Table of contents

1. About this document

1.1. Product support

This document supports the following devices (See Firmware change log for firmware details):

Product	Hardware	Firmware	Supported
SF40/C	1	1.4.0	Yes
SF40/C	1	1.3.0	Yes
SF40/C	1	1.2.0	Yes
SF40/C	1	1.1.2	Yes
SF40/C	1	1.1.1	Yes
SF40/C	1	1.1.0	Yes
SF40/C	1	1.0.1	No

1.2. Revision history

Revision	Date	Notes
6	04/02/2019	Updated dimension drawing.
5	23/01/2019	Added change log for version V1.4.0. Updated link to Upgrader tool.
4	08/11/2018	Added change log for version V1.3.0. Updated link to Upgrader tool. Modified command `Distance [105]` .
3	02/10/2018	Added change log for version V1.1.2 & V1.2.0. Updated link to Upgrader tool.
2	27/09/2018	Added change log for version V1.1.1. Command `Distance [105]` now shows the correct read/write byte count.
1	26/09/2018	Added change log for version V1.1.0. Added link/description for Upgrader tool. Modified command `Distance [105]` . Updated FDA approval. Updated range and height specifications.
0	4/09/2018	Initial pre-release draft.

1.3. 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 the accuracy of the information contained herein.

2. Firmware change log

2.1. Upgrading

NOTE: The Upgrader tool only supports Microsoft Windows® at this time.

1. Download the LightWare Upgrader tool here: http://support.lightware.co.za/LightWareUpgrader-1.26.0.rar
2. Unzip the downloaded file to a location on your PC.
3. Run the file `LightWareUpgrader.exe` in the unzipped folder.
4. Connect your LightWare device via USB to your PC.
5. Click the COM port that appears.
6. If the device is not the latest version you can click the `Upgrade` button to begin the process.
7. Wait until the upgrade has completed successfully and click `OK` .

2.2. 1.4.0

Changes

- Improved noise issues.

2.3. 1.3.0

Changes

- Modified `Distance [105]` command to report the closest obstacle angle in 10ths of a degree.

2.4. 1.2.0

Features

- Added a minimum distance parameter for flight controller compatibility mode.

2.5. 1.1.2

Fixes

- Flight controller compatibility mode now correctly uses the orientation setting.

2.6. 1.1.1

Fixes

- Version `1.1.0` had the LWNX protocol disabled by default. It is now active by default.

2.7. 1.1.0

Features

- Modified `Distance [105]` command to accept a minimum range parameter.
- Modified `Distance [105]` command to output angle to closest distance measurement.
- Less aggressive power draw on the 5V line during start-up.
- Supports flight controller proximity detection when disabling LWNX mode.

Fixes

- The `Point start index` field of the `Distance output [48]` command now correctly takes the `Output rate [108]` into account.

2.8. 1.0.1

Notes

- Initial release.

3. Safety instructions

3.1. Overview

The SF40/C is a laser rangefinder that 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 at a distance of less than 15 meters. Notwithstanding the safety rating, avoid looking into the beam and switch the unit off when working in the area.

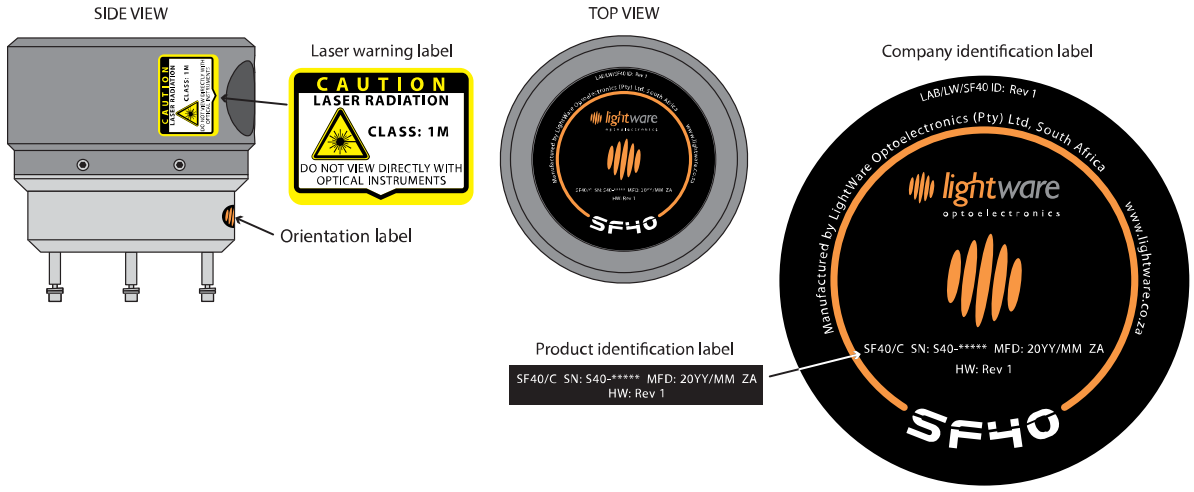
CAUTION -- The use of optical instruments with this product will increase eye hazard.

The SF40/C should not be disassembled or modified in any way. The laser eye safety rating depends on the mechanical integrity of the optics and electronics so if these are damaged do not continue using the SF40/C. There are no user serviceable parts and maintenance or repair must only be carried out by the manufacturer or a qualified service agent.

No regular maintenance is required for the SF40/C but if the lenses start to collect dust then they may be wiped with suitable lens cleaning materials. Make sure that the SF40/C is switched OFF before looking into the lenses.

The SF40/C should be mounted using the four holes provided in the circuit board. Do not hold or clamp the lens tubes as this may cause damage and adversely affect the laser safety rating.

3.2. Labeling



3.3. Laser radiation information

Specification	Value / AEL	Notes
Eye safety classification	Class 1M	
Laser wavelength	905 nm	
Pulse width	< 20 ns	
Pulse frequency	< 36 kHz	
Peak power	< 10 W	50 millimeter aperture at 2 meters
Average power	< 0.6 mW	7 millimeter aperture
Average energy per pulse	< 300 nj	
NOHD	15 m	Distance beyond which binoculars with may be used safely

4. Specifications

	SF40/C
Range	0.2 ... 100 meters (natural targets)
Revolutions per second	5.5
Resolution	3 cm
Readings per revolution	3638
Readings per second	20010
Motor power supply voltage	12.0 V DC (+-10%)
Motor power supply current	0.5 A (max)
Laser power supply voltage	5.0 V DC (+-10%)
Outputs & interfaces	Serial UART 3.3 V TTL
Dimensions	70 mm (height) x 79 mm (largest diameter)
Weight	256 g
Laser power	20 W (peak), 11 mW (average), Class 1M
Optical aperture	51 mm
Beam divergence	0.2°
Operating temperature	0 ... 40°C
Approvals	FDA: 1410968-002 (2018/09)

5. Functional description

5.1. Overview

The SF40/C uses a scanning laser rangefinder to measure on a 360 degree disc with a radius of 100 meters. Collected data is stored in memory and continually refreshed as the laser scans around. The speed of rotation is 5.5 revolutions per second at a resolution of +- 3 cm. This data can be used to trigger predefined alarm zones, give information about target distances in any direction or streamed to a host controller with selectable output rates.

5.2. Data streaming

Measurement data accumulated by the SF40/C can be output to a host controller for immediate or deferred analysis. The rate at which data is output can be selected as 20010, 10005, 6670 or 2001 points per second.

5.3. Alarms

Seven configurable alarms zones can be set within the measuring plane to alert obstacle proximity. Each zone can be set with an individualized alarm distance, angular width and aiming direction. Typically, one zone would cover 360 degrees around the vehicle at close range to alert when people get too close to the moving parts. Additionally, a forward looking alarm zone is used to detect obstacles in the direction of motion. Other alarm zones can check that specific directions are clear of obstructions before course changes are made.

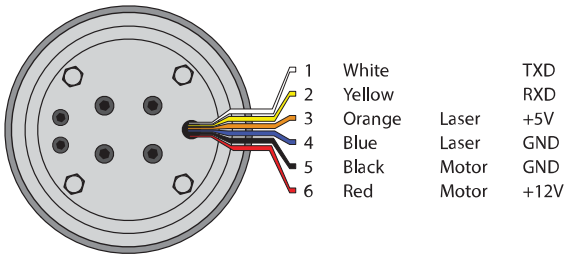
The status of the alarms can be read from the serial port through a command or from the streaming data. Once the SF40/C is running the alarms are updated continuously without the need for any external commands.

5.4. Virtual laser range finder

The virtual laser range finder (VLRF) tool is used to find the distance in any direction on the measuring plane. VLRF can assist with keeping station at a fixed distance from a target or measuring how far away an obstacle is. Any number of VLRFs can be created that aim in different directions. This allows for accurate position holding within a confined space and provides confirmation of GPS location using adjacent buildings or other known structures as reference points.

6. Hardware

6.1. Connector wiring



Serial UART (Pins 1 & 2)

- 3.3 V TTL UART
- Supports baud rates from 115200 to 921600

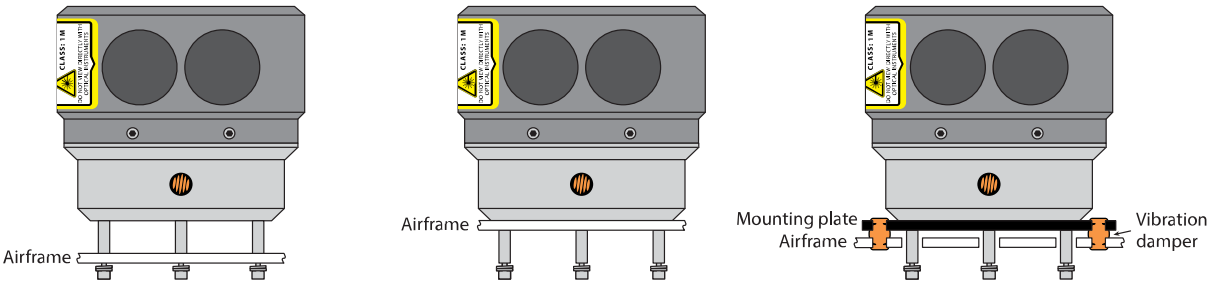
Laser power supply (Pins 3 & 4)

- Power supply should provide 5 V +- 10%
- Power supply current capacity should be at least 0.2 A

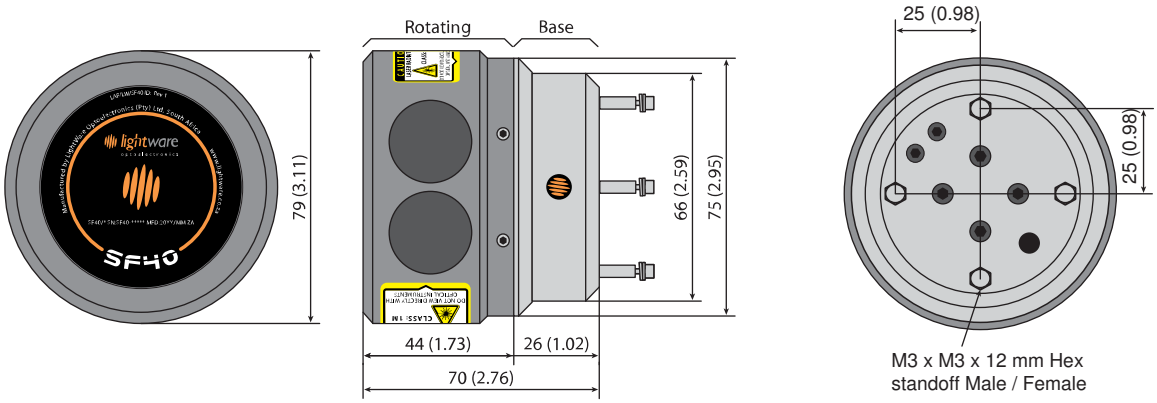
Motor power supply (Pins 5 & 6)

- Power supply should provide 12 V +- 10%
- Power supply current capacity should be at least 0.5 A

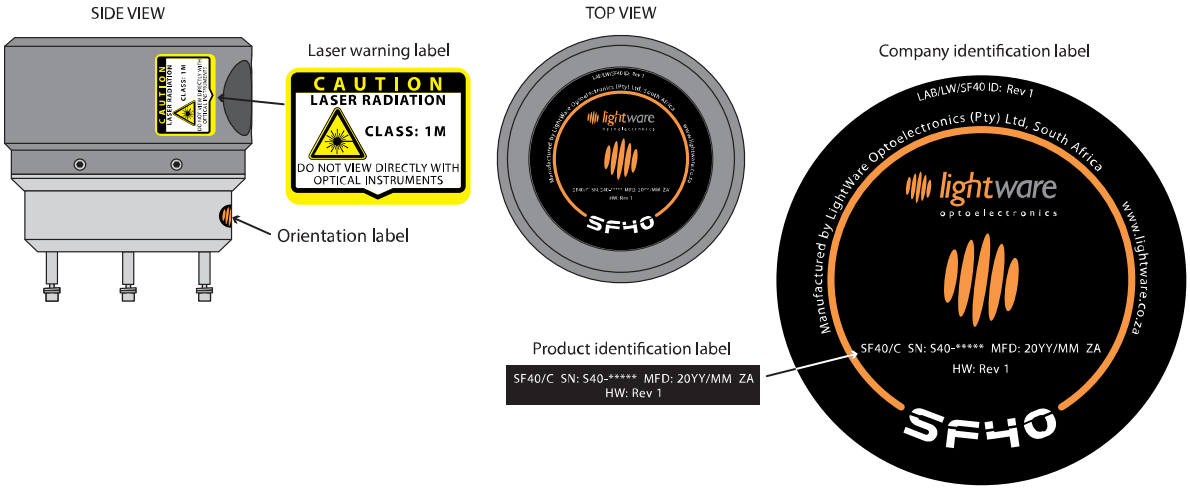
6.2. Mounting on an airframe



6.3. Dimension drawings



6.4. Labeling


SIDE VIEW
Laser warning label
CAUTION
LASER RADIATION
CLASS: 1M
DO NOT VIEW DIRECTLY WITH OPTICAL INSTRUMENTS
Orientation label
TOP VIEW
Company identification label
LAB/LW/SF40 ID: Rev 1
Manufactured by LightWare Optoelectronics (Pty) Ltd, South Africa
www.lightware.co.za
SF40/C SN: S40-***** MFD: 20YY/MM ZA
HW: Rev 1
SF40
Product identification label
SF40/C SN: S40-***** MFD: 20YY/MM ZA
HW: Rev 1

7. Serial interface

7.1. Overview

The SF40/C uses a 3.3 V (5 V tolerant) TTL serial UART for communication. Communication is performed using encapsulated packets for both sending and receiving data. Every packet that is sent to the SF40/C is known as a `request` and a request will always be replied to with a `response`. There are cases where the SF40/C will send a request packet to the host, these packets are considered `streaming` packets do not require a response from the host.

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: `921600`
- Data: `8 bit`
- Parity: `none`
- Stop: `1 bit`
- Flow control: `none`

> **The SF40/C can operate in baud rates as low as `115200` but maximum data output will require higher rates.**

7.2. 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 veryify that the payload length is sane (0 to 1023 bytes) 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
	Flags high byte								Flags low byte							
	Payload length (1 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, therefore the payload length has a minimum of 1.

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 checkum calculation.

7.3. Checksum

Each packet has a 2 byte checksum which is used to validate data integrity. The algorithm is `CRC-16-CCITT` `0x1021` (idendtical to the one used for the XMODEM protocol).

The CRC must be correctly formed for the SF40/C 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

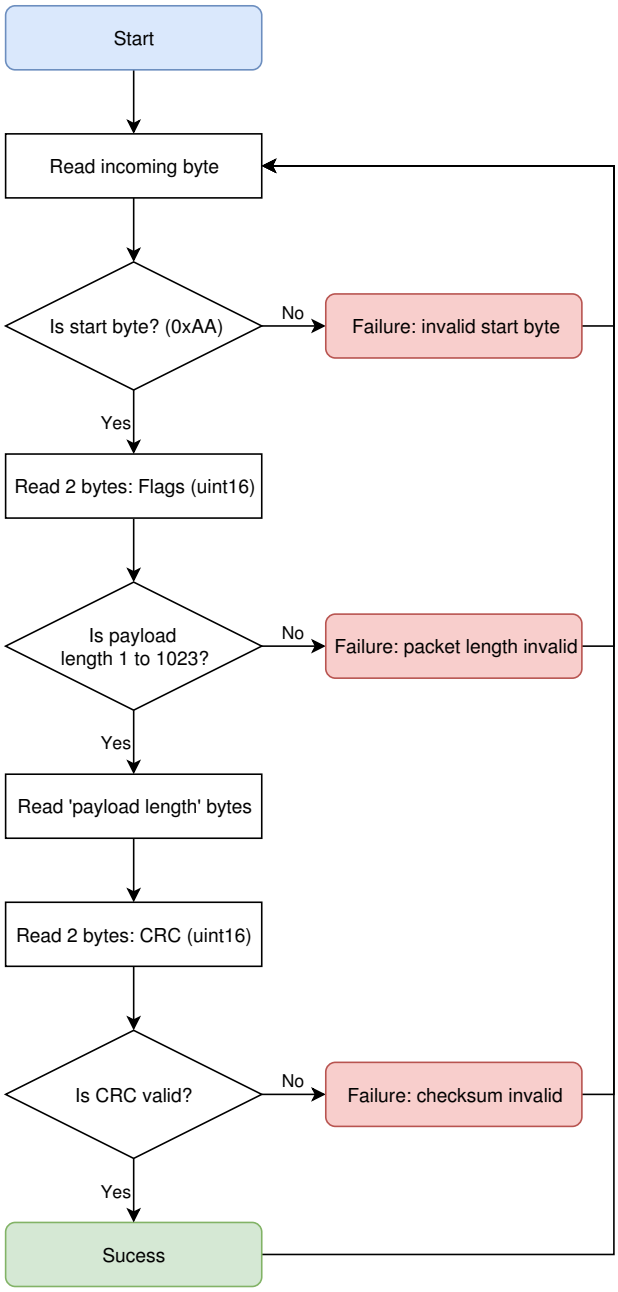
```
function createCRC(data, size) {
    let crc = 0;

    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;
}
```

7.4. Receiving packets

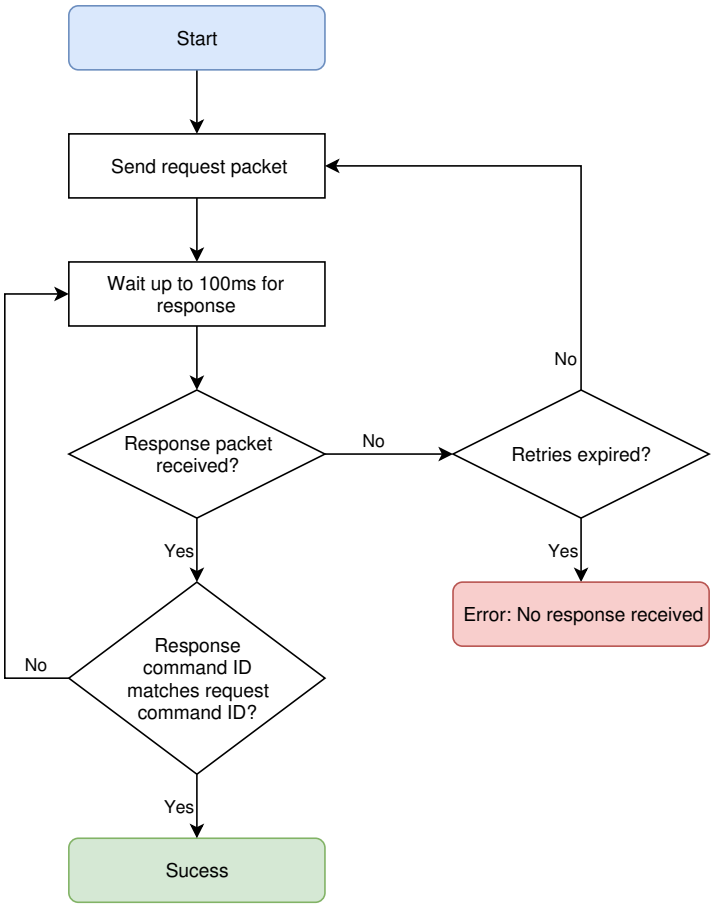
Here is the process for reading the raw serial byte stream and identifying packets. Once a packet has been successfully read it can be processed based on its command ID.



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.

7.5. Handling request & response

Every request sent to the SF40/C 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.

8. Command list

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 persistable parameters	W	-	2	-
14	Reset	Restart the unit	W	-	2	-
16	Stage firmware	Upload firmware file pages	RW	4	130	-
17	Commit firmware	Apply staged firmware	RW	4	0	-
20	Incoming voltage	Measured incoming voltage	R	4	-	-
30	Stream	Current data stream type	RW	4	4	N
48	Distance output	Streaming distance data	-	-	-	-
50	Laser firing	Laser firing state	RW	1	1	N
55	Temperature	Measured temperature	R	4	-	-
90	Baud rate	Serial baud rate	RW	1	1	Y
105	Distance	Single direction distance	RW	12	6	N
106	Motor state	Motor state	R	1	-	-
107	Motor voltage	Measured motor voltage	R	2	-	-
108	Output rate	Streaming point output rate	RW	1	1	Y
109	Forward offset	Forward orientation offset	RW	2	2	Y
110	Revolutions	Number of revolutions	R	4	-	-
111	Alarm state	Triggered state of alarms	R	1	-	-
112	Alarm 1	Alarm 1 configuration	RW	7	7	Y
113	Alarm 2	Alarm 2 configuration	RW	7	7	Y
114	Alarm 3	Alarm 3 configuration	RW	7	7	Y
115	Alarm 4	Alarm 4 configuration	RW	7	7	Y
116	Alarm 5	Alarm 5 configuration	RW	7	7	Y
117	Alarm 6	Alarm 6 configuration	RW	7	7	Y
118	Alarm 7	Alarm 7 configuration	RW	7	7	Y

9. Detailed command descriptions

9.1. Product name [0]

A `16 byte string` indicating the product model name. This will always be `SF40` followed by a null terminator. You can use this to verify the SF40/C is connected and operational over the selected interface.

Read	Write	Persists
`16 byte string`	-	-

9.2. Hardware version [1]

The hardware revision number as a `uint32` .

Read	Write	Persists
`uint32`	-	-

9.3. Firmware version [2]

The version of currently installed firmware represented as `4 bytes` . This can be used to identify the product for API compatibility.

1	2	3	4
Patch	Minor	Major	Reserved

Read	Write	Persists
`4 bytes`	-	-

9.4. Serial number [3]

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

Read	Write	Persists
`16 byte string`	-	-

9.5. UTF8 text message [7]

Serial interface only

A null terminated ASCII string. The SF40/C will send this command when it needs to communicate a human readable message.

Read	Write	Persists
-	-	-

9.6. User data [9]

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

Read	Write	Persists
`16 bytes`	`16 bytes`	Yes

9.7. Token [10]

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`	-	-

9.8. Save parameters [12]

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`	-

9.9. Reset [14]

Writing the safety `token` to this command will restart the SF40/C.

Read	Write	Persists
-	`uint16`	-

9.10. Stage firmware [16]

The first part of uploading firmware to the SF40/C is to stage the data. This command accepts pages of the firmware, each `128 bytes` long, and an index to indicate which page is being uploaded. Pages are created by dividing the firmware upgrade file into multiple 128 byte chunks.

When writing to this command, use the following data structure:

Byte offset	Data type	Name	Description
0x00	`int16`	Page index	The index of the page currently being uploaded
0x02	`128 bytes`	Page data	The byte data of the page currently being uploaded

When reading this command, or analyzing its response after writing a page, the packet will contain an `int32` error code:

Value	Description
0 to 1000	Index of successfully written page
-1	Page length is invalid
-2	Page index is out of range
-3	Flash failed to erase
-4	Firmware file has invalid header
-5	Flash failed to write
-6	Firmware is for a different hardware version
-7	Firmware is for a different product

Read	Write	Persists
`int32`	`130 bytes`	-

9.11. Commit firmware [17]

The second part of uploading firmware to the SF40/C is to commit the staged data. Once the firmware data has been fully uploaded using the Stage firmware [16] command, then this command can be written to (with 0 bytes).

When reading this command, or analyzing its response after writing, the packet will contain an `int32` error code:

Value	Description
-1	Firmware integrity check failed
1	Firmware integrity check passed and firmware committed

Once the firmware is committed, a reboot is required to engage the new firmware. This can be done by cycling power to the SF40/C or by sending the Reset [14] command.

After the unit has rebooted the firmware version should be checked to ensure the new firmware is installed.

Read	Write	Persists
`int32`	`0 bytes`	-

9.12. Incoming voltage [20]

The incoming voltage is directly measured from the incoming 5 V line. The response from reading this command is in counts. To convert the counts to voltage use the following equation:

`voltage` = (`counts` / 4095.0) x 2.048 x 5.7

Read	Write	Persists
`uint32`	-	-

9.13. Stream [30]

The SF40/C 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
3	Distance output [48]

Read	Write	Persists
`uint32`	`uint32`	No

9.14. Distance output [48]

This command contains measurement results over a period of time. If Stream [30] is set to `3` then this command will automatically output as measurements are taken.

Please note that the rate at which this command is output will vary based on Output rate [108].

Each distance output packet contains distance measurement data for a consecutive series of points. There is a maximum of 200 points per packet. A packet will only contain points from the same revolution.

The data is composed as follows:

Byte offset	Data type	Name	Description
0x00	`1 byte`	Alarm state	State of each alarm as described in Alarm state [111]
0x01	`uint16`	Points per second	Number of points per second.
0x03	`int16`	Forward offset	Orientation offset as described in Forward offset [109]
0x05	`int16`	Motor voltage	Motor voltage as described in Motor voltage [107]
0x07	`uint8`	Revolution index	Increments as each new revolution begins. Note that this value wraps to 0 after 255.
0x08	`uint16`	Point total	Total number of points this revolution.
0x0A	`uint16`	Point count	Number of points in this packet.
0x0C	`uint16`	Point start index	Index of the first point in this packet.
0x0E	`Point count` x `int16`	Point distances	Array of distances [cm] for each point.

By using the `Point start index` and `Point total` you can determine the angle in degrees that each point in the packet was measured at.

- `Point index` of `Nth` point in packet = `Point start index` + `N`
- `Point angle [degrees]` = (`Point index` / `Point total`) * 360

Read	Write	Persists
-	-	-

9.15. Laser firing [50]

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

9.16. Temperature [55]

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

Read	Write	Persists
`uint32`	-	-

9.17. Baud rate [90]

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]
4	115200
5	230400
6	460800
7	921600

Read	Write	Persists
`uint8`	`uint8`	Yes

9.18. Distance [105]

NOTE: The data structure for reading/writing this command has changed since firmware 1.0.1.

Reading this command will return the `average`, `closest` and `furthest` distance within an angular view pointing in a specified direction. When writing to this command you can specify the direction and angular width of the view that the results are calculated from. The response to the write command is the same as the read command. Readings below the `Minimum distance` will be ignored.

Data response when reading or writing:

Byte offset	Data type	Description
0x00	`int16`	Average distance [cm]
0x02	`int16`	Closest distance [cm]
0x04	`int16`	Furthest distance [cm]
0x06	`int16`	Angle to closest distance [10ths of a degree]
0x08	`uint32`	Calculation time [us]

Data for request when writing:

Byte offset	Data type	Description
0x00	`int16`	Direction [degrees]
0x02	`int16`	Angular width [degrees]
0x04	`int16`	Minimum distance [cm]

Read	Write	Persists
`12 bytes`	`6 bytes`	N

9.19. Motor state [106]

Reading this command will return the current state of the motor. This can be useful to debug or check start-up conditions.

Value	Description
1	Motor is preparing for start-up.
2	Motor is waiting for first 5 revolutions to occur.
3	Motor is running normally.
4	Motor has failed to communicate.

Read	Write	Persists
`uint8`	-	-

9.20. Motor voltage [107]

Reading this command will return the voltage drawn by the motor in mV.

Read	Write	Persists
`uint16`	-	-

9.21. Output rate [108]

The output rate controls the amount of data sent to the host when distance output streaming is enabled.

Value	Points per second
0	20010
1	10005
2	6670
3	2001

Read	Write	Persists
`uint8`	`uint8`	Yes

9.22. Forward offset [109]

The forward offset affects the position of the `0 degree` direction. The `orientation` label on the front of the SF40/C marks the default `0 degree` direction.

Read	Write	Persists
`int16`	`int16`	Yes

9.23. Revolutions [110]

Reading this command will return the number of full revolutions since start-up.

Please note that this value will reset to zero after 4294967295 revolutions.

Read	Write	Persists
`uint32`	-	-

9.24. Alarm state [111]

Reading this command will return a byte with the current state of all alarms. Each bit represents 1 of the 7 alarms, if the bit is set then the alarm is currently triggered. The most significant bit is set when any alarm is currently triggered.

Bit	Description
0	Alarm 1
1	Alarm 2
2	Alarm 3
3	Alarm 4
4	Alarm 5
5	Alarm 6
6	Alarm 7
7	Any alarm

Read	Write	Persists
`1 byte`	-	-

9.25. Alarm 1 [112]

This is the same for all 7 alarms.

By reading this command the configuration for this alarm is retrieved as follows:

Byte offset	Data type	Name	Description
0x00	`uint8`	Enabled	`1` is enabled, `0` is disabled.
0x01	`int16`	Direction	Primary direction in degrees.
0x03	`int16`	Width	Angular width in degrees around the primary direction.
0x05	`int16`	Distance	Distance at which alarm is triggered.

The same data bytes as specified above can be written to this command to set the alarm configuration.

Read	Write	Persists
`7 bytes`	`7 bytes`	Yes

9.26. Alarm 2 [113]

By reading this command the configuration for this alarm is retrieved as follows:

Byte offset	Data type	Name	Description
0x00	`uint8`	Enabled	`1` is enabled, `0` is disabled.
0x01	`int16`	Direction	Primary direction in degrees.
0x03	`int16`	Width	Angular width in degrees around the primary direction.
0x05	`int16`	Distance	Distance at which alarm is triggered.

The same data bytes as specified above can be written to this command to set the alarm configuration.

Read	Write	Persists
`7 bytes`	`7 bytes`	Yes

9.27. Alarm 3 [114]

By reading this command the configuration for this alarm is retrieved as follows:

Byte offset	Data type	Name	Description
0x00	`uint8`	Enabled	`1` is enabled, `0` is disabled.
0x01	`int16`	Direction	Primary direction in degrees.
0x03	`int16`	Width	Angular width in degrees around the primary direction.
0x05	`int16`	Distance	Distance at which alarm is triggered.

The same data bytes as specified above can be written to this command to set the alarm configuration.

Read	Write	Persists
`7 bytes`	`7 bytes`	Yes

9.28. Alarm 4 [115]

By reading this command the configuration for this alarm is retrieved as follows:

Byte offset	Data type	Name	Description
0x00	`uint8`	Enabled	`1` is enabled, `0` is disabled.
0x01	`int16`	Direction	Primary direction in degrees.
0x03	`int16`	Width	Angular width in degrees around the primary direction.
0x05	`int16`	Distance	Distance at which alarm is triggered.

The same data bytes as specified above can be written to this command to set the alarm configuration.

Read	Write	Persists
`7 bytes`	`7 bytes`	Yes

9.29. Alarm 5 [116]

By reading this command the configuration for this alarm is retrieved as follows:

Byte offset	Data type	Name	Description
0x00	`uint8`	Enabled	`1` is enabled, `0` is disabled.
0x01	`int16`	Direction	Primary direction in degrees.
0x03	`int16`	Width	Angular width in degrees around the primary direction.
0x05	`int16`	Distance	Distance at which alarm is triggered.

The same data bytes as specified above can be written to this command to set the alarm configuration.

Read	Write	Persists
`7 bytes`	`7 bytes`	Yes

9.30. Alarm 6 [117]

By reading this command the configuration for this alarm is retrieved as follows:

Byte offset	Data type	Name	Description
0x00	`uint8`	Enabled	`1` is enabled, `0` is disabled.
0x01	`int16`	Direction	Primary direction in degrees.
0x03	`int16`	Width	Angular width in degrees around the primary direction.
0x05	`int16`	Distance	Distance at which alarm is triggered.

The same data bytes as specified above can be written to this command to set the alarm configuration.

Read	Write	Persists
`7 bytes`	`7 bytes`	Yes

9.31. Alarm 7 [118]

By reading this command the configuration for this alarm is retrieved as follows:

Byte offset	Data type	Name	Description
0x00	`uint8`	Enabled	`1` is enabled, `0` is disabled.
0x01	`int16`	Direction	Primary direction in degrees.
0x03	`int16`	Width	Angular width in degrees around the primary direction.
0x05	`int16`	Distance	Distance at which alarm is triggered.

The same data bytes as specified above can be written to this command to set the alarm configuration.

Read	Write	Persists
`7 bytes`	`7 bytes`	Yes