SUTTER INSTRUMENT®

LAMBDA SC

SMARTSHUTTER CONTROL SYSTEM

EXTERNAL CONTROL QUICK REFERENCE

REV. 2.00B (20190606)

Controlling the Lambda SC externally via computer is accomplished by sending commands over the serial or USB interface between the computer and the equivalent connector on the rear panel of the Lambda SC controller.

Serial Ports: Refer to the main Lambda SC operation manual for descriptions on how to externally-control the instrument using the serial (RS-232, 9-pin DSUB connector) interface.

Table 1. RS-232 interface serial port settings.

Property	Setting
Data ("Baud") Rate (bits per second (bps))	9600
Data Bits	8
Stop Bits	1
Parity	None
Flow Control	None

USB Interface: The USB device driver for Windows is downloadable from Sutter Instrument's web site (www.sutter.com). The Lambda SC requires USB CDM (Combined Driver Model) Version 2.10.00 or higher. The CDM device driver for the Lambda SC consists of two device drivers: 1) USB device driver, and 2) VCP (Virtual COM Port) device driver. Install the USB device driver first, followed by the VCP device driver. The VCP device driver provides a serial RS-232 I/O interface between a Windows application and the Lambda SC. Although the VCP device driver is optional, its installation is recommended even if it is not going to be used. Once installed, the VCP can be enabled or disabled.

The CDM device driver package provides two I/O methodologies over which communications with the Lambda SC can be conducted: 1). USB Direct, or 2). Serial RS-232 asynchronous via the VCP device driver. The first method requires that the VCP device driver be disabled (or not installed). The second method requires that the VCP be installed and enabled.

Virtual COM Port (VCP) Serial Port Settings: The following table lists the required RS-232 serial settings for the COM port (COM3, COM5, etc.) generated by the installation of the VCP device driver.

Table 2. USB-VCP interface serial port settings.

Property	Setting
Data ("Baud") Rate (bits per second (bps))	128000 or 9600
Data Bits	8
Stop Bits	1
Parity	None
Flow Control	None

The settings shown in the above table can be set in the device driver's properties (via the Device Manager if in Windows) and/or programmatically in your application.

Serial Port vs. USB-VCP (Virtual Serial) Port vs. USB (D2XX): External communications with the Lambda SC can be accomplished in one of three ways,

1. RS-232 DB9 connector via 9-pin cable to a serial port on the computer.
2. USB-VCP USB connector via USB cable to the computer's USB bus, with the USB VCP device driver installed and enabled. The same RS-232 protocol is used with the VCP interface as over the 9-pin cable to the RS-232 port.
3. USB (Direct) via USB cable to the computer, with USB VCP device driver disabled or not installed.

<*>

Handshaking: Command sequences do not have terminators. If a command sequence just sent to the controller is determined to be valid, the entire sequence is immediately echoed back to the computer. All commands return an ASCII CR (Carriage Return; 13 decimal, 0D hexadecimal) to indicate that the task associated with the command has completed. When the Lambda SC completes the task associated with a command, it sends ASCII CR back to the host computer indicating that it is ready to receive a new command. If a command returns data, the last byte returned is the task-completed indicator.

Commands: Each command sequence consists of at least one byte, the first of which is the "command byte". Those commands that have parameters or arguments require a sequence of bytes that follow the command byte. No delimiters are used between command sequence arguments. Every command and command sequence ends with a terminator byte containing an ASCII CR (13 decimal, 0D hexadecimal). Although most command bytes can be expressed as ASCII displayable/printable characters, the rest of a command sequence must always be expressed as a sequence of unsigned byte values (0-255 decimal; 00 – FF hexadecimal, or 00000000 – 11111111 binary). Each byte in a command sequence being transmitted to the controller must contain an unsigned binary value. Attempting to code command sequences as "strings" is not advisable. Any command data being returned from the controller must also be received and initially treated as a sequence of unsigned byte values. Groups of contiguous bytes can later be combined to form larger values, as appropriate (e.g., 2 bytes into 16-bit "word" or "short", or 4 bytes into a 32-bit "long" or "double word"). For the Lambda SC, all Galvo DAC values are stored as "unsigned short" (16-bit) values. A 16-bit value is transmitted and received to and from the controller as two contiguous bytes.

"Unsigned" means the value can only be positive; negative values are not permitted. A U16 consists of two contiguous bytes, with a byte/bit-ordering format of Little Endian ("Intel") (most significant byte (MSB) in the first byte and least significant (LSB) in the last byte). If the platform on which your application is running is Little Endian, then no byte order reversal of galvo DAC values is necessary. Examples of platforms using Little Endian formatting include any system using an Intel processor (including Microsoft Windows and Apple Mac OS X), and most Linux distributions running on Intel/AMD processor based systems.

If the platform on which your application is running is "Big Endian" ("Motorola"), then these U16 position values must have their bytes reverse-ordered after receiving from, or before sending to, the Lambda SC. Examples of Big Endian platforms include most all non-Intel-based systems, LabVIEW (regardless of system & operating system), and Java (programming language/environment).

Command Reference: The following table lists all the external-control commands for the Lambda SC. The commands apply equally to the parallel, serial, and USB ports.

Table 3. Commands.

Dec.	Hex.	Bin.	Description
0 - 169	00 - A9	00000000 - 10101001	
170	AA	10101010	Open Shutter
171	AB	10101011	
172	AC	10101100	Close Shutter
185 - 190	AD - BE	10101101 - 10111110	
191	BF	10111111	Stop Free Run
192 - 203	C0 – CB	11000000 - 11001011	
204	CC	11001100	Get status
205	CD	11001101	
206	CE	11001110	Power On all motors
207	CF	11001111	Power Off all motors
208 - 219	D0 – DD	11000000 - 11001011	
220	DC	11001100	Set Fast mode
221	DD	11001101	Set Soft mode
222	DE	11001110	Set Neutral density mode (number of microsteps (1 – 144) is specified in the next byte that follows this command)
223 - 237	DF - ED	11011111 - 11101101	
238	EE	11101110	Set controller to On Line
239 - 249	EF – F9	11101111 - 11111001	

Dec.	Hex.	Bin.	Description				
250	FA	11111010	**Timer, TTL control, configuration (2- or more-byte) commands:**				
			Followed by:				**Description:**
			Num. Of bytes	**Sub-Command (2nd Byte)**			
				Dec.	**Hex.**	**Bin.**	
				0 – 15	00 – 0F	00000000 - 00001111	
			5	16 - 22	10 - 15	00010000 - 00010101	Set Delay Timer for 0 – 5 hours (hours specified in lower nibble (4 bits)), followed by 4 bytes specifying (in order) minutes (0 – 59), seconds (0 – 59), and several millisecond and sub-millisecond multipliers. See "Command byte stream structure for timers" table for details of how this timer is set.
				23 - 31	16 – 1F	00010110 - 00011111	
			5	32 - 37	20 - 25	00100000 - 00100101	Set Exposure Timer for 0 – 5 hours (hours specified in lower nibble (4 bits)), followed by 4 bytes specifying (in order) minutes (0 – 59), seconds (0 – 59), and several millisecond and sub-millisecond multipliers. See "Command byte stream structure for timers" table for details of how this timer is set.
				33 – 159	26 – 90	00100001 - 10010000	
			1	160	A0	10100000	Trigger Pulse/TTL IN disconnected
			1	161	A1	10100001	Trigger on High: Shutter is normally closed; is open while TTL IN is high.
			1	162	A2	10100010	Trigger on Low: Shutter is normally open; is closed while TTL IN is low.
			1	163	A3	10100011	Toggle on Rising Edge: On the rising edge of TTL IN going high, shutter opens if closed and closes if open.
			1	164	A4	10100100	Toggle on Falling Edge: On the falling edge of TTL IN going low, shutter closes if open and opens if closed. **Please refer to note below.**
				165 – 175	A5 – AF	10100101 – 10101111	
			1	176	B0	10110000	No Sync out
			1	177	B1	10110001	Sync out high for Open Shutter
			1	178	B2	10110010	Sync out low for Open Shutter
				179 – 191	B3 - BF	10110011 - 10111111	
			1	192	C0	11000000	Restore factory default configuration
			1	193	C1	11000001	Save the current configuration to the controller
				194 – 239	C2 – EF	11000010 - 11101111	
			3	240	F0	11110000	Set repeat cycle numbers for Free Run. Followed by two bytes containing the number of repeat cycles. Both bytes are treated as a single 16-bit "word" type, thereby allowing from 0 to 65,535 possible values. Any value over 65,000 (65,001 – 65,535), however, sets the Free Run to continuous (cycles repeat forever until the Stop Free Run command is issued).

Dec.	Hex.	Bin.	Description				
			1	241	F1	11110001	Free Run when power on (repeat number not needed)
			1	242	F2	11110010	Free Run on trigger pulse (repeat number must be set first)
			1	243	F3	11110011	Free Run to Go (repeat number must be set first)
				244 – 255	F4 – FF	11110100 - 11111111	
251	FB	11111011	Reset to latest saved configuration on controller				
252	FC	11111100					
253	FD	11111101	Get controller type & configuration				
254 - 255	FE - FF	11111110 - 11111111					

NOTE: The "TTL IN Falling Edge Triggers SmartShutter to Toggle Open/Close" command exists in versions of the Lambda SC programmed with Version 1.08 or later of the firmware. Before attempting to use this command, it is strongly advised that the version of the firmware with which the connected Lambda SC is programmed is first checked by issuing the "Get Controller Type and Configuration" command.

NOTES:

1. A short delay (usually around 1 ms) is recommended between commands (after the reception of one command and the sending of the next command).
2. All values greater than one byte (8 bits) transmitted to, and received from, the LAMBDA 421 consist of two bytes ordered in "Little Endian" (least significant byte last) format (see NOTE 3). The value stored in these two bytes is always unsigned, meaning that the value will always be positive (negative values are not allowed). These two bytes are converted to and from 16-bit "unsigned short" (C/C++) or "U16" (LabVIEW) value storage entities.
3. "Little Endian" means that the least significant byte is last (last to send and last to receive). Byte-order reversal may be required on some platforms (e.g., LabVIEW always handles "byte strings" in "Big Endian" byte order, requiring that the two bytes containing a DAC value be reverse-ordered before conversion into a 16-bit "unsigned short" (or "word") value). Whereas Microsoft Windows, Intel-based Apple Macintosh systems running Mac OS X, and some Intel/AMD processor based Linux distributions handle byte storage in Little-Endian byte order so byte reordering is not necessary before converting to/from 16-bit "short" or "word" values.

Table4. Timer command time encoding

Byte Offset	**Byte Value** (Decimal, hexadecimal, & binary)	**Upper & Lower Nibble Values** (Decimal, hexadecimal, & binary)	**Description**
0	250 FA 1111 1010	N/A	Lambda SC Special Command.
1	10 – 15, 20 – 25 0001 0000 – 0001 0101. 0010 0000 – 0010 0101	1 - 2 1 - 2 0001 - 0010	1 = Set Delay Timer 2 = Set Exposure Timer
		0 – 5 0 – 5 0000 - 0101	Hours (0 through 5) *
2	0 – 59 00 – 3B 00000000 - 00111011	N/A	Minutes (0 through 59)
3	0 – 59 00 – 3B 00000000 - 00111011	N/A	Seconds (0 through 59)
4	0 – 9, 16 – 25, 32 – 41, 48 – 57, 64 – 73, 80 – 89, 96 – 105, 112 – 121, 128 – 137, 144 – 153. 00 – 09, 10 – 19, 20 – 29, 30 – 39, 40 – 49, 50 – 59, 60 – 69, 70 – 79, 80 – 89, 90 - 99 0000 0000 – 0000 1001, 0001 0000 – 0001 1001 0010 0000 – 0010 1001, 0011 0000 – 0011 1001 0100 0000 – 0100 1001, 0101 0000 – 0101 1001 0110 0000 – 0110 1001, 0111 0000 – 0111 1001 1000 0000 – 1000 1001, 1001 0000 – 1001 1001	0 – 9 0 – 9 0000 - 1001	Milliseconds: 100s digit (0 through 9)
		0 – 9 0 – 9 0000 - 1001	Milliseconds: 10s digit (0 through 9)
5	0 – 9, 16 – 25, 32 – 41, 48 – 57, 64 – 73, 80 – 89, 96 – 105, 112 – 121, 128 – 137, 144 – 153. 00 – 09, 10 – 19, 20 – 29, 30 – 39, 40 – 49, 50 – 59, 60 – 69, 70 – 79, 80 – 89, 90 - 99 0000 0000 – 0000 1001, 0001 0000 – 0001 1001 0010 0000 – 0010 1001, 0011 0000 – 0011 1001 0100 0000 – 0100 1001, 0101 0000 – 0101 1001 0110 0000 – 0110 1001, 0111 0000 – 0111 1001 1000 0000 – 1000 1001, 1001 0000 – 1001 1001	0 – 9 0 – 9 0000 - 1001	Milliseconds: 1s digit (0 through 9)
		0 – 9 0 – 9 0000 - 1001	Milliseconds: 0.1s digit (0 through 9)

NOTE: If hours are set to 5, then all other time fields must be set to zero. In other words, 5 hrs 0 min. 0 sec. 0.0 ms is the maximum time for the timer commands (or 4 hrs., 59 min., 59 sec., 999.9 ms PLUS 0.1 ms).

Table 5. Status command return codes and data.

Order	Category	Byte Offset	Value (Decimal, hexadecimal, & binary)	Description
1	**Command echo** (1 byte)	0	204 CC 11001100	The Status command byte code echoed back.
2	**Shutter Open/Closed State** (1 byte)	1	170 AA 10101010	**Open:** Shutter is in the opened state.
			172 AC 10101100	**Closed:** Shutter is in the closed state.
3	**Shutter Mode** (1 byte)	2	219 DB 11011011	**Not Connected:** Indicates that no SmartShutter is connected.
			220 DC 11011100	**Fast:** Indicates that the SmartShutter is in fast mode.
			221 DD 10111011	**Soft:** Indicates that the SmartShutter is in soft mode.
			222 DE 10111100	**Neutral Density:** Indicates that the SmartShutter is in neutral-density mode. A second byte follows with number of microsteps (1 – 144).
		NOTE: From this point onwards in this table, byte numbers shown in parenthesis are true only if Shutter Mode (Byte 3) = Neutral Density (222 decimal (DE hexadecimal)).		
4	**Neutral Density Microsteps** (1 byte)	(3)	1 – 144 01 – 90 00000001 - 10010000	**Neutral Density Microsteps:** Contains the number of microsteps (1 through 144) for the SmartShutter's neutral density mode. **Note that this byte is present (and at this position) in Status structure only if the Shutter Mode (Byte 3) contains the value for Neutral Density (222 decimal (DE hexadecimal)).**
5	**Lead-in byte for Lambda SC Special commands** (1 byte)	3/(4)	250 FA 11111010	Lead-in byte for all Lambda SC-specific special commands (TTL IN, TTL OUT, Timers, and Free Run). This lead-in byte is expressed only once in the status return data structure, and this is where it occurs.
6	**TTL IN Setting** (1 byte)	4/(5)	160 A0 10100000	**Disabled**: Trigger/toggle control of the shutter via TTL IN is turned off.
			161 A1 10100001	**Trigger on High:** Shutter is normally closed; opens while high signal is present on TTL IN.

Order	Category	Byte Offset	Value (Decimal, hexadecimal, & binary)	Description
			162 A2 10100010	**Trigger on Low:** Shutter is normally open; closes while high signal is not present on TTL IN.
			163 A3 10100011	**Toggle on Rising Edge:** On the rising edge of TTL IN going high, shutter opens if closed and closes if open.
			164 A4 10100100	**Toggle on Falling Edge:** On the falling edge of TTL IN going low, shutter closes if open and opens if closed.
7	**TTL OUT Setting** (1 byte)	5/(6)	176 B0 10110000	**Disabled:** TTL OUT synch signal is disabled.
			177 B1 10110001	**High on Shutter Open:** TTL OUT is set to high when shutter opens.
			178 B2 10110010	**Low on Shutter Open:** TTL OUT is set to low when shutter opens.
8	**Delay Timer** (Time to when shutter opens; 5 bytes)	6/(7)	0 or 1 0x or 1x 0000xxxx or 0001xxxx	**Enabled/Disabled (Upper Nibble):** If 0 (disabled), all five Delay Timer bytes can be ignored. If 1 (enabled), remaining five bytes (and respective nibbles where applicable) may contain values for hours, minutes, seconds, and milliseconds.
			0 – 5 x0 – x5 xxxx0000 – xxxx0101	**Hours (Lower Nibble):** Hours (0 – 5).
		7/(8)	0 – 59 00 – 38 00000000 - 00111011	**Minutes:** (0 – 59)
		8/(9)	0 – 59 00 – 38 00000000 - 00111011	**Seconds:** (0 – 59)
		9/(10)	0 – 9 0x – 9x 0000xxxx – 1001xxxx	**Milliseconds 100s Digit (Upper Nibble):** 0 – 9
			0 – 9 x0 – x9 xxxx0000 –xxxx1001	**Milliseconds 10s Digit (Lower Nibble):** 0 – 9.
		10/(11)	0 – 9 0x – 9x 0000xxxx – 1001xxxx	**Milliseconds 1s Digit (Upper Nibble):** 0 – 9.

Order	Category	Byte Offset	Value (Decimal, hexadecimal, & binary)	Description
			0 – 9 x0 – x9 xxxx0000 –xxxx1001	**Milliseconds 0.1s Digit (Lower Nibble):** 0 – 9.
9	**Exposure Timer** (Time during which the shutter remains open; 5 bytes)	11/(12)	0 or 1 0x or 1x 0000xxxx or 0001xxxx	**Enabled/Disabled (Upper Nibble):** 0 (disabled) or 1 (enabled). If disabled, all five Exposure Timer bytes can be ignored. Otherwise, the remaining five bytes (and respective nibbles where applicable) may contain values for hours, minutes, seconds, and milliseconds.
			0 – 5 x0 – x5 xxxx0000 – xxxx0101	**Hours (Lower Nibble):** (0 – 5).
		12/(13)	0 – 59 00 – 38 00000000 - 00111011	**Minutes**: (0 – 59)
		13/(14)	0 – 59 00 – 38 00000000 - 00111011	**Seconds**: (0 – 59)
		14/(15)	0 – 9 0x – 9x 0000xxxx – 1001xxxx	**Milliseconds 100s Digit (Upper Nibble):** 0 – 9.
			0 – 9 x0 – x9 xxxx0000 –xxxx1001	**Milliseconds 10s Digit (Lower Nibble):** 0 – 9.
		15/(16)	0 – 9 0x – 9x 0000xxxx – 1001xxxx	**Milliseconds 1s Digit (Upper nibble):** 0 – 9.
			0 – 9 x0 – x9 xxxx0000 –xxxx1001	**Milliseconds 0.1s Digit (Lower Nibble):** 0 – 9.
10	**Free Run** (3 bytes)	16/(17)	241 F1 11110001	**Run On Power On**
			242 F2 11110010	**Run On Trigger Pulse**
			243 F3 11110011	**Run Now**

Order	Category	Byte Offset	Value (Decimal, hexadecimal, & binary)	Description
		17/(18)	(Upper Byte) 0 - 255 00 – FF 00000000 - 11111111	**Number of repeat cycles for Free Run**. These two contiguous bytes combined (joined) contain the number of repeat cycles as a 16-bit integer ranging from 0 to 65,000. Above 65,000 (65,001 to 65,535), Free Run is set up for Continuous operation (i.e., infinite repeat cycles).
		18/(19)	(Lower Byte) 0 - 255 00 – FF 00000000 - 11111111	
11	**Terminator** (1 byte)	19/(20)	13 0D 00001101	**Terminator:** ASCII CR (Carriage Return) used to indicate the end of the status data.

Table 6. "Get Controller Type and Configuration" command return codes and data

Total num. bytes	Description					
	Category	Subcategory	Length (in bytes)	Byte Offset	Possible values	
					ASCII string	Meaning
14	**Command echo back**		1	0	ý	253 decimal; FD hexadecimal. *
	Controller Type and firmware version	**Controller Type**	2	1	SC	Lambda SC, with firmware version V and subversion SS (e.g., “SC-v1.05”).
		Filler text	2	3	-v	
		Firmware version	4	5	V.SS	
	Shutter Type		4	9	S-IQ	*SmartShutter*
	Command return data terminator		1	13		ASCII carriage return; 13 decimal, 0D hexadecimal.

***NOTE: The character shown in the “ASCII string” column for the command echo is a typical visual representation of the byte value 253 decimal (FD hexadecimal) on both Windows and Linux platforms. However, other platforms may display a different character or nothing at all. For the command return data terminator (ASCII carriage return (13 decimal, 0D hexadecimal), generally no character will be displayed, although the carriage return is acted upon in most cases in text-based console programs.**

NOTES:

NOTES: