# Summary Data Protocol and Structure

Revision: 01

Release Date: 01-10-2021

Part #: PG-RESBIT-GEN2,GEN3,GEN4

# Table of Contents

**Summary Data Structure** **3**

**Event Types** **4**

ResBit Awake Event (0x0000) 4

Data 4

Generic Trigger Event (0x0001) 5

Data 5

Tilt Angle Event (0x0002) 5

Data (Legacy) 5

Blob Uint32 Data Event (0x0003) 6

Data 6

Blob Float Data Event (0x0004) 6

Data 6

**Summary Transfer Protocol** **7**

Workflow 7

Packet Structure 8

Response Structure 8

Response Types 9

Resend Packets (0x00) 9

# Revision History

| | |
|---|---|
| **Revision Date** 01-OCT-2021 | **Description** *01:* Initial Release |

# Summary Data Structure

| Name | Description | Size (Bytes) | Data Type |
|---|---|---|---|
| Event Id | The identifier for the event | 2 | uint16_t |
| Timestamp | The timestamp of the summary event. Unix Timestamp. | 4 | uint32_t |
| Size | The total size of the summary data | 1 | uint8_t |
| Data | A block of that that can be used for whatever the summary event needs. | 20 | uint8_t[20] |

# Event Types

The existing Resbit event types are documented below.

| Name | Description | ID | Total Data Size (Bytes) |
|---|---|---|---|
| ResBit Awake | Resbit Awakes | 0x0000 | 4 |
| Generic Trigger | Count generalized events | 0x0001 | 4 |
| Tilt Angle (Legacy) | Tilt angle (Legacy) | 0x0002 | 16 |
| Blob Uint32 Data | Variable length blob of uint32 data. | 0x0003 | 4-252 |
| Blob Float Data | Variable length blob of float data. | 0x0004 | 4-252 |

## ResBit Awake Event (0x0000)

This event is generated every time a resbit device goes to sleep. It contains how long the device was awake.

### Data

| Name | Description | Size (Bytes) | Data Type |
|---|---|---|---|
| Time Awake | The amount of time (in seconds) the resbit device was awake | 4 | uint32_t |

# Generalized Trigger Event (0x0001)

This event is generated each time the ResBit device goes to sleep. It reports the number of trigger events that occurred during the prior wake period.

A trigger event is defined as the sensor or generalized model exceeding the “Threshold Value” parameter in the Resbit device.cfg file. (Example: Classifier predicts pour.)

{"Events":[{"studyId":"Particle Boron","participantId":"e00fce6813aaf698e3186a44","deviceID":"69002f000f50535252313920","data":{"eventFields":[{"name":"Awake Time","value":5}],"eventType":0,"eventDataSize":4,"eventWakeupTime":1631656289}},

{"studyId":"Particle Boron","participantId":"e00fce6813aaf698e3186a44","deviceID":"69002f000f50535252313920","data":{"eventFields":[{"name":"Trigger Pulls","value":0}],"eventType":1,"eventDataSize":4,"eventWakeupTime":1631656289}}]}

## Data

| Name | Description | Size (Bytes) | Data Type |
|---|---|---|---|
| Number of Trigger Pulls | Number of times a event was counted | 4 | uint32_t |

# Tilt Angle Event (0x0002) (LEGACY - DO NOT USE)

This event is generated every time a resbit device goes to sleep. It reports the start, stop, min, and max tilt angle values during the prior wake period.

## Data

| Name | Description | Size (Bytes) | Data Type |
|---|---|---|---|
| Start Tilt Angle | The tilt angle (in radians) at the beginning of the wake event. | 4 | float (single) |
| Stop Tilt Angle | The tilt angle (in radians) at the end of the wake event. | 4 | float (single) |
| Min Tilt Angle | The minimum tilt angle (in radians) measured during the wake event. | 4 | float (single) |
| Max Tilt Angle | The maximum tilt angle (in radians) measured during the wake event. | 4 | float (single) |

# Blob Uint32 Data Event (0x0003)

This event is generated every time a resbit device goes to sleep. It transfers a variable-length blob of uint32 data.

SEE EXAMPLE: EDGE AUDIO NODE (NEXT PAGE)

## Data

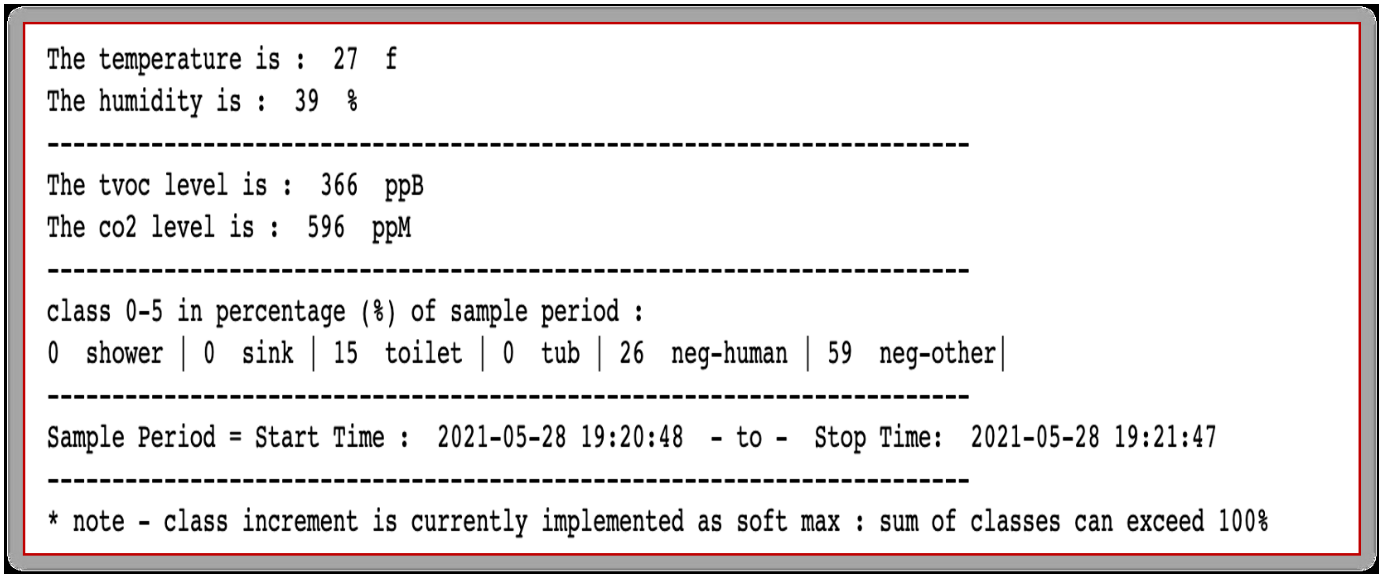
| Name | Description | Size (Bytes) | Data Type |
|---|---|---|---|
| Blob data | Blob data | 4 - 252 | uint32_t [1] - uint32_t [63] |

# Blob Float Data Event (0x0004)

This event is generated every time a resbit device goes to sleep. It transfers a variable-length blob of float data.

## Data

| Name | Description | Size (Bytes) | Data Type |
|---|---|---|---|
| Blob data | Blob data | 4 - 252 | float [1] - float [63] |



```
The temperature is :  27  f
The humidity is :  39  %
----------------------------------------------------------------------
The tvoc level is :  366  ppB
The co2 level is :  596  ppM
----------------------------------------------------------------------
class 0-5 in percentage (%) of sample period :
0  shower | 0  sink | 15  toilet | 0  tub | 26  neg-human | 59  neg-other|
----------------------------------------------------------------------
Sample Period = Start Time :  2021-05-28 19:20:48  - to -  Stop Time:  2021-05-28 19:21:47
----------------------------------------------------------------------
* note - class increment is currently implemented as soft max : sum of classes can exceed 100%
```

# **Decoding** UINT32 String, Bytes Allocation

{"data":{"eventFields":[{"name":"UINT32 Data Blob","value":"HBoAAJABAAAAAAMAAAAtAkwAfALxVbBgK1awYA=="}],"eventWakeupTime":1622169131,"eventType":3,"eventDataSize":28},"studyId":"test","deviceId":"200040001950523364e363320"}

- Base64 Text = HBoAAJABAAAAAAMAAAAtAkwAfALxVbBgK1awYA==
- Encoded Hexadecimal = 1c 1a 00 00 90 01 00 00 00 00 03 00 00 00 2d 02 4c 00 7c 02 f1 55 b0 60 2b 56 b0 60

| Temperature C | Humidity % | TVOC (PPB) | | CO2 (PPM) | | Class 00 - Shower | | Class 01 - Sink | | Class 02 - Toilet | | Class 03 - Tub | | Class 04 - Human | | Class 05 - Other | | Count of Cycles | | Start Time | | | | Stop Time | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1c | 1a | 00 | 00 | 90 | 01 | 00 | 00 | 00 | 00 | 03 | 00 | 00 | 00 | 2d | 02 | 4c | 00 | 7c | 02 | f1 | 55 | b0 | 60 | 2b | 56 | b0 | 60 |

# Summary Transfer Protocol

Summary data will be transferred using the ResBit Summary Service. Data will be sent in chunks with syncing communication between each chunk. Each chunk will be broken down into individual tagged packets that will be streamed to the client device through BLE characteristic notification. No syncing will occur between packet sending.

Chunks will not contain incomplete summary event data. Any dummy bytes at the end of a chunk (due to packets' fixed-length) will be set to '0'. Chunks should be parsed individually into summary event data rather than concatenating (unless dummy bytes are removed prior to concatenation).

## Workflow

1. Client connects to device
2. Client reads "ResBit Serial Number" characteristic for the ResBit serial number from the "ResBit Summary Service"
3. Client subscribes to the *"Data"* characteristic notifications
4. Client subscribes to the *"Transfering"* characteristic notifications
5. (Optional) Client subscribes to the *"Transfer Error"* characteristic notifications
6. (Optional) Client subscribes to the *"Transfer Summary Data"* characteristic notifications
7. Client writes a 1 into the *"Transfer Summary Data"* characteristic to begin data transfer
8. Client Begins Transfering all stored summary data (described below):
    a. *"Transfering"* now has a value of 1
    b. Device will burst write a number of packets through the Data characteristic.
        i. Client will be given these values through the characteristic notification. Data from notification will need to be quickly read and cached to be reconstructed later.
    c. *"Transfering"* now has a value of 0
    d. After sending all the packets, the device will wait up to 2 seconds for a response written to the *"Ack/Nack"* characteristic.
        i. Client will check all cached packets to make sure all packets were received.
            1. This can be done by checking the Packet Index field on the packets. If packets 0-(X-1) are all cached all packets have been received
            2. The total number of packets can be determined by reading the "Total Packets" field on any of the packets
        ii. If all packets have been received Client writes a 1 into *"Ack/Nack"*

iii. If some packets are missing, Client writes a “Resend Packets” response with the missing packets into *“Response”* then writes a 2 into *“Ack/Nack”* (refer to “Response Structure” and “Resend Packets”).

1. Device will resend missing packets

a. *“Transfering”* will be 1 while sending packets and 0 when done.

b. This will repeat until all packets in the chunk are received successfully.

e. If summary data remains, the device will move on to transferring the next chunk (repeat steps 8.a through 8.d).

9. “Transfer Summary Data” value will become 0

## Packet Structure

| Name | Description | Size (Bytes) | Data Type |
|---|---|---|---|
| Total Packets | The total number of packets in the chunk | 1 | uint8_t |
| Packet Index | The index of the packet relative to the start of the chunk | 1 | uint8_t |
| Data | The data within the packet | 18[1,2] | uint8_t[18] |

Notes:

[1] Packets may contain incomplete summary events. Every packet in a chunk should be read and concatenated (based on Packet Index) prior to parsing into summary event data.

[2] Any ‘dummy’ bytes at the end of a packet (i.e. if there are 11 bytes of summary data in the packet, there will be 7 dummy bytes) will be set to 0. This will only occur in the last packet of a chunk.

## Response Structure

| Name | Description | Size (Bytes) | Data Type |
|---|---|---|---|
| Response Type | The response type. Identifies what kind of | 1 | uint8_t |

| | | | |
|---|---|---|---|
| | response is being returned (see Response Types). | | |
| Response Data | The data of the response. | 19 | uint8_t[19] |

## Response Types

### Resend Packets (0x00)

| Name | Description | Size (Bytes) | Data Type |
|---|---|---|---|
| Number of Packets | The number of packets that need to be resent | 1 | uint8_t |
| Packet Indices | The indices of the packet to resend. Up to 18 packets. Don’t have to write the entire array (dummy bytes will be ignored). | 18 | uint8_t[18] |

Example:

| | Response Packet | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Field | Response Type | Response Data | | | | | | | | | | | | | | | | | | |
| Byte Value | 0 | 3 | 0 | 4 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |

Legend:

| | |
|---|---|
| 🟨 | Response type: Resend Packets |
| 🟦 | Number of packets: 3 |
| 🟩 | Packet indices: 0, 4, 9 |
| ▨ | Ignored |