

## Gobotics Sonar-I Protocol Document (Rev. A5)

### Sonar Serial Protocol I

**Mode 1)** The sonar module enters into Mode 1 upon power up or reset. The sonar module automatically sends out a sonar distance message every second in inches.

**Mode 2)** The sonar module enters into Mode 2 when it receives a valid command message. Once the sonar module enters into Mode 2 it remains in Mode 2 until power cycled. Upon entering Mode 2, the module stops sending data automatically and waits for a valid command.

#### Example data values:

Data is in BCD format represented under HIGH\_DATA and LOW\_DATA respectively. BCD data format for millimeters is XXXX. For inches, the data format is XXX.Y, where Y is 1/10 of an inch.

##### Example 1: (Millimeters)

10 millimeters = 00 10

100 millimeters = 01 00

##### Example 2: (Inches)

10.0 Inches = 01 00

10.5 Inches = 01 05

## **MODE 1**

### **Response Message:**

#### **0xFA, High\_data, Low\_data, Status\_byte, Chksum**

Where:

0xFA is header value;

High\_data and Low\_data are distance data bytes (in inches by default);

Distance data bytes are in BCD format. (0000 – 9999) See page 3.

Status\_byte is bitmapped as follows:

Bit 0 = Mode bit, set if in Mode 2, clear if in Mode 1

Bit 1 = 1 if result is a multiple (averaged) ping, 0 if single ping

Bit 2 = 0 if requested ping, 1 if auto-ping

Bit 3 = 0 if inches, and 1 if in millimeters

Bit 4 = 1 if response for COM test, otherwise =0

Bit 5 = 1 result is in error (optional)

Chksum: This is checksum of the response message ANDed by 0x7F.

In Mode 1, the module still listens to Mode 2 query commands. If it receives a Mode 2 command, the module switches to Mode 2, and waits for a command.

## **MODE 2**

### **Command Query Format:**

#### **0xF5, Command\_byte, Data\_byte, Chksum**

Where:

0xF5: This is the header of the command message;

Command\_byte: This value contains the command type;

Data\_byte: This byte contains data if needed by command, otherwise, it is zero;

Chksum: This is checksum of the command string ANDed by 0x7F.

#### **Commands (byte code):**

0x01: Ping once and return a measurement

0x03: Ping multiple times, then stop, return value with best accuracy  
Data byte: n (BCD, contains the number of pings to average).

0x04: Ping automatically, continuously return value every n pings  
Data byte: n (ex: if n=0x20, then return message after 20 pings)

0x05: Stop pinging automatically.

0x08: Units command, set data\_byte=1 for millimeter, =0 for inches;

0x10: respond immediately for COM test, bit 4 =1 in status, all data =0.

### Command sequence examples:

Mode 1: Mode 1 is the mode the module enters at start-up. This is indicated by bit0=0 of the status byte.

Example, target is at 100 inches:

Module sends:

0xFA, 0x01, 0x00, 0x04, 0x7F

:

0xFA, 0x01, 0x00, 0x04, 0x7F

:

0xFA, 0x01, 0x00, 0x04, 0x7F

Notes:

1. The module sets bit 2 of status byte since the mode is auto-ping at start-up.
2. If error is detected, then bit 5 is set.
3. All other status bits are zero in mode 1.

Mode 2: The module enters mode 2 if it receives a valid command message.

Computer sends command to ping once:

0xF5, 0x01, 0x00, 0x76

Module responds with message after the echo is returned (ex: 24 inches):

0xFA, 0x00, 0x24, 0x01, 0x1F

Module waits for command (it's now in mode 2).

Computer sends command to ping plus change scale to millimeters:

0xF5, 0x09, 0x00, 0x7E

Module responds with the message after the echo is returned (ex: 112 mm)

0xFA, 0x01, 0x12, 0x09, 0x16

**Allowable values for the sonar result:**

1. Maximum distance: (pings that do not return an echo return the maximum range and return with status error bit 5 =1. (“9999” with error bit set).
2. Minimum distance: If it is possible to detect “too close”, a value of 0000 is reported, and the status error bit is set (bit 5 of status byte). (support error bit if possible).

**Notes about the checksum:**

We have modified the checksum to be the sum of all bytes in a message, then the checksum result is ANDed by the value 0x7F. This clears the high order bit so that the checksum can never be interpreted as a header.

**Checksum:** Add all bytes including header and command, take least significant byte only, then AND the byte with 0x7f.