Lovense Hardware and Protocol Documentation

Release 1.0.0

February 22, 2017

Contents:

CHAPTER 1

Models

Each Lovense toy is identifiable by a Model Number assigned to the toy. This model number is used in bluetooth discovery, as well as in libraries to discern toy capabilities.

Name	Form Factor	Model	Capabilities
Nora	Rabbit Vibe	A or C	Vibration, Rotation, Accelerometer
Max	Onahole	B	Vibration, Inflation, Accelerometer
Lush	Vaginal Insertable Vibe	S	Vibration
Hush	Buttplug	Z	Vibration
Edge	Prostate Stimulator	Unknown	Vibration
Ambi	Bullet Vibe	Unknown	Vibration
Domi	Magic Wand Style Vibe	Unknown	Vibration
Osci	G-Spot Vibrator	Unknown	Vibration

CHAPTER 2

Bluetooth Details

While all lovense toys use the same protocol, they can communicate over bluetooth differently, depending on when they were released.

When discovering or pairing with Lovense toys, toys identify using the format "LVS-?001". The ? will be a letter pertaining to the model specifier. For instance, when discovering the Hush Buttplug, it will show up as "LVS-Z001", as "Z" is the model code for the Hush.

Bluetooth 2.0 Toys

The first toys released by Lovense used both Bluetooth 2.0 SPP (emulating a serial port) and Bluetooth LE. This was most likely due to the sparse mobile support of BTLE when they were released.

These toys include:

- Max
- Nora

When paired with a system via Bluetooth 2.0, these toys identify as a serial port. These toys are also capable of using Bluetooth 4.0, as outlined in the next section.

Bluetooth LE Toys

Starting with the Lush, all toys released by Lovense use only Bluetooth LE.

These toys include:

- Lush
- Hush
- Edge
- Ambi
- Osci
- Domi

These toys have GATT characteristics to mimic the RX/TX setup of the serial port style control of the old toys. The GATT service and characteristic IDs differ between the Max/Nora and all toys released afterward.

Service UUID (Max/Nora):

```
0000fff0-0000-1000-8000-00805f9b34fb
```

TX Characteristic UUID (Max/Nora):

```
0000fff1-0000-1000-8000-00805f9b34fb
```

TX Characteristic UUID (Max/Nora):

```
0000fff2-0000-1000-8000-00805f9b34fb
```

Service UUID (All Others):

```
6e400001-b5a3-f393-e0a9-e50e24dcca9e
```

TX Characteristic UUID (All Others):

```
6e400002-b5a3-f393-e0a9-e50e24dcca9e
```

RX Characteristic UUID (All Others):

```
6e400003-b5a3-f393-e0a9-e50e24dcca9e
```

CHAPTER 3

Lovense Protocol

Protocol Rules

- Commands and replies are strings, using semicolons to mark their end.
- All commands start with a command identifier word, then possibly either specifiers or levels, delimited by colons. e.g. “Vibrate:5;” would set vibration to 5.
- Replies are in the context of the command (i.e. sending “Battery;” will just return a number, like “85;”), but can still be colon delimited lists.
- Commands that do not return a context specific value will return “OK;” on success, “ERR;” on error.

Command List

The following is the known command table for all toys. Anything send or received over the serial port is in quotes to denote communication, but should not be sent using quotes if you are implementing your own version of this protocol. Commands with ”:x” mean that the x should be replaced with a number, the range of which is mentioned in the description.

Get Device Information

Returns toy model type (see models for model letters), Firmware version, and bluetooth MAC address, as a colon delimited list

Availability: All toys

Command Format

```
DeviceType;
```

Return Example

```
C:11:0082059AD3BD;
```

Denotes Nora toy, running v1.1 firmware, BT Addr of 00:82:05:9A:D3:BD

Get Battery Level

Returns the battery level of the toy as an integer percentage from 0-100.

Availability: All toys

Command Format

```
Battery;
```

Return Example

```
85;
```

Denotes 85% battery remaining.

Turn Off Power

Turns off power to the toy.

Availability: All toys

Command Format

```
PowerOff;
```

Return Example

```
OK;
```

Device Status

Retreive the status of the toy.

Availability: All toys

Command Format

```
Status:1;
```

Return Example

```
2;
```

Status Codes:

- 2: Normal

Set Vibration Speed

Changes the vibration speed for the toy. Takes integer values from 0-20.

Availability: All toys

Command Format

```
Vibrate:10;
```

Sets vibration speed to 10 (50%).

Return Example

```
OK;
```

Start Accelerometer Data Stream

Starts a stream of accelerometer data. Will send constantly until stop command is sent. Incoming accelerometer data starts with the letter G, followed by 3 16-bit little-endian numbers.

Availability: Max, Nora

Command Format

```
StartMove:1;
```

Return Example

```
GEF008312ED00;
```

Denotes [0x00EF, 0x1283, 0x00ED] accelerometer readings.

Stop Accelerometer Data Stream

Stops stream of accelerometer data.

Availability: Max, Nora

Command Format

```
StopMove:1;
```

Return Example

```
OK;
```

Change Rotation Direction

Changes the direction of rotation for the toy.

Availability: Nora

Command Format

```
RotateChange;
```

Return Example

```
OK;
```

Set rotation speed

Changes the rotation speed of the Nora toy. Takes integer values from 0-20.

Availability: Nora

Command Format

```
Rotate:10;
```

Sets rotation speed to 10 (50%).

Return Example

```
OK;
```

Set Absolute Air Level

Changes the inflation level of the Max toy. Takes integer values from 0-5.

Availability: Max

Command Format

```
Air:Level:3;
```

Sets air level to 3 (60%).

Return Example

```
OK;
```

Set Relative Inflation Level

Inflates relative to current level, i.e. if currently inflation level is 3, and "Air:In:1;" is sent, will inflate to 4.

Availability: Max

Command Format

```
Air:In:1;
```

Sets air level to 1 level more inflated than it was.

Return Example

```
OK;
```

Set Relative Deflation Level

Deflates relative to current level, i.e. if currently inflation level is 3, and "Air:Out:1;" is sent, will deflate to 2.

Availability: Max

Command Format

```
Air:Out:1;
```

Sets air level to 1 level deflated than it was.

Return Example

```
OK;
```

This document contains information about Lovense sex toy hardware, and the common bluetooth protocol used in all Lovense toys. This information can be used to access Lovense toys and create new libraries for controlling features and receiving information from them.

CHAPTER 4

Lovense Library Info

lovesense is a library for controlling Lovense sex toys. The library allows users to control all aspects of the toy (vibration/rotation/inflation, depending on the toy), as well as retrieving information like device type, status, battery level, and accelerometer readings.

The library is currently available in the following languages:

- Python
- Max/MSP
- Rust (with C Headers)
- Javascript/Node.js

If you need an implementation in a language not currently supported by lovesense, please file an issue on the github tracker of the lovesense-docs projects

Please note that this project has no direct relation to the Lovense company. These drivers have been developed/supported by the open source community. Lovense has had no direct participation in this project and most likely will not be able to answer questions or provide support for any of the Lovesense drivers.

If you require commercial support for programming for Lovense products, it is recommended you go through the Official Lovense Developer Program.

CHAPTER 5

Disclaimer

The Lovesense project is in no way affiliated with Lovense or any of its partners. The documentation and libraries here have been produced via clean room reverse engineering methods, and are provided with no guarantees, as outlined by the license agreement. Usage of these libraries and information is in no way condoned by Lovense and may void the warranty of your toy.

CHAPTER 6

License

Lovesense documentation is covered under the Creative Commons 4.0 Attribution License.