)			
		minimum	maximum	default
	slope = scaling factor (Ampere/a.u.)	(-125 to -0.008) ∪ (0.008 to 125)		1
	Offset = bias current (A)	-2000	2000	0
Return value	SetFieldToCurrent xxxxx			
	Response time: 0s	minimum	maximum	

SetMaxCurrent				
Sets the maximum output current				
Command	SetMaxCurrent(maxI)			
		minimum	maximum	default
	maxI = Maximum output current (A)	0	20/50*)	20/50*)
Return value	SetMaxCurrent xxxxx			
	Response time: 0s	minimum	maximum	

*) Dependent on hardware type

SetMaxVoltage				
Sets the maximum output voltage				
Command	SetMaxVoltage(maxV)			
		minimum	maximum	default
	maxV = Maximum output Voltage (Volt)	0	200	100/120*)
Return value	SetMaxVoltage xxxxx			
	Response time: 0s	minimum	maximum	

*) Dependent on hardware type

SetOperationMode

The current source is operated in a digital or analogue mode.

Command	SetOperationMode(<mode>)		
		list	default
	<mode> = The actual operation mode	analog,digital	digital
Return value	SetOperationMode xxxxx		
	Response time: 0s	minimum	maximum

SetOutputState

The current source output can be set active or can be switched off.

Command	SetOutputState(<mode>)		
		list	default
	<mode> = The actual output mode	active,off	off
Return value	SetOutputState xxxxx		
	Response time: 0s	minimum	maximum

SetPID

Stores PID values in one of the four PID type

Command	SetPID(<type>,p1,p2,p3)			
		minimum	maximum	default
	<type> = PID type	standard,fast,smooth,ext		standard
	p1 = proportional factor	0	800	0
	p2 = integral factor	0	4E6	0
	p3 = slewrate	1	5E5	1
Return value	SetPID xxxxx			
	Response time: 0s	minimum	maximum	

SetReferenceImpedance

Set the impedance and maximum impedance deviations values for the inspection of an attached load

Command	SetReferenceImpedance(R_{ref},L_{ref},deltaR_{ref},deltaL_{ref})			
		minimum	maximum	default
	R_{ref} = Resistive load reference (R)	0	1000	0
	L_{ref} = Inductive load reference (L)	0	1000	0
	deltaR_{ref} = Max. resistive load deviation (R)	0	1000	0
	deltaL_{ref} = Max. inductive load deviation (L)	0	1000	0
Return value	SetReferenceImpedance xxxxx			
	Response time: 0s	minimum	maximum	

SetReferenceImpedance

Sets impedance values for the inspection reference of an attached load.
The resistive deviation is 0.1×R, the inductive deviation is 0.1×L.

Command	SetReferenceImpedance(R_{ref},L_{ref})			
		minimum	maximum	default
	R_{ref} = Resistive load reference	0	1000	0
	L_{ref} = Inductive load reference	0	1000	0
Return value	SetReferenceImpedance xxxxx			
	Response time: 0s	minimum	maximum	

SetReferenceSensorParameters

An external sensor can be used as feedback signal for the current source output in (a.u.) by: SensorOutput (V) = sens × CurrentOutput (a.u.) + offset.

Command	SetReferenceSensorParameters(sens,offset)			
		minimum	maximum	default
	sens = sensor sensitivity (Volt/a.u.)	(-125 to -0.008) ∪ (0.008 to 125)		1
	offset = sensor offset (Volt)	-2000	2000	0
Return value	SetReferenceSensorParameters xxxxx			
	Response time: 0s	minimum	maximum	

SetSequenceLine

Program a sequence of values to an internal address. The nominal output value is applied for “time” seconds when the output value equals “amp” within the “tolerance”.

Command	SetSequenceLine(<address>,amp,tolerance, toltime,<type>,<marker>)			
		minimum	maximum	default
	<address> = internal memory address	1	64k	1
	amp = scaled output (a.u.)	-2000	2000	0
	tolerance = maximum output deviation(a.u.)	0	2000	0
	toltime = time of applied value in tolerance (sec.)	0	1000	0
	<type> = PID type	standard,fast,smooth,ext		standard
	<marker> = analogue marker output	0,1		0
Return value	SetSequenceLine xxxxx			
	Response time: 0s	minimum	maximum	

SetStartupMode

The current source can be operated in a digital or analogue mode. The operation mode at start up can be set

Command	SetStartupMode(<mode>)		
		list	default
	<mode> = The operation mode at start up	analog,digital	digital
Return value	SetStartupMode xxxxx		
	Response time: 0s	minimum	maximum

SetUserValue

Set a value in a.u. at the current source output. The output state is set to “active”.

Command	SetUserValue(value,slewrate,<type>,<contr>)			
		minimum	maximum	default
	value = nominal value in a.u. at output	-2000	2000	0
	slewrate = maximum slewrate (a.u./s)	0.1	1E5	1E5
	<type> = PID type	standard,fast,smooth,ext		standard
	<contr> = intern or external feedback	intern,extern		intern
Return value	SetUserValue xxxxx			
	Response time: 0s	minimum	maximum	

SetValue

Set a current at the current source output. The output state is set to “active”.

Command	SetValue(value)			
		minimum	maximum	default
	value = nominal current value at output	-2000	2000	0
Return value	SetValue xxxxx			
	Response time: 0s	minimum	maximum	

StartSequence

Starts or restarts the sequence of programmed lines from address <start> to address <stop>. The progress mode is defined by <proc> (time or trigger)

Command	StartSequence(<start>,<stop>,<rep>,<proc>,runtime,<contr>,time-out)			
		minimum	maximum	default
	<start> = first sequence address	1	64k	1
	<stop> = last sequence address	1	64k	1
	<rep> = number of repetitions	1	1024	1
	<proc> = mode of progress	trigger,time		time
	runtime = duration of a sequence (s)	0.0001	10000	0.05
	<contr> = intern or external feedback	intern,extern		intern
	Time-out = maximum time for the full process (s)	0.0001	10000	100
Return value	StartSequence xxxxx			
	Response time: 0s	minimum	maximum	

StopSequence

Stops a running sequence process and switch the output state to “off”

Command	StopSequence()		
Return value	StopSequence xxxxx		
	Response time: 0s	minimum	maximum