1. **PURPOSE**

The purpose of this document is to record the source files and function definitions for the Sparrow Link Python API.

1. **SCOPE**

The scope of this record contains the version history for each version of released API source files. For detailed release notes, see the information in the version history table below.

1. **REFERENCE DOCUMENTS**

PDPROJ-12 Sparrow Link Project

PLAN-044 Sparrow Link Project Plan

PLAN-053 Sparrow Link Firmware Development Plan

1. **Version History**

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **API Version** | **Details** |
| 0 | 0.0.17 | Baseline Release |
| 1 | 0.0.18 | Allows the command ReadHubVersion() to accommodate Hub FW versions with two digits. |
| 2 | 0.0.24 | Addresses Jira entries SPL-38 and SPL-51 per CCB on 6/2/2025. |

See the attached Zip file, SFT-025 Sparrow Link Python API for the API source files.

Sparrow Link Python API Documentation

SparrowLinkHub.SparkHub()

Contents

[Serial Communication 3](#_Toc203051538)

[BLE Connection 3](#_Toc203051539)

[Sparrow Link Pulse Generator 5](#_Toc203051540)

[Device Information 5](#_Toc203051541)

[System Information 5](#_Toc203051542)

[System Time 7](#_Toc203051543)

[History 8](#_Toc203051544)

[Device Configuration 8](#_Toc203051545)

[Factory Reset 8](#_Toc203051546)

[Lock Screen 9](#_Toc203051547)

[Stimulation 10](#_Toc203051548)

[Amplitude 10](#_Toc203051549)

[Impedance 11](#_Toc203051550)

[Frequency 11](#_Toc203051551)

[Pulse Width 12](#_Toc203051552)

[Duty Cycle 12](#_Toc203051553)

[Ramping 13](#_Toc203051554)

[Sham & Error Notifications 14](#_Toc203051555)

[Triggering 16](#_Toc203051556)

# Serial Communication

Serial communication is used as the communication line between the Sparrow Link Hub and the connected computer. Set the name of the COM port, then open the communication line to the Hub, and Connect to the Sparrow Link Pulse Generator at the beginning of each script.

|  |  |  |
| --- | --- | --- |
| **SetPort()** | | |
| **Parameters** | **Input** | **Description** |
| Port | Enter the name of the COM port the Hub uses as a string.  Example: SetPort(“COM1”) | Plug the Hub into your computer and find the COM port number for the device named:  “Silicon Labs CP210x USB to UART Bridge”  Install any necessary drivers. |

|  |  |  |
| --- | --- | --- |
| **Open()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Opens the serial port to enable communication with the Hub.  Please note that you must use the SetPort() call to set the serial communication port prior to opening the line of communication. |

|  |  |  |
| --- | --- | --- |
| **Close()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Closes the serial port to disable communication with the Hub. |

# BLE Connection

To begin interfacing with the Sparrow Link Pulse Generator using the Sparrow Link Hub, use the BLE connection methods to establish a pairing between the Pulse Generator and the Hub. When done, Disconnect the pairing.

Note: After a factory reset or update to the Sparrow Link Pulse Generator, it may take two attempts to connect to the app or Hub. If the first attempt fails, cancel BLE advertising from the Pulse Generator and restart the connection steps.

When accurate timestamps of device interactions are needed, the ConnectSync() call is recommended. This call ensures that the Pulse Generator's internal clock is set to the same date and time settings as the computer connected to the Hub.

|  |  |  |
| --- | --- | --- |
| **Connect(serial)** | | |
| **Parameters** | **Input** | **Description** |
| Serial | Wrap the 6-digit serial number (SN) displayed on the Sparrow Link Pulse Generator in quotations. | Connects the Sparrow Link Hub to the Sparrow Link Pulse Generator.  Steps to connect:   1. Enable Bluetooth Low Energy (BLE) on the Sparrow Link Pulse Generator by navigating to the BLE screen and selecting ‘OK’ to turn on Bluetooth. 2. After BLE is enabled, enter the 6-digit number shown on the Pulse Generator screen into the Connect() call, as shown in the example.   Example: Connect(“123456”) or Connect(‘123456’) |

|  |  |  |
| --- | --- | --- |
| **ConnectSync(serial)** | | |
| **Parameters** | **Input** | **Description** |
| Serial | Wrap the 6-digit serial number (SN) displayed on the Sparrow Link Pulse Generator in quotations. | Connects the Sparrow Link Hub to the Sparrow Link Pulse Generator and sets the Sparrow Link Pulse Generator system time to match the date and time of your computer.  Steps to connect:   1. Enable Bluetooth Low Energy (BLE) on the Sparrow Link Pulse Generator by navigating to the BLE screen and selecting ‘OK’ to turn on Bluetooth. 2. After BLE is enabled, enter the 6-digit number shown on the Pulse Generator screen into the ConnectSync() call as shown in the example.   Example: Connect(“123456”) or Connect(‘123456’) |

|  |  |  |
| --- | --- | --- |
| **Disconnect()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | End the pairing between the Sparrow Link Pulse Generator and the Sparrow Link Hub. |

# Sparrow Link Pulse Generator

## Device Information

Use the following calls to get information about the system setup or state of the paired Sparrow Link Pulse Generator.

### System Information

|  |  |  |
| --- | --- | --- |
| **ReadSerial()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Returns the last 6 digits of the serial number (SN) for the paired Pulse Generator.  Example: ‘123456’ |

|  |  |  |
| --- | --- | --- |
| **ReadModel()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Returns the model number of the paired Pulse Generator.  Example: Sparrow Link Pulse Generator s return ‘910’ |

|  |  |  |
| --- | --- | --- |
| **ReadManufacturer()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Returns the full serial number of the paired Pulse Generator.  Example: ‘X2150211201’ |

|  |  |  |
| --- | --- | --- |
| **ReadFirmware()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Returns the firmware version of the paired Pulse Generator.  Example: ’03.00.12’ |

|  |  |  |
| --- | --- | --- |
| **ReadHardware()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Returns the hardware version of the paired Pulse Generator.  Example: ’02.00.12’ |

|  |  |  |
| --- | --- | --- |
| **ReadHubVersion()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Returns the firmware version of the Hub.  Example: ’0.0.9’ |

### System Time

Use the following calls to set or check the system time of the paired Sparrow Link Pulse Generator.

|  |  |  |
| --- | --- | --- |
| **SetSystemTime(seconds)** | | |
| **Parameters** | **Input** | **Description** |
| Seconds | Seconds of time since epoch | Sets the date and time to the parameter entered. |

|  |  |  |
| --- | --- | --- |
| **SetSystemCurrentTime()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Sets the date and time to match the system date and time of the computer powering the Hub. |

|  |  |  |  |
| --- | --- | --- | --- |
| **ReadSystemTime ()** | | | |
| **Parameters** | **Input** | **Description** | **Returns** |
| None | None | Gets the date and time set on the Pulse Generator. | Example; ‘TimeStamp,MM/DD/YYYY HH:MM:SS,’ |

### History

Device history captures Pulse Generator use and changes to device settings. Call Clear History to erase all device history records.

To retrieve the device logs from the Sparrow Link Pulse Generator, call Save History or connect the Pulse Generator to the Sparrow Link Mobile App and use the Share Logs feature to download the logs to the tablet or mobile device running the mobile app. The logs will be downloaded in a .CSV format to the local folder or the device’s internal storage, respectively.

|  |  |  |
| --- | --- | --- |
| **ClearHistory()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Erases all device history records. |

|  |  |  |
| --- | --- | --- |
| **SaveHistory()** | | |
| **Parameters** | **Input** | **Description** |
| ‘filename.csv’, Input  \*Note: Change ‘filename’ to name of choosing with **.csv** appended | ‘Today’ | Saves device history records as a csv from that day, based on the system date at which the computer is set. |
| ‘All’ | Saves all device history records as a csv. |
| 'mmddyyyy','mmddyyyy' | Saves device history records between the two dates as a csv. |

## Device Configuration

### Factory Reset

Factory reset can be useful when reusing Pulse Generators across multiple use cases or end users. Alternatively, to maintain custom device settings while clearing the device history, see Clear History under the History section.

|  |  |  |
| --- | --- | --- |
| **FactoryReset()** | | |
| **Parameters** | **Input** | **Description** |
| None | None | Factory Reset will reset the Pulse Generator to the manufacturer's default settings. All custom settings and device history logs will be erased. |

### Lock Screen

The Pulse Generator screen locks, just like your phone, to prevent accidental button presses and changes to stimulation while wearing the Pulse Generator in an ambulatory setting. By default, the screen will dim after 15 seconds of inactivity and locks after 30 seconds of inactivity. Use the following commands to modify the screen behavior as needed.

|  |  |  |
| --- | --- | --- |
| **SetDimTimeMs(value)** | | |
| **Parameters** | **Input** | **Description** |
| Value | Input a value from 0 – 24 hours in **milliseconds** | Sets the duration of inactivity required before the Pulse Generator screen will dim. |

|  |  |  |
| --- | --- | --- |
| **SetLockTimeMs(value)** | | |
| **Parameters** | **Input** | **Description** |
| Value | Input a value from 0 – 24 hours in **milliseconds** | Sets the duration of inactivity required before the Pulse Generator screen will lock.  Note, inputting the value of 0 will disable the lock screen but will not disable the screen from dimming. |

### Stimulation

#### Starting and Stopping Stimulation

|  |  |  |
| --- | --- | --- |
| **StartStimulation(channel)** | | |
| **Parameters** | **Input** | **Description** |
| Channel | 0 | Toggles ON stimulation for the **Inner** channel. |
| 1 | Toggles ON stimulation for the **Outer** channel. |

|  |  |  |
| --- | --- | --- |
| **StopStimulation(channel)** | | |
| **Parameters** | **Input** | **Description** |
| Channel | 0 | Toggles OFF stimulation for the **Inner** channel. |
| 1 | Toggles OFF stimulation for the **Outer** channel. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ReadStimulationState(channel)** | | | | |
| **Parameters** | **Input** | **Description** | **Returns** | **Description** |
| Channel | 0 | Gets the stimulation state for the **Inner** channel. | 0 or 1 | 0 means the channel is OFF  1 means the channel is ON  Note: Channels can be toggled ON with an amplitude of 0. Use the ReadAmplitude() call to determine the amplitude delivered. |
| 1 | Gets the stimulation state for the **Outer** channel. | 0 or 1 | 0 means the channel is OFF  1 means the channel is ON  Note: Channels can be toggled ON with an amplitude of 0. Use the ReadAmplitude() call to determine the amplitude delivered. |

### Amplitude

|  |  |  |
| --- | --- | --- |
| **SetAmplitude(channel,amplitude)** | | |
| **Parameters** | **Input** | **Description** |
| Channel | 0 | Sets the amplitude for the **Inner** channel. |
| 1 | Sets the amplitude for the **Outer** channel. |
| Amplitude | An integer from 0-50. | Sets the amplitude for the given channel. An amplitude input of 0 sets the amplitude to 0.0 mA. An amplitude input of 50 sets the amplitude to 5.0 mA.  Note: Changes to the amplitude while stimulation is ON will be instantaneous and will not ramp. |

|  |  |  |
| --- | --- | --- |
| **ReadAmplitude(channel)** | | |
| **Parameters** | **Input** | **Description** |
| Channel | 0 | Gets the amplitude for the **Inner** channel as shown on the Pulse Generator. Returns an integer from 0-50. |
| 1 | Gets the amplitude for the **Outer** channel as shown on the Pulse Generator. Returns an integer from 0-50. |

### Impedance

|  |  |  |
| --- | --- | --- |
| **ReadImpedanceValue(channel)** | | |
| **Parameters** | **Input** | **Description** |
| Channel | 0 | Returns the current impedance for the **Inner** channel in Ohms.  Example: 10000 |
| Channel | 1 | Returns the current impedance for the **Outer** channel in Ohms.  Example: 10000 |

### Frequency

|  |  |  |
| --- | --- | --- |
| **SetFrequency(channel,freq)** | | |
| **Parameters** | **Input** | **Description** |
| Channel | 0 | Sets the frequency for the **Inner** channel. |
| 1 | Sets the frequency for the **Outer** channel. |
| Frequency | 1, 5, 10, 15, 20, 25, 30, 40, 50, 75,100,125,150 | Sets the frequency for the given channel in **Hz**.  Note: See the Sparrow Link Instructions for Use for documentation on our frequency combinations that are unsupported or output a nominal value +/-15% of the selected frequencies. |

|  |  |  |  |
| --- | --- | --- | --- |
| **ReadFrequency(channel)** | | |  |
| **Parameters** | **Input** | **Description** | **Returns** |
| Channel | 0 | Gets the frequency for the **Inner** channel. | Returns the **Inner** channel frequency in **Hz**.  Example: 15 |
| 1 | Gets the frequency for the **Outer** channel. | Returns the **Outer** channel frequency in **Hz**.  Example: 100 |

### Pulse Width

|  |  |  |
| --- | --- | --- |
| **SetPulseWidth(channel,pw)** | | |
| **Parameters** | **Input** | **Description** |
| Channel | 0 | Sets the pulse width for the **Inner** channel. |
| 1 | Sets the pulse width for the **Outer** channel. |
| pw | 50, 100, 150, 250, 350, 500, 750 | Sets the pulse width for the given channel in **µS**. |

|  |  |  |  |
| --- | --- | --- | --- |
| **ReadPulseWidth(channel)** | | | |
| **Parameters** | **Input** | **Description** | **Returns** |
| Channel | 0 | Gets the pulse width for the **Inner** channel. | Returns the **Inner** channel set pulse width in **µS**.  Example: 250 |
| 1 | Gets the pulse width for the **Outer** channel. | Returns the **Outer** channel set pulse width in **µS**. |

### Duty Cycle

|  |  |  |
| --- | --- | --- |
| **SetCycleState(value)** | | |
| **Parameters** | **Input** | **Description** |
| Value | 0 | Disables the duty cycle or session intervals.  **Note:** Use this call to disable the duty cycle prior to using the trigger function. If left enabled, the duty cycle will continue while triggering. |
| 1 | Enables the duty cycle. The duty cycle is enabled by default. |

|  |  |  |
| --- | --- | --- |
| **SetSessionIntervalTimeMs(milliseconds)** | | |
| **Parameters** | **Input** | **Description** |
| Milliseconds | 102 ms - 24 hours | Set the ON portion of the duty cycle. Enter the number of **milliseconds** that stimulation shall be ON while in cycling mode. |

|  |  |  |
| --- | --- | --- |
| **SetSessionIntervalBreakMs(milliseconds)** | | |
| **Parameters** | **Input** | **Description** |
| Milliseconds | 0 ms - 24 hours | Set the OFF portion of the duty cycle. Enter the number of **milliseconds** that Stimulation shall be off while in cycling mode. |

|  |  |  |  |
| --- | --- | --- | --- |
| **ReadCycleState()** | | | |
| **Parameters** | **Input** | **Description** | **Returns** |
| None | None | Use this call to determine if the Duty Cycle state is enabled/disabled. | **True**: The Cycle state is enabled, and duty cycling will occur as configured. |
| **False**: The Cycle state is disabled, and duty cycling will not occur. |

|  |  |  |  |
| --- | --- | --- | --- |
| **ReadSessionIntervalTime()** | | | |
| **Parameters** | **Input** | **Description** | **Returns** |
| None | None | Gets the ON portion of the duty cycle in milliseconds. | A value in the range of 100 ms - 24 hours in **milliseconds**. |

|  |  |  |  |
| --- | --- | --- | --- |
| **ReadSessionIntervalBreak()** | | | |
| **Parameters** | **Input** | **Description** | **Returns** |
| None | None | Gets the OFF portion of the duty cycle in milliseconds. | A value in the range of 10 ms - 24 hours in **milliseconds**. |

### Ramping

|  |  |  |
| --- | --- | --- |
| **SetStepType(value)** | | |
| **Parameters** | **Input** | **Description** |
| value | 0 | Sets the step type to the **default** ramping algorithm of 0.1 mA increase per second. |
| 1 | Enables the setting of a **custom** ramp-up time.  **Note:** You must enable this setting before setting a custom ramp time with the SetFixedTimeMs() call. |

|  |  |  |
| --- | --- | --- |
| **SetFixedTimeMs(milliseconds)** | | |
| **Parameters** | **Input** | **Description** |
| Milliseconds | 50 mS – 60 seconds | **Sets a custom fixed ramp-up time.** When stimulation is toggled ON, stimulation will ramp-up to the target strength in the duration input in milliseconds. **Enter all inputs in milliseconds.** |

|  |  |  |
| --- | --- | --- |
| **SetRampDownMs (Milliseconds)** | | |
| **Parameters** | **Input** | **Description** |
| Milliseconds | 50 mS – 3 seconds | **Sets a custom fixed ramp-down time.** When stimulation is toggled OFF, stimulation will ramp-down to 0 mA in the duration input in milliseconds. **Enter all inputs in milliseconds.** |

### Sham & Error Notifications

|  |  |  |
| --- | --- | --- |
| **SetSham (sham, message)** | | |
| **Parameters** | **Input** | **Description** |
| sham | 0 | Disable the sham state. Shammed stimulation is disabled by default. |
| 1 | Enable the sham state.  **Note:** Before setting a sham channel state with the SetShameState() call, you must enable this setting. |
| message | 0 | Disable Error Notifications. |
| 1 | Enable Error Notifications. Error notifications are enabled by default. |

|  |  |  |
| --- | --- | --- |
| **SetShamState(value)** | | |
| **Parameters** | **Input** | **Description** |
| Value | 0 | Disable sham for all channels. |
| 1 | Enable sham for ONLY the Inner channel. The Inner channel will not output stimulation. |
| 2 | Enable sham for ONLY the Outer channel. The Outer channel will not output stimulation. |
| 3 | Enable sham for BOTH the Inner and Outer channels. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ReadShamMessage ()** | | | |  |
| **Parameters** | **Input** | **Description** | **Returns** | **Description** |
| None | None | Gets the global Sham state for stimulation and Error Notifications. | 256, 257, 1 or 0 | **256** means that the Sham state is disabled and Error Notifications are ON.  **257** means that the Sham State is Enabled and Error Notifications are ON.  **1** means that the Sham state is Enabled and Error Notifications are OFF.  **0** means that the Sham state is disabled and Error Notifications are OFF. |

|  |  |  |
| --- | --- | --- |
| **ReadShamState()** | | |
| **Parameters** | **Return** | **Description** |
| None | 0 | Sham is disabled for all channels. |
| 1 | Sham is enabled for ONLY the Inner channel. The Inner channel will not output stimulation. |
| 2 | Sham is enabled for ONLY the Outer channel. The Outer channel will not output stimulation. |
| 3 | Sham is enabled for BOTH the Inner and Outer channels. |

### Triggering

When the Sparrow Link Pulse Generator is set to Trigger Mode, the screen will lock and read “ Trigger Mode Enabled.” This feature blinds and prevents an end user from viewing or modifying the stimulation.

Always configure the stimulation parameters as desired before entering Trigger Mode. This includes:

* Cycle times
* Ramp times
* Pulse width
* Frequency
* Amplitude

Before entering Trigger mode, turn off all stimulation. If the stimulation is ON when entering Trigger mode, it will continue to cycle as set by the cycle ON and OFF times. Changes to stimulation other than Trigger ON and OFF may result in failed commands or dropped connections.

|  |  |  |
| --- | --- | --- |
| **SetTriggerMode(value)** | | |
| **Parameters** | **Input** | **Description** |
| Value | 0 | Exit the trigger state. |
| 1 | Enable triggering for the Inner channel via Channel A on the Hub. |
| 2 | Enable triggering for the Outer channel via Channel B on the Hub |
| 3 | Enable triggering for BOTH the Inner and Outer channels via Channel A & B on the Hub. |

If you want to run triggering from a console or script, enable trigger mode, then use the SetTrigger() command below to toggle stimulation ON and OFF. You do not need to use the SetTrigger() command when triggering from the BNC connections, channels A or B.

|  |  |  |
| --- | --- | --- |
| **SetTrigger(value)** | | |
| **Parameters** | **Input** | **Description** |
| Value | 0 | Turn the trigger OFF. Stimulation will turn off. |
| 1 | Trigger ON the Inner channel. |
| 2 | Trigger ON the Outer channel. |
| 3 | Trigger ON BOTH the Inner and Outer channels. |

To improve response rates while rapidly triggering, use SetTriggerLog() below to turn OFF the logging of triggering events. By default, logging of trigger events will be ON.

|  |  |  |
| --- | --- | --- |
| **SetTriggerLog(value)** | | |
| **Parameters** | **Input** | **Description** |
| Value | 0 | Turn OFF the logging of Trigger events. |
| 1 | Turn ON the logging of Trigger events |

Revision History

|  |  |  |
| --- | --- | --- |
| Rev | Reason for Revision | CO No. |
| 0 | Initial Release | See GG |