Diode Current Source
DSQ3K
User Manual

Doc. No.	ARGOS 545-8300-0001
Issue	1.0
Date	2011-06-20

Prepared	J. Ziegleder	2010/09/30
	Name	Date
Approved	N. Surname	yyyy/mm/dd
	Name	Date
Released	N. Surname	yyyy/mm/dd
	Name	Date

TABLE OF CONTENTS

Change Record

Issue	Date	Section/ Paragraph Affected	Reasons / Remarks	Name
1.0	30.09.2010	all	created	J. Ziegleder
	5.11.2010	SCPI commands	prepared	S. Ovari
	8.11.2010	SCPI commands	included	J. Ziegleder
	29.04.2011	Chapter 6 & 8	modificated	J. Ziegleder
	20.06.2011	Chapter 6 & 8	modificated	J. Ziegleder

1 Scope

This manual describes the use of the diode current source DSQ3K.

2 Applicable documents

No.	Title	Number & Issue
AD 1	LBT Laser Phase A study report	1.0
AD 2	ARGOS FDR documentation	1.0
AD 3	NE-4100 Series User's Manual	Ninth Edition, June 2008
AD 4	LUXEON Rebel Color Datasheet DS65	09/10/07
AD 5	ARGOS FDR 011	1.0
AD 6		
AD 7		
AD 8		
AD 9		
AD 10		
AD 11		
AD 12		
AD 13		
AD 14		
AD 15		
AD 16		
AD 17		

3 Overview

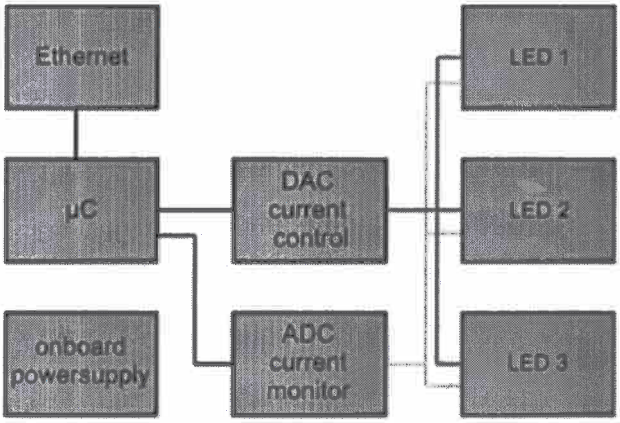

Figure 1: Scheme of the 3 channel LED current source

The DSQ3K is a 3U rack mounted three channel diode current source. This plug-in unit is fully remote controlled via an Ethernet interface, and operated by a single 24VDC voltage. Great importance was attached to stability, ease of use and EMC immunity. Light weight and low power dissipation were achieved by using a dedicated LED switching regulator.
Figure 1 shows the scheme of the device. A microcontroller cares about all internal control processes and the communication to the user. Hardware protection of the circuit is implemented to prevent failures due to software errors, both on user and on device side. The communication is simplified by the use of SCPI (= Standard Commands for Programmable Instruments) for remote control.
The user has the possibility to:

- Set the output current, each port individually
- Switch the channels on and off.
- Monitor the diode current
- Read out onboard diagnostics

See AD 5, p.27f for the motivation to build this current source.

4 Specifications

Outputs

Number of output channels	3
Current range per channel	35mA – 700mA
Resolution	8bit
Current in off-state	0mA
Allowed forward voltage of load	2.4V - 12V
PWM switching frequency and duty cycle	Not yet implenented

Inputs

Power	24VDC +5% / -10%, 0.5A. Via backplane
Interface	Ethernet 10/100 Mbit/s. Moxa RealCom or TCP connection

Miscellaneous

Dimensions	170mm x 60.62mm (12HP) x 127.4mm (3U)
Operating temperature	+5°C… +40°C
Max. rel. humidity	5%...80% (non condensing)
Weight	TBD
Main Fuses	Polyswitch; trip 3A, hold 1.5A @20°C
Cooling	Air ventilated

Subject to change without notice.

5 Important Hints

This is a non-commercial electronic device. Handle with appropriate care and obey the relevant safety rules. Do not operate outside the specifications; do not cover the ventilating slots.

6 Controls, connectors and user-serviceable parts

6.1 Front Panel

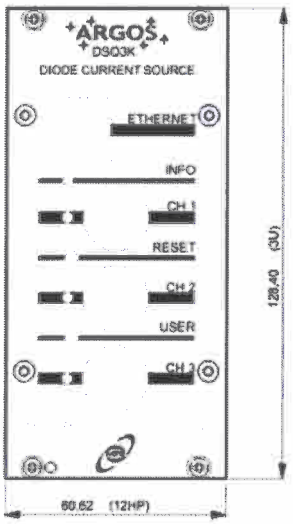

Figure 2: Front of the device.

Ethernet
Control interface. The internal device server module NE-4100T from the company "Moxa" is operated in RealCom mode. TCP server mode is also possible.

Info
Off in normal operation. A continous signal indicates that the device runs in service mode. Flashing light indicates an error.

CH 1, CH2, CH3
The fibre couplers are the optical output of the integrated LED modules. A small signal indicator is illuminated when the LED is switched on.

Tighten the fibre connector carefully. Unscrewing might accidentally unscrew the optics and parts could fall into the device!

RESET
Press the recessed reset button to perform a hardware reset of the microcontroller and the device server module.

USER
Only for service purposes.

6.2 Rear panel

The MAC address of the device is labelled to the rear panel.

Connector: DIN 41612 type D ac, 32pin

	A2	C2	
	A4	C4	
DC IN +	A6	C6	DC IN +
DC IN -	A8	C8	DC IN -
	A10	C10	
	A12	C12	
	A14	C14	
	A16	C16	
	A18	C18	
	A20	C20	
	A22	C22	
	A24	C24	
	A26	C26	
	A28	C28	
	A30	C30	
P.E.	A32	C32	P.E.

Table 1. Pinout of the rear panel connector.

Voltage between DC IN and P.E. must not exceed 26V.

6.3 Inside the device

Despite from the optics, there are no user service able parts inside. Disconnect all connectors and ensure an ESD conforming workplace while the enclosure is open.

Opening
To open the enclosure, open the 4 screws of the cover (2 on front panel, 2 on rear panel). The cover is the aluminium half, most distant to the backplane connector. Be careful with the fan and the P.E. cable, both are connected to the cover. These cables can be unplugged. Remove the perforated metal plates if necessary.
Usually there is no need to remove the LED modules. If you have to, remove the front panel completely and pull the electronics board out of the enclosure. The LEDs are fixed with srews which are on the bottom side. Pull the module carefully to the front side of the board.

Reassemble
Put every part in the place where you dissembled it. Do not forget the conenctors of the fan, P.E. and the temperature sensor.

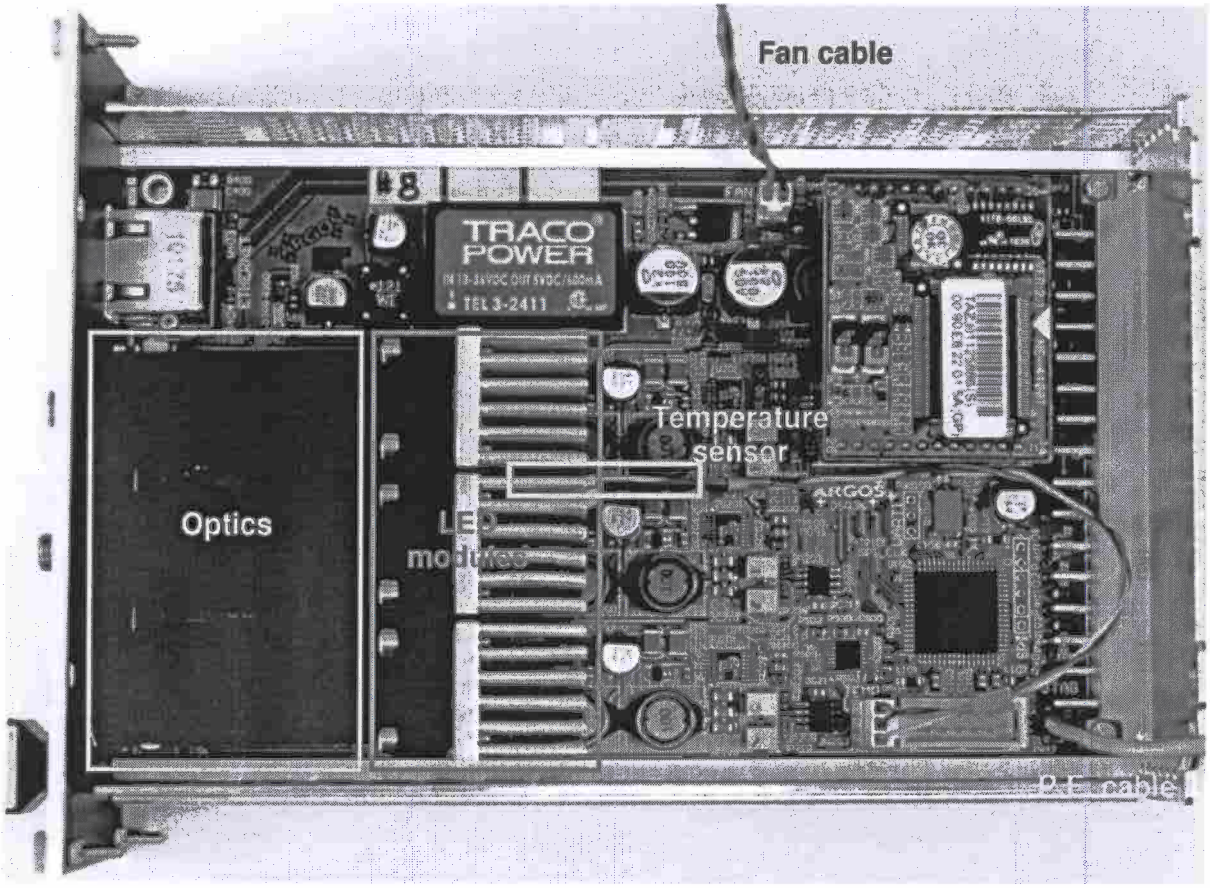

Figure 3: The LED modules are mounted to the electronics board. By opening the cover of the enclosure, the optics are accessible.

7 Operating

- Connect an adequate power supply to the rear connenctor. Two LEDs inside the device indicate that the circuit is powered. Ensure that the ventilation slots are not covered.
- Use a commen patch cable to connect the ethernet port to your network. See AD 3 how to set the IP address. Ensure that the device server operates in RealCom mode or telnet mode. You can find the MAC address on the rear panel.
- Connect optical fibres to the optic modules.
- Use a terminal to send commands or use high level software, if available.
- Have fun.

8 Remote control

The DSQ3K uses SCPI (= Standard Commands for Programmable Instruments) for remote control. Remote control is possible via the built-in Ethernet interface.

The following commands are based on the Standard Commands for Programmable Instruments (SCPI) including standard commands and specific commands to control the three LED channels and to set/get different system parameters.

- Every transmittion and every receipt ends with a linefeed ('\n').
- A delay of 25ms after every command helps avoiding time-out problems.

8.1 Standard Commands

Description of the following Symbols:

Symbol	Description
< >	variable, predefined element
=	equality, identical to
|	or
( )	group of elements, commentary
[ ]	optional elements
{ }	multitude of elements
' '	example

Command	Reply	Description
***CLS**	None	Clear Status Command
***ESR?**	Range: 0-255 8Bit Register	Standard Event Status Register Query
***ESE <value>**	None	Standard Event Status Enable Command
***ESE?**	Range: 0-255 16Bit Register	Standard Event Status Enable Query
STAT:QUES:EVEN?	Range: 0-65535 16Bit Register	Questionable Event Status Register
STAT:QUES:ENAB <value>	None	Status Questionable Enable Register
STAT:QUES:ENAB?	Range: 0-65535 16Bit Register	Status Questionable Enable Query
STATus:PRESet	None	Status Questionable Enable Reset
***STB?**	Range: 0-255 8Bit Register	Read Status Byte Query
***SRE <value>**	None	Service Request Enable Command
***SRE?**	Range: 0-255 8Bit Register	Service Request Enable Query
***IDN?**	char [31]	Identification Query
***OPC <value>**	None	Operation Complete Command
***OPC?**	{'0'|'1'}	Operation Complete Query
***RST**	None	Reset Command
***TST?**	None	Self-Test Query
***WAI**	None	Wait-to-Continue Command

All the commands followed by a <value> need one ***prefix blank*** ahead the value!

Closer description of the registers:

- ***CLS**
  This command clears all status data structures in a device.

  Standard Event Status Register
  Questionable Status Register
  Status Byte Register
  Error/Event Queue

- ***ESR?**
  Standard Event Status Register Query

  This command reads out the value of the standard event status register and clears the register.
  Range: 8Bit Register 0-255

Standard Event Status Register	
1	**Not used**
2	**Not used**
4	**Not used**
8	**Not used**
16	**Execution Error**
32	**Command Error**
64	**Not used**
128	**Power ON**

Example: ***ESR?** **//Reply: "32" → command error**
***ESR?** **//Reply: "0"**

- ***ESE <value>**
  Standard Event Status Enable Command

  This command enables the bits of the Standard Event Status Register. If the value is smaller then 100 fill in one prefix zero, if the value is smaller then 10, fill in two prefix zeros.

  Range: 8Bit Register 0-255

Standard Event Status Enable Register (Default value = 176)	
1	**0**
2	**0**
4	**0**
8	**0**
16	**1**
32	**1**
64	**0**
128	**1**

  The default value is 176. If this value has to be changed, the sum of the specific bits is the value for the command.

  Example: ***ESE 000** **// all errors are disabled and they won't have any effect // on the system**

- ***ESE?**
  Standard Event Status Enable Query

  This command reads out the value of the standard event status enable register.
  Range: 8Bit Register 0-255

  Example: ***ESE 176**
  ***ESE? //Reply: "176"**

- **STATus:QUEStionable:EVENt?**
  Questionable Event Status Register

  This command reads out the value of the questionable event status register and delets the register. There are two possible events: **current overload** and **temperature overrun**
  Range: 16Bit Register 0-65535

Questionable Event Status Register	
1	**Not used**
2	**Current overload**
4	**Not used**
8	**Not used**
16	**Temperature overrun**
32	**Not used**
....	**Not used**
....	**Not used**
....	**Not used**
....	**Not used**
32768	**Not used**

Example: **STAT:QUES:EVEN?** **//Reply: 18 → current overload and temperature overrun**

If the corresponding bits in the Status Questionable Enable Register are enabled, the errors in the Questionable Event Status Register will have an effect on the system.
If the corresponding bits are disabled, the errors in the Questionable Event Status Register won't have any effect on the system.

- **STATus:QUEStionable:ENABle <value>**
  Status Questionable Enable Register

  This command enables the bits of the Status Questionable Enable Register.
  If the value is smaller then 10000 fill in one prefix zero, if the value is smaller then 1000, fill in two prefix zeros and so on.
  The following bits can be selected:

Standard Event Status Enable Register (Default value = 00018)	
1	**0**
2	**1**
4	**0**
8	**0**
16	**1**
32	**0**
....	**0**
32768	**0**

- **STATus:QUEStionable:ENABle?**
  Status Questionable Enable Query

  This command reads out the value of the status questionable enable register.
  Range: 16Bit Register 0-65535

  Example: **STATus:QUEStionable:ENABle 00018** **// 00018 or 00002 or 00000**
  STATus:QUEStionable:ENABle? **// Reply: "18"**

- **STATus:PRESet**
  Status Questionable Enable Reset

  This command delets the Status Questionable Enable Register. If this command is sent, the Status Questionable Register has no effect on the system and overload current and a temperature overrun will be ignored.

- ***STB?**
  Status Byte Query

  This command reads out the value of the status byte register. If there is an event written to the **questionable data** register or a **standard event** occurs or a **message is available**, the corresponding bit in the Status Byte Register is set.

Status Byte Register	
1	**Not used**
2	**Not used**
4	**Not used**
8	**Questionable Data**
16	**Message available**
32	**Standard Event**
64	**Request Service (not available yet)**
128	**Not used**

Example: ***STB?** **// Reply: "16"**
// message available

- ***SRE <value>**
  Service Request Enable Command

  This command enables the Request Service Bit of the Status Byte Register. If there is an event written to the status byte register and the corresponding bits are set by the Service Request Enable Command, the corresponding bit (Request Service) in the Status Byte Register is set.
  If the value is smaller then 100 fill in one prefix zero, if the value is smaller then 10, fill in two prefix zeros.
  Range: 8Bit Register 0-255

  Example: ***SRE 120** **// all the available information of the Status Byte is**
  // enabled

Status Byte Enable Register (Default value = 120)	
1	**0**
2	**0**
4	**0**
8	**1**
16	**1**
32	**1**
64	**1**
128	**0**

- ***SRE?**
  Service Request Enable Query

  This command reads out the value of the service request enable register.
  Range: 8Bit Register 0-255

  Example: ***SRE 008**
  ***SRE?** **// Reply: "8"**

- ***IDN?**
  Identification Query

  This command reads out the identification of the hardware. The id is limited to 31 characters (char [30] + '\0'). In this case the id is:

  "MPE,DSQ3K,SN01,10-10-01-001 „

- ***OPC <value> and *OPC?**
  Operation Comlpete Command and Operation Complete Query

  The *OPC command has no effect on the system and the *OPC? sends back the value '1'.

- ***RST**
  Reset Command

  This command deletes the settings which are done by the user and the system is set back to its start conditions. **This is no hardware reset!**

- ***TST?**
  Hardware Test

  This command causes a test sequence of the system and checks all the relevant parameters. During the test there are different current measures. If there are any unexpected measured current values, the current overload bit in the questionable status register is set and can be detected by the corresponding command.
  This procedure will take about 10s. Do not send any commands in the meanwhile!

- **WAI**
  Wait
  This command has no effect on the system!

8.2 System specific Commands

The diode current source consists of three different channels and offers the following opportunities:

- **Select one of the three channels**
- **Set the current for the diode**
- **Turn on/off the selected channel**
- **Measure the current of the chosen channel**
- **Measure the air temperature inside the device**
- **Get information about the system**

The following commands control the three channels :

Command	Description
INSTrument[:SELect] {OUTPut1|OUTPut2|OUTPut3|OUT1|OUT2|OUT3}	Selection of one of the three channels
INSTrument:NSELect { 1 | 2 | 3 }	Delection of the channel
INSTrument[:SELect]?	Query of the channel selection **Reply: {OUTP1|OUTP2|OUTP3}**
[SOURce:]CURRent[:LEVel] < current >	Set the current of the chosen channel **Range: 0.021 – 0.701 [A]**
CURRent?	Query of the selected current **Reply: 0.000 – 0.701 [A]**
OUTPut[:STATe] {OFF|ON|0|1}	Turning on/off the outputs
OUTPut[:STATe]?	Query whether the output is on or off **Reply: {ON|OFF}**
MEASure[:SCALar]:CURRent[:DC]?	Reading back the current of the selected channel **Reply: 0.000 – 0.800 [A]** Note: The current measurement is not very acurate. The device is calibrated and the actual current is very constant over a wide range of temperature.
MEASure:TEMPerature?	Reading back the temperature **Reply: 0 – 150 [°C]**
SYSTem:ERRor?	Reading out the errors **Reply: 0, no error** **-100, command error** **-310, system error** **-311, memory error** **-350, queue overflow** **-500, power on**

All the commands followed by a <value> or an expression {....} need one ***prefix blank*** ahead the value or the expression!

- **Command for the channel selection**

  INSTrument[:SELect] {OUTPut1|OUTPut2|OUTPut3|OUT1|OUT2|OUT3}

  This command selects one of the three channels. The user has different opportunities to choose one channel e.g. select channel 1:

  Example: **INST OUTP1** **// Channel 1 is selected**
  INST OUT1
  INSTrument OUTPut1
  INSTrument:SELect OUT1

- **Command for channel deselection**

  INSTrument:NSELect { 1 | 2 | 3 }

  This command deselects the chosen channel and following channel specific commands like setting the current or measuring the current won't have any effect.

  Example: **INST:NSEL 1** **// Channel 1 is no longer selected**

- **Command for instrument query**

  INSTrument[:SELect]?

  This command reads back the selected channel. The reply is whether OUTP1, OUTP2 or OUTP3.

  Example: **INST OUTP1**
  INST? **// Reply: OUTP1**

- **Command for setting the current**

  [SOURce:]CURRent[:LEVel] < current >

  This command sets the current of the selected channel. The minimum is 0.021 [A] up to 0.701 [A].

  Example: **INST OUTP1**
  CURR 0.123 **// the current is 0.123 [A] for channel 1**

- **Command for current query**

  CURRent?

  This command reads back the selected current of the chosen channel.

  Example:
  ```
  INST OUTP1
  CURR 0.123        // the current is 0.123 [A] for channel 1
  CURR?             // Reply: 0.123
  ```

- **Command for turning on/off the LED**

  OUTPut[:STATe] {OFF|ON|0|1}

  This command turns on/off the selected LED channel.

- **Command for output query**

  OUTPut[:STATe]?

  This command sends back ON or OFF whether the channel is activated or not.

  Example:
  ```
  INST OUTP1
  OUTP?             // Reply: OFF
  OUTP ON           // turns on channel 1 (after reset the value for
                    // the current is set to minimum)
  OUTP?             // Reply: ON
  ```

- **Command Measure**

  - **measure of current**

    MEASure[:SCALar]:CURRent[:DC]?

    This command sends back the measured current of the selected channel. If no channel is selected, there won't be any reply.

  Note: The current measurement is not very acurate. The device is calibrated and the actual current is very constant over a wide range of temperature.

  - **measurement of temperature**

MEASure:TEMPerature?

This command sends back the measured temperature from the temperature sensor.

Example:

```
INST OUTP1
CURR?                  // Reply: 0.000 (default value after reset)
CURR 0.234
CURR?                  // Reply: 0.234
OUTP ON
MEAS:CURR?             // Reply: 0.234
OUTP OFF
INST:NSEL 1
```

- **Command for system information**

SYSTem:ERRor?

This command sends back information about the system. The following commands are possible replies for the system request

Reply:

0, no error
-100, command error
-310, system error
-311, memory error
-350, queue overflow
-500, power on

After a reset the command "-500, power on" is always set. If there is any problem with the memory after reset the second command will be "-311, memory error". A wrong command will result in a command error (-100, command error) record in the queue.

There is a maximum of 20 informations which can be read out by the user. Every call of the command SYSTem:ERRor? quits the error meassage.

8.3 More Examples

Example 1:

```
INST OUT1          // Channel 1 is selected
INST?              // Reply: OUTP1
CURR 0.500         // current = 500mA
CURR?              // Reply: 0.500
OUTP ON            // enables the LED driver channel 1
MEAS:CURR?         // Reply: 0.500
MEAS:TEMP?         // Reply: 35[°C]
INST:NSELect 1     // Channel 1 is no longer selected
```

Example 2:

```
WRONG_COMMAND...   // a wrong command is sent
SYSTem:ERRor?      // Reply after Reset:   -500, power on
SYSTem:ERRor?      // Reply:               -100, command error
SYSTem:ERRor?      // Reply:               0, no error
```

Example 3: //the temperature rises above the maximum value (default value = 90°C)

```
*STB?              // Reply: 56
*STB?              // Reply: 56
SYST:ERR?          // Reply: -500, power on
SYST:ERR?          // Reply: 0, no error
*ESR?              // Reply: 128 (power on bit is set)
*ESR?              // Reply: 0 (standard event register is cleared by first *ESR?)
STAT:QUES:EVEN?    // Reply: 16 (temperature overrun bit is set)
STAT:QUES:EVEN?    // Reply: 0

// after reading out the status questionable event register the register is
// cleared and the system runs in normal mode
```

APPENDIX (1):

List of acronyms

µC	Microcontroller
LED	Light Emitting Diode
P.E.	Protective Earth
mA	milli ampere

End of document