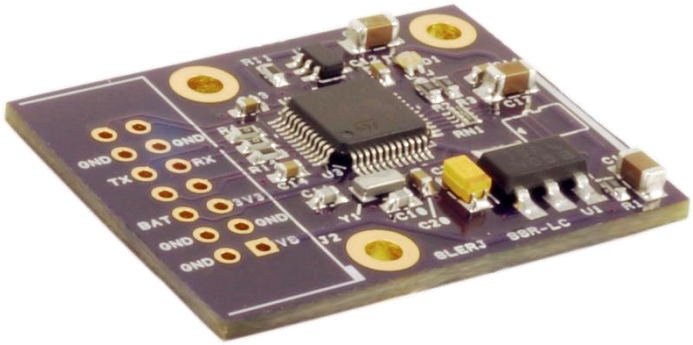
SLERJ



SSR-LC User's Manual

Revision E

2 September 2019

CONTENTS

Warranty

The SSR-LC Serial Data Recorder is warranted against defects in materials and manufacturing for a period of 90 days from the date of purchase. In the event of a product failure due to materials or workmanship, Slerj will, at its discretion, repair or replace the product. For warranty service, return the defective produce to Slerj, shipping prepaid, for prompt repair or replacement. Slerj, its suppliers, and its licensors shall in no event be liable for any damages arising from the use of or inability to use this product. This includes business interruption, loss of business information, or other loss which may arise from the use of this product.

SLERJ PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF SLERJ PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE UNDERTAKEN SOLELY AT THE CUSTOMER'S OWN RISK. Should a customer purchase or use Slerj products for any such unauthorized application, the customer shall indemnify and hold Slerj and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs damages and attorney fees which could arise.

1 Introduction

1.1 Description

The SSR-LC is a low cost addition to the Slerj serial data logger product line. It records an asynchronous serial channel at up to 230400 baud onto removable microSD media and provides most of the advanced features of the 3-channel Slerj SSR-1. It is available in both 5V TTL compatible and RS-232 versions. Streams can be recorded automatically at power up, on command through a digital or PWM input, or using an interactive shell interface.

1.2 Features

- Up to 230.4k baud recording
- RS-232 or 5V TTL compatible (order option)
- Powered from supply of 4.5–15 VDC or directly from 3.3 VDC
- Small size: 1.40 x 1.10 x 0.20 inches (36 x 28 x 5 mm) (non-header version)
- Supports microSD, microSDHC, and microSDXC cards
- exFAT, FAT32, FAT16, and FAT12 file system support
- Long File Name support
- Support for raw and time-tagged recording
- Battery backed real-time clock (with externally supplied battery voltage)
- Flexible record control: digital input, PWM input, software controlled, or automatic
- Interactive shell for configuration and file system operations
- Flexible recording modes (overwrite/append, user defined path and file names, etc.)

2 Getting Started

2.1 Device Versions

The SSR-LC is available in the following configurations:

SSR-LC-TS

- S = no header (solder pads)
  H = 14 pin header, dual row, 0.100 inch (2.54 mm) pitch
- T = 5V TTL compatible (3.3V CMOS)
  R = RS-232

2.2 The SSR-LC Hardware

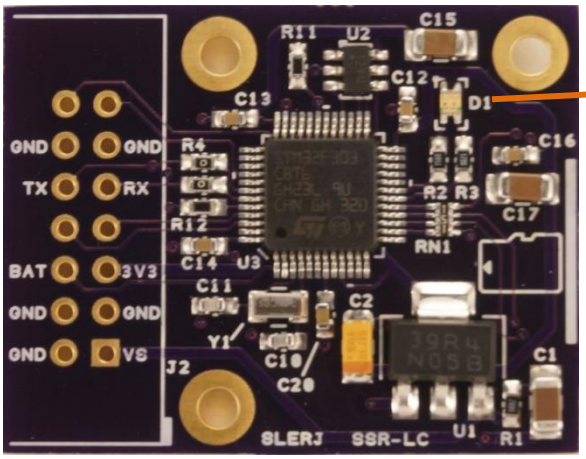


Figure 1. Top View (SSR-LC-TS shown)

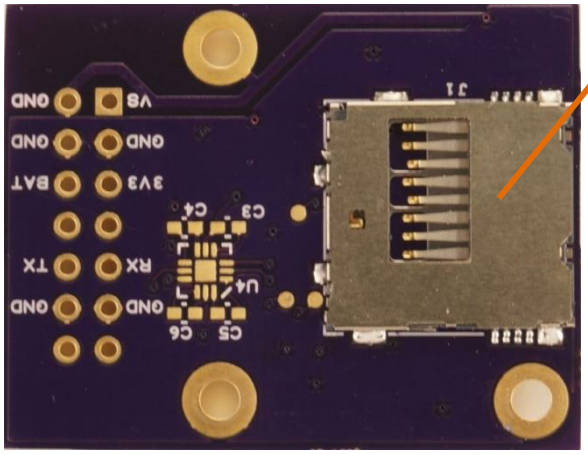


Figure 2. Bottom View (SSR-LC-TS shown)

CAUTION: Like most electronic components, the SSR-LC can be damaged by electrostatic discharge. Observe typical precautions for handling electrostatic discharge sensitive devices.

2.3 Connecting the SSR-LC

The pins of the main connector are described in Table 1. The SSR-LC is highly configurable and has a variety of settings that can be changed through a shell interface (see Section 4 Interactive Shell) and stored in on-board, non-volatile memory. Typical connections are shown in Figure 3 and Figure 4. In the default configuration (as shipped), the SSR-LC records when the PWM/digital input command pin (PDI, pin 13) is connected to ground and the shell is not active (pin 7 is allowed to float high). To access the SSR-LC interactive shell, tie pin 7 to ground. Default serial parameters are 115200 baud, 8 data bits, no parity, and 1 stop bit. While the shell is active (pin 7 is low), all received bytes goes to the shell and no data is recorded.

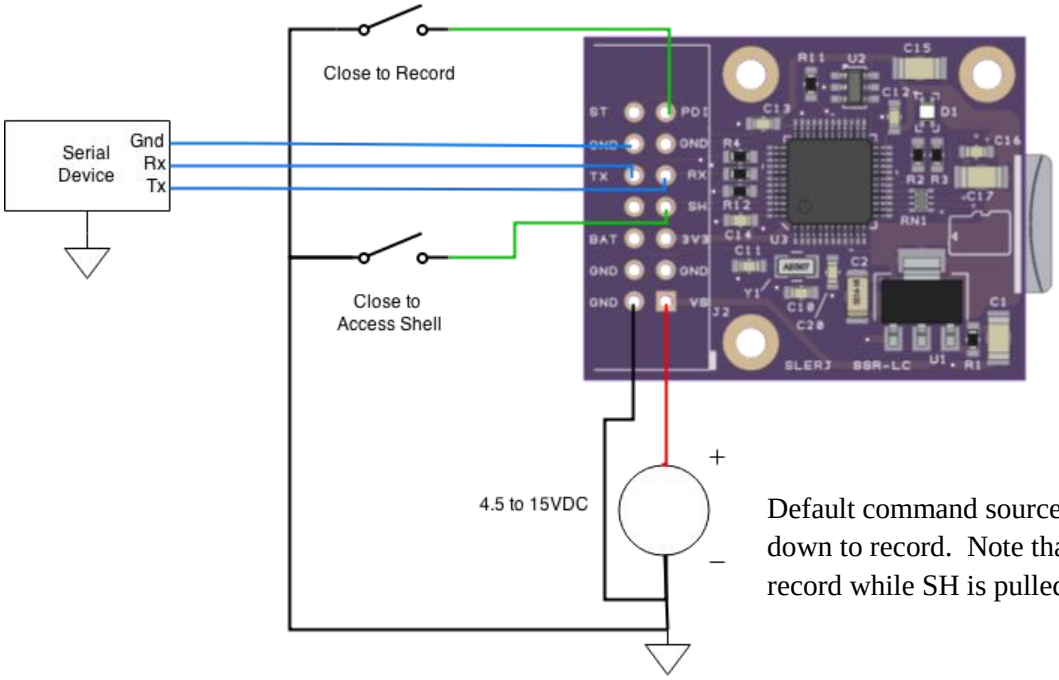
WARNING: SSR-LC-R device pins 9 and 10 are RS-232 voltage level compatible. SSR-LC-T device pins 9 and 10 are 5 Volt tolerant, TTL compatible, 3.3V CMOS. Do not connect an RS-232 device to pins 9 and 10 of an SSR-LC-T. Note that the RS-232 compatible SSR-LC-R can be identified by the presence of an IC installed on the bottom of the board at U4.

NOTE: A proper serial connection to the SSR-LC requires transmit, receive, and ground connections. The ground connection is critical if the SSR-LC is powered from a different supply than the device with which it is communicating.

Table 1. Main Connector Pins

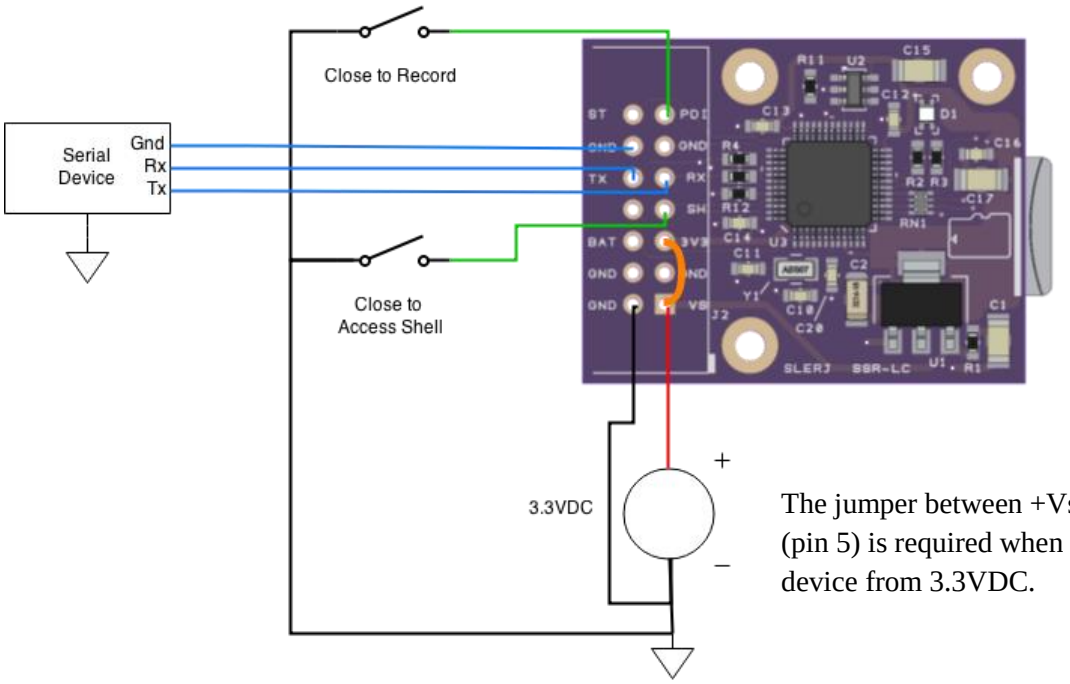
Pin [1]	ID	Description
1	Vs	Supply voltage (4.5-15 VDC, or 3.3 VDC [2])
2	GND	Supply return (tied to GND onboard the SSR-LC)
3,4	GND	Ground connection
5	3V3	Tied to the 3.3 VDC on-board supply
6	BAT	Battery voltage supply for the real-time clock
7 [3]	SH	Shell Select – interactive shell accessible when low
8 [3]	res	Reserved
9 [4]	RX	Asynchronous serial receiver input
10 [5]	TX	Asynchronous serial transmitter output
11,12	GND	Ground connection
13 [3]	PDI	PWM / Digital input record command
14 [6]	ST	Record status – High level indicates the device is recording

[1] See Section 6.2 Electrical for detailed electrical specifications.
[2] A 3.3 VDC supply can be used by connecting both the Vs and 3V3 pins to 3.3 VDC. If 3V3 is held at 3.3 VDC, Vs must not be held at ground.
[3] 5V tolerant, TTL compatible, 3.3V CMOS input. Internally pulled up to 3.3 VDC.
[4] RS-232 compatible on SSR-LC-R. 5V tolerant, TTL compatible, 3.3V CMOS input on SSR-LC-T.
[5] RS-232 compatible on SSR-LC-R. 3.3V CMOS output on SSR-LC-T.
[6] 3.3V CMOS output



Default command source is ***-dig***. Pull PDI down to record. Note that the device will not record while SH is pulled low for shell access.

Figure 3. Typical connections using a 4.5 to 15 VDC supply



The jumper between +Vs (pin 1) and 3V3 (pin 5) is required when powering the device from 3.3VDC.

Figure 4. Typical connections using a 3.3VDC supply

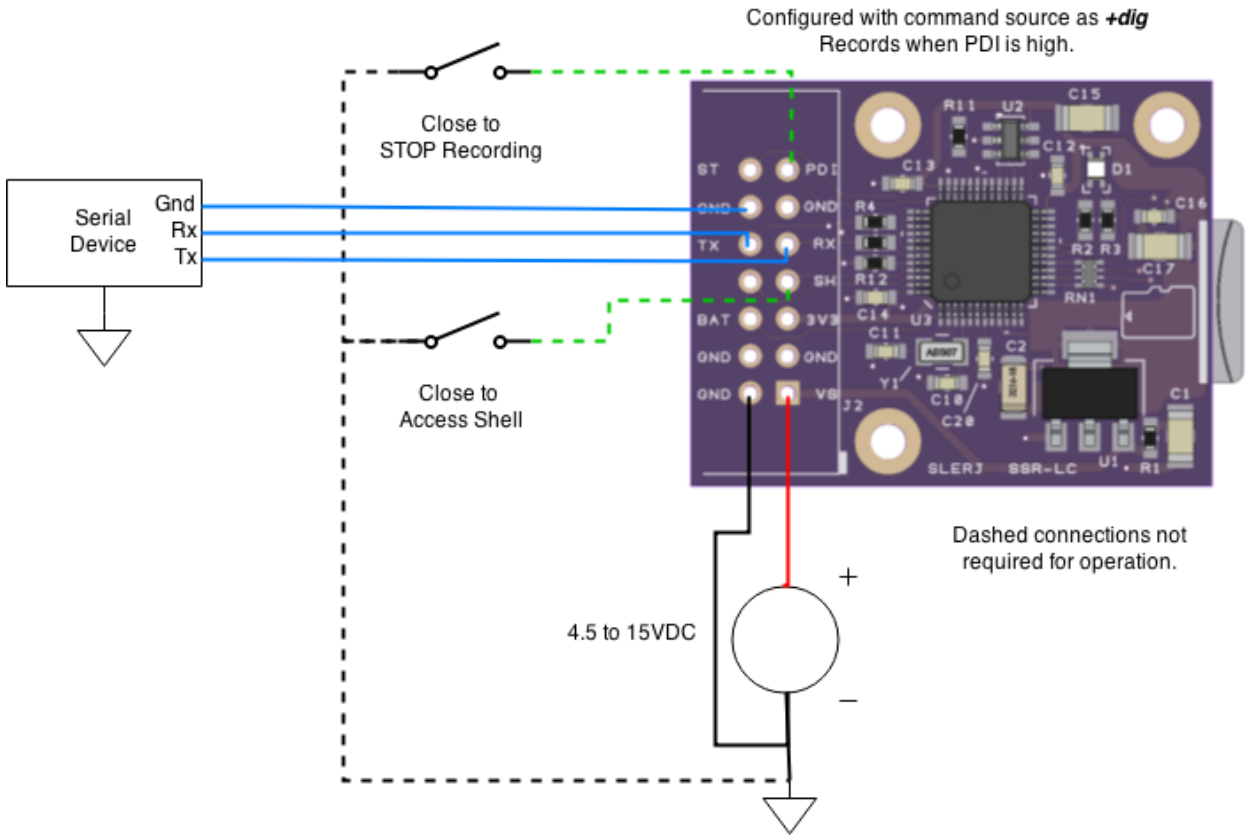


Figure 5. Connections when configured with command source as +dig

The device can be configured to record automatically at power up by configuring command source as ***+dig*** and connecting as shown in Figure 5. In this case, only the connections to the power supply and the device to be recorded are required. Both PDI and SH have internal pull up resistors (approximately 40kΩ), and will float high. With SH high (shell access not requested), and PDI high with command source set to ***+dig***, the device will begin recording at power up. Note that the SSR-LC serial parameters (baud, data bits, parity, stop bits) must match the device to be recorded.

The internal 40kΩ pull up resistors of the SSR-LC input pins may not be sufficient to keep the pins high in electrically noisy environments. In those cases, it is recommended to use stronger pull-ups (e.g., 4.7 kΩ) sourced from the on-board 3.3V supply (3V3, pin 5).

2.4 Using the SSR-LC

The SSR-LC is shipped in default configuration, with serial parameters set to 115200 baud, 8 data bits, no parity, and 1 stop bit. The device is configured to record when PDI is pulled low and SH is high. The interactive shell is accessible through the serial interface when SH is low.

On power up, the SSR-LC displays a boot loader announcement and device details. If SH is low, the shell prompt will be presented. A typical power-on sequence would produce output similar to:

```
Slerj Boot Loader v1.0.0
MK:Slerj
HW:SSRLC
MG:1662473
MD:SSR-LC
SN:1

SSR-LC Shell [Firmware 2.1.0]
>
```

Details of the shell interface are provided in Section 4 Interactive Shell. As an example of shell usage, consider changing the command source to ***+dig*** so that the device records when the PDI pin is high. In the following sequence, *<enter>* means pressing the Enter/Return key to execute the command in the shell. With the SSR-LC connected to a terminal program and the shell active, type

`config` *<enter>*

to show the current SSR-LC configuration.

To change the command source to ***+dig***, type

`config src +dig` *<enter>*

Verify that the configuration has been changed using:

`config` *<enter>*

To save the modified configuration in on-board non-volatile memory so that it is preserved across power cycles, type:

`config save` *<enter>*

To confirm that the configuration has been saved, reboot the SSR-LC and verify configuration using:

`reset` *<enter>*
`config` *<enter>*

Note that 'cfg' is an alias for 'config' and can be used as a shortcut.

3 Functional Overview

The SSR-LC consists of an asynchronous serial channel, a data recording subsystem, a user interface (shell) module, a real-time clock, and digital input/output for status and control. Specific behavior for each functional element is defined by parameters that can be inspected and modified using the user shell interface.

3.1 Serial Channel

The behavior of the serial channel is defined by the following parameters:

- Baud rate – 600 to 230400 baud, inclusive
- Bits – 8 or 7 (note that 7 bit frames require even or odd parity)
- Parity – Even, odd, or none
- Stop – 1, 1.5 or 2 stop bits.
- Echo – (Boolean) Echoes received characters out through the transmitter.

The single serial channel of the SSR-LC is attached to ***either*** the data recording subsystem ***or*** the user interface module based on the level of the SH input pin.

3.2 Data Recording Subsystem

In addition to the basic serial channel parameters above, when a channel is attached to the data recording subsystem (SH is high), several other configurable parameters apply:

- Command Source
  The *Command Source* parameter determines how the device is commanded to record. Options include a discrete digital signal, a PWM signal, or software controlled through the user interface module (shell). *Command Source* can be set to one of the following:
  - **-soft** – The device records when the *Soft Command* parameter (defined in the next major bullet) is *true*. With –soft, *Soft Command* is set to *false* at startup, and the PDI pin is ignored.
  - **+soft** – The device records when the *Soft Command* parameter is *true*. With +soft, *Soft Command* is set to *true* at startup, and the PDI pin is ignored. This selection for *Command Source* causes the device to automatically record at startup.
  - **-dig** – The device records when the PWM/digital input pin (PDI) is low.
  - **+dig** – The device records when the PWM/digital input pin (PDI) is high.
  - **-pwm** – Recording starts when the pulse width on the PWM/digital input pin (PDI) is 1ms ± 250µs. Recording stops when the pulse width on PDI is 2ms ± 250µs.
  - **+pwm** – Recording starts when the pulse width on the PWM/digital input pin (PDI) is 2ms ± 250µs. Recording stops when the pulse width on PDI is 1ms ± 250µs.

The PWM input is designed to work with the type of signal used by hobby servos. The signal is considered valid when high going pulses are present with a pulse width between 750µs and 2250µs, and a period of 4ms to 65ms. The current record state for a channel using the PWM input is changed only when a valid PWM signal is present that meets the requirements of the selected PWM *Command Source* (+pwm or –pwm, specified above).

The PDI pin is 5 Volt tolerant and internally pulled to 3.3V. See Section 6.2 Electrical for specifications.

- Soft Command
  Soft Command is a Boolean software parameter that is used when the *Command Source* parameter is ±*soft*. When *Command Source* is ±*soft* and *Soft Command* is *true*, the device records. The *Soft Command* parameter can be set through the user interface module (shell).

- File Type
  The SSR-LC supports three archive types: raw, tagged line[1], and time tagged archives.
  - When file type is *raw*, bytes are written to file just as they are received, and no timestamp information is attached to the data.
  - When file type is *tl* (tagged line), text timestamps are inserted into the stream at the first printable character following a newline or carriage return. This file type is convenient for line-oriented data, but has a number of limitations as compared to the time tagged archive. Since received data is modified with timestamp strings prior to recording, the original data stream is not preserved. The timestamp format is non-configurable, YYMMDDhhmmss.sss with a trailing space. Since this mode adds 17 characters to every line received, it can significantly inflate the volume of data that written to the card. A series of 4 byte lines into the serial channel becomes a series of 21 byte writes to the card, and in extreme cases (short lines at high speed) could exceed the write bandwidth of the card. This mode is not appropriate for binary data.
  - When the file type is *tt* (time tagged), bytes are encapsulated into an archive file structure that associates a timestamp with each group of received bytes with a resolution of 2ms. The time tagged archive format overcomes all of the limitations of the tagged line format, but requires post processing to retrieve the data. The archive is not intended to be human readable. A utility to perform the post processing, STTP, is provided with source code. For details, see Section 5 Time Tagged Archives.

[1] Support for tagged line archives was added in firmware 1.2.1.

- File Mode
  Supported file creation modes are *retry*, *overwrite*, and *append*. When file mode is *retry*, the SSR-LC will continue to retry the file creation operation until it succeeds. File creation can fail if a file with the same name already exists. This mode is a useful complement to user definable file paths (next bullet). *Overwrite* will cause an existing file to be replaced by a newly commanded recording. *Append* will cause new data to be appended to an existing file. For both *overwrite* and *append* modes, if the file specified by File Path does not exist, it is created.

- File Path
  The File Path parameter holds a path template that specifies the name and location of the file to be created when recording is commanded. A path template is a normal path string that has replaceable fields defined in Table 2 below. A field is identified in the template by a backslash followed by one of the field identifiers, or several consecutive identifiers encapsulated in square brackets. For example, the path template /st[hms].dat would be translated to the path /st083000.dat if the time is 08:30:00 when a file is created. Similarly, the path template /gps/nmea\4.txt would be replaced by /gps/nmeaXXXX.txt where XXXX is a number that is incremented on each attempt to open the file. Currently, path templates of up to 29 bytes are supported, and the resulting path (with fields replaced) must be no more than 64 bytes.

Table 2. Path Template Field Codes

Field Identifier	Replaced With
Y	year [00-99]
M	month [01-12]
D	day [01-31]
h	hour [00-23]
m	minute [00-59]
s*	second [00-59]
t	tenth of second [0-9]
y	year (4 digit) [2001-2099]
X	hex digit month [1-C]
d	day of year [001-366]
2	two digit sequence number [00-99]
3	three digit sequence number [000-999]
4	four digit sequence number [0000-9999]

* This field identifier is lower case.

- File Size
  The SSR-LC supports automatic file close and reopen when a size (or time) threshold is reached. Threshold values of 1 MB, 2 MB, 4 MB, 8 MB, 16 MB, 32 MB, 64 MB, 128

MB, 256 MB, 512 MB, and 1024 MB are supported. Additionally, the File Size parameter can be set to *Hour*, *Day*, or *Week*, causing new files to be started based on time instead of size. By default, the File Size threshold is *off*, and no automatic close/reopen operations are performed.

3.3 User Interface Module

The user interface module provides user access to file system operations, device status, and configuration. More information on the interactive shell can be found in Section 4.

3.4 Real-Time Clock

The Real-Time Clock (RTC) maintains calendar time for the SSR-LC. An off-board backup battery (typically Lithium 3V) connector to BAT and GND on the main connector allows the RTC to keep time across power cycles. RTC time is used for file creation and received data timestamps.

3.5 Digital I/O

Digital input and output lines are provided for hardware access to recording control and status. On the main connector, PDI provides record control as discussed in Section 3.2, Data Recording Subsystem. Additionally, a status line is provided (ST) to indicate when the channel is recording. A bi-color (red and green) LED on the SSR-LC (Figure 1) provides status. The green segment flashes to indicate reception of serial data on the channel and indicates when the shell is active. The red segment illuminates solid red when the device is recording. The following LED flash codes indicate device status.

Table 3. LED Flash Patterns

Status	Flash Pattern	Description
RECORDING	Red: Solid illumination	Device is recording (if shell is not active).
READY	Red: Short flash every 4 seconds	Indicates that a record channel is ready to record when commanded.
CARD ERROR	Red: Long flash followed by two short flashes every 2 seconds	Indicates that the card is missing or an unrecoverable error has occurred.
CARD FULL	Red: Two long flashes every 2 seconds	Indicates that the card is full.
SHELL ACTIVE	Green: Solid every other second with flashes during the *off* seconds to indicate serial activity.	Indicates that the serial port is connected to the user interface module instead of the data recording subsystem. No data is recorded.

4 Interactive Shell

The interactive shell is designed to provide easy access to the SD card file system, device status, and configuration options. Entering '`?`' or '`help`' at the command prompt provides information about using the shell. Each command can be followed by '`?`' to retrieve help information. Multiple commands can be separated by a semi-colon. All commands are case sensitive. For example:

```
>cls ?
Usage: cls
  Clears the screen.
  Aliases: clear

>date;time
20130327
102840
>
```

The shell supports line editing and keeps a history of recently used commands. The ANSI escape sequences shown in Table 4 are supported.

Table 4. Shell Line Editing Sequences

Keyboard Key	Alternate Sequence[1]	Function
Up-arrow	^p	Recall the previous command to the command line.
Down-arrow	^n	Recall the next command to the command line. This is only available when up-arrow has been used to recall a previous command.
Home	^a	Move the cursor to the start of the command line.
End	^z	Move the cursor to the end of the command line.
Left-arrow	^k	Move the cursor left one character.
Right-arrow	^l	Move the cursor right one character.
Ctrl + Left-arrow	^b	Move the cursor left (backward) one word.
Ctrl + Right-arrow	^f	Move the cursor right (forward) one word.

[1] The caret (^) indicates use of the Ctrl key with the letter.

In the description of individual commands below, the following conventions are used:

[ ] indicates optional parameters
{ } identifies a set of choices separated by | (choose one)
< > indicates a variable defined in the help text

4.1 System Commands

System commands provide access to general system functions including the real-time clock and operational status.

Table 5. System Commands

Command	Aliases	Description
cls	clear	Clears the screen.
date [*yyyymmdd*]		Sets the current date to the year, month, and day specified. If no date is specified, this command returns the current date.
help	?	Provides help for using the shell.
reset		Performs a device reset.
status	stat	Displays device status (date/time, inputs, record channels).
time [*hhmmss*][*ap*]		Sets the current time using the hour, minute, and second specified. The hour is assumed to be in 24 hour format. However, the time may be appended with an 'a' or 'p' to explicitly identify AM or PM if a 12 hour format is used.

4.2 File Commands

File commands provide access to the SD card file system. FAT12, FAT16 and FAT32 volumes are supported, and long filenames are supported on FAT32. Many file system commands require a *path*. Both relative and absolute paths are supported in the shell. Directories are separated by a forward slash (/).

Table 6. File Commands

Command	Aliases	Description
chdir <*path*>	cd	Changes the current working directory.
del <*path*>	rm	Removes a file or an empty directory.
df		Prints the volume size and free space -.
dir [*path*]	ls	Lists the contents of a directory. If no path is provided, this command lists the contents of the current directory.
mkdir <*path*>	md	Creates a directory.
pwd		Prints the current working directory.
ren <*path1*> <*path2*>	mv,rn	Moves or renames a file or directory from *path1* to *path2*. [NOTE: Do not move open files]
touch <*path*>		Updates the timestamp on a file or directory.
sz <*path*>		Send a file to the connected terminal[2] using the zmodem protocol.

[2] The zmodem file transfer has been tested successfully with several freely available terminal emulators, including ExtraPutty, Teraterm, SyncTERM, and qodem.

4.3 Device Configuration

Device configuration is manipulated through the user interface module. The current working configuration is held in system memory (RAM) and can be saved to non-volatile memory for preservation across resets. On startup, if the contents of the non-volatile configuration memory are valid, the stored configuration is loaded and used by the SSR-LC. The shell provides access to device configuration through the following commands:

Table 7. Configuration Commands

Command	Description
config	Prints the current configuration (including all channels).
config save	Saves the working configuration to non-volatile memory.
config load	Retrieves the stored configuration from non-volatile memory.
config erase	Erases the non-volatile configuration memory, but does not change the current working configuration. When non-volatile configuration memory has been erased, configuration defaults are loaded at startup.
config [*args*]	Provides access to serial channel and record parameter configuration. If no arguments are specified, this command prints the current configuration. Specific configuration commands are in Table 8.
config leds[1] {on | off | onx}	Allows the on-board LEDs to be turned off for power savings. The *onx* option causes the LEDs to be active for 10 seconds after power is applied.

[1] Option added in firmware 2.1.0

In addition to the global configuration commands presented in Table 7, there are several serial channel and record parameter configuration commands. The commands in the following table are entered as part of a command line 'config *command*'.

Table 8. Channel Configuration Commands

Command	Alias	Description
baud <*rate*>		Sets baud to *rate* (600 to 115200).
bits[1] {8 | 7[2]}		Sets data bits to 8 or 7[2].
parity { E | O | N | e | o | n }		Sets parity to even, odd, or none.
stop { 1 | 1.5 | 2 }		Sets the number of stop bits.
echo <*bool*[3]>		Enables echoing of received characters.
source [{ + | - }][4]{ soft | dig | pwm }	src	Sets the command source.
soft <*bool*>		Sets the soft command parameter.
file type { raw | tl | tt }		Selects between raw, tagged line, and time-tagged archives.
file mode { retry | append | overwrite }		Sets the channel file mode.
file path <*path*>		Sets the channel file path template to *path*. See Section 3.2 Data Recording Subsystem for more information on path templates.

Command	Alias	Description
file size { off | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | hour | day | week }		Sets the file size threshold. See Section 3.2 Data Recording Subsystem for more information on file size thresholds.

[1] Option added in firmware 2.1.0
[2] The SSR-LC does not support 7 data bits with no parity. A 7 bit data frame must have parity enabled.
[3] *bool* denotes a Boolean expression, and may be specified using { y | Y | t | T | true | yes | on } for affirmative and { n | N | f | F | false | no | off } for negative.
[4] The { + | - } prefix is optional. If not specified, + is assumed.

Note that multiple channel configuration commands may be specified together. For example, to set the baud, parity and stop parameters with a single command, type

```
config baud 38400 parity N stop 1
```

Also note that the file option commands are two-word commands (don't omit the word 'file'). For example, to set the file type to *raw*, type

```
config file type raw
```

4.4 Capturing the Shell

To allow access to configuration without knowing the current serial channel settings, the shell can be captured by the user during startup. The feature is inherited from the SSR-1, where it is necessary when all channels are configured for recording. For the SSR-LC, it is useful only if the current channel settings are unknown. The capture mechanism operates prior to loading stored configuration data, so the serial port always operates at 115200 baud, 8 bits, no parity, and 1 stop bit for capture.

The process for capture involves presentation of the boot loader message at startup, the user quickly typing the string *config*, the device replying with an upper case, four character challenge string, and the user echoing the challenge string back to the device in lower case. Details are as follows:

- On startup, the SSR-LC displays the boot loader message and device information.
- An 800ms window begins in which a valid character of the capture sequence must be received from the user. If an invalid character is received, or 800ms elapses, the capture sequence is aborted and the device boots normally according to its stored configuration. Each valid character received resets the capture window to 800ms. Since it is difficult to judge the time from power on until the device is ready to begin receiving the capture sequence, the valid capture sequence includes up to 5 lower case *z* characters prior to the string *config*. A typical capture will involve hitting *z* while powering on the device until the *z* character is echoed from the device, then typing *config*.
- When the previous step has been completed, the SSR-LC will send a random challenge string consisting of 4 upper case characters, and a new capture window of 5 seconds is

established. The user must type those same characters in lower case to complete the capture process. If the challenge string is not answered in 5 seconds, the capture process is aborted and the SSR-LC starts normally.

5 Time Tagged Archives

Often, it is important to know not only what serial data was transferred, but when it was transferred. This is useful in both the analysis of communication systems and in the ability to reconstruct streams as they originally occurred. The SSR-LC supports time tagged archives, in which received bytes are tagged with the time they were received. When the device is configured to record time tagged archives, the received bytes are encapsulated in packets prior to being written to the file system. Two types of packets are currently defined: the data packet and a time correlation packet. All multi-byte words in the archive are big endian.

> Note that a software utility, including source code, is provided at slerj.com for parsing time tagged archives into a variety of useful formats. See section 5.3 The STTP Utility for details.

5.1 Data Packet

The data packet uses the system free running clock as the time stamp source. Bytes are grouped into 2ms windows for stamping. Each data packet begins with a base time stamp that identifies the whole second in which the data was collected. The base time stamp is followed by a series of frames that are composed of an incremental time stamp (fractional second within the base time stamp window) and a group of bytes that were received during the increment. The data packet is terminated by a checksum.

Table 9. Time Tagged Data Packet

Element	Bytes	Description
Packet Header 0x82 0xA2	2	Packet start sequence.
Run Time	4	Current run time in seconds.
Frame repeated until an mSec_Count value of 0xFFFF is encountered.		
mSec_Count	2	Fractional second and number of bytes for this frame. bits 15-7: milliseconds / 2 bits 6-0: number of bytes to follow (*n*)
Data	*n*	The *n* bytes that were received in the time window leading up to this packet frame.
End Sequence	2	0xFFFF (invalid mSec_Count)
Checksum	2	Fletcher checksum calculated between Run Time and End Sequence, inclusive.

5.2 Time Correlation Packet

The time correlation packet associates the free running clock timer with the real-time clock. A time correlation packet is written when the recording is started, every 10 minutes, and as the recording is stopped.

Table 10. Time Tagged Time Correlation Packet

Element	Bytes	Description
Packet Header 0x82 0xA3	2	Packet start sequence.
Run Time	4	Current run time in milliseconds.
RTC Time	6	Real-Time Clock word 0: bits 15-4: year (2001 – 2099) bits 3-0: month (1 – 12) word 1: bits 15-11: day (1 – 31) bits 10-6: hour (0 – 23) bits 5-0: minute (0 – 59) word 2: bits 15-10: second (0 – 59) bits 9-0: milliseconds (0 – 999)
Checksum	2	Fletcher checksum calculated between Run Time and RTC Time, inclusive.

5.3 The STTP Utility

The SLERJ Time Tagged Parser is a Windows command line utility (sttp.exe) provided with the SSR-LC to parse time tagged archives into various output types. Source code is provided under a non-restrictive (MIT) license so that it can be freely modified and incorporated into user applications. The utility has a number of functions including the ability to extract the raw data (without timestamps), extract line-oriented data prepended with configurable timestamps, extract only portions of the data in intervals or windows, and extract text representations of the packets stored in the archive. For more details on line-oriented extraction, see Application Note *AN002 – STTP Timestamped Lines*.

Usage of the sttp utility is summarized by its help output:

```
usage: sttp.exe [options] <infile>
    Version 1.5, Feb 21 2018 20:30:23
    options:
      -h                Include headers in tcp and dat files.
      -r <raw_file>     Write raw stream data to raw_file
      -t <tcp_file>     Write Time Correlation Packets to tcp_file
      -d <dat_file>     Write Tagged data to dat_file
      -m <mxd_file>     Write both TCPs and tagged data to mxd_file
      -n <txt_file>     Write timestamped line text file
      -N "string"       Date format string for tagged line (strftime)
      -S                Supress milliseconds in tagged line output
    Interval extraction options for timestamped line output:
         For the arguments below, 'N' is assumed to be in seconds
         unless suffixed with 'L', which denotes lines.  For example
         '-i 30' denotes an interval of 30 seconds, where '-i 30L'
         Denotes an interval of 30 timestamped lines of data.
      -k,--skip N       Skip N seconds/lines of data before output
      -i,--interval N   Extract excerpts at intervals of N seconds/lines
      -w,--window N     Extract N seconds/lines at each interval
      -v,--nwins M      Process M windows (default=0, to end of file)
```

The tagged line option can extract line-oriented data with user defined timestamps (see the readme.txt file that comes with the STTP utility for details). For example, consider an archive that contains data from an instrument that produces output:

```
S D  0.0000122 kg
S D  0.0000122 kg
S D  0.0000122 kg
S D  0.0000123 kg
```

Extracting this data with options `-n outputfile.txt -N "%m/%d/%Y %H:%M:%S."` would produce an output file:

```
02/03/2014 21:47:38.915 S D  0.0000122 kg
02/03/2014 21:47:39.013 S D  0.0000122 kg
02/03/2014 21:47:39.111 S D  0.0000122 kg
02/03/2014 21:47:39.207 S D  0.0000123 kg
```

For the options that extract the archive packets (-t, -d, and –m), the outputs are space delimited text files. Data bytes are represented as a series of hexadecimal text characters. An example of each of the textual output files is below.

Time Correlation Packet output example:

```
RunTime(ms) Year Month Day Hour Minute Second
4196 2013 3 25 9 52 4.625
604196 2013 3 25 10 2 3.628
1204196 2013 3 25 10 12 2.486
```

Tagged Data output example:

```
RunTime(ms) count HexBytes
4196 20 322E323530333630652B303520322E3339343433
4198 23 30652D3034202D312E343530303639652D303420322E37
4200 23 3637343235652D303420312E373134373036652D303120
```

Mixed output example:

```
A3 4196 2013 3 25 9 52 4.625
A2 4196 20 322E323530333630652B303520322E3339343433
A2 4198 23 30652D3034202D312E343530303639652D303420322E37
…
A2 604194 23 3032202D352E353633313634652D303120312E32323636
A3 604196 2013 3 25 10 2 3.628
A2 604196 23 3330652D303220332E313334343333652B303020302037
```

6 Specifications

6.1 Environmental

All devices manufactured and shipped after October 2017 support extended temperature operation (-40 to 85C).

6.2 Electrical

	Min	Typical	Max	Unit
Supply Voltage	4.5		15	VDC
Supply Current	5 VDC Supply			
Idle[1]		22		mA
Recording[2]		44		mA
Recording[2] – LEDs Off		32		
	12 VDC Supply			
Idle[1]		22		mA
Recording[2]		44		mA
Recording[2] – LEDs Off		32		
BAT Supply Voltage	1.65		3.6	VDC
Digital Input Characteristics (RX (SSR-LC-T), PDI, SH, Res)				
Low level input voltage			1.37	V
High level input voltage	1.85			V
Schmitt trigger hysteresis		100		mV
Weak pull-up equivalent resistor	25	40	55	kΩ
Digital Output Characteristics (TX (SSR-LC-T), ST)				
Low level output voltage (±8mA)			0.4	V
High level output voltage (±8mA)	2.9			V
Low level output voltage (±20mA)			1.3	V
High level output voltage (±20mA)	2.0			V
RS-232 Transmitter Characteristics (TX (SSR-LC-R))				
Transmitter Output Voltage Range	-15		15	V
Transmitter Output Voltage into 3kΩ Load	±5	±5.7		V
RS-232 Receiver Characteristics (RX (SSR-LC-R))				
Receiver Input Voltage Range	-25		25	V
Positive going input threshold voltage		1.6	2.4	V
Negative going input threshold voltage	0.6	1.3		V

[1] SanDisk Industrial 8GB Class 10 microSDHC card inserted, but no data being received.
[2] SanDisk Industrial 8GB Class 10 microSDHC card inserted, recording full stream at 115200 baud.

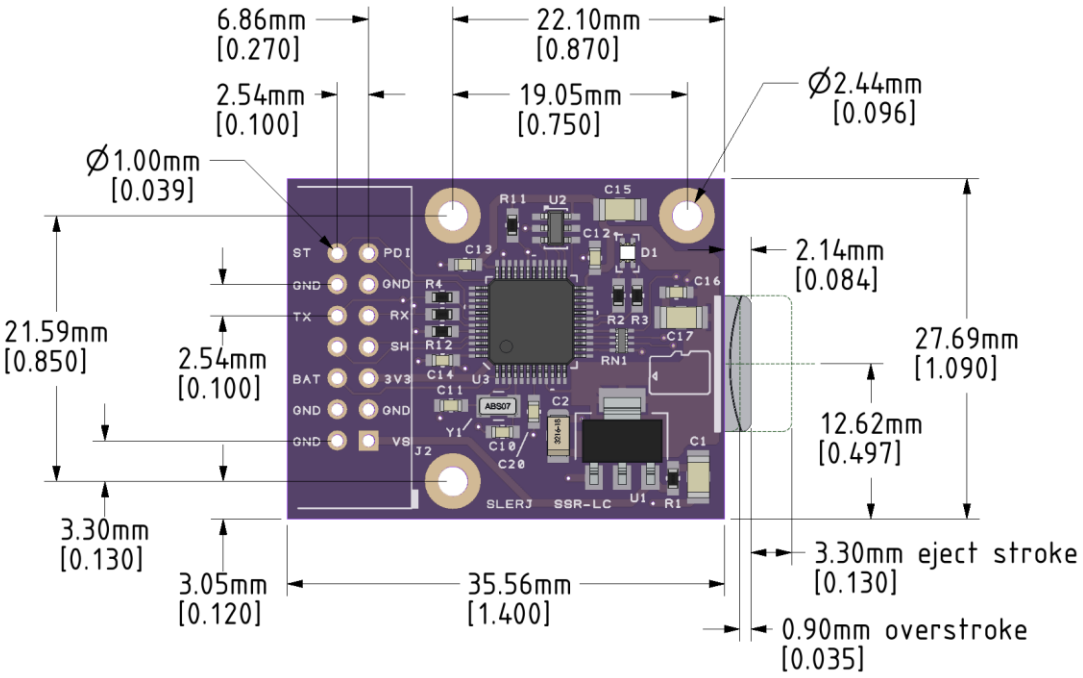
6.3 Mechanical

Dimensions: 1.40 x 1.09 x 0.20 inches (36 x 28 x 5 mm)

5mm height is without header installed

Hole Pattern: 0.750 x 0.850 inches (19.05 x 21.59 mm)

Hole Diameter: 0.096 inches (2.44 mm) – Designed for No. 2 hardware.



7 Revision History

Date	Rev.	Changes
29 April 2014		draft release
25 Sept. 2014	A	Removed field 'c' from Table 2 because the SSR-LC has only a single channel, so that providing a channel number for file names is meaningless. Also, the File Path section of 3.2 Data Recording Subsystem was updated to use an example without the 'c' field.
17 April 2015	B	Table 2: changed pin 2 ID from Vret to GND to match the board silk screen. Added dimensioned figure of the board. Updated dimensions – corrected erroneous hole pattern metric values Added typical connection figures in 2.3 Connecting the SSR-LC Added example of shell interaction in 2.4 Using the SSR-LC Clarified the Soft Command parameter in section 3.2 Data Recording Subsystem. Added reference to the new STTP utility and application note in section 5.3 The STTP Utility
12 May 2015	C	Additional clarification of the Soft Command parameter in section 3.2 Data Recording Subsystem. Typological corrections and terminology sync with SSR-1 User Manual.
27 Feb 2018	D	Updated maximum baud rate to 230400 to match recent firmware versions. Clarified wording of several parts of section 3 Functional Overview. Added the new sz command to Table 6. File Commands Added description of the new file type 'Tagged Line' introduced in firmware 1.2.1. Clarified section 4.4 Capturing the Shell Updated section 5.3 The STTP Utility to reflect version 1.5 of STTP. Added environmental specifications section.
2 Sep 2019	E	Updated for capabilities in the 2.1.0 firmware: Added exFAT as a supported file system. Added shell configuration commands for LEDs and channel data bits. Updated power consumption specifications based on firmware 2.1.0