Tap BLE API Documentation

Version 1.0.1

Table of contents

General description

Bluetooth Low Energy (or BLE) is a new standard that emphasizes connectivity with low power devices.

The Tap is a BLE v4.2 device, backwards compatible with BLE v4.0, in *peripheral* mode. Every BLE v4.0 *central* device can connect to the Tap device.

Usage of the keyboard and mouse is done over the standard HID over GATT interface, which most of mainstream devices, such as Windows 8.1/10 laptops or iPhones since 4S, support. Custom protocol can be used to interact with the Tap without using the HID interface for any kind of application.

Device discovery

Scanning

The Tap is advertising when it's not connected.

The advertised data contains a 128-bit custom UUID to distinguish it from other BLE devices. The device name itself is not unique to the device as it can be modified by the user.

The 128-bit UUID is: *C3FF0001-1D8B-40FD-A56F-C7BD5D0F3370*

Connecting & pairing

The device is connectable as long as it's not already connected to a peer device.

For proper usage with the Tap, the Tap must be also paired (bonded) and not just connected.
To detect a Tap in a paired devices list, use the Tap UUID for filtering.

If it's currently connected, it must be disconnected from the active peer.
Disconnected on the peer can be done by turning off the *Bluetooth connectivity* or by hitting a *disconnect* button, if applicable.

Usage of API

Tap is communicating over GATT for information exchange between the Tap and the central device.

The Tap is a GATT server and is exposing several standard services and 2 custom ones.
Any central device may communicate with the API after connection.

For certain operations, higher ATT MTU (Maximum transmission unit) of 185 is required and expected from the central to support.
Which operation require that are TBA.

Types of API

Battery level

A standard Battery Service is available, according to BLE specifications.

Any bonded central device may read the battery level from the Battery Level characteristic.

Device information

A standard Device Information Service is available, according to BLE specifications.

Any bonded central device may read these fields:

- Manufacturer Name String
- Serial Number String
- Hardware Revision String
- Firmware Revision String

Tap proprietary

Custom 128-bit service contains the main API for the Tap. The UUID of the service is: *C3FF0001-1D8B-40FD-A56F-C7BD5D0F3370*

The following characteristics are available:

- Tap data. UUID: *C3FF0005-1D8B-40FD-A56F-C7BD5D0F3370*. Properties: Notify

Support service is being used for the API. It's UUID: *6E400001-B5A3-F393-E0A9-E50E24DCCA9E*.
There's a single support characteristic in the support service. UUID: *6E400002-B5A3-F393-E0A9-E50E24DCCA9E*. Properties: Write

Tap proprietary protocol

General description

The is extracting a handful of gestures from its sensor using complicated algorithms.
Our SDK exposes those via the Tap protocol and allows anyone to receive events on any gesture detection.

The Tap has 2 usage modes: Text mode and Controller mode. By default, the device is in Text mode and all normal functions, such as mouse and multi-taps are mantained.
When the propritrary protocol is desired, the Tap must enter Controller mode by application request.
During Controller mode, the Tap has reduced functionally but will send events according to the protocol.
When done with Controller mode, the app must tell the Tap to exit the controller mode.

Events are sent as an array of bytes from specific characteristic under the Tap Service. Multi-byte fields are in Little Endian format.

Controller mode

The Tap must be entered into controller mode by performing a write action on the support characteristic with the *Enter controller mode* 4 byte magic packet.
To remain in controller mode, the *enter controller mode* magic packet must be sent **every 10 seconds**.
To exit out of controller mode, the *exit controller mode* magic packet must be written.

Packet description	Byte 0	Byte 1	Byte 2	Byte 4
Enter controller mode	0x03	0x0C	0x00	0x01
Exit controller mode	0x03	0x0C	0x00	0x00

During the controller mode, the Tap has reduced functionality to not potentially interfere with applications.
For example, multi taps are not supported and Switch & Shift behavior.
Make sure to exit out of controller mode when the application closes.

Tap data

Once in controller mode and Tap data characteristic's notifications are enables, Tap data characteristic will send tap events via notifications.

Packet format:

Byte 0	Byte 1	Byte 2
Tap code Type: Unsigned Int8	Tap interval in milliseconds Type: Unsigned Int16	

The tap code is a binary combination of the tapping fingers with the following mapping:

- (LSB) Bit 0: thumb finger
- Bit 1: index finger
- Bit 2: middle finger
- Bit 3: ring finger
- Bit 4: pinky finger
- Bit 5: unused
- Bit 6: unused
- (MSB) Bit 7: unused

For example, when the user tapped all fingers, the tap code is 31 (or 0x1F). When the user tapped both thumb and index (character N), the tap code is 3 (0x03).

When the tap is in left hand mode, the finger-to-bit mapping remains relevant.
Thumb tap on the right hand with the right-hand setting will set the same bit as the thumb tap on the left-hand with the left hand setting.

The tap interval is the period from the previous tap in milliseconds. It's saturated at 65535 milliseconds and will not grow further.