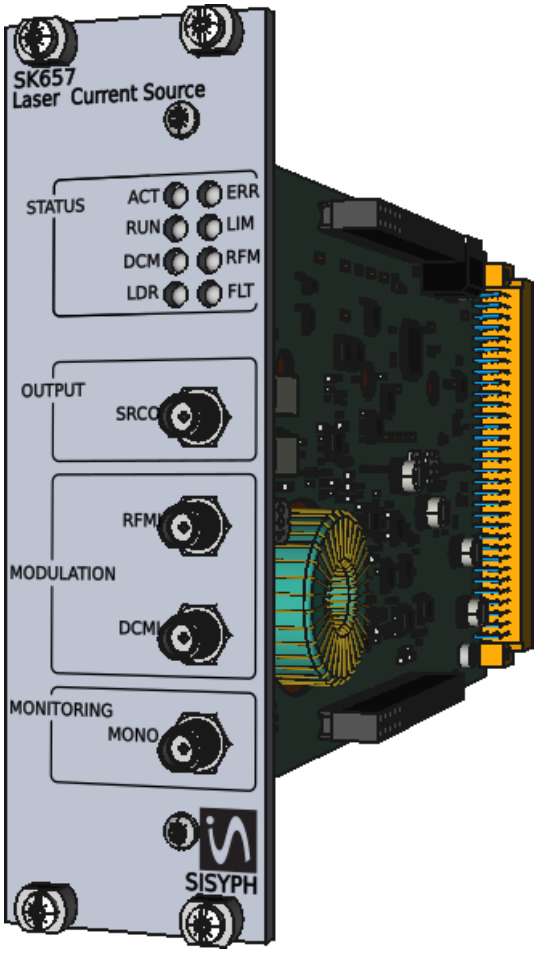
Programming Guide

SK657*Gavarnie*
Laser Diode Current Controller

SK-Series Modules



SISYPH
Signals & Systems for Physics

www.sisyph.com

General Information

Important Notice

Signals and Systems for Physics
206 rue Alfred Kastler
17190 Saint-Georges-d'Oléron
France
Phone (+33) 781 547 391
www.sisyph.com

Scope

This document describes operating the module over the serial interface.

Contents

1 Introduction

Remote operation of the SK657 is through a simple command language documented in this chapter. Both set and query forms of most commands are supported, allowing the user complete control of the module from a remote computer.

1.1 Power-on Configuration

The settings for serial interface are 9600 baud with no parity and no hardware flow control, and local echo disabled (`CONS 0`).
Most of the instrument settings are stored in non-volatile memory and can be retrieved using the appropriate commands. At power-on the instrument returns to the state noted in the command descriptions. Reset values (`*RST` command) of parameters are shown in **boldface**.

1.2 Buffers

The instrument stores incoming bytes from the host interface in a 128-byte input buffer. Characters accumulate in the input buffer until a command terminator (either `<CR>` or `<LF>`) is received, at which point the message is parsed and executed. Query responses from the instrument are sent when they are ready without any flow control nor output buffering. The input buffer is automatically flushed upon detecting an overflow, and an error is recorded in the EVTS status register.

1.3 Command syntax

The four letter mnemonic (shown in CAPS) in each command sequence specifies the command. The rest of the sequence consists of parameters. The command parser accepts only uppercase mnemonics.
Commands may take either set or query form, depending on whether the `?` character follows the mnemonic. *Set only* commands are listed without the `?`, *query only* commands show the `?` after the mnemonic, and *optionally query* commands are marked with a `(?)`. Parameters shown in `{ }` and `[ ]` are not always required. Parameters in `{ }` are only required to set a value, and should be omitted for queries. Parameters in `[ ]` are optional in both set and query commands. Parameters listed without any surrounding characters are always required. Do not send `()` or `{ }` or `[]` as part of the command. Multiple parameters are separated by commas. Multiple commands may be sent on one command line by separating them with semicolons `;` so long as the input buffer does not overflow. Commands are terminated by either `<CR>` or `<LF>` characters. Null commands and whitespace are ignored. Execution of the command does not begin until the command terminator is received.
The following table summarizes the notation used in the command descriptions:

Symbol	Definition
b	Boolean
i, m, n	Unsigned integers
u, v	Signed integers
(?)	Required for queries; illegal for set commands.
p	Parameter always required.
{*p*}	Required parameter for set commands; illegal for queries.
[*p*]	Optional parameter for both set and query forms.

1.4 Examples

Each command is provided with a simple example illustrating its usage. In these examples, all data sent by the host computer to the instrument are set as `straight teletype font`, while responses received the host computer from the instrument are set as *`slanted teletype font`*. The usage examples vary with respect to set/query, optional parameters, and token formats. These examples are not exhaustive, but are intended to provide a convenient starting point for user programming.

2 List of Commands

This section provides syntax and operational descriptions for remote commands.

2.1 Instrument Settings commands

The Instrument Settings commands provide control of the instrument's physical parameters.

IFIN (Laser Biasing Current - Fine Value)

Group	Settings commands
Action	Set/Query
Syntax	`IFIN(?){`*m*`}`
Description	Set (query) fine current {to m}, in µA (12-bit DAC).
Allowed range	$\mathbf{0} \leq m \leq 10000$
Reset value	**0**
Power-on value	Restored from non-volatile memory (*cf* *SAV).
Example	`IFIN 5000; IFIN?`
	5000
Related commands	ICRS, ILIM

ICRS (Laser Biasing Current - Coarse Value)

Group	Settings commands
Action	Set/Query
Syntax	`ICRS(?){`*m*`}`
Description	Set (query) the coarse current {to m}, in mA (12-bit DAC).
Allowed range	$0 \leq m \leq 500$
Reset value	**200**
Power-on value	Restored from non-volatile memory (*cf* *SAV).
Example	`ICRS 250; ICRS?`
	250
Related commands	IFIN ILIM

ILIM (Current Limitation)

Group	Settings commands
Action	Set/Query
Syntax	`ILIM(?){`*m*`}`
Description	Set (query) the current limitation {to m}, in mA.
Allowed range	$0 \leq m \leq 1000$
Reset value	**250**
Power-on value	Restored from non-volatile memory (*cf* *SAV).
Example	`ILIM 600; ILIM?`
	600
Related commands	IFIN ICRS

2.2 Instrument Configuration commands

The Instrument Configuration commands provide control of the instrument's physical functionalities.

LDEN (Laser Diode Enable)

Group	Configuration commands
Action	Set/Query
Syntax	`LDEN(?){b}`
Description	Set (query) the laser output state {to b}. When $b = 0$, the output current is set to 0 mA and the SRCO connector's terminals are shorted. When $b = 1$, the slow turn-on procedure is engaged : after a 5-s delay period, during which the RUN indicator will blink, the shorting relay will be heard to open, and the laser current source will quickly ramp up to the programmed current setting. To abort turning the laser on during this delay, the `LDEN 0` must be invoked.
Allowed range	$b \in \{\mathbf{0}, 1\}$
Reset value	**0**
Power-on value	0
Example	`LDEN 1`
Related commands	IFIN ICRS

REAR (Rear Output Enable)

Group	Configuration commands
Action	Set/Query
Syntax	`REAR(?){b}`
Description	Set (query) the rear output state {to b}. When $b = 0$, the front-panel connector (SRCO) is used to deliver the source's current to the laser. When $b = 1$, either the expansion connector (EXP-AIO) or the user's terminal blocks (UTB) can be used to connect the laser to the current source.
Allowed range	$b \in \{\mathbf{0}, 1\}$
Reset value	**0**
Power-on value	0
Example	`REAR 1; REAR?` *1*
Related commands	

DCME (DC-Modulation Enable)

Group	Configuration commands
Action	Set/Query
Syntax	`DCME(?){b}`
Description	Set (query) the DC-modulation state {to b}. When $b = 0$ (resp. 1), the DC-modulation circuit is disabled (resp. enabled). The signal source should be selected using the DCMS command prior enabling this functionality. *Do not activate the modulation until the laser has been properly biased.*
Allowed range	$b \in \{\mathbf{0}, 1\}$
Reset value	**0**
Power-on value	0
Example	`DCME 1`
Related commands	DCMS

RFME (RF-Modulation Enable)

Group	Configuration commands
Action	Set/Query
Syntax	`RFME(?){b}`
Description	Set (query) the RF-modulation state {to b}. When $b = 0$ (resp. 1), the RF-modulation circuit is disabled (resp. enabled). *Do not activate the modulation until the laser has been properly biased.*
Allowed range	$b \in \{\mathbf{0}, 1\}$
Reset value	**0**
Power-on value	0
Example	`RFME 1; RFME?` *1*
Related commands	

FPSE (Front-Panel Switch Enable)

Group	Configuration commands
Action	Set/Query
Syntax	`FPSE(?){b}`
Description	Set (query) the front-panel switch operation {to b}.
	When $b = 0$ (resp. 1), the optional switch functionality is disabled (resp. enabled). In both cases, the LDEN command still operates to control the laser state.
Allowed range	$b \in \{0, \mathbf{1}\}$
Reset value	**1**
Power-on value	1
Example	`FPSE 1; FPSE?`
	1
Related commands	LDEN

ILKE (Interlock Switch Enable)

Group	Configuration commands
Action	Set/Query
Syntax	`ILKE(?){b}`
Description	Set (query) the state of the safety interlock {to b}.
	When $b = 0$ (resp. 1), the interlock functionality is disabled (resp. enabled).
Allowed range	$b \in \{0, \mathbf{1}\}$
Reset value	**1**
Power-on value	1
Example	`ILKE 1; ILKE?`
	1
Related commands	

DCMS (DC-Modulation Source Select)

Group	Configuration commands
Action	Set/Query
Syntax	`DCMS(?){`*n*`}`
Description	Set (query) the DC-modulation source {to n}. This command does not enable the modulation circuit. For this purpose, the DCME command must be used. Note that when the DCM jumper is mounted on the DCMI location, the input source multiplexer is bypassed and the signal from the DCMI front-panel connector is directly routed to the modulation circuit. This provides the maximal bandwidth. In this case, the DCMS command has no effect.
Allowed range	$n \in \{0, 1, 2, 3, \mathbf{4}\}$ where: $0 \longleftrightarrow$ user terminal block, $1 \longleftrightarrow$ backplane line, $2 \longleftrightarrow$ expansion board connector, $3 \longleftrightarrow$ front-panel SMA connector, $\mathbf{4} \longleftrightarrow$ ground voltage (0 V).
Reset value	**4**
Power-on value	Restored from non-volatile memory (*cf* *SAV).
Example	`DCMS 1; DCMS?` *1*
Related commands	DCME

MONS (Monitoring Output Source Select)

Group	Configuration commands
Action	Set/Query
Syntax	`MONS(?){`*`n`*`}`
Description	Set (query) the monitoring output source {to n}. The selected signal is then available at the MONO output connector.
Allowed range	$n \in \{0, 1, 2, \mathbf{3}\}$ where: $0 \longleftrightarrow$ laser voltage buffer, $1 \longleftrightarrow$ laser current sensor, $2 \longleftrightarrow$ /STATUS signal, $\mathbf{3} \longleftrightarrow$ ground voltage (0 V).
Reset value	**3**
Power-on value	Restored from non-volatile memory (*cf* *SAV).
Example	`MONS 1; MONS?` *1*
Related commands	

VCMP (Compliance Voltage - Trip Point)

Group	Configuration commands
Action	Set/Query
Syntax	`VCMP(?){m}`
Description	Set (query) the maximal allowed compliance voltage {to m}, in mV (8-bit DAC). When the laser diode voltage is above the threshold, the current source is immediately shut down and its output terminals are shorted.
Allowed range	$1000 \leq m \leq$ **5000**
Reset value	**5000**
Power-on value	Restored from non-volatile memory (*cf* *SAV).
Example	`VCMP 3000; VCMP?` *3000*
Related commands	

2.3 Instrument Monitoring commands

The Instrument Monitoring commands provide the host computer with the last measurements of the instrument's physical parameters.

ADCR (Last ADC Conversion Reading)

Group	Monitoring commands
Action	Query only
Syntax	`ADCR?`*`n`*
Description	Return the last ADC conversion result of the channel n, in mV.
Allowed range	$n \in \{0, 1, 2, 3, 4\}$ where: $0 \longleftrightarrow$ laser voltage buffer, $1 \longleftrightarrow$ laser current sensor, $2 \longleftrightarrow$ internal negative voltage (maintenance), $3 \longleftrightarrow$ current limiter trip point (maintenance), $4 \longleftrightarrow$ ground voltage (0 V).
Example	`ADRC? 1` *`503`*
Related commands	

2.4 Status commands

The Status commands query and configure registers associated with status reporting of the instrument.

*CLS (Clear Status Registers)

Group	Status commands
Action	Query only
Syntax	`*CLS`
Description	Clear immediately all status registers, which are : LEXE, LCMD, LINS, LURQ, INSS, OVLS, COMS and EVTS.
Example	`*CLS`
Related commands	

MSTS (Master Summary Status)

Group	Status commands
Action	Query only
Syntax	`MSTS? [`*`n`*`]`
Description	Return the Master Summary Status register [bit-mask n]. The execution of the MSTS? query – without the optional bit-mask n – always causes the /STATUS signal to be de-asserted. Note that MSTS? n will not clear /STATUS, even if bit $i \mid n = 2^i$ is the only bit presently causing the /STATUS signal.
Power-on value	0
Example	`MSTS?; MSTS? 128;` *129* *128*
Related commands	MSTE

MSTE (Master Summary Enable)

Group	Status commands
Action	Set/Query
Syntax	`MSTE(?)[`*n*`]{`*m*`}`
Description	Set (query) the Master Summary Enable register [bit-mask n] {to bit-mask m}. The set-form command will clear the bits outside the bit-mask.
Power-on value	0
Example	`MSTE 128; MSTE?` *128*
Related commands	MSTS

EVTS (Event Status)

Group	Status commands
Action	Query only
Syntax	`EVTS? [`*n*`]`
Description	Read the Event Summary Status register [bit-mask *n*].
Power-on value	1
Example	`EVTS?`
	4
Related commands	EVTE

EVTE (Event Enable)

Group	Status commands
Action	Set/Query
Syntax	`EVTE(?)[`*n*`]{`*m*`}`
Description	Set (query) the Event Summary Enable register [bit-mask *n*] {to bit-mask *m*}. The set-form command will clear the bits outside the bit-mask.
Power-on value	0
Example	`EVTE 4; EVTE?` *4*
Related commands	EVTS

COMS (Communications Status)

Group	Status commands
Action	Query only
Syntax	`COMS? [`*n*`]`
Description	Read the Communications Status register [bit-mask n].
Power-on value	0
Example	`COMS?`
	0
Related commands	COME

COME (Communications Enable)

Group	Status commands
Action	Set/Query
Syntax	`COME(?)[`*n*`]{`*m*`}`
Description	Set (query) the Communications Enable register [bit-mask n] {to bit-mask m}. The set-form command will clear the bits outside the bit-mask.
Power-on value	0
Example	`COME 1`
Related commands	COMS

OVLS (Overload Status)

Group	Status commands
Action	Query only
Syntax	`OVLS? [`*n*`]`
Description	Read the Overload Status register [bit-mask n].
Power-on value	0
Example	`OVLS?`
	2
Related commands	OVLE, OVLC.

OVLE (Overload Enable)

Group	Status commands
Action	Set/Query
Syntax	`OVLE(?)[`*n*`]{`*m*`}`
Description	Set (query) the Overload Enable register [bit-mask n] {to bit-mask m}. The set-form command will clear the bits outside the bit-mask.
Power-on value	0
Example	`OVLE 2`
Related commands	OVLS, OVLC.

OVLC (Overload Condition)

Group	Status commands
Action	Query only
Syntax	`OVLC? [`*n*`]`
Description	Read the Overload Condition register [bit-mask *n*].
	The values of the bits in the OVLC condition register are determined by the current (real-time) condition of the events defined in the OVLS status register.
	Reading the condition register does not affect the register.
Power-on value	0
Example	`OVLC?`
	2
Related commands	OVLS, OVLE.

INSS (Instrument Status)

Group	Status commands
Action	Query only
Syntax	`INSS? [`*n*`]`
Description	Read the Instrument Status register [bit-mask *n*].
Power-on value	0
Example	`INSS?`
	1
Related commands	LINS, INSE, INSC.

INSE (Instrument Enable)

Group	Status commands
Action	Set/Query
Syntax	`INSE(?)[`*n*`]{`*m*`}`
Description	Set (query) the Instrument Enable register [bit-mask n] {to bit-mask m}. The set-form command will clear the bits outside the bit-mask.
Power-on value	0
Example	`INSE 2`
Related commands	LINS, INSS, INSC.

INSC (Instrument Condition)

Group	Status commands
Action	Query only
Syntax	`INSC? [`*n*`]`
Description	Read the Instrument Condition register [bit-mask n]. The values of the bits in the INSC condition register are determined by the current (real-time) condition of the events defined in the INSS status register. Reading the condition register does not affect the register.
Power-on value	0
Example	`INSC?` *2*
Related commands	LINS, INSE, INSS.

LINS (Last Instrument Error Status)

Group	Status commands
Action	Query only
Syntax	`LINS?`
Description	Query the last execution instrument error. LINS? returns the unique code number associated with this event.
Valid codes are	0 ⟷ no execution error since last LINS?
Power-on value	0
Example	`LINS?` *0*
Related commands	INSC, INSE, INSS.

2.5 Interface commands

The Interface commands provide control over the interface between the instrument and the host computer.

*RST (Reset)

Group	Interface commands
Action	Set only
Syntax	`*RST`
Description	Reset the instrument to its default configuration. When a parameter is affected by the *RST command, its value is reset according to the information given by the Reset value field within the related command section. Whereas status registers are unaffected by *RST, the content of some conditions registers may have been modified upon resetting the intrument.
Example	`*RST`
Related commands	*RCL, *SAV.

*OPC (Operation Complete)

Group	Interface commands
Action	Set/Query
Syntax	`*OPC(?)`
Description	Set the OPC flag in the EVTS register.
	The query form *OPC? returns 1 when complete, but does not affect the EVTS register.
Example	`*OPC?`
	1
Related commands	

CONS (Console Mode)

Group	Interface commands
Action	Set/Query
Syntax	`CONS(?){`*n*`}`
Description	Set (query) the Console mode {to n}. When $n = 0$ (resp. 1), the console mode is disabled (resp. enabled). `CONS 1` causes each character received to be returned to the host computer.
Allowed range	$n \in \{\mathbf{0}, 1\}$
Reset value	**0**
Power-on value	0
Example	`CONS 1` *1*
Related commands	

*IDN (Identify)

Group	Interface commands
Action	Query only
Syntax	`*IDN?`
Description	Read the device identification string. This string is formatted as: Signals and Systems for Physics, model SK657, hw Rppx, fw Rqqy, s/n dddddd. In this string, SK657 is the model number, Rnnx and Rppy are revision numbers identifying the hardware or the firmware versions and dddddd refers to the 6-digit serial number.
Example	`*IDN?` *`Signals and Systems for Physics, model SK657, hw R24A, fw R24A, s/n 12356.`*
Related commands	

LURQ (Last User Request Status)

Group	Interface commands
Action	Query only
Syntax	`LURQ?`
Description	Query the last user's request. `LURQ?` returns the unique code number associated with this event.
Valid codes are	0 ⟷ no user request since last `LURQ?`; 1 ⟷ front-panel switch : output enable; 2 ⟷ front-panel switch : output disable.
Power-on value	0
Example	`LURQ?` *1*
Related commands	

LCMD (Last Command Error Status)

Group	Interface commands
Action	Query only
Syntax	`LCMD?`
Description	Query the last command error. `LCMD?` returns the unique code number associated with this event :
Valid codes are	0 ⟷ no execution error since last `LCMD?` 1 ⟷ illegal (unknown) command; 2 ⟷ illegal query; 3 ⟷ illegal set (read-only command); 4 ⟷ extra parameter; 5 ⟷ missing parameter; 6 ⟷ null command.
Power-on value	0
Example	`*RST?;LCMD?` *2*
Related commands	

LEXE (Last Execution Error Status)

Group	Interface commands
Action	Query only
Syntax	`LEXE?`
Description	Query the last execution error. `LEXE?` returns the unique code number associated with this event :
Valid codes are	0 ⟷ no execution error since last `LEXE?`; 1 ⟷ invalid parameter; 2 ⟷ argument value out-of-range; 3 ⟷ some parameters have been adapted or clamped; 4 ⟷ a conflict due to the current operation has been avoided; 5 ⟷ no change upon executing the command; 6 ⟷ the operation was aborted due to a fault condition.
Power-on value	0
Example	`CONS2;LEXE?;LEXE?` *1* *0*
Related commands	

TERM (Response Termination)

Group	Interface commands
Action	Set/Query
Syntax	`TERM(?){`n`}`
Description	Set (query) the termination sequence {to n}.
	The termination sequence is appended to all query responses sent by the instrument. It is constructed of ASCII character(s) `<CR>` (carriage return) or `<LF>` (line feed).
Allowed range	$n \in \{1, 2, 3, 4\}$ where :
	$1 \longleftrightarrow$ `<CR>` character appended;
	$2 \longleftrightarrow$ `<LF>` character appended;
	$3 \longleftrightarrow$ both `<CR>` and `<LF>` characters appended;
	$4 \longleftrightarrow$ no character appended.
Reset value	**3**
Power-on value	3
Example	`TERM?`
	3
Related commands	

2.6 Memory commands

The Memory commands allow the User to save and recall the instrument's settings in non-volatile memory.

*RCL (Recall Settings)

Group	Memory commands
Action	Set only
Syntax	`*RCL`
Description	Recall the settings stored in the non-volatile memory.
Example	`*RCL`
Related commands	*RST, *SAV.

*SAV (Save Current Settings)

Group	Memory commands
Action	Set only
Syntax	`*SAV`
Description	Save the current settings in the non-volatile memory.
Example	`*SAV`
Related commands	*RCL, *RST.

3 Status Model

The complete block diagram of the status register array is available online at the related product page. There are four categories of registers in this model :

Last Event registers These four read registers (LINS, LCMD, LURQ and LEXE) store the last event that they monitor. A query command i) return the last registered event since the previous query and ii) clears the register's content.

Condition registers These read-only registers correspond to the real-time condition of some underlying physical properties under monitoring. Queries return the latest value of the property, and have no other effect.
Condition register names end with C.

Status registers These read-only registers record the occurrence of defined events. If the event occurs, the corresponding status bit is set to 1. Upon querying a status register, any set bits within it are cleared. These are sometimes known as sticky bits since once set, a bit can only be cleared by reading its value. Status register names end with S.

Enable registers These read/write registers define a bitwise mask for their corresponding status register. If any bit position is set in a status register while the same bit position is also set in the enable register, then the corresponding summary bit is set in either the Event Summary or Master Summary register. Enable register names end with E.

3.1 Master Summary Status (MSTS)

The Master Summary Status (MSTS) is the top-level summary register of the status model. When masked by the Master Summary Status Enable (MSTE) register, a bit set in the Status Byte causes the /STATUS signal to be asserted on the DIN41612 connector. This register is queried with the MSTS?[n] command.

Weight $n = 2^i$	**Bit** i	**Flag**	**Description**
1	0	MSS	Master Summary Status. Indicates whether one or more of the enabled status messages in the Status Byte register is true.
2	1	RFU	Undefined (read 0).
4	2	RFU	Undefined (read 0).
8	3	RFU	Undefined (read 0.)
16	4	COM	Communication Summary Bit. Indicates whether one or more of the enabled flags in the Communication Status register has become true.
32	5	EVT	Event Summary Bit. Indicates whether one or more of the enabled flags in the Event Status register is true.
64	6	INS	Instrument Summary Bit. Indicates whether one or more of the enabled flags in the Instrument Status register is true.
128	7	OVL	Overload Summary Bit. Indicates whether one or more of the enabled flags in the Overload Status register is true.

3.2 Master Summary Enable (MSTE)

Each bit in the MSTE register corresponds one-to-one with a bit in the MSTS register, and acts as a bitwise AND of the MSTS flags to generate the MSS flag. Bit 0 of the MSTE is undefined—setting it has no effect, and reading it always returns 0. This register is set and queried with the MSTE(?) command and cleared at power-on.

3.3 Event Status (EVTS)

The Event Status register consists of 8 event flags. These flags are set by the corresponding event, and cleared only by reading or with the *CLS command ("sticky bits"). Querying the single bit i with the command EVTS? n where the bit-mask $n = 2^i$ will only clear the bit i. For instance, issuing the command `EVTS?128` will clear the bit 7 (INS) only.

Weight $n = 2^i$	**Bit** i	**Flag**	**Description**
1	0	PON	Power On event. Indicates that an off-to-on transition has occurred.
2	1	OPC	Operation Complete. Set by the *OPC command.
4	2	CMD	Command Error event. Indicates an error detected by the command parser. The error code can be queried with LCMD?
8	3	EXE	Execution Error event. Indicates an error in a command that was successfully parsed. The error code can be queried with LEXE?
16	4	RXQ	Reception Buffer event. Indicates that the RX buffer has been cleared.
32	5	TXQ	Transmission Buffer event. Indicates that the TX buffer has been cleared.
64	6	URQ	User Request event. Indicates that a User request has occured. The request code can be queried with LURQ?
128	7	INS	Instrument event. Indicates whether one or more of the enabled flags in the Instrument Status register is true.

3.4 Event Enable (EVTE)

Each bit in the EVTE register corresponds one-to-one with a bit in the EVTS register, and acts as a bitwise AND of the EVTS flags to generate the EVT flag in the Master Summary Status (MSTS) register. This register is set and queried with the EVTE command and cleared at power-on. For instance, issuing the command `EVTE 128` enables the bit 7 (INS) only.

3.5 Communication Status (COMS)

The Communication Status register consists of 8 event flags. These flags are set by the corresponding event, and cleared only by reading or with the *CLS command ("sticky bits"). Querying the single bit i with the command COMS? n where the bit-mask $n = 2^i$ will only clear the bit i.
Because the COMS register is not used here, querying this register always returns 0. Therefore, the corresponding summary bit in the MSTS register (bit COM) is never set whatever the value of the COME register.

Weight	**Bit**	**Flag**	**Description**
$n = 2^i$	i		
1	0	RFU	Undefined (read 0).
2	1	RFU	Undefined (read 0).
4	2	RFU	Undefined (read 0).
8	3	RFU	Undefined (read 0).
16	4	RFU	Undefined (read 0).
32	5	RFU	Undefined (read 0).
64	6	RFU	Undefined (read 0).
128	7	RFU	Undefined (read 0).

3.6 Communication Enable (COME)

Each bit in the COME register corresponds one-to-one with a bit in the COMS register, and acts as a bitwise AND of the COMS flags to generate the COM flag in the Master Summary Status (MSTS) register. This register is set and queried with the COME command and cleared at power-on. *Because the COMS register is not used, its value is always 0. Therefore, the corresponding summary bit in the MSTS register (COM bit) is never set whatever the value of the enable register (COME).*

3.7 Instrument Status (INSS)

The Instrument Status register consists of 8 event flags. These flags are set by the corresponding event, and cleared only by reading or with the *CLS command ("sticky bits"). Querying the single bit i with the command INSS? n where the bit-mask $n = 2^i$ will only clear the bit i. For instance, issuing the command `INSS?32` will clear the bit 5 (IPWR) only.

Weight $n = 2^i$	Bit i	Flag	Description
1	0	STAB	Laser Current Stable event. Indicates that the ramping sequence is ended.
2	1	RFU	Undefined (read 0).
4	2	ILKO	Interlock event. Indicates an opened switch has been detected. This bit is automatically set until the switch is closed.
8	3	RFU	Undefined (read 0).
16	4	XPWR	External Power Supplies event. Indicates that an external power supply has crossed its undervoltage trip point.
32	5	IPWR	Internal Power Supplies event. Indicates that a internally generated power supply has crossed its undervoltage trip point. This bit is automatically set until the safe power level has been restored. This bit may have been set after a power-on sequence.
64	6	RFU	Undefined (read 0).
128	7	LDEN	Laser Enabled event. Indicates that the laser has been successfully connected to the current source.

3.8 Instrument Enable (INSE)

Each bit in the INSE register corresponds one-to-one with a bit in the INSS register, and acts as a bitwise AND of the INSS flags to generate the INS flag in the Master Summary Status (MSTS) register. This register is set and queried with the INSE command and cleared at power-on.

3.9 Instrument Condition (INSC)

Each bit in the INSC register corresponds one-to-one with a bit in the INSS register. The bits in the INSC register reflect the real-time values of their corresponding signals. Reading the entire register, or individual bits within it, does not affect the value of INSC. This register is queried with the INSC command and cleared at power-on.

3.10 Overload Status (OVLS)

The Overload Status register consists of 8 event flags. These event flags are set by the corresponding event, and cleared only by reading or with the *CLS command ("sticky bits"). Querying the single bit i with the command OVLS? n where the bit-mask $n = 2^i$ will only clear the bit i. For instance, issuing the command `OVLS?2` will clear the bit 1 (VCMP) only.

Weight $n = 2^i$	**Bit** i	**Flag**	**Description**
1	0	ILIM	Current Limitation event. Indicates a limitation of the laser current.
2	1	VCMP	Compliance Voltage event. Indicates that current source's compliance voltage has crossed its maximal allowed level.
4	2	RFU	Undefined (read 0).
8	3	RFU	Undefined (read 0).
16	4	RFU	Undefined (read 0).
32	5	RFU	Undefined (read 0).
64	6	RFU	Undefined (read 0).
128	7	RFU	Undefined (read 0).

3.11 Overload Enable (OVLE)

Each bit in the OVLE register corresponds one-to-one with a bit in the OVLS register, and acts as a bitwise AND of the OVLS flags to generate the OVL flag in the Master Summary Status (MSTS) register. This register is set and queried with the OVLE command and cleared at power-on.

3.12 Overload Condition (OVLC)

Each bit in the OVLC register corresponds one-to-one with a bit in the OVLS register. The bits in the OVLC register reflect the real-time values of their corresponding signals. Reading the entire register, or individual bits within it, does not affect the value of OVLC. This register is queried with the OVLC command and cleared at power-on.

3.13 Last Command Error (LCMD)

The LCMD register holds the last error detected by the command parser. The related error code can be retrieved by the command `LCMD?`. When such an error is detected, the corresponding bit in the Event Status register is set (bit CMD in EVTS).

3.14 Last Execution Error (LEXE)

The LEXE register holds the last error detected during the execution of a command. The related error code can be retrieved by the command `LEXE?`. When such an error is detected, the corresponding bit in the Event Status register is set (bit EXE in EVTS).

3.15 Last Instrument Error (LINS)

The LINS register holds the last error detected during the operation of the instrument. The related error code can be retrieved by the command `LINS?`. When such an error is detected, the corresponding bit in the Event Status register is set (bit INS in EVTS).

3.16 Last User Request (LURQ)

The LURQ register holds the last User's request. The related request code can be retrieved by the command `LURQ?`. When such a request is reported, the corresponding bit in the Event Status register is set (bit URQ in EVTS).

4 Index of commands

5 Document Revision History

5.1 Version Number

This document is identified by SK657-SU03-P25A.

5.2 Revision History

P24A – 2024-05-20

Initial version

P25A – 2025-12-16

Updated index

Updated address