

**AFRIDEV-V2 REMOTE SENSOR**

**MESSAGE SPECIFICATION**

**Version 1.6**

*Prepared by David Laone*

*MindInspire Inc.*

*Updated by Robert Lieb*

*Intelligent Product Solutions*

*Last Updated*

Oct 19, 2019

Abbreviations and Definitions

|  |  |
| --- | --- |
| Afridev-V1 | First generation of the Afridev Sensor (product ID = 0) |
| Afridev-V2 | Second generation of the Afridev Sensor (product ID = 3) |
| IoT Platform | The Charity Water cloud based server application that monitors and controls all the Sensors. |
| BodyTrace | The company that implements the cloud based server to support cellular communication with the Sensors. The IoT Platform communicates with the BodyTrace server. |
| OTA | Over the Air – refers to messages received by the Sensor from the IoT Platform |
| Opcode | Identifies a message type received from the IoT Platform |
| NMEA | National Marine Electronics Association (for GPS) |

TABLE OF CONTENTS

1 MESSAGE Architecture 5

2 Sensor Initiated MessageS 6

2.1 Sensor Initiated Message 6

2.2 Common Message Header Structure 7

2.3 Final Assembly Message (msgId = 0x00) 8

2.3.1 Raw Packet Example 8

2.4 Monthly Check-in Message (Type = 0x05) 9

2.4.1 Raw Packet Example 9

2.5 Activated Message (Type = 0x07) 10

2.5.1 Raw Packet Example 10

2.6 GPS Location Message (Type = 0x08) 11

2.6.1 Raw Packet Example 12

2.7 Daily Water Log Message (msgId = 0x21) 13

2.7.1 Raw Packet Example 13

2.8 Sensor Data Message (Type = 0x22) 15

2.8.2 Raw Packet Example 17

2.9 SOS Message (Type = 0x23) 18

2.9.1 Raw Packet Example 18

2.10 Timestamp Message (Type = 0x24) 20

2.10.1 Raw Packet Example 20

3 IoT Platform Initiated MessageS 22

3.1 IoT Platform Initiated Message Summary 22

3.2 OTA Command Structure 23

3.3 OTA Replies (Sent by the Sensor to the IoT Platform) 23

3.4 GMT Clock Update (opcode=0x01) 25

3.4.1 Command Message Format (IoT Platform to Sensor) 25

3.4.2 Response Message Format (Sensor to IoT Platform) 25

3.4.3 Command Raw Packet Example 26

3.4.4 Response Raw Packet Example 26

3.5 Storage Clock Alignment (opcode=0x02) 28

3.5.1 Example Packets 28

3.5.2 Command Message Format (IoT Platform to Sensor) 28

3.5.3 Response Message Format (Sensor to IoT Platform) 29

3.6 Reset Data (opcode=0x03) 30

3.6.1 Command Message Format (IoT Platform to Sensor) 30

3.6.2 Response Message Format (Sensor to IoT Platform) 30

3.6.3 Command Raw Packet Example 30

3.6.4 Response Raw Packet Example 31

3.7 Reset Red Flag Alarm Condition (opcode=0x04) 32

3.7.1 Command Message Format (IoT Platform to Sensor) 32

3.7.2 Response Message Format (Sensor to IoT Platform) 32

3.7.3 Command Raw Packet Example 32

3.7.4 Response Raw Packet Example 33

3.8 Activate Device (opcode=0x05) 34

3.8.1 Command Message Format (IoT Platform to Sensor) 34

3.8.2 Response Message Format (Sensor to IoT Platform) 34

3.8.3 Command Raw Packet Example 34

3.8.4 Response Raw Packet Example 35

3.9 De-Activate Device (opcode=0x06) 36

3.9.1 Command Message Format (IoT Platform to Sensor) 36

3.9.2 Response Message Format (Sensor to IoT Platform) 36

3.9.3 Command Raw Packet Example 36

3.9.4 Response Raw Packet Example 37

3.10 Update Transmission Rate (opcode=0x07) 38

3.10.1 Command Message Format (IoT Platform to Sensor) 38

3.10.2 Response Message Format (Sensor to IoT Platform) 38

3.10.3 Command Raw Packet Example 39

3.10.4 Response Raw Packet Example 39

3.11 Reset Device (opcode=0x08) 40

3.11.1 Command Message Format (IoT Platform to Sensor) 40

3.11.2 Response Message Format (Sensor to IoT Platform) 40

3.11.3 Command Raw Packet Example 41

3.11.4 Response Raw Packet Example 41

3.12 GPS Request (opcode=0x0D) 42

3.12.1 Command Message Format (IoT Platform to Sensor) 42

3.12.2 Response Message Format (Sensor to IoT Platform) 42

3.12.3 Response Raw Packet Example 43

3.13 GPS Set Measurement Criteria (opcode=0x0E) 45

3.13.1 Command Message Format (IoT Platform to Sensor) 45

3.13.2 Response Message Format (Sensor to IoT Platform) 45

3.13.3 Response Raw Packet Example 46

3.15 Sensor Request (opcode=0x0F) 48

3.15.1 Command Message Format (IoT Platform to Sensor) 48

3.15.3 Response Message Format (Sensor to IoT Platform) 50

3.15.4 Command Raw Packet Example 50

3.15.5 Response Raw Packet Example 50

3.16 Firmware Update (opcode=0x10) 53

3.16.1 Command Message Format (IoT Platform to Sensor) 53

3.16.2 Response Message Format (Sensor to IoT Platform) 53

3.16.3 Error Code Definitions 54

# MESSAGE Architecture

The Sensor and IoT Platform exchange messages in an asynchronous fashion using the cellular system. A key design factor is that command and status request messages sent by the IoT Platform to the Sensor will only be received when the Sensor enables its cellular modem to transmit a message. As shown in Figure 1‑1, the BodyTrace server acts as the “go-between” for the Senor and IoT Platform communication. Messages sent by the IoT platform are queued by the BodyTrace server until the Sensor modem has successfully connected to the local Cellular Network. In this model, the communication model between the Sensor and the IoT Platform must be considered non-real-time. The design strategy of how the Sensor operates and is controlled must take into account this communication model.

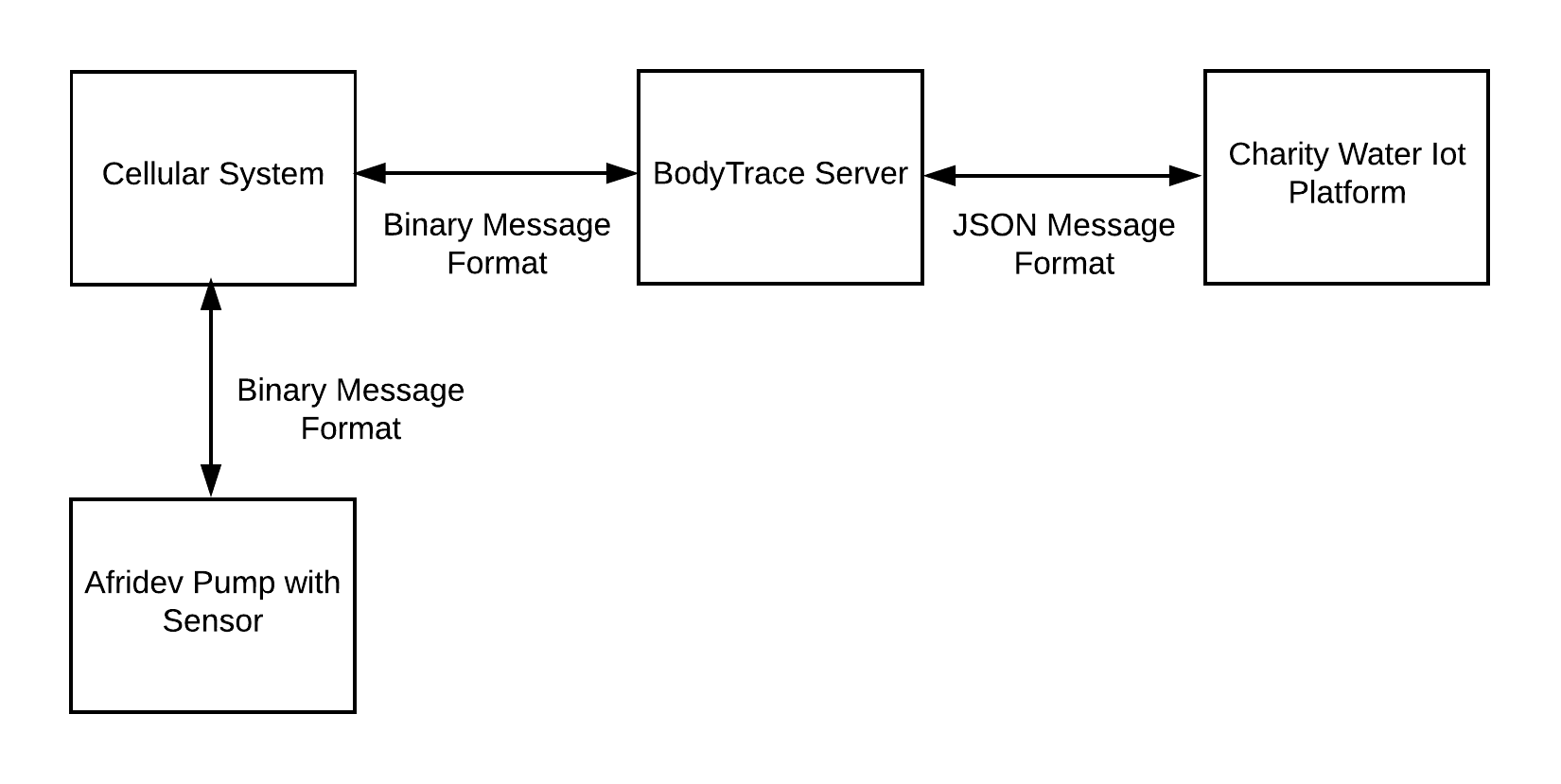


Figure 1‑1 System Block Diagram

# Sensor Initiated MessageS

This section specifies the format and content for all messages sent by the Sensor.

## Sensor Initiated Message

Summary below is a summary of the messages sent by the Sensor to the IoT platform.

|  |  |  |
| --- | --- | --- |
| msgId | Name | Description |
| 0x00 | Final Assembly | Mechanism to get IoT platform to send a GMT time update message to the Sensor |
| 0x03 | OTA Reply | Acknowledge to IoT platform that an OTA message was received. Also returns status information based on the OTA message type. |
| 0x05 | Monthly Check-In | Unit check-in with IoT Platform on a monthly basis (every 28 days) |
| 0x07 | Activated | Message sent when the Sensor becomes activated |
| 0x8 | GPS Location | Message containing GPS location data |
| 0x21 | Daily Water Log | Message containing water statistics for one day |
| 0x22 | Sensor Data | Message containing snapshot of Water Detection software data |
| 0x23 | SOS | SOS mode when firmware is corrupted or invalid. Only sent by the boot loader. |
| 0x24 | Timestamp | Debug message that shows the internal clocks within the Afridev2 unit |

Table 1 Sensor Initiated Messages

## Common Message Header Structure

All transmitted messages sent by the Sensor have a common header format. It consists of 16 bytes. Specific and custom data to each message type follows the common header structure.

Note: For all 16 bit and 32 bit data parameters, the most significant byte is sent first.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Byte Offset | Field | Description | Size | Note |
| 0 | msgType | BodyTrace message type | 1 byte | Always set to 1 |
| 1 | msgId | Message identifier | 1 byte |  |
| 2 | productId | Afridev 2 Product Identifier | 1 byte | Always set to 3 |
| 3 | gmtSecond | GMT second (0-59) | 1 byte | Binary |
| 4 | gmtMinute | GMT minute (0-59) | 1 byte | Binary |
| 5 | gmtHour | GMT hour (24 hour, 0-23) | 1 byte | Binary |
| 6 | gmtDay | GMT day | 1 byte | Binary |
| 7 | gmtMonth | GMT month | 1 byte | Binary |
| 8 | gmtYear | GMT year (tens only, i.e. 15, 16) | 1 byte | Binary |
| 9 | fwVersionMajor | x of x.y version number | 1 byte |  |
| 10 | fwVersionMinor | y of x.y version number | 1 byte |  |
| 11-12 | dayCount | Count of days since activation | 2 bytes | MSB first |
| 13 | storageWeek | The current Storage Clock’s week number (rolls over after 255) | 1 byte |  |
| 14 | storateDay | The current Storage Clock’s day number (0-6) | 1 byte |  |
| 15 | Reserved |  | 1 bytes | Set to 0xA5 |

## Final Assembly Message (msgId = 0x00)

The Sensor sends a Final Assembly message approximately 30 seconds after is starts. The purpose of this message is twofold:

1. It alerts the IoT Platform that the Sensor has restarted and is up and running.
2. It acts as the catalyst to have the IoT Platform send the Sensor a GMT Update message. In most cases, the Sensor will not receive the GMT Update message until the next time it transmits a message.

The Final Assembly message contains no custom data other than the standard message header.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Byte Offset | Field | Description | Size | Note |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x00 |
| 2-15 | Remainder of common 16 byte message header | | | |

### Raw Packet Example

01 00 03 15 02 00 01 01 0f 01 03 00 00 00 00 a5

#### Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x00 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x15 |
| GMT Minutes | 1 | 0x02 |
| GMT Hour | 1 | 0x00 |
| GMT Day | 1 | 0x01 |
| GMT Month | 1 | 0x01 |
| GMT Year | 1 | 0x0f |
| Firmware Version Major | 1 | 0x01 |
| Firmware Version Minor | 1 | 0x03 |
| Activated Day Count | 2 | 0x00 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |

## Monthly Check-in Message (Type = 0x05)

The monthly check-in message is used to ensure that the unit will always communicate to the IoT Platform at least once a month. This allows the unit to download any OTA messages from the IoT Platform even if the unit is not yet activated. Note that the monthly check-in message is based on a storage month (28 days). The unit will not send the monthly check-in message if the Sensor has sent water log messages within the last 28 days.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x05 |
| 2-15 | Remainder of common 16 byte message header | | | |

### Raw Packet Example

01 00 03 15 02 00 01 01 0f 01 03 00 00 00 00 a5

#### Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x05 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x15 |
| GMT Minutes | 1 | 0x02 |
| GMT Hour | 1 | 0x00 |
| GMT Day | 1 | 0x01 |
| GMT Month | 1 | 0x01 |
| GMT Year | 1 | 0x0f |
| Firmware Version Major | 1 | 0x01 |
| Firmware Version Minor | 1 | 0x03 |
| Activated Day Count | 2 | 0x00 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |

## Activated Message (Type = 0x07)

When the device becomes activated it will send this message at the end of the day.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x07 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  | Start message data | | | |
| 16-17 | Total Water Volume | Total water volume calculated over the day in liters | 2 bytes | Binary, MSB first. |
|  | End message data (total data bytes: 2) | | | |
|  | End message (total message bytes: 18) | | | |

### Raw Packet Example

01 07 03 20 28 01 01 01 11 01 02 00 01 00 01 a5 00 7c

#### Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x07 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x20 |
| GMT Minutes | 1 | 0x28 |
| GMT Hour | 1 | 0x01 |
| GMT Day | 1 | 0x01 |
| GMT Month | 1 | 0x01 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x01 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0001 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x01 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Liter Sum for the Day | 2 | 0x007C |

## GPS Location Message (Type = 0x08)

This message is used to return the latest GPS measurement data. The GPS data is primarily retrieved from the NMEA GGA sentence. One additional value is added: how long the GPS device was on for the measurement (in seconds).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x08 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  | Start message data | | | |
| 16 | hours | From GGA message | 1 byte |  |
| 17 | minutes | From GGA message | 1 byte |  |
| 18-21 | latitude | From GGA message (x 1000), | 4 bytes | MSB first Signed number |
| 22-25 | longitude | From GGA message (x 1000) | 4 bytes | MSB first Signed number |
| 26 | fix quality | From GGA message | 1 byte |  |
| 27 | satellites tracked | From GGA message | 1 byte |  |
| 28 | hdop value | From GGA message (x 10) | 1 byte |  |
| 29 | reserved | Future, | 1 byte | Value = 0x0 |
| 30-31 | measurement time in seconds | How long the GPS device was on (in seconds) | 2 bytes | MSB first |
|  | End message data (total data bytes: 16) | | | |
|  | End message (total message bytes: 32) | | | |

### Raw Packet Example

01 08 03 30 08 00 01 01 11 00 02 00 00 00 00 a5

0e 39 02 37 0b 5f f8 c1 23 1c 01 0a 08 00 01 78

#### Raw Packet Breakout

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Field Bytes | Value | Notes |
| msgType | 1 | 0x01 |  |
| msgId | 1 | 0x08 |  |
| productId | 1 | 0x03 |  |
| GMT Seconds | 1 | 0x30 |  |
| GMT Minutes | 1 | 0x08 |  |
| GMT Hour | 1 | 0x00 |  |
| GMT Day | 1 | 0x01 |  |
| GMT Month | 1 | 0x01 |  |
| GMT Year | 1 | 0x11 |  |
| Firmware Version Major | 1 | 0x00 |  |
| Firmware Version Minor | 1 | 0x02 |  |
| Activated Day Count | 2 | 0x0000 |  |
| Storage Week Number | 1 | 0x00 |  |
| Storage Day Number | 1 | 0x00 |  |
| Reserved | 1 | 0xa5 |  |
|  |  |  |  |
| GGA hours | 1 | 0xe |  |
| GGA minutes | 1 | 0x39 |  |
| GGA latitude x 1000 | 4 | 0x02370b5f | 3716.1823 |
| GGA longitude x 1000 | 4 | 0xf8c1231c | -12156.0292 |
| GGA Fix Quality | 1 | 0x1 |  |
| GGA Satellites Tracked | 1 | 0xa |  |
| GGA hdop x 10 | 1 | 0x08 | 0.8 |
| reserved | 1 | 0x00 |  |
| GPS Device On Time | 2 | 0x0178 (376 seconds) | 376 seconds |

## Daily Water Log Message (msgId = 0x21)

Water data for one day is contained in the daily water log message.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x21 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  | Start message data | | | |
| 16-63 | litersPerHour | Per-hour count of milliliters / 32 | [24] \* 2 bytes = 48 bytes |  |
| 64-65 | totalLiters | Daily liter sum | 2 bytes |  |
| 66-67 | averageLiters | Red flag average | 2 bytes |  |
| 68 | redFlag | Red flag alarm | 1 byte |  |
| 69 | reserved | Not used | 1 byte |  |
| 70-71 | unknowns | Daily count of unknown readings | 2 bytes |  |
| 72-83 | padSubmerged | Per-sensor submerged count | [6] \* 2 bytes = 12 bytes |  |
| 84-127 | reserved | Not used | 44 bytes |  |
|  | End message data (total data bytes: 112) | | | |
|  | End message (total message bytes: 128) | | | |

### Raw Packet Example

01 21 03 00 00 00 01 01 11 01 02 00 01 00 01 a5

00 00 00 00 00 00 00 00 00 00 00 00 02 28 02 37

02 39 02 36 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 02 22 02 1f 02 2f 00 00 00 00 00 00

00 7c 00 7c 00 ff 00 00 03 40 03 ac 04 12 04 E8

05 a0 06 90 ff ff ff ff ff ff ff ff ff ff ff ff

ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

#### Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x21 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x00 |
| GMT Minutes | 1 | 0x0 |
| GMT Hour | 1 | 0x00 |
| GMT Day | 1 | 0x01 |
| GMT Month | 1 | 0x01 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x01 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0001 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x01 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Liters Per Hour [24] | 24 x 2 |  |
| Total Liters | 2 | 0x007c |
| Average Liters | 2 | 0x007c |
| Red Flag Alarm | 1 | 0x00 |
| Reserved | 1 | 0xff |
| Unknowns | 2 | 0x0000 |
| Pad Submerged Count | 6 x 2 |  |
| Reserved | 44 | 0xff |

**Liters Per Hour (48 bytes, 24 values, 16 bits each)**

0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0228, 0x0237,

0x0239, 0x0236, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,

0x0000, 0x0000, 0x0222, 0x021f, 0x022f, 0x0000, 0x0000, 0x0000

**Sub-Merged Pads (12 bytes: 6 values, 16 bits each)**

0x0340, 0x03ac, 0x0412, 0x04E8, 0x05a0, 0x0690

## Sensor Data Message (Type = 0x22)

This message is used to provide a snapshot of the Water Detection functionality. There are two major sections of data:

* The calibration data shows the capacitance measurements of “Air” at time of manufacturing. This data is needed to restore the target values in case there is water over the pads at the time of power-up.
* The pad data shows the targets and temperatures for each of the pads

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x21 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  | Start message data | | | |
| 16-27 | padBaseline | Baseline Air Capacitance values stored in Factory | [6] \* 2 bytes = 12 bytes |  |
| 28-39 | airDeviation | Baseline Air Deviation values for Water Detection | [6] \* 2 bytes = 12 bytes |  |
| 40-41 | tempBaseline | Temperature of Pads at Baseline measurement | 2 bytes | Signed |
| 42-43 | tempCurrent | Current Pad Temperature value | 2 bytes | Signed |
| 44-45 | seqUnknowns | Number of sequential “Unknown” states in a row | 2 bytes |  |
| 46-57 | Pad0Stats | Current water detection stats | 12 bytes | See definition below |
| 58-69 | Pad1Stats | Current water detection stats | 12 bytes | See definition below |
| 70-81 | Pad2Stats | Current water detection stats | 12 bytes | See definition below |
| 82-93 | Pad3Stats | Current water detection stats | 12 bytes | See definition below |
| 94-105 | Pad4Stats | Current water detection stats | 12 bytes | See definition below |
| 106-117 | Pad5Stats | Current water detection stats | 12 bytes | See definition below |
| 118-119 | unkownLimit | Number of Sequential unknowns before resetting water detection  0 =disable | 2 bytes |  |
| 120-121 | totalFlow | Numberof ml pumped in the current pumping session | 2 bytes |  |
| 122-123 | downspoutRate | Maximum Rate of Flow of downspout pipe used to calibrate Flow Calculation in ml | 2 bytes |  |
| 124-125 | waterLimit | Number of 2 second periods that water is seen flowing without a break before the detection software is reset  0 = disable | 2 bytes |  |
| 126-127 | waterResets | Number of times the water detection software was reset since power-up | 2 bytes |  |
| 128-129 | trickleVolume | Estimated number of ml of water that is standing at the bottom of the well head | 2 bytes |  |
|  | End message data (total data bytes: 114) | | | |
|  | End message (total message bytes: 130) | | | |

Each pad has 12 bytes of “Pad Stats” reported in this message. The format of this data is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0-1 | lastMean | The last mean capacitance value measured (over 4 samples) | 2 bytes |  |
| 2-3 | airTarget | The target capacitance value for air given the current temperature | 2 bytes |  |
| 4-5 | airTemp | The temperature when the air target was set | 2 bytes | Signed |
| 6-7 | waterTarget | The target capacitance value for water given the current temperature | 2 bytes |  |
| 8-9 | waterTemp | The target when the water target was set | 2 bytes | Signed |
| 10 | State | Current state of the pad:  w or W = Water a or A = Air  ? = unsure | 1 byte |  |
| 11 | numSamp | Number of allowed samples in the lastMean calculation | 1 byte |  |

### Raw Packet Example

01 22 03 34 03 00 01 01 12 00 94 00 00 00 00 a5

60 9c 58 89 e1 94 8e 8e 18 a0 38 9d bf 03 d8 02

48 03 f6 02 a9 03 34 03 6b 00 8a 00 00 00 c5 9b

e8 9b 89 00 9c 9b 89 00 00 04 43 89 4f 89 89 00

fe 88 89 00 00 04 a5 94 b9 94 89 00 7a 94 89 00

00 04 50 8e 66 8e 89 00 31 8e 89 00 00 04 e2 9f

ff 9f 89 00 b1 9f 89 00 00 04 1f 9d 30 9d 89 00

e0 9c 89 00 00 04 00 00 43 0b d5 01 96 00 5c 00

#### Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x22 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x34 |
| GMT Minutes | 1 | 0x03 |
| GMT Hour | 1 | 0x00 |
| GMT Day | 1 | 0x01 |
| GMT Month | 1 | 0x01 |
| GMT Year | 1 | 0x12 |
| Firmware Version Major | 1 | 0x00 |
| Firmware Version Minor | 1 | 0x94 |
| Activated Day Count | 2 | 0x0000 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
| padBaseline | 12 | 0x9c60 0x8958 0x94e1 0x8e8e 0xa018 0x9d38 |
| airDeviation | 12 | 0x03bf 0x02d8 0x0348 0x02f6 0x03a9 0x0334 |
| tempBaseline | 2 | 0x006b |
| tempCurrent | 2 | 0x008a |
| seqUnknowns | 2 | 0x0000 |
| Pad0Stats | 12 | 0x9bc5 0x9be8 0x0089 0x9b9c 0x0089 0x00 0x04 |
| Pad1Stats | 12 | 0x8943 0x894f 0x0089 0x88fe 0x0089 0x00 0x04 |
| Pad2Stats | 12 | 0x94a5 0x94b9 0x0089 0x947a 0x0089 0x00 0x04 |
| Pad3Stats | 12 | 0x8e50 0x8e66 0x0089 0x8e31 0x0089 0x00 0x04 |
| Pad4Stats | 12 | 0x9fe2 0x9fff 0x0089 0x9fb1 0x0089 0x00 0x04 |
| Pad5Stats | 12 | 0x9d1f 0x9d30 0x0089 0x9ce0 0x0089 0x00 0x04 |
| unkownLimit | 2 | 0x0000 |
| totalFlow | 2 | 0x0b43 |
| downspoutRate | 2 | 0x01d5 |
| waterLimit | 2 | 0x0096 |
| trickleVolume | 2 | 0x005c |

## SOS Message (Type = 0x23)

The SOS Message is only sent by the bootloader due to a catastrophic failure. The bootloader will send the SOS message every 12 hours if it detects that there is not a valid application firmware image to execute. The most likely cause for this to happen is if the firmware-upgrade sequence fails.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x06 |
| 2 | productId | Product Identifier | 1 byte | Value = 0x03 |
| 3-8 | reserved | Message pattern 55 55 55 55 55 55 | 6 byte | Fixed value |
| 9 | blVersionMajor | Bootloader version | 1 byte |  |
| 10 | blVersionMinor | Bootloader version | 1 byte |  |
| 11-15 | reserved | Message pattern 55 55 55 55 55 | 5 bytes | Fixed value |
|  |  |  |  |  |
| 16 | rebootReason | Last reboot reason code | 1 byte |  |
| 17 | appRecordIsGood | Identifies if the boot loader thinks the application record is good | 1 byte |  |
| 18-19 | blrRecordCount | Attempted boots with failure | 2 bytes |  |
| 20 | newFwCopyResult | Result of the FW copy attempt from backup image to main image | 1 byte |  |
| 21 | newFwIsReady | Value of fwIsReady flag in the appRecord | 1 byte |  |
| 22-23 | newFwCrc | Value of newFwCrc in the appRecord | 2 bytes |  |

### Raw Packet Example

01 23 03 34 03 00 01 01 12 02 05 00 00 00 00 a5

01 01 00 00 01 01 36 2a

#### Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x23 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x34 |
| GMT Minutes | 1 | 0x03 |
| GMT Hour | 1 | 0x00 |
| GMT Day | 1 | 0x01 |
| GMT Month | 1 | 0x01 |
| GMT Year | 1 | 0x12 |
| Firmware Version Major | 1 | 0x02 |
| Firmware Version Minor | 1 | 0x05 |
| Activated Day Count | 2 | 0x0000 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
| rebootReason | 1 | 0x01 |
| appRecordIsGood | 1 | 0x01 |
| blrRecordCount | 2 | 0x0000 |
| newFwCopyResult | 1 | 0x01 |
| newFwIsReady | 1 | 0x01 |
| newFwCrc | 2 | 0x2a36 |

## Timestamp Message (Type = 0x24)

This is a debug message that is sent to observe the states of the internal clocks of the Afridev Firmware

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x24 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  | Start message data | | | |
| 16-19 | SysTime | Number of seconds the unit is up since boot time | 4 bytes | Binary, MSB first. |
| 20 | RTC Second | Real Time Clock Second | 1 byte | Binary |
| 21 | RTC Minute | Real Time Clock Minute | 1 byte | Binary |
| 22 | RTC Hour | Real Time Clock Hour | 1 byte | Binary |
| 23 | RTC Day | Real Time Clock Day | 1 byte | Binary |
| 24 | RTC Month | Real Time Clock Month | 1 byte | Binary |
| 25 | RTC Year | Real Time Clock Year | 1 byte | Binary |
| 26 | Storage Second | Number of seconds of water data collected | 1 byte | Binary |
| 27 | Storage Minute | Number of minutes of water data collected | 1 byte | Binary |
| 28 | Storage Hour | Number of hours of water data collected | 1 byte | Binary |
| 29 | Unused |  | 1 byte |  |
|  | End message data (total data bytes: 14) | | | |
|  | End message (total message bytes: 30) | | | |

### Raw Packet Example

01 24 03 01 00 0d 13 0a 13 03 0c 00 02 00 02 a5

b3 c6 02 00 01 00 0d 13 0a 13 01 00 02 00

#### Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x24 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x01 |
| GMT Minutes | 1 | 0x00 |
| GMT Hour | 1 | 0x0d |
| GMT Day | 1 | 0x13 |
| GMT Month | 1 | 0x0a |
| GMT Year | 1 | 0x13 |
| Firmware Version Major | 1 | 0x03 |
| Firmware Version Minor | 1 | 0x0c |
| Activated Day Count | 2 | 0x0002 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x02 |
| Reserved | 1 | 0xa5 |
| SysTime | 4 | 0x b3 c6 02 00 |
| RTC Second | 1 | 0x01 |
| RTC Minute | 1 | 0x00 |
| RTC Hour | 1 | 0x0d |
| RTC Day | 1 | 0x13 |
| RTC Month | 1 | 0x0a |
| RTC Year | 1 | 0x13 |
| Storage Second | 1 | 0x01 |
| Storage Minute | 1 | 0x00 |
| Storage Hour | 1 | 0x02 |
| Unused | 1 | 0x00 |

# IoT Platform Initiated MessageS

This section specifies the format and content for all messages sent by the IoT Platform to the Sensor.

## IoT Platform Initiated Message Summary

Below is a summary of the messages sent by the IoT platform to the Sensor. When sent by the IoT platform, the messages will be queued by the BodyTrace server. On the next occurrence that the Sensor connects to the Network to send a message, the Sensor will retrieve the queued messages. For each message sent by the IoT platform and received by the Sensor, the Sensor will respond with an OTA Reply message. The OTA Reply message acts as an acknowledgement to all messages received by the Sensor and may also contain response data. The response data is based on the specific message opcode sent by the IoT platform.

|  |  |  |
| --- | --- | --- |
| Opcode | Name | Action |
| 0x01 | GMT Clock Update | Advance GMT Clock by the specified time |
| 0x02 | Storage Clock Alignment | Align storage clock to local time zone |
| 0x03 | Reset Data | Erase all logged data and de-activate Sensor |
| 0x04 | Reset Red Flag | Reset global Red Flag event |
| 0x05 | Activate Device | Activate device. This enables the device to send the water log messages at the set transmission rate |
| 0x06 | Deactivate Device | De-activate device. |
| 0x07 | Set Transmission Rate | Set how often to send the water log messages. Set in terms of days. |
| 0x08 | Reset Device | Perform a microprocessor reset |
| 0x0D | GPS request | Request the last measured GPS coordinates or request a new GPS measurement be performed |
| 0x0E | GPS Measurement Parameters | Set the threshold of key parameters used by the Sensor to determine that a valid GPS fix has been received |
| 0x0F | Sensor Request | Configures and controls the Water Sensing features of the product. |
| 0x10 | Firmware Update | Sends new firmware to the Sensor |

## OTA Command Structure

All messages sent by the IoT Platform to the Sensor have a common header format. It consists of 3 bytes. Specific and custom data to for each message type immediately follows the common header.

Note on the msgNumber field:

The msgNumber field value is a counter that is incremented for each message sent by the IoT Platform to a specific Sensor. It acts as a unique message identifier.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Field** | **Description** | **Size** | **Notes** |
| 0 | Opcode | Specifies the message type | 1 byte |  |
| 1-2 | msgNumber | Unique identification number that will be echoed back to IoT Platform in the OTA Reply message sent by the Sensor | 2 bytes | MSB first |
|  |  |  |  |  |
| 3-N | Message Based | Dependent on message | N-3 bytes |  |

## OTA Replies (Sent by the Sensor to the IoT Platform)

After the Sensor receives any message, the Sensor sends an OTA Reply message back to the IoT Platform to confirm that the message was received.

Each message received by the Sensor contains a message opcode and a unique two-byte message number. At a minimum, each OTA Replay will contain the standard 16-byte header, the received opcode, the received message number and a status byte. Some OTA replies will also contain response data. The custom data contained in the reply is dependent on the message opcode received. The size of the OTA message is always 48 bytes, regardless if all bytes are used to contain response data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | Opcode | Specifies the message type sent by the IoT Platform | 1 byte |  |
| 17-18 | msgNumber | Message number echoed back to server | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |
|  |  |  |  |  |
| 20-47 | Message Type Dependent | Message Type Dependent | 28 bytes |  |

## GMT Clock Update (opcode=0x01)

Upon receiving this message, the unit will advance its internal clock ahead by the given amounts.

The example packet below advances the clock by 4 days, 2 hours, 22 minutes, and 51 seconds. The opcode is 0x1, and the message number is 0x1122.

0x01 0x11 0x22 0x33 0x16 0x02 0x00 0x04

Notes:

* 1 byte of Seconds, 1 byte of Minutes, 1 byte of Hours, and 2 bytes of days allows for advancement of 179 years.
* A storage clock alignment to GMT midnight will be performed at each GMT update

If the unit receives multiple GMT Clock Update messages within the same modem session, then it will accept the one with the highest msgNumber and reject all others with lower msgNumber values.

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Field** | **Description** | **Size** | **Note** |
| 0 | Opcode | Specifies the message type | 1 byte | Value = 0x01 |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |
|  |  |  |  |  |
| 3 | gmtSecond | seconds to advance GMT time | 1 byte |  |
| 4 | gmtMinute | minutes to advance GMT time | 1 byte |  |
| 5 | gmtHour | hours (24 hour) to advance GMT time | 1 byte |  |
| 6 | gmtDay | days to advance GMT time | 2 bytes |  |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x01 |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |
|  |  |  |  |  |
| 20 | accept | 1 = accept, 0xff = reject | 1 byte |  |
| 20 | gmtSecond | Echo seconds to advance GMT time | 1 byte |  |
| 21 | gmtMinute | Echo minutes to advance GMT time | 1 byte |  |
| 22 | gmtHour | Echo hours (24 hour) to advance GMT time | 1 byte |  |
| 23 | gmtDay | Echo days to advance GMT time | 2 bytes |  |

### Command Raw Packet Example

01 00 bf 32 23 17 00 a2

#### Command Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msg opcode | 1 | 0x01 |
| msg id | 1 | 0x00bf |
| gmtBinSecondsOffset | 1 | 0x32 |
| gmtBinSecondsOffset | 1 | 0x23 |
| gmtBinHoursOffset | 1 | 0x17 |
| gmtBinDaysOffset | 1 | 0x00a2 |

### Response Raw Packet Example

01 03 03 0e 26 17 0c 06 11 03 02 00 01 00 00 a5

01 00 bf 01 01 32 23 17 00 a2 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#### Response Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x03 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x0e |
| GMT Minutes | 1 | 0x26 |
| GMT Hour | 1 | 0x17 |
| GMT Day | 1 | 0x0c |
| GMT Month | 1 | 0x06 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x03 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0001 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Echo msg opcode | 1 | 0x01 |
| Echo msg id | 1 | 0x00bf |
| status | 1 | 0x01 |
| accept | 1 | 0x01 |
| Echo gmtBinSecondsOffset | 1 | 0x32 |
| Echo gmtBinSecondsOffset | 1 | 0x23 |
| Echo gmtBinHoursOffset | 1 | 0x17 |
| Echo gmtBinDaysOffset | 1 | 0x00a2 |

## Storage Clock Alignment (opcode=0x02)

The Sensor provides the means to align its Storage clock to the local time zone where the Sensor is located using the Storage Clock Alignment message. This is desired so that the hourly water statistics contained in a daily water record correlates with 12:00AM to 11:59 PM of the region that the Sensor is located.

Notes:

* Once this message is received by the Sensor, it will not store any water data until the alignment time is reached
* Receiving this message also erases all Red Flag information and previously stored water data
* If the local timezone is located in an area where the GMT offset is negative, then the message must contain offset values set as 24 - offset. For example, in San Francisco the local timezone offset is GMT - 7 hours. So the offset sent in the message must be set to 0 seconds, 0 minutes and 17 hours.

### Example Packets

In the following examples, the 02 is the opcode value. The 11 22 is the message number (0x1122).

* Ethiopia (+3 GMT): 02 12 34 00 00 03
* San Francisco (-7 GMT): 02 12 34 00 00 11

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x02 |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |
|  |  |  |  |  |
| 3 | gmtSecondOffset | GMT Offset Seconds | 1 byte |  |
| 4 | gmtMinuteOffset | GMT Offset Minutes | 1 byte |  |
| 5 | gmtHourOffset | GMT Offset 24 Hour | 1 byte |  |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x02 |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |
|  |  |  |  |  |
| 20 | gmtSecondOffset | Echo GMT Offset Seconds | 1 byte |  |
| 21 | gmtMinuteOffset | Echo GMT Offset Seconds | 1 byte |  |
| 22 | gmtHourOffset | Echo GMT Offset 24 Hour | 1 byte |  |
| 23 | gmtSecondAlign | GMT second alignment time | 1 byte |  |
| 24 | gmtMinuteAlign | GMT minute alignment time | 1 byte |  |
| 25 | gmtHourAlign | GMT hour (24 hour) alignment time | 1 byte |  |

## Reset Data (opcode=0x03)

* Resets and erases all stored red flag data
* Resets and erases all stored water data
* De-activates the unit

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x03 |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x03 |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |

### Command Raw Packet Example

03 12 34

#### Command Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msg opcode | 1 | 0x03 |
| msg id | 1 | 0x1234 |

### Response Raw Packet Example

01 03 03 0e 26 17 0c 06 11 03 02 00 01 00 00 a5

03 12 34 01 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#### Response Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x03 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x0e |
| GMT Minutes | 1 | 0x26 |
| GMT Hour | 1 | 0x17 |
| GMT Day | 1 | 0x0c |
| GMT Month | 1 | 0x06 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x03 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0001 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Echo msg opcode | 1 | 0x03 |
| Echo msg id | 1 | 0x1234 |
| status | 1 | 0x01 |

## Reset Red Flag Alarm Condition (opcode=0x04)

* Resets and erases all stored red flag data

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x04 |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x04 |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |

### Command Raw Packet Example

04 12 35

#### Command Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msg opcode | 1 | 0x04 |
| msg id | 1 | 0x1235 |

### Response Raw Packet Example

01 03 03 0e 26 17 0c 06 11 03 02 00 01 00 00 a5

04 12 35 01 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#### Response Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x03 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x0e |
| GMT Minutes | 1 | 0x26 |
| GMT Hour | 1 | 0x17 |
| GMT Day | 1 | 0x0c |
| GMT Month | 1 | 0x06 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x03 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0001 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Echo msg opcode | 1 | 0x04 |
| Echo msg id | 1 | 0x1235 |
| status | 1 | 0x01 |

## Activate Device (opcode=0x05)

* Set the unit to “activated”.

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x05 |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x05 |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |

### Command Raw Packet Example

05 12 36

#### Command Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msg opcode | 1 | 0x05 |
| msg id | 1 | 0x1236 |

### Response Raw Packet Example

01 03 03 0e 26 17 0c 06 11 03 02 00 01 00 00 a5

05 12 36 01 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#### Response Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x03 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x0e |
| GMT Minutes | 1 | 0x26 |
| GMT Hour | 1 | 0x17 |
| GMT Day | 1 | 0x0c |
| GMT Month | 1 | 0x06 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x03 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0001 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Echo msg opcode | 1 | 0x05 |
| Echo msg id | 1 | 0x1236 |
| status | 1 | 0x01 |

## De-Activate Device (opcode=0x06)

* Set the unit to de-activated.

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x06 |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x06 |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |

### Command Raw Packet Example

06 12 37

#### Command Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msg opcode | 1 | 0x06 |
| msg id | 1 | 0x1237 |

### Response Raw Packet Example

01 03 03 0e 26 17 0c 06 11 03 02 00 01 00 00 a5

06 12 37 01 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#### Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x03 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x0e |
| GMT Minutes | 1 | 0x26 |
| GMT Hour | 1 | 0x17 |
| GMT Day | 1 | 0x0c |
| GMT Month | 1 | 0x06 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x03 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0001 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Echo msg opcode | 1 | 0x06 |
| Echo msg id | 1 | 0x1237 |
| status | 1 | 0x01 |

## Update Transmission Rate (opcode=0x07)

Transmission Rate: Updates the rate at which the device transmits data. This is a variable timer that will transmit all daily water logs at the set interval. For example, if it is set to 10, the unit will transmit every 10 days. The default setting is set to 7 days. Valid range is from 1 and 28 days (up to 4 weeks).

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x07 |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |
|  |  |  |  |  |
| 3 | Transmission Rate | The number of days in between each transmission of the daily water logs | 1 byte | 1-42 |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x07 |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |
|  |  |  |  |  |
| 20 | Transmission Rate | Echo the number of days in between each transmission of the daily packets | 1 byte |  |

### Command Raw Packet Example

07 12 38 1C

#### Command Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msg opcode | 1 | 0x07 |
| msg id | 1 | 0x1238 |
| transmissionRate | 1 | 0x1C |

### Response Raw Packet Example

01 03 03 0e 26 17 0c 06 11 03 02 00 01 00 00 a5

07 12 38 01 1C 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#### Response Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x03 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x0e |
| GMT Minutes | 1 | 0x26 |
| GMT Hour | 1 | 0x17 |
| GMT Day | 1 | 0x0c |
| GMT Month | 1 | 0x06 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x03 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0001 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Echo msg opcode | 1 | 0x07 |
| Echo msg id | 1 | 0x1238 |
| status | 1 | 0x01 |
| Echo transmissionRate | 1 | 0x1C |

## Reset Device (opcode=0x08)

* Performs a MSP430 reboot after modem communication is complete (after the OTA reply is sent).

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x08 |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |
|  |  |  |  |  |
| 3 | keyByte0 |  | 1 byte | value = 0xAA |
| 4 | keyByte1 |  | 1 byte | value = 0x55 |
| 5 | keyByte2 |  | 1 byte | value = 0xCC |
| 6 | keyByte3 |  | 1 byte | value = 0x33 |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x08 |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |

### Command Raw Packet Example

08 56 78 aa 55 cc 33

#### Command Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msg opcode | 1 | 0x08 |
| msg id | 2 | 0x5678 |
| Key | 4 | 0xaa55cc33 |

### Response Raw Packet Example

01 03 03 1e 05 00 01 01 11 00 02 00 00 00 00 a5

08 56 78 01 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#### Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x03 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x1e |
| GMT Minutes | 1 | 0x05 |
| GMT Hour | 1 | 0x00 |
| GMT Day | 1 | 0x01 |
| GMT Month | 1 | 0x01 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x00 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0000 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Echo msg opcode | 1 | 0x08 |
| Echo msg id | 1 | 0x5678 |
| status | 1 | 0x01 |

## GPS Request (opcode=0x0D)

* This message is used to request GPS data from the Sensor.
* The message can request that the Sensor return existing GPS measurement data (request = 0), or to perform a new GPS measurement (request = 1).
* If requesting existing GPS measurement data, then the data will be returned as part of the OTA response.
* If the request is to perform a new GPS measurement, the Sensor will perform the GPS measurement and send a GPS Location message following the GPS measurement.

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x0D |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |
|  |  |  |  |  |
| 3 | Request Type | 0: Return existing GPS data  1: Perform a new GPS measurement | 1 byte |  |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x0D |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |
|  |  | If request type = 0, GPS data is returned |  |  |
| 20 | hours | Hours from GGA message | 1 byte |  |
| 21 | minutes | Minutes from GGA message | 1 byte |  |
| 22-25 | latitude | Latitude from GGA message, scaled x 1000 | 4 bytes | MSB first Signed number |
| 26-29 | longitude | Longitude from GGA message, scaled x 1000 | 4 bytes | MSB first Signed number |
| 30 | fixQuality | Fix Quality from GGA message | 1 byte |  |
| 31 | satellitesTracked | Number of satellites from GGA message | 1 byte |  |
| 32 | hdop | Horizontal dilution of position from GGA message, scaled x 10 | 1 byte |  |
| 33 | reserved | future, | 1 byte | Value = 0x0 |
| 34-35 | measTime | How long the GPS device was on (in seconds) | 2 bytes | MSB first |

#### Command Raw Packet Example

0d 12 34 01

#### Command Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msg opcode | 1 | 0x0D |
| msg id | 2 | 0x1234 |
| option | 1 | 0x01 |

### Response Raw Packet Example

01 03 03 28 05 00 01 01 11 00 02 00 00 00 00 a5

0d 12 34 01 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#### Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x03 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x28 |
| GMT Minutes | 1 | 0x05 |
| GMT Hour | 1 | 0x00 |
| GMT Day | 1 | 0x01 |
| GMT Month | 1 | 0x01 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x00 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0000 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Echo msg opcode | 1 | 0x0d |
| Echo msg id | 1 | 0x1234 |
| status | 1 | 0x01 |

## GPS Set Measurement Criteria (opcode=0x0E)

This message is used to set the thresholds used by the Sensor during a GPS measurement to identify if a valid GPS measurement has been received. By default, the following values are used:

* Minimum number of satellites: 6
* Maximum HDOP value (x10): 30
* Minimum measurement time in seconds: 300 (5 minutes)

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x0D |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |
|  |  |  |  |  |
| 3 | numberOfSatellites | Minimum number of satellites to consider a valid fix | 1 byte |  |
| 4 | maxHdop | Maximum horizontal dilution of position (HDOP) value to consider a valid fix (scaled by factor of 10) | 1 byte |  |
| 5-6 | minMeasTime | Minimum measurement time in seconds to consider a valid fix | 2 bytes |  |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x0E |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |
|  |  |  |  |  |
| 20 | numberOfSatellites | Echo the received minimum number of satellites to consider a valid fix | 1 byte |  |
| 21 | maxHdop | Echo the received maximum HDOP value to consider a valid fix (scaled by factor of 10) | 1 byte |  |
| 22-23 | minMeasTime | Echo the received minimum measurement time in seconds to consider a valid fix | 2 bytes |  |

#### Command Raw Packet Example

0E 12 34 0A 0A 01 68

#### Command Raw Packet Breakout

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Field Bytes | Value | Notes |
| msg opcode | 1 | 0x0D |  |
| msg id | 2 | 0x1234 |  |
| option | 1 | 0x01 |  |
| numSats | 1 | 0xA | Satellites >= 10 |
| hdop | 1 | 0xA | HDOP < 1.0 |
| minMeasTime | 2 | 0x168 | Meas Time >= 6 minutes |

### Response Raw Packet Example

01 03 03 0a 01 00 01 01 11 00 02 00 00 00 00 a5

0e 12 34 01 0a 0a 01 68 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#### Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x03 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x0a |
| GMT Minutes | 1 | 0x0a |
| GMT Hour | 1 | 0x00 |
| GMT Day | 1 | 0x01 |
| GMT Month | 1 | 0x01 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x00 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0000 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Echo msg opcode | 1 | 0x0e |
| Echo msg id | 1 | 0x1234 |
| status | 1 | 0x01 |
| Echo numSats | 1 | 0xa |
| Echo hdop | 2 | 0xa |
| Echo minMeasTime | 2 | 0x0168 |

## Sensor Request (opcode=0x0F)

This message is used to configure and control the operation of the water sensing feature of the product. The following operations can be requested:

|  |  |  |
| --- | --- | --- |
| Request Number | Request Data | Action |
| 0 | n/a | Request Sensor Data (see Sensor Data message (0x22) Format |
| 1 | n/a | Reset Calibration Data to Current Pad Data |
| 2 | n/a | Reset Water Detection |
| 3 | Unknown Limit | Set Unknown Limit: The number of times an Unkown is seen in a row before Water Detection Data is reset automatically. (disabled by default) |
| 4 | Enable/Disable | Report Water Flow Data Now: When enabled, every time water flow starts and finishes, a SENSOR DATA message is sent. |
| 5 | Downspout Rate | Set Downspout Rate: This is used to calibrate the Flow Rate calculation in the field. |
| 6 | Water Limit | This is used to set the number of consecutive minutes that water is detected before it is flagged as a possible hardware failure (disabled by default) |
| 7 | Wake Time | This is used to set the number of seconds that the sensor is “dry” before entering low power mode.  When try over this time, the unit sleeps for 20 seconds at a time until water is seen or there is a need to send a message or read a GPS location |
| 8 | NOP Response | This is used by the backend server to trigger the setting of the clock without having to reset the entire unit. |

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x0F |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |
|  |  |  |  |  |
| 3 | Request Type | 0: Request Sensor Data (see Sensor Data message (opcode 0x22)  1: Overwrite Factory Data with current pad data  2: Reset Water Detection  3: Set Unknown Limit: This is the number of times an Unknown is seen in a row before automatic “Reset Water Detection”  4: Report Water Flow Data Now  5: Set Downspout Rate  6: Set Water Limit  7: Set Wake Time  8: NOP Response | 1 byte |  |
| 4-5 | Request Data | For Request Types 0-2: Not Applicable  For Request Type 3:  Unknown Limit: 0 = Feature Disabled; 1-65535 number of unknowns  For Request Type 4:  0 = Disabled, 1= Enabled  For Request Type 5:  Downspout Rate: Allowed values 200-800  For Request Type 6:  Water Limit: Allowed values: 0 – 65535 (2 sec resolution) 1800 = 1 hour, 900 = 30 minutes, 0 = feature disabled  For Request Type 7:  Wake Time: Allowed Values: 0 –65535 (0.5 sec resolution) 7200 = 1 hour; 3600 = 30 minutes, 0 = feature disabled  For Request Type 8:  NOP Data: Allowed Values 0 -65535 | 2 bytes |  |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | Opcode | Opcode received in the command message | 1 byte | Value = 0x0F |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | Status | 1 = success, 0xff = failure | 1 byte |  |
|  |  |  |  |  |
| 20 | Request Type | Echo the request type provided in the request | 1 byte |  |
| 21-22 | Request Data | For Request Types 0-7:  Echo the data provided in the request  For Request Type 8:  Send the current RTC value: A binary value = hour \* 256 + minutes | 2 bytes |  |

### Command Raw Packet Example

0F 12 38 00 00 00

#### Command Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msg opcode | 1 | 0x0F |
| msg id | 1 | 0x1238 |
| requestType | 1 | 0x00 |
| requestData | 2 | 0x0000 |

### Response Raw Packet Example

01 03 03 0e 26 17 0c 06 11 03 02 00 01 00 00 a5

0F 12 38 01 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

#### Response Raw Packet Breakout

|  |  |  |
| --- | --- | --- |
| Field | Field Bytes | Value |
| msgType | 1 | 0x01 |
| msgId | 1 | 0x03 |
| productId | 1 | 0x03 |
| GMT Seconds | 1 | 0x0e |
| GMT Minutes | 1 | 0x26 |
| GMT Hour | 1 | 0x17 |
| GMT Day | 1 | 0x0c |
| GMT Month | 1 | 0x06 |
| GMT Year | 1 | 0x11 |
| Firmware Version Major | 1 | 0x03 |
| Firmware Version Minor | 1 | 0x02 |
| Activated Day Count | 2 | 0x0001 |
| Storage Week Number | 1 | 0x00 |
| Storage Day Number | 1 | 0x00 |
| Reserved | 1 | 0xa5 |
|  |  |  |
| Echo msg opcode | 1 | 0x0F |
| Echo msg id | 1 | 0x1238 |
| status | 1 | 0x01 |
| Echo requestType | 1 | 0x00 |
| Echo requestData | 1 | 0x0000 |

## Firmware Update (opcode=0x10)

* Updates the firmware

### Command Message Format (IoT Platform to Sensor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | opcode | Specifies the message type | 1 byte | Value = 0x10 |
| 1-2 | msgNumber | Identification number that will be echoed back to IoT Platform in reply | 2 bytes |  |
|  |  |  |  |  |
| 3 | keyByte0 | Msg Validation Byte 0 | 1 byte | value = 0x31 |
| 4 | keyByte1 | Msg Validation Byte 1 | 1 byte | value = 0x41 |
| 5 | keyByte2 | Msg Validation Byte 2 | 1 byte | value = 0x59 |
| 6 | keyByte3 | Msg Validation Byte 3 | 1 byte | value = 0x26 |
| 7 | numSections | Total number of sections | 1 byte | always 1 |
| 8 | sectionStart | Firmware Section Start Byte | 1 byte | value = 0xA5 |
| 9 | sectionNumber | Section Number | 1 byte | always 0 |
| 10-11 | startAddress | Firmware start address in flash (MSB first) | 2 bytes | value = 0x9000 |
| 12-13 | firmwareLength | Length of firmware part only (in bytes, MSB first) | 2 bytes | value = 0x5000 |
| 14-15 | crc16 | CRC16 of the firmware data  Algorithm = ANSI, POLY=0x8005 | 2 bytes | calculated  CRC16 |
| 16-  20495 | firmwareData | Firmware Application  20480 (0x5000) bytes | 20480 bytes |  |

### Response Message Format (Sensor to IoT Platform)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Byte Offset** | **Field** | **Description** | **Size** | **Note** |
| 0 | msgType | BodyTrace message type | 1 byte | Value = 0x01 |
| 1 | msgId | Message identifier | 1 byte | Value = 0x03 |
| 2-15 | Remainder of common 16 byte message header | | | |
|  |  |  |  |  |
| 16 | opcode | Opcode received in the command message | 1 byte | Value = 0x10 |
| 17-18 | msgNumber | Identification number that was received in the command | 2 bytes |  |
| 19 | status | 1 = success, 0xff = failure | 1 byte |  |
|  |  |  |  |  |
| 20 | errorCode | Error code if a failure occurs  255 = Modem Error ( More than 3 retries, or expected Modem reply was not received  254 = Section Header in image is Incorrect  253 = Parameters are Incorrect (Start/End Address is wrong, or Bad Data length field in image)  252 = CRC Error (the image’s CRC is not correct)  251 = Timeout (transport took > 10 minutes) | 1 byte |  |
| 21-22 | msgCrc16 | The CRC16 received in the message | 2 bytes |  |
| 23-24 | calcCrc16 | Calculated CRC16 performed over the programmed flash. Should match the CRC16 sent in the message | 2 bytes |  |

### Error Code Definitions

|  |  |
| --- | --- |
| FW\_UP\_ERR\_NONE | 0 |
| FW\_UP\_ERR\_MODEM | -1 |
| FW\_UP\_ERR\_SECTION\_HEADER | -2 |
| FW\_UP\_ERR\_PARAMETER | -3 |
| FW\_UP\_ERR\_CRC | -4 |
| FW\_UP\_ERR\_TIMEOUT | -5 |