

gForce Armband Data Protocol

Version 2.3

2017/4/18

Copyright

Shanghai OYMotion Technologies, Inc.

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | | **Update** | **Author** |
| 2017/04/18 | | V2.3 | Translate to English based on V2.2 | Neo Ni |
| 2017/02/10 | | V2.2 | Renamed Message to Event | Zhou Yu |
| 2016/11/24 | | V2.1 | Added device notification | Kai Zhou |
| 2016/11/18 | | V2.0 | Added Package ID | Kai Zhou |
| 2016/09/30 | | V1.0 | First release version | Kai Zhou |

**Manu**

[Revision History 2](#_Toc480291604)

[Fig List 4](#_Toc480291605)

[1. Brief 5](#_Toc480291606)

[2. Event Packet Format 6](#_Toc480291607)

[2.1 Event Data 6](#_Toc480291608)

[2.1.1 Quaternion Array 6](#_Toc480291609)

[2.1.2 Gesture 7](#_Toc480291610)

[2.1.3 Status 7](#_Toc480291611)

[2.2 Example 7](#_Toc480291612)

Fig List

[Fig 2‑1 Event Format 6](#_Toc480291550)

[Fig 2‑2 Event Type 6](#_Toc480291551)

[Fig 2‑4 Event Type = Quaternion (float) 6](#_Toc480291552)

[Fig 2‑5 Event Type = Gesture 7](#_Toc480291553)

[Fig 2‑6 Event Type = Status 7](#_Toc480291554)

[Fig 2‑7 Example 1 7](#_Toc480291555)

# Brief

This document is to explain the data format and protocol the gForce Armband is implemented.

The user data and information gForce Armband sends out are called “Event”. “Event” includes “Quaternion”, “Gesture”, “Armband Status”, “Sensor Raw Data”, etc. Developer’s App can be implemented differently on the “Event” it. “Event” is transmitted thru “HID” or “Simple GATT” profile. But the “Event” data packet formats are consistent between these two profiles.

When “HID” profile is used for “Event”, “Vendor Usage”（Usage = Vendor Usage）mode is actually implemented. “HID” version is not suggested for new developers if Quaternion is necessary in your application.

**Important Note**:

In terms of versioning of this document, the major number denotes the version of the protocol itself, while the minor number denotes the version of the changes to this document but not to the protocol.

# Event Packet Format

Fig 2‑1 Event Format

|  |  |  |
| --- | --- | --- |
| Byte[0] | Byte[1] | Byte[2:] |
| |  |  | | --- | --- | | Bit[0:6] | Bit[7] | | Event Type | PackageID Flag | | Length | When PackageID Flag=1, Byte[2]=Package ID, Byte[3:]= Event Data  When PackageID Flag=0, Byte[2:]= Event Data |

Event Type: is the bit[6:0] of the first byte in the whole packet and is used to indicate “Event”:

Fig 2‑2 Event Type

|  |  |
| --- | --- |
| Event type | Contents |
| 0x02 | Quaternion Array (float, [w, x, y, z]) |
| 0x0F | Gesture |
| 0x14 | Status Notify |
|  |  |

PackageID Flag: Bit[7] of the first byte, indicates the following bytes contain “Package ID” byte or not. When the flag is 1, Byte[2] is “Package ID”, and Byte[3:N] are “Event Data”, else Byte[2:N]are “Event Data”.

Length: Byte[1] in the whole packet, indicates the total bytes count from Byte[2] and following.

Package ID: Byte[2] in the whole packet if it exists. This is a one-byte sequential timestamp prefix to all “Event”. This is added to help check if there’s “Event” lost, as a mark to the quality of BLE transmit.

Event Data： Indicates all different types of events such like “Quaternion Array”, “Gesture Value”, “Status”, “MEMS Raw Data”, “EMG Raw Data”, etc.

## Event Data

### Quaternion Array

Fig 2‑4 Event Type = Quaternion (float)

|  |  |  |  |
| --- | --- | --- | --- |
| Byte[0-3] | Byte[4-7] | Byte[8-11] | Byte[12-15] |
| W | X | Y | Z |

### Gesture

Fig 2‑5 Event Type = Gesture

|  |  |
| --- | --- |
| Byte[0] | Remark |
| 0x00 | Relax (No gesture) |
| 0x01 | Fist |
| 0x02 | Spread Fingers |
| 0x03 | Wave-In |
| 0x04 | Wave-Out |
| 0x05 | Pinch |
| 0x06 | Gun-Hand |
| 0xFF | Unknown Gesture |

### Status

“Status” event is used to notify host (user’s App) when there’s some status change in the Armband, the content is one byte, right now only “Re-Center” is implemented.

Fig 2‑6 Event Type = Status

|  |  |  |
| --- | --- | --- |
|  | Value | Remark |
| Bit0 | 1 | Re-center.  Set the current direction that gForce Armband’s USB port pointing to as the Quaternion base [w=1, x=0, y=0, z=0].  Press and release quickly the button on the main block when gForce is ON to re-center. |

## Example

* The following examples show data packet format when PackageID Flag is 0 or 1, and the data content is “Gesture = Fist” (0x01):

Fig 2‑7 Example 1

|  |  |  |  |
| --- | --- | --- | --- |
| PackageID Flag | Byte[0] | Byte[1] | Byte[2:3] |
| 1 | 0x8F | 2 | Byte[2]=Package ID(0x00~0xFF), Byte[3]=gForce Data = 0x01 |
| 0 | 0x0F | 1 | Byte[2]=gForce Data=0x01; NO Byte[3] |