

## SF000 LiDAR sensor

### MicroLiDAR for obstacle detection



FM 654831

#### Disclaimer

Information found in this document is used entirely at the reader's own risk and whilst every effort has been made to ensure its validity, neither LightWare Optoelectronics (Pty) Ltd nor its representatives make any warranties with respect to the accuracy of the information contained herein.

## Overview

The SF000/B is an ultra-small and light laser LiDAR sensor designed for size and weight constraint applications. Using a time-of-flight system to make very fast, accurate distance measurements. Accuracy is not affected by the colour or texture of the surface, nor the angle of incidence of the laser beam. The SF000/B is virtually immune to background light, wind and noise, making it an ideal sensor to detect unexpected obstacles.

The SF000/B can take up to 388 readings per second and the configurable features and multiple hardware interfaces make the SF000/B easy to connect with different types of controllers.

## Product support

This document supports the following devices:

Product	Hardware	Firmware	Supported
SF000/B	1	1.2.0	Yes

Revision	Date	Notes
0	17/09/2021	Initial release

<b>Table of contents</b>	
<b>Overview</b>	<b>2</b>
Product support	2
<b>Table of contents</b>	<b>2</b>
<b>Specifications</b>	<b>5</b>
<b>Quickstart guide</b>	<b>6</b>
<b>Package contents</b>	<b>11</b>
<b>Hardware</b>	<b>12</b>
Dimensions	12
Components	12
Optical assembly	13
Heatsink & EMI shield	13
Connectors & indicators	14
Communication cable pinout	14
<b>Installation</b>	<b>15</b>
Mounting	15
<b>Operating concepts</b>	<b>17</b>
Distance measuring	17
Parameters	18
<b>Firmware change log</b>	<b>20</b>
Upgrading firmware	20
1.2.0	20
<b>Safety &amp; maintenance</b>	<b>21</b>
Safety overview	21
Laser eye safety	21
Electrical safety	22
Mechanical safety	22
Laser radiation information	23
Service & maintenance	23
Cleaning	23
<b>Serial interface</b>	<b>24</b>
Overview	24
Packets	25
Checksum	25
Receiving packets	27
Handling request & response	28
<b>I2C interface</b>	<b>29</b>
Overview	29
<b>Command list</b>	<b>30</b>
<b>Command descriptions</b>	<b>32</b>
0. Product name	32
1. Hardware version	32

---

2. Firmware version	32
3. Serial number	33
7. UTF8 text message	33
9. User data	33
10. Token	34
12. Save parameters	34
14. Reset	34
27. Distance output	35
30. Stream	36
44. Distance data in cm	37
45. Distance data in mm	38
50. Laser firing	39
57. Temperature	39
66. Update Rate	40
74. Noise	41
75. Zero offset	41
76. Lost signal counter	41
79. Baud rate	42
80. I2C address	43
82. Median filter enable	43
83. Median filter size	44
84. Smoothing filter enable	44
85. Smoothing factor	45
93. Rolling average enable	45
94. Rolling average size	46
<b>Document revision</b>	<b>47</b>

**Specifications**

<b>Performance</b>	
<b>Range</b>	0.2 ... 50 m ( white wall in daylight conditions)
<b>Resolution</b>	1 cm
<b>Update rate</b>	48 ... 388 readings per second
<b>Accuracy</b>	±10 cm
<b>Connections</b>	
<b>Power supply voltage</b>	4.5 V ... 5.5 V
<b>Power supply current</b>	100 mA
<b>Outputs &amp; interfaces</b>	Serial and I2C (3.3 V TTL, 5V tolerant)
<b>Mechanical</b>	
<b>Dimensions</b>	24 mm x 34 mm x 21 mm
<b>Weight</b>	9.7 g (excluding cables)
<b>Optical</b>	
<b>Laser safety</b>	Class 1M (Refer to <a href="http://www.lightware.co.za/safety">www.lightware.co.za/safety</a> for full details)
<b>Optical aperture</b>	10 mm
<b>Beam divergence</b>	< 1°
<b>Environmental</b>	
<b>Operating temperature</b>	-10 ... +40°C
<b>Shipping Temperature</b>	- 40 ... + 80 °C
<b>Approvals</b>	FDA: 1710193-000 (2020/09)
<b>Enclosure rating</b>	N/A
<b>Accessories</b>	
<b>Main cable</b>	Digikey: 455-3238-ND
<b>Breakout board</b>	SKU: ACC.BOB.000
<b>USB adaptor</b>	SKU: ACC.USB.000
<b>Default settings</b>	
<b>Serial port settings</b>	115200 baud, 8 data bits, 1 stop bit, no parity, no handshaking
<b>I2C address</b>	0x66 (Hex), 102 (Dec)
<b>Update rate</b>	40 readings per second
<b>Main cable connections</b>	
<b>1</b>	Servo driver
<b>2</b>	SDA (I2C data line must be used with pin 4)
<b>3</b>	TXD/SCL (Serial/I2C)
<b>4</b>	RXD/SDA (Serial/I2C)
<b>5</b>	GND
<b>6</b>	+5V (4.5V - 5V operating range)

## Quickstart guide

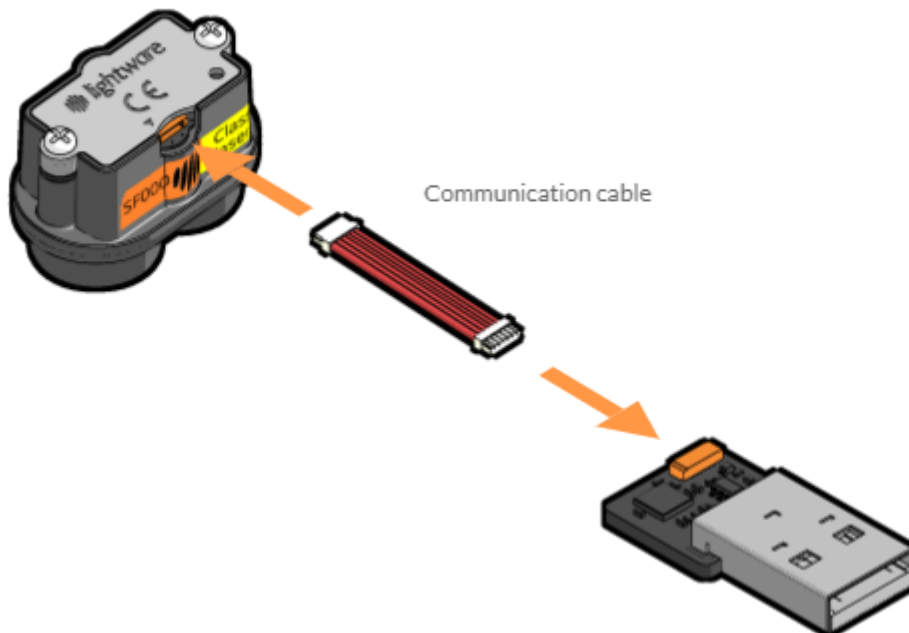
LightWare Studio is an application (available for Windows, macOS, and Linux) that can configure, update, and visualize data for the SF000/B.

In this guide we will use LightWare Studio to view distance data from the SF000/B.

LightWare Studio can be found [here](#). Download the version compatible with your operating system and proceed with installation. You can safely install over an existing version of LightWare Studio if you are upgrading.

Insert the communication cable into the SF000/B and a serial to USB adaptor\*\*.

You will need a serial to USB adaptor to connect the SF000/B to a computer. Any serial TTL 3.3 V USB adaptor will work, this guide uses one available from LightWare LLC [here](#).

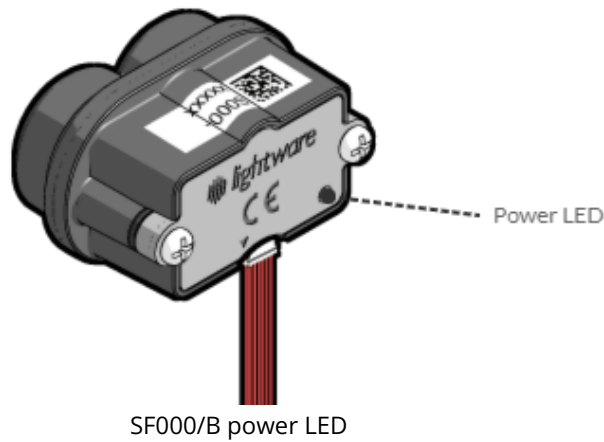


SF000/B being connected to a USB adaptor

Insert the USB adaptor into your computer.

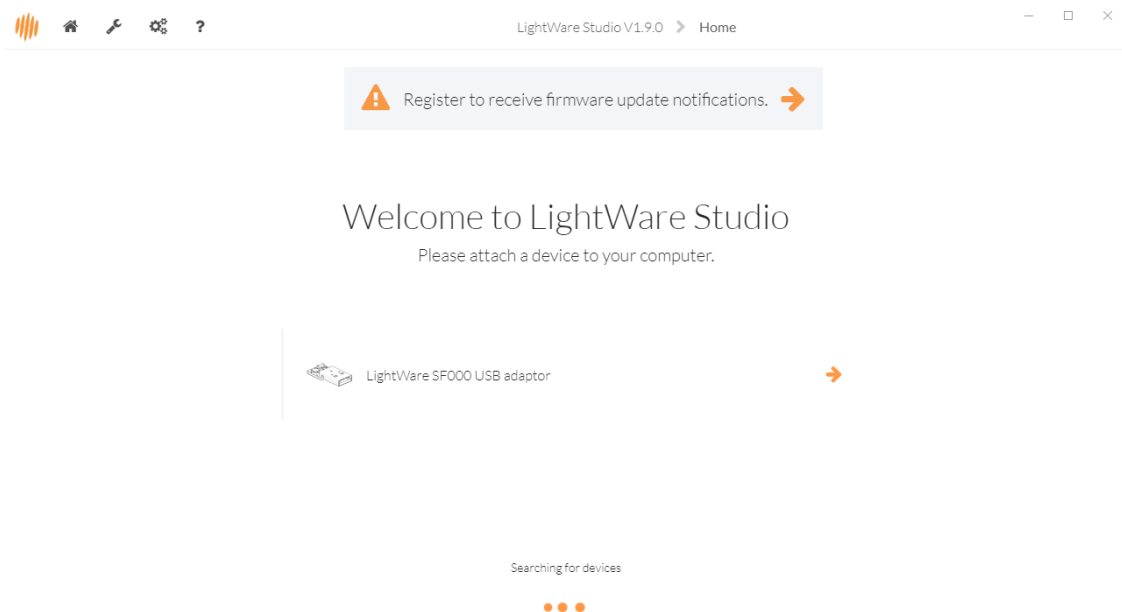
The power LED on the back of the SF000/B will light up when the USB adaptor has been connected to a computer.

\*\*USB adaptor sold separately.



Windows users: Please wait for Windows to install the generic communication driver after connecting the SF000/B for the first time.

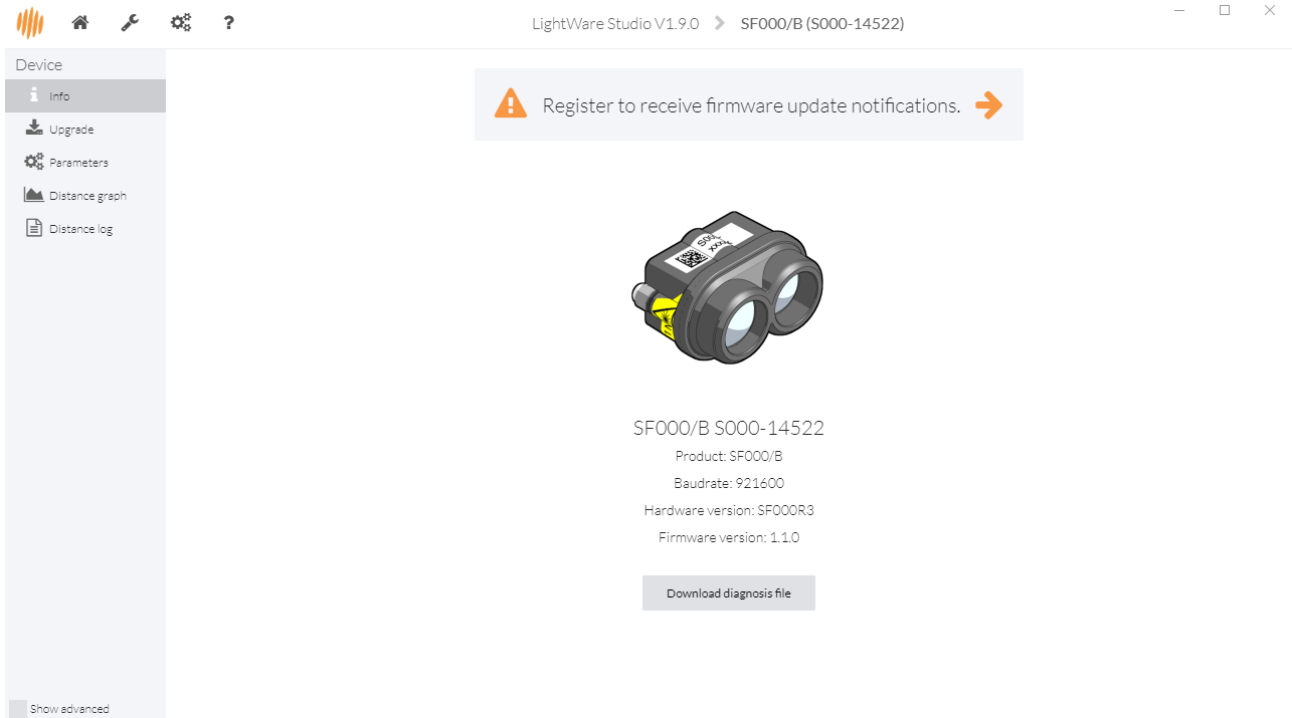
Run LightWare Studio. You will be presented with a home screen that shows devices connected to your computer. Your USB adaptor will be recognized and displayed here.



LightWare Studio device connection screen

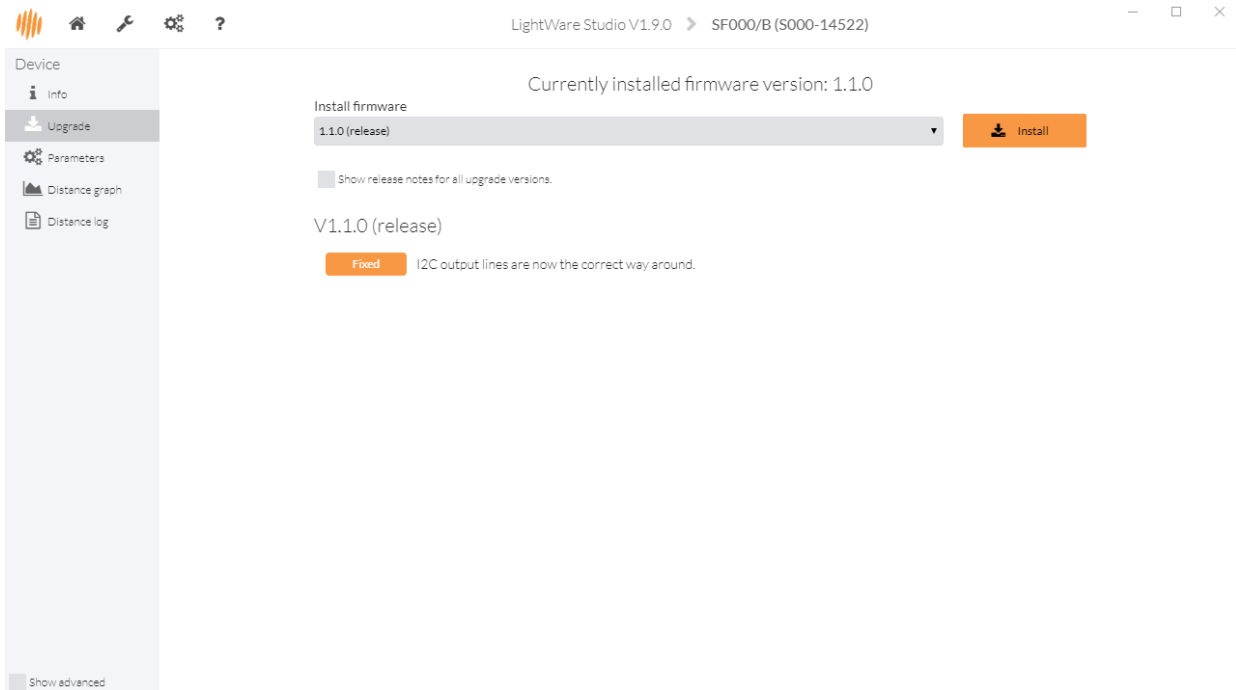
You can register your device by clicking on the Register to receive firmware update notifications banner.

Click on the USB adaptor to establish a connection. From here you can access tools to inspect and configure the SF000/B.



LightWare Studio device information screen

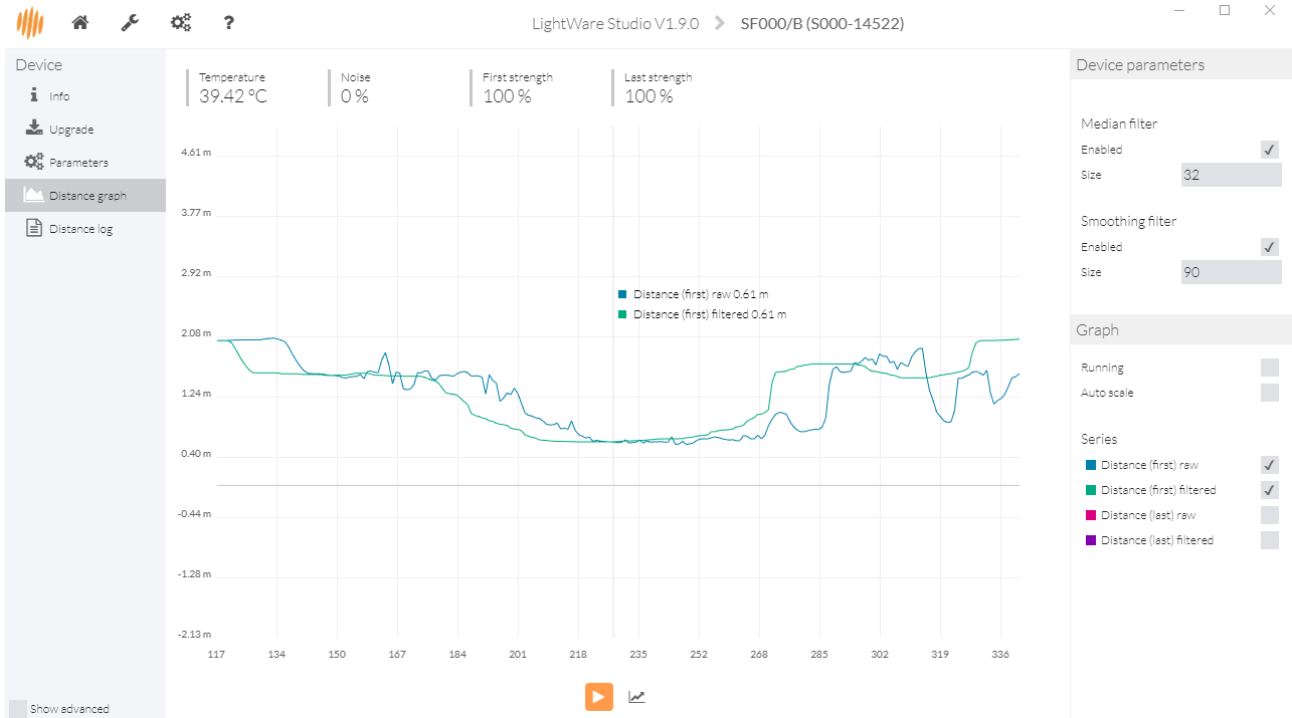
Click on the Upgrade tool in the left panel. It is recommended to make sure your SF000/B has the latest firmware. You can see the changes that have been made to each version, and the option of downgrading is also available. If you choose to upgrade, then click the Install button and follow the instructions.



LightWare Studio device upgrade screen

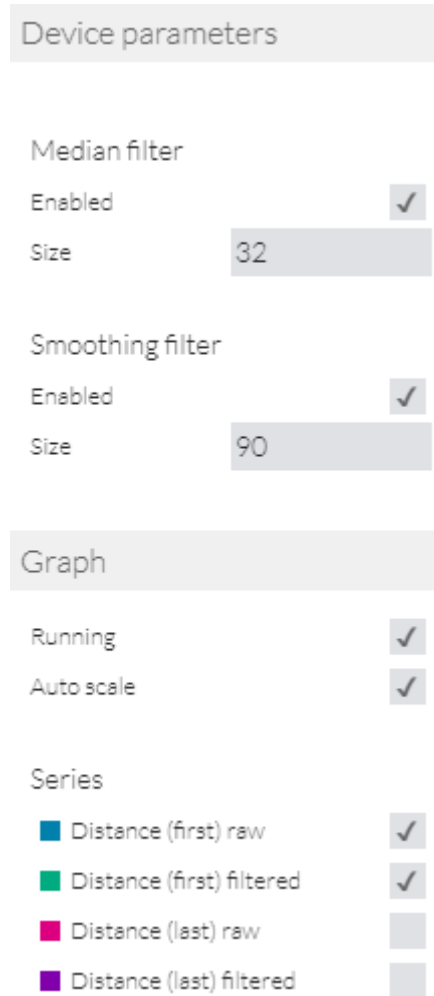


Click the Distance tool in the left panel. This tool shows you distance readings gathered by the SF000/B. Feel free to point the SF000/B at various surfaces to measure their distances.



LightWare Studio SF000/B scanning screen

Parameters are shown in the panel on the top right of the LightWare Studio window. Modify these parameters to fit your application.



LightWare Studio SF000/B scanning parameters

For a detailed breakdown on what the parameters do and how they affect the operation of the SF000/B, see the [operating concepts](#) section.

### What's next?

The next step is to integrate the SF000/B into your platform of choice. Please follow one of the integration guides, or consult the communication protocol reference for building your own system to interact with the SF000/B.

### Package contents

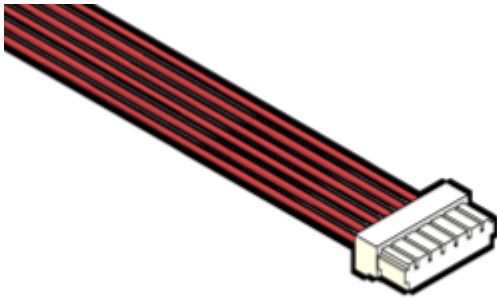
The SF000/B box includes:



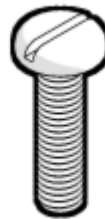
1 x SF000/B MicroLiDAR unit



1 x SF000/B mounting bracket



2 x Communication cable (Digikey: 455-3238-ND)



5 x M3 x 12 SS Cap Screw



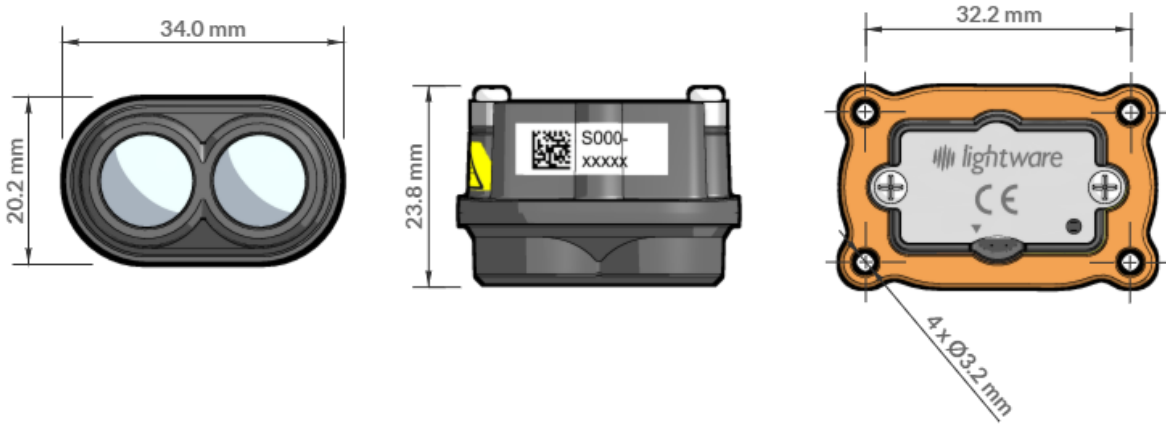
5 x M3 SS Hex Nut



5 x M3 SS Washer

**Hardware**

**Dimensions**



SF000/B dimensions

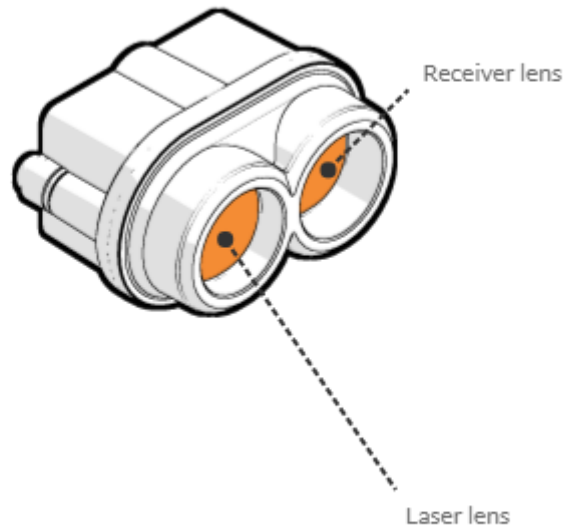
**Components**



SF000/B view from above and below

### Optical assembly

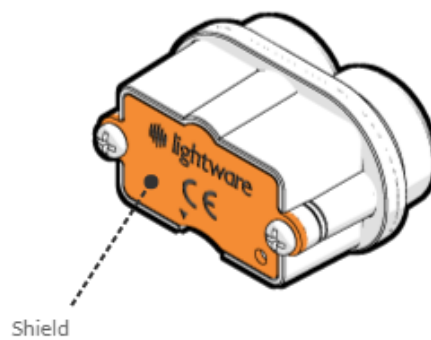
The optical assembly is comprised of the laser and receiver lenses.



SF000/B optical assembly

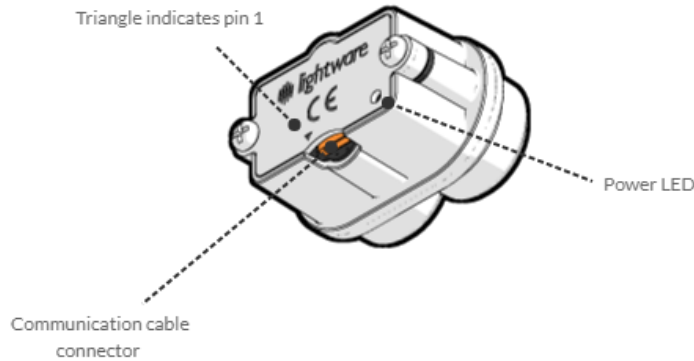
### Heatsink & EMI shield

The shield lowers EMI radiation entering or leaving the SF000/B. It also acts as a heatsink to draw heat away from the SF000/B.



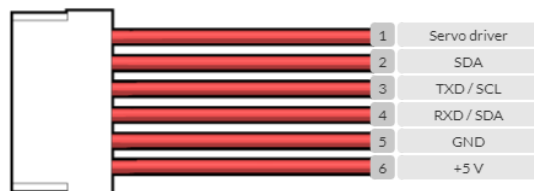
SF000/B shield

## Connectors & indicators



SF000/B connectors & indicators

## Communication cable pinout



Communication cable (Digikey: 455-3238-ND) pinout

Pin	Function	Notes
1	Servo driver	
2	SDA	I2C data line (must be used with pin 4)
3	TXD/SCL	TXD when using Serial, SCL when using I2C.
4	RXD/SDA	RXD when using Serial, SDA when using I2C.
5	GND	
6	+5 V	4.5 V ... 5.5 V, 100 mA (typical).

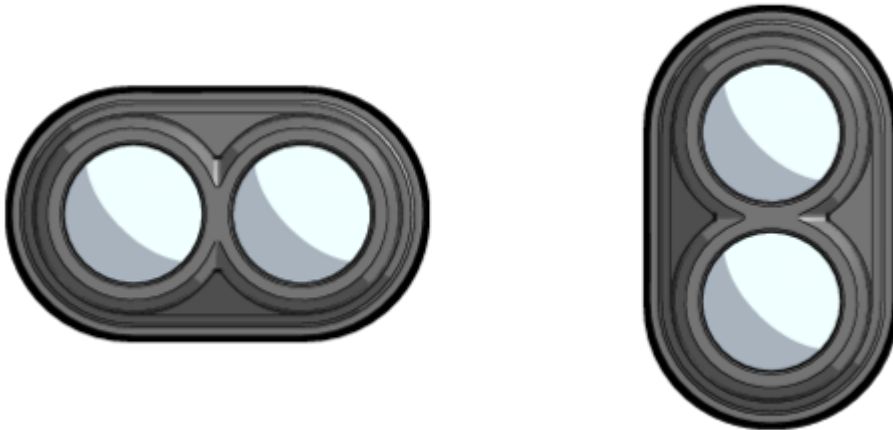
Please note that when connecting to the device using I2C, both pins 2 and 4 need to be used together as the SDA line.

## Installation

### Mounting

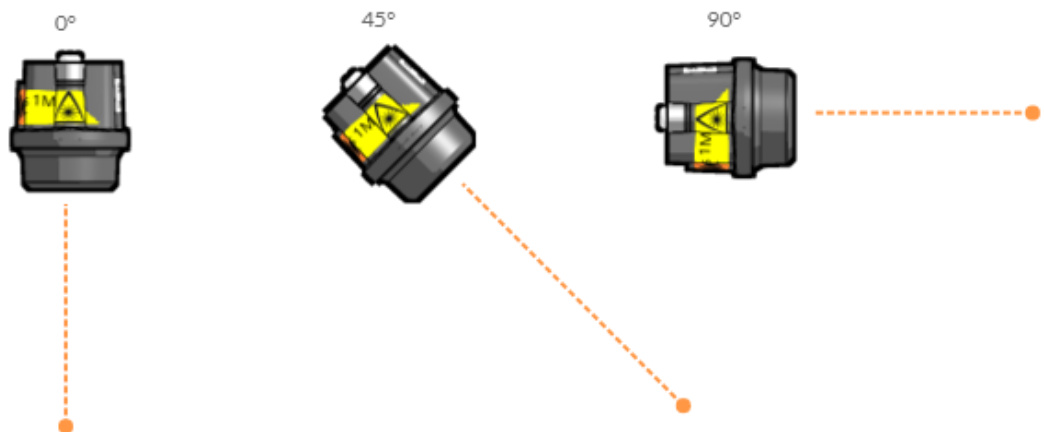
Make sure the SF000/B is functional before installation. You can use LightWare Studio to verify operation. See the [quick start with LightWare Studio](#) for details.

The SF000/B requires a clear line-of-sight to measure distance to a target surface. It can be mounted with a vertical or horizontal lens orientation.



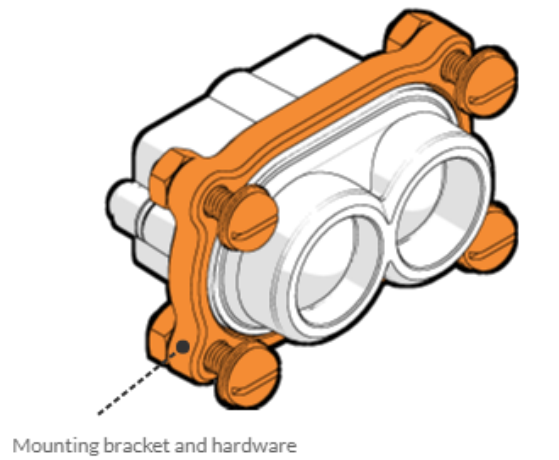
SF000/B mounting orientations

The SF000/B can be mounted in a downward facing orientation for altimetry, terrain following or precision landing applications. For terrain following, install the LiDAR sensor at an angle to reduce lag time in reaction. The angle depends on the speed traveled, and the overall system lag, but should be between 20 and 45 degrees. The SF000/B can be mounted forward facing orientation for sense-and-avoid or position-hold applications.



SF000/B mounting directions

The SF000/B includes a mounting bracket and mounting hardware which can be used to securely attach the LiDAR to the required surface.



Mounting bracket installed onto the SF000/B

**Take note:**

- Make sure the SF000/B is securely mounted to prevent false readings or damage.
- Allow for proper ventilation.
- Secure the communication cable to prevent it from pulling on the connector.

Ensure that nothing is in the path of the laser beam. Ensure that no shiny or highly reflective surfaces are near the path of the beam.

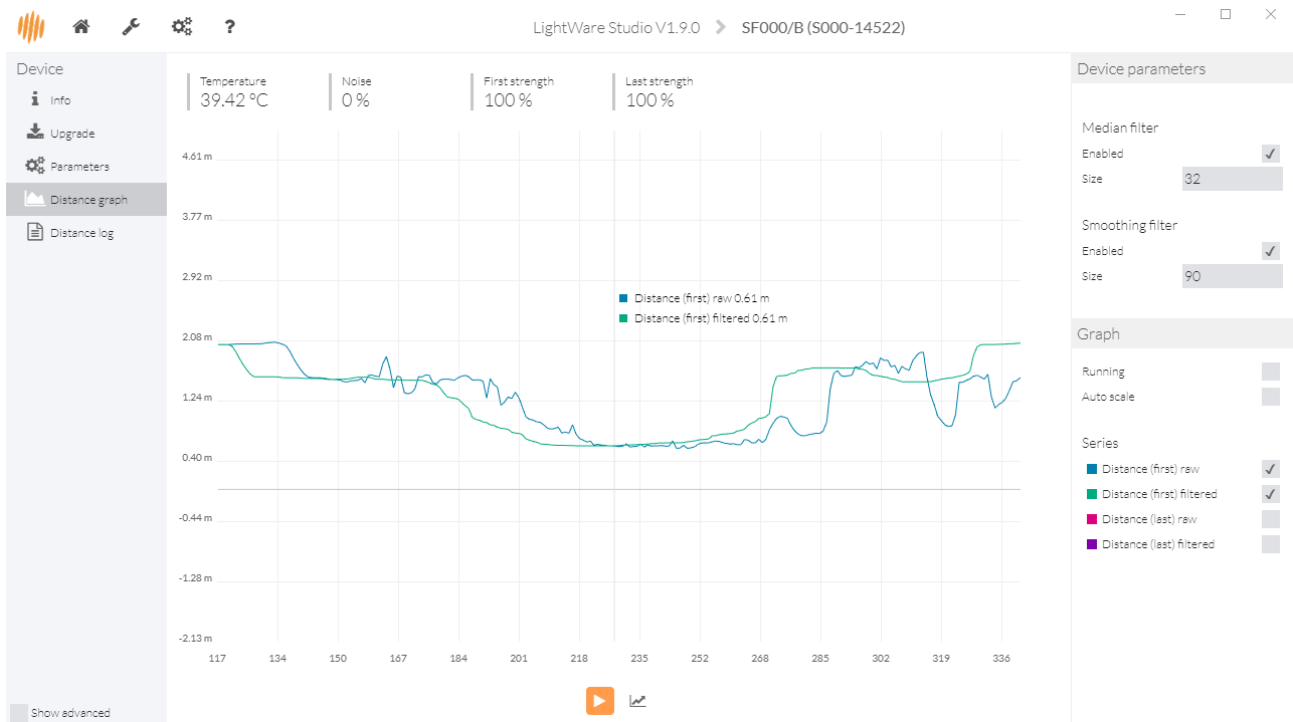


## Operating concepts

### Distance measuring

The SF000/B is a single point distance measuring LiDAR. It uses time of flight technology to determine the distance of a target at a configurable update rate of 48 Hz to 388 Hz.

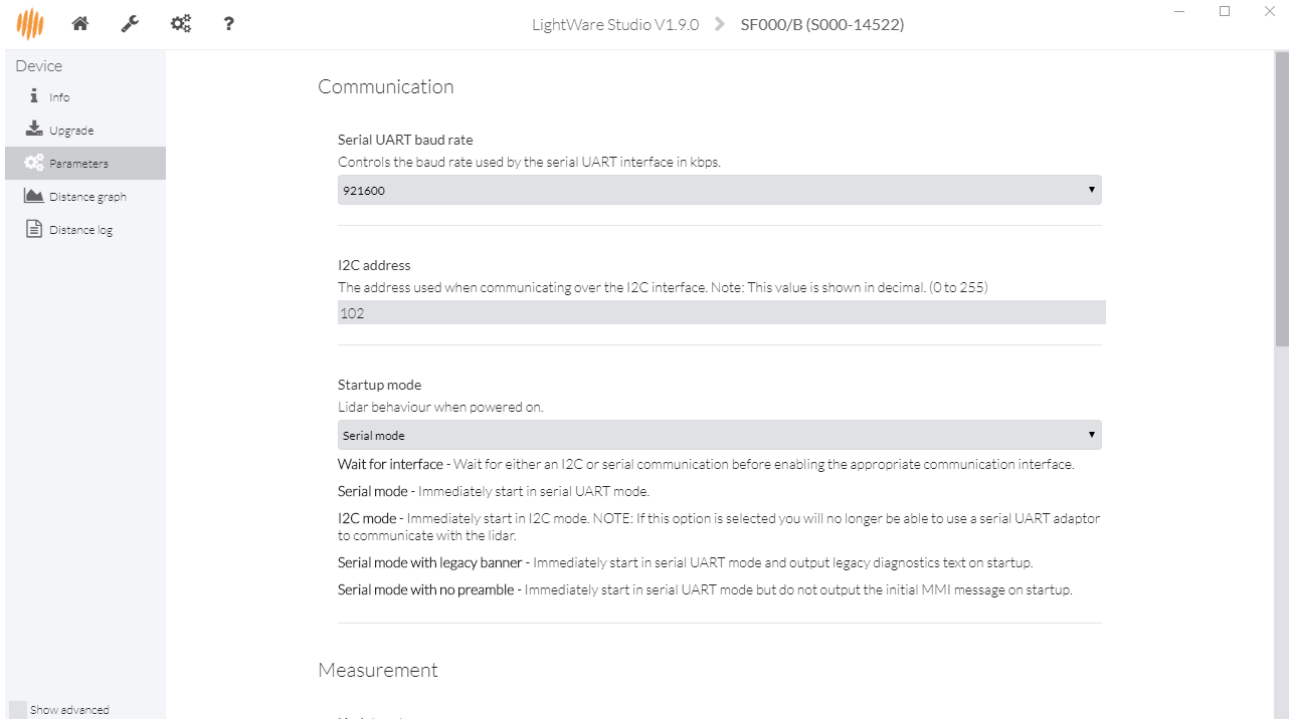
The SF000/B can be connected to a host controller with the serial or I2C interface. The serial port (3.3 V logic level, 5 V tolerant) has a configurable baud rate. The I2C serial bus (3.3 V logic level and 3.3 k pull up resistors, 5 V tolerant) has a configurable address and is an alternative to the serial port when multiple devices are connected on a common bus.



LightWare Studio showing measurements from the SF000/B

## Parameters

SF000/B parameters can be adjusted with LightWare Studio or from the platform of your choice through the serial or I2C communication interfaces.



LightWare Studio parameters screen

**Serial port baud rate**

Controls the baud rate used by the serial UART interface in kbps.  
(The baud rate should be as high as possible to accommodate high measurement update rates.)

**I2C address**

The address used to communicate on the I2C bus.

**Startup mode**

Determines the communication behaviour of the SF000/B when turned on.

**Update rate**

The number of measurements per second. From 48 Hz to 388 Hz.

**Zero distance offset**

An offset applied to the measured distance value.

**Lost signal threshold**

The number of failed measurements before a loss of signal is reported.

**Enable median filter**

The median filter is used to disregard short unwanted measurements.

**Median filter size**

Determines response time of median filter.

**Enable rolling average filter**

The rolling average filter averages over a specified number of distance measurements.

**Rolling average size**

The number of distance measurements to use for the rolling average filter.

**Enable smoothing filter**

The smoothing filter is used to remove noise from the measurements.

**Smoothing filter strength**

Determines response of the smoothing filter.

## Firmware change log

### Upgrading firmware

The SF000/B firmware can be upgraded by using *LightWare Studio*. Download *LightWare Studio* [here](#).

Follow the [Quick start with LightWare Studio](#) guide for details on downloading and using *LightWare Studio*.

### 1.2.0

Notes:

- Initial release.

## Safety & maintenance

### Safety overview

The SF000/B should not be disassembled or modified in any way. The laser eye safety rating depends on the mechanical integrity of the optics and electronics. There are no user serviceable parts and maintenance or repair must only be carried out by the manufacturer or a qualified service agent.

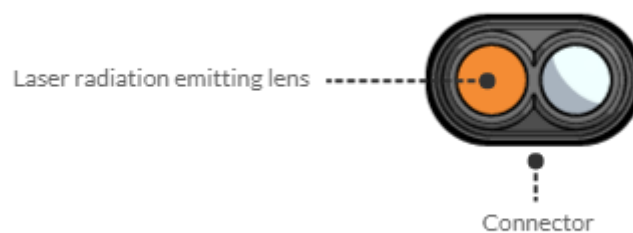
Always observe the following safety precautions when operating the SF000/B. Failure to comply with these precautions or specific warnings given in this product guide violates the safety standards of the intended use for the sensor and may damage the sensor. LightWare LLC assumes no liability for failure to comply with these requirements.

The SF000/B is not field serviceable. For any repairs, the equipment should be completely isolated, removed then packaged carefully. Please visit LightWare LLC's [refund policy](#) for Return Merchandise Authorization (RMA) details, or contact [support@lightwarelidar.com](mailto:support@lightwarelidar.com).

### Laser eye safety

The SF000/B emits ionizing laser radiation. The level of the laser emission is Class 1M which indicates that the laser beam is safe to look at with the unaided eye, but must not be viewed using binoculars or other optical devices. Notwithstanding the safety rating, avoid looking into the beam and switch the unit off when working in the area.

The laser eye safety rating depends on the mechanical integrity of the optics and electronics, if these are damaged do not continue using the SF000/B.



Laser emitting lens

The SF000/B is rated laser Class 1M eye safe. Class 1M laser is safe for all conditions of use however it is not safe to view the laser through magnifying optics such as microscopes, binoculars or telescopes.

The use of optical instruments with this product will increase eye hazard.



Class 1M laser radiation label

### Electrical safety

- Check all electrical connections are isolated and that there are no exposed wires.
- Make sure the power supplied to the device does not exceed the maximum rated voltages which are specified in the technical specifications section.
- Keep the device free of moisture as this device is not IP rated.
- The electrical wiring layout of this device is provided in the hardware section.

### Mechanical safety

- Ensure that there is adequate airflow permitted for the sensor.
- The mechanical dimensions for mounting the SF000/B are provided in the hardware overview.

**Laser radiation information**

Specification	Value/AEL	Notes
Eye safety classification	Class 1M	
Laser wavelength	905 nm	
Pulse width	15 ns	
Pulse frequency	20 kHz	
Average power	< 3.0 mW	
NOHD	15 m	Distance beyond which binoculars may be used safely.

Approximate values only. Please contact LightWare LLC if further information is required.

**Service & maintenance**

The SF000/B is not field serviceable. For any repairs, the equipment should be completely isolated, removed then packaged carefully. Please visit LightWare LLC's [refund policy](#) for Return Merchandise Authorization (RMA) details, or contact [support@lightwarelidar.com](mailto:support@lightwarelidar.com)

**Cleaning**

Use a dry and clean microfiber cloth to gently wipe across the lenses of the sensor, taking care not to scratch the surface of the lenses.

## Serial interface

### Overview

We suggest using the pre-built APIs for communicating with the SF000/B where possible.

The SF000/B uses a packet based binary protocol which can be accessed over the serial and I2C interfaces. All higher level APIs (C, Python, JavaScript) use this protocol to function.

If you require more control than the existing APIs offer, or need to port the protocol to a different platform, then you can use the information here to build a compatible system.

Communication is performed using encapsulated packets for both sending and receiving data. Every packet that is sent to the SF000/B is known as a request and a correctly formatted request will always be replied to with a response. There are cases where the SF000/B will send a request packet to the host, these packets are considered streaming packets as they arrive without a direct request from the host - they do not require a response from the host.

Streaming data is only available through the Serial interface.

Requests are made using one of the available [commands](#) and are either flagged as read or write. When a read request is issued then the response will contain the requested data. When a write request is issued then the contents of the response will vary depending on the command.

Default serial interface properties

- Baud rate: 115200
- Data: 8 bit
- Parity: none
- Stop: 1 bit
- Flow control: none



### Packets

A packet for both requests and responses is composed of the following bytes:

	Header			Payload		Checksum	
**Byte:**	Start	Flags Low	Flags High	ID	Data 0 .. N	CRC Low	CRC High

The Start byte is always 0xAA and indicates the beginning of a packet. It is important to verify that the payload length is between (inclusive) 0 to 1023 and that the checksum is valid before processing a packet, rather than just relying on the start byte.

The Flags bytes form a 16 bit integer that represents the payload length and read/write status of the packet. The payload length is inclusive of the ID byte and the required number of data bytes. The write bit is set to 1 to indicate write mode, or 0 to indicate read mode.

**Bit:**	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	Payload length (0 to 1023)										Reserved				W	

The ID byte represents which command the request/response relates to.

There will be between 1 and 1023 Payload bytes (inclusive) depending on the command type. Each command under the [detailed command descriptions](#) section documents how the data bytes are used. The ID byte will always be present in the payload.

The CRC bytes form a 16 bit checksum value used to validate the integrity of the packet data. Every byte in the packet except for the CRC itself is included in the checksum calculation.

### Checksum

Each packet has a 2 byte checksum which is used to validate data integrity. The algorithm is CRC-16-CCITT 0x1021. The CRC must be correctly formed for the SF000/B to accept and process packets. Below are some examples in various languages for CRC calculation:

## C/C++

```
uint16_t createCRC(uint8_t* Data, uint16_t Size)
{
    uint16_t crc = 0;

    for (uint32_t i = 0; i < Size; ++i)
    {
        uint16_t code = crc >> 8;
        code ^= Data[i];
        code ^= code >> 4;
        crc = crc << 8;
        crc ^= code;
        code = code << 5;
        crc ^= code;
        code = code << 7;
        crc ^= code;
    }

    return crc;
}
```

## JavaScript

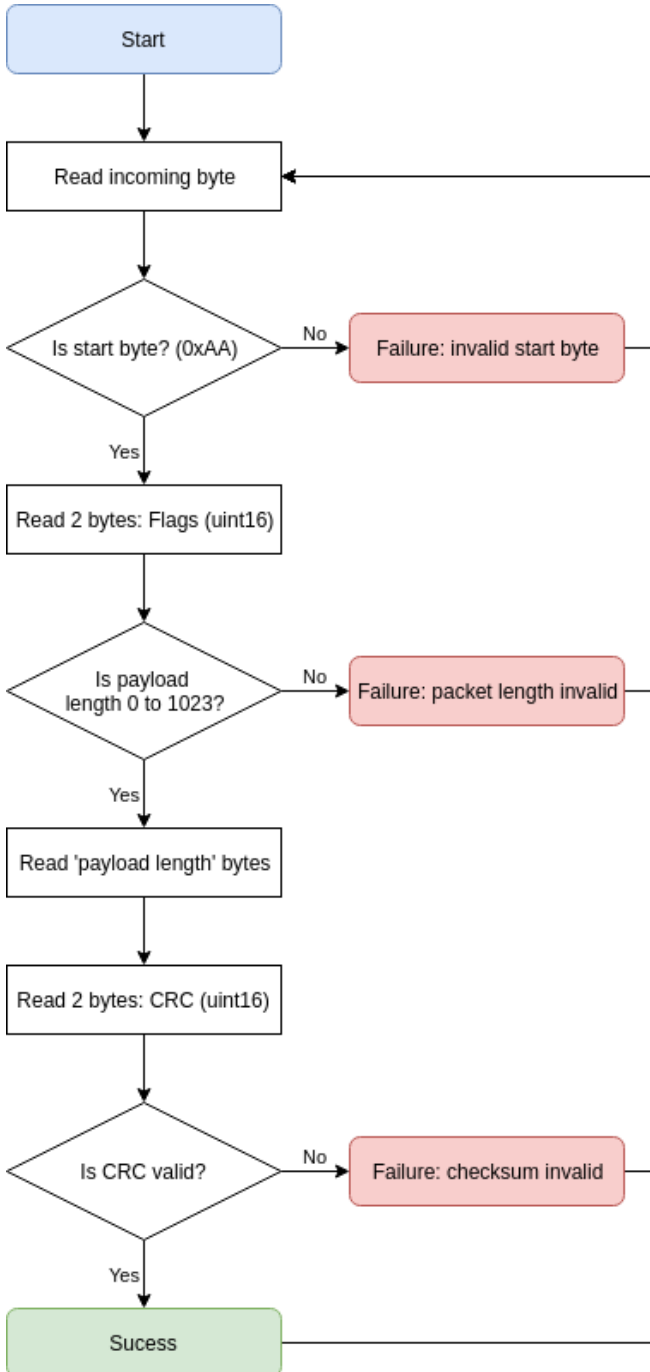
```
function createCRC(data, size) {
    let crc = 0;

    for (let i = 0; i < size; ++i) {
        let code = crc >>> 8 & 0xFF;
        code ^= data[i] & 0xFF;
        code ^= code >>> 4;
        crc = crc << 8 & 0xFFFF;
        crc ^= code;
        code = code << 5 & 0xFFFF;
        crc ^= code;
        code = code << 7 & 0xFFFF;
        crc ^= code;
    }

    return crc;
}
```

### Receiving packets

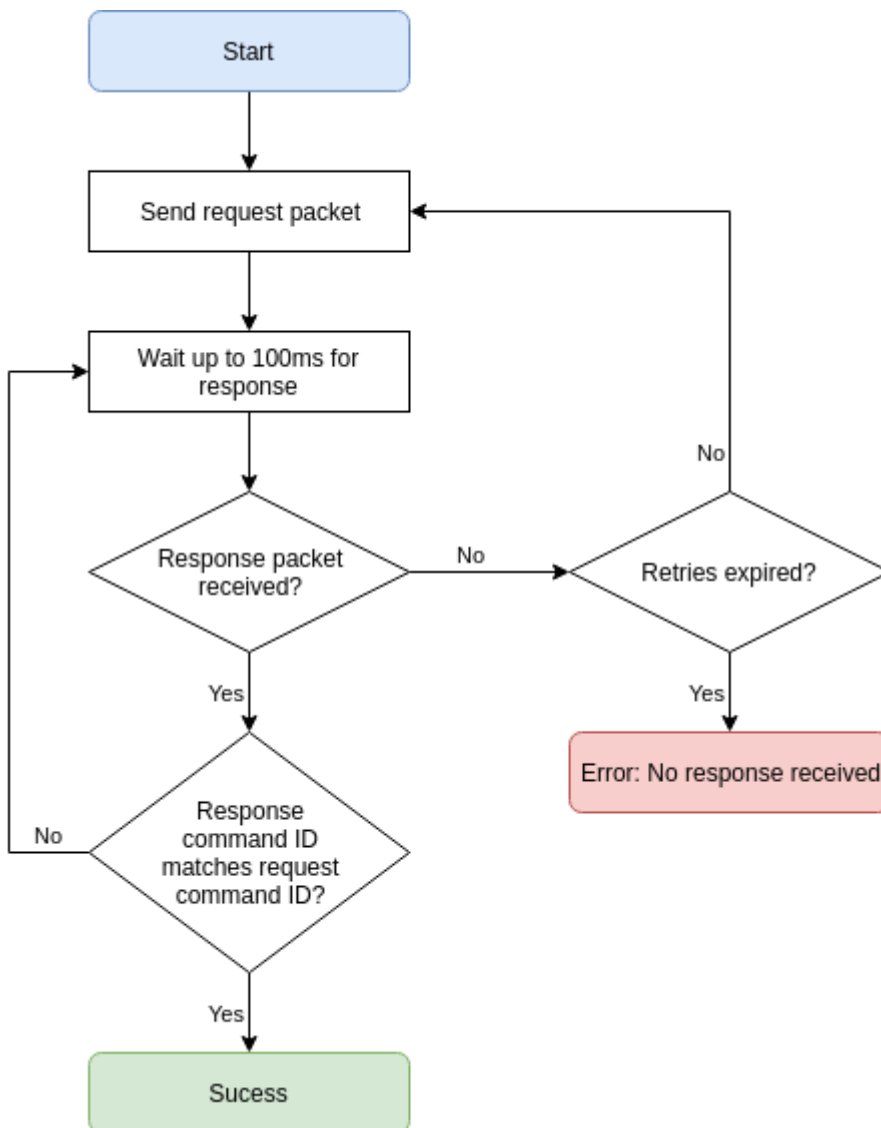
Here is the process for reading the raw serial byte stream and identifying packets. Once a packet has been successfully read it can be processed based on its command ID.



If the packet length or checksum is invalid then it is technically more correct to roll the incoming stream back to when the start byte was found. However in practice this has little appreciable impact.

### Handling request & response

Every request sent to the SF000/B will receive a response, it is often useful to use the response as a way to determine if the request was received and processed. Here is the recommended procedure for sending a command request and reading the response:



The values used for timeout or number of retries should be tuned to the specific application.

## I2C interface

### Overview

We suggest using the pre-built APIs for communicating with the SF000/B where possible.

The SF000/B uses a packet based binary protocol which can be accessed over the serial and I2C interfaces. All higher level APIs (C, Python, JavaScript) use this protocol to function.

If you require more control than the existing APIs offer, or need to port the protocol to a different platform, then you can use the information here to build a compatible system.

The SF000/B will always be the slave on the I2C interface. Therefore data will only be transmitted when requested by the master.

Streaming data is only available through the Serial interface.

Requests are made using one of the available [commands](#). When a read request is issued then the response will contain the requested data. When a write request is issued then there is no response generated.

#### Default I2C interface properties

- Address: 0x66

**Command list**

If a command is not readable or writable then it can only be received from the SF000/B and not sent to it.

ID	Name	Description	RW	Read bytes	Write bytes	Persists
0	Product name	Product name	R	16	-	-
1	Hardware version	Hardware revision	R	4	-	-
2	Firmware version	Firmware revision	R	4	-	-
3	Serial number	Serial number	R	16	-	-
7	UTF8 text message	Human readable text message	-	-	-	-
9	User data	16 byte store for user data	RW	16	16	Y
10	Token	Next usable safety token	R	2	-	-
12	Save parameters	Store persistent parameters	W	-	2	-
14	Reset	Restart the unit	W	-	2	-
27	Distance output	Distance output configuration	RW	4	4	N
30	Stream	Current data stream type	RW	4	4	N
44	Distance data in cm	Measurement distance data in cm	R	<i>varies</i>	-	-
45	Distance data in mm	Measurement distance data in mm	R	<i>varies</i>	-	-
50	Laser firing	Is laser firing?	RW	1	1	N

57	Temperature	Measured temperature	R	4	-	-
66	Update rate	Data sampling update rate	RW	1	1	Y
74	Noise	Measured background noise	R	4	-	-
75	Zero offset	This adjusts the zero distance position in mm	RW	4	4	Y
76	Lost signal counter	Set number of lost signal conditions	RW	4	4	Y
79	Baud rate	Serial baud rate	RW	1	1	Y
80	I2C address	I2C address	RW	1	1	Y
82	Median filter enable	Enables median filter	RW	1	1	Y
83	Median filter size	Size of median filter	RW	4	4	Y
84	Smoothing filter enable	Enables smoothing filter	RW	1	1	Y
85	Smoothing factor	Strength of smoothing filter	RW	4	4	Y
93	Rolling average enable	Enables rolling average	RW	1	1	Y
94	Rolling average size	Size of rolling average	RW	4	4	Y

## Command descriptions

### 0. Product name

A 16 byte string indicating the product model name. This will always be SF000/B followed by a null terminator. You can use this to verify the SF000/B is connected and operational over the selected interface.

Read	Write	Persists
16 byte string	-	-

### 1. Hardware version

The hardware revision number as a uint32.

Read	Write	Persists
uint32	-	-

### 2. Firmware version

The version of currently installed firmware is represented as 4 bytes. This can be used to identify the product for API compatibility. The [product support](#) section details which firmware versions this document applies to.

1	2	3	4
Patch	Minor	Major	Reserved

Read	Write	Persists
4 bytes	-	-



### 3. Serial number

A 16 byte string (null terminated) of the serial identifier assigned during production.

Read	Write	Persists
16 byte string	-	-

### 7. UTF8 text message

*Serial interface only*

A null terminated ASCII string. The SF000/B will send this command when it needs to communicate a human readable message.

Read	Write	Persists
-	-	-

### 9. User data

This command allows 16 bytes to be stored and read for any purpose.

Read	Write	Persists
16 byte	16 bytes	Yes

### 10. Token

Current safety token required for performing certain operations. Once a token has been used it will expire and a new token is created.

Read	Write	Persists
uint16	16 bytes	Yes

### 12. Save parameters

Several commands write to parameters that can persist across power cycles. These parameters will only persist once the `Save parameters` command has been written with the appropriate `token`. The safety token is used to prevent unintentional writes and once a successful save has completed the token will expire.

Read	Write	Persists
-	uint16	-

### 14. Reset

Writing the `safety token` to this command will restart the SF000/B.

Read	Write	Persists
-	uint16	-

**27. Distance output**

This command configures the data output when using the 44. Distance data command. Each bit toggles the output of specific data.

Bit	Output
0	First return raw
1	First return filter
2	First return strength
3	Last return raw
4	Last return filter
5	Last return strength
6	Background noise
7	Temperature
8	Yaw angle

Read	Write	Persists
uint32	uint32	No

### 30. Stream

The SF000/B can continuously output data without individual request commands being issued. Reading from the `Stream` command will indicate what type of data is being streamed. Writing to the `Stream` command will set the type of data to be streamed.

Value	Streamed data
0	disabled
5	44. Stream distance data cm

Streaming commands will only output on the serial and USB interface. While it is possible to read and write the Stream command over I2C, the resulting streamed data will not be retrievable.

Read	Write	Persists
uint32	uint32	No

#### 44. Distance data in cm

This command contains distance data as measured by the SF000/B. The data included will vary based on the configuration of the 27. Distance output command.

This command can be read at any time however if 30. Stream is set to 5 then this command will automatically output at the measurement update rate.

The data will be packed in order based on the bits set in the `Distance output` parameter.

Data output bit	Description	Size
0	First return raw [cm]	int16
1	First return filtered [cm]	int16
2	First return strength [%]	int16
3	Last return raw [cm]	int16
4	Last return filtered [cm]	int16
5	Last return strength [%]	int16
6	Background noise	int16
7	Temperature [1/100 degC]	int16
8	Yaw angle [1/100 deg]	int16

Read	Write	Persists
<i>varies</i>	-	-

**45. Distance data in mm**

This command contains distance data as measured by the SF000/B. The data included will vary based on the configuration of the 27. Distance output command. The data will be packed in order based on the bits set in the Distance output parameter.

Data output bit	Description	Size
0	First return raw [mm]	int32
1	First return filtered [mm]	int32
2	First return strength [%]	int32
3	Last return raw [mm]	int32
4	Last return filtered [mm]	int32
5	Last return strength [%]	int32
6	Background noise	int32
7	Temperature [1/100 degC]	int32
8	Yaw angle [1/100 deg]	int32

Read	Write	Persists
<i>varies</i>	-	-

### 50. Laser firing

Reading this command will indicate the current laser firing state. Writing to this command will enable or disable the firing of the laser.

Value	Description
0	Disabled
1	Enabled

Read	Write	Persists
uint8	uint8	No

### 57. Temperature

Reading this command will return the temperature in 100ths of a degree.

Read	Write	Persists
uint32	-	-

**66. Update Rate**

This controls the update rate of the SF000/B.

Reading this command will return the current update rate. Writing this command will set the update rate.

The update rate is selected from the following table:

Command value	Update rate samples/second
1	48
2	55
3	64
4	77
5	97
6	129
7	194
8	388

Read	Write	Persists
uint8	uint8	Yes



**74. Noise**

Reading this command will return the level of background noise.

Read	Write	Persists
uint32	-	-

**75. Zero offset**

Changing this offset value will change the Zero distance position for the output. This value is written and read in mm.

Read	Write	Persists
int32	int32	Yes

**76. Lost signal counter**

The lost signal counter is used to determine the number of lost signal returns that needs to be generated before a lost signal indication is output on the distance value. The lost signal indication on the distance output value is -1000.

Read	Write	Persists
uint32	uint32	Yes

**79. Baud rate**

The baud rate as used by the serial interface. This parameter only takes effect when the serial interface is first enabled after power-up or restart.

Reading this command will return the baud rate. Writing to this command will set the baud rate.

Value	Baud rate [bps]
0	9600
1	19200
2	38400
3	57600
4	115200
5	230400
6	460800
7	921600

Read	Write	Persists
uint8	uint8	Yes

**80. I2C address**

Reading this command will return the I2C address. Writing this command will set the I2C address.

The I2C address value is in decimal.

Read	Write	Persists
uint8	uint8	Yes

**82. Median filter enable**

Reading this command will return the status of the median filter. Writing this command will set the status of the median filter.

Value	Description
0	disabled
1	enabled

Read	Write	Persists
uint8	uint8	Yes

### 83. Median filter size

Reading this command will return the size of the median filter. Writing this command will set the size of the median filter.

The valid range is 3 to 32.

Read	Write	Persists
uint32	uint32	Yes

### 84. Smoothing filter enable

Reading this command will return the status of the smoothing filter. Writing this command will set the status of the smoothing filter.

Value	Description
0	disabled
1	enabled

Read	Write	Persists
uint8	uint8	Yes

**85. Smoothing factor**

Reading this command will return the strength of the smoothing filter. Writing this command will set the strength of the smoothing filter.

The valid range is 0 to 100.

Read	Write	Persists
uint32	uint32	Yes

**93. Rolling average enable**

Reading this command will return the status of the rolling average filter. Writing this command will set the status of the rolling average filter.

Value	Description
0	disabled
1	enabled

Read	Write	Persists
uint8	uint8	Yes

#### 94. Rolling average size

Reading this command will return the size of the rolling average filter. Writing this command will set the size of the rolling average filter.

The valid range is 2 to 32.

Read	Write	Persists
uint32	uint32 2	Yes

---

## Document revision

Revision	Date	Comments
Rev 5	2022/02/15	PDF format of guide release
Rev 4	2022/01/12	Introduction of online guide format
Rev 3	2021/06/17	Typo on page 1 and 3 constraint to constrained
Rev 2	2021/02/17	Breakout Board section has been inserted
Rev 1	2020/11/18	FDA accession number added,notification that I2C pins 2 and 4 need to be used as for SDA
Rev 0	2020/08/03	First edition