

# RS1xx LoRa Protocol

## Sentrius™ RS1xx Sensor

Application Note

v2.6

## 1 INTRODUCTION

The goal of this document is to detail the messages sent between the RS1xx sensor and a LoRa network server.

## 2 CONVENTIONS

### 2.1 Data Formats

#### 2.1.1 Data Types

Data types are based on the C language definitions for fixed point data types defined in *stdint.h* for C99 and newer compilers. These define whether a value is signed and how many bits are used to represent its value. For example, *int8\_t* is a signed value represented in eight bits and *uint32\_t* is a 32-bit unsigned value. The *\_t* at the end designates a type defined (rather than native) data type.

#### 2.1.2 Data Format

Temperature and humidity values are represented as two-byte values. The first byte is the fractional portion of the value and the second byte is the integer portion of the value. For example, a temperature of 27.43 degrees C has a fractional portion of 43 and an integer portion of 27. A temperature of -15.87 C would have a fractional portion of -87 and an integer portion of -15.

The following are the equations for temperature and humidity:

- $\text{Temp} = \text{Integer Portion} + (\text{Fractional Portion}/100)$
- $\text{Humidity} = \text{Integer Portion} + (\text{Fractional Portion}/100)$

---

**Note:** Temperature uses signed eight-bit values, while humidity uses unsigned eight-bit values. However, because humidity cannot be negative, the sign bit is never set and there is no practical difference between using signed or unsigned values for humidity.

---

## 3 SERVER-TO-SENSOR MESSAGES

### 3.1 Message Options

All Server-to-Sensor messages have the same options byte format. The options byte is always at byte index 1.

| Byte Index | Name    | Description                                 | Type | Default Value | Example Value | Min Value |
|------------|---------|---|------|---------------|---------------|-----------|
| [1]        | Options | Mote Response Options:                      | enum | 0x00          | 0x00          | 0x03      |
|            |         | 0   None                                    |      |               |               |           |
|            |         | 1   Send simple configuration next uplink   |      |               |               |           |
|            |         | 2   Send advanced configuration next uplink |      |               |               |           |
|            |         | 3   Send firmware version next uplink       |      |               |               |           |

### 3.2 Message Name: Generic Data Retrieval Notification

#### Notification

Method Name: GenericDataRetrieval

#### Notes

Message Length = 2 bytes

Server Method: asGenericRequest

Direction: Downlink

#### Parameters

| Byte Index | Name    | Description                                     | Type | Default Value | Example Value | Min. Value |
|------------|---------|---|------|---------------|---------------|------------|
| [0]        | MsgType | Message type                                    | 0x01 | 0x01          | 0x01          | 0x01       |
| [1]        | Options | See the <a href="#">Message Options</a> section | enum | 0x00          | 0x00          | 0x03       |

#### Example Message

Sent MSB First.

01 00

### 3.3 Message Name: Set UTC Notification

#### Notification

Method Name: SetUTC

#### Notes

Message Length = 8 bytes

Server Method: generateRTCDownlink

Direction: Downlink

#### Parameters

| Byte Index | Name        | Description                                     | Type    | Default Value | Example Value | Min Value |
|------------|-------------|---|---------|---------------|---------------|-----------|
| [0]        | MsgType     | Message type                                    | 0x02    | 0x02          | 0x02          | 0x02      |
| [1]        | Options     | See the <a href="#">Message Options</a> section | enum    | 0x00          | 0x00          | 0x03      |
| [2]        | RTC_Year    | Year  | uint8_t | 0x11          | 0x00          | 0xFF      |
| [3]        | RTC_Month   | Month   | uint8_t | 0x01          | 0x00          | 0x0C      |
| [4]        | RTC_Day     | Day   | uint8_t | 0x01          | 0x00          | 0x1F      |
| [5]        | RTC_Hours   | Hours (24-hour format)                          | uint8_t | 0x05          | 0x00          | 0x17      |
| [6]        | RTC_Minutes | Minutes   | uint8_t | 0x0F          | 0x00          | 0x3B      |
| [7]        | RTC_Seconds | Seconds   | uint8_t | 0x1E          | 0x00          | 0x3B      |

#### Example Message

Sent MSB First.

```
02 00 11 01 01 05 0F 1E
```

## 3.4 Message Name: Sensor Config Notification

### Notification

Method Name: Config

### Notes

Message Length = 16 bytes.

Server Method: *asSetSensorParameters*

Direction: Downlink

*ReadSensorPeriod* determines the time between individual temperature and humidity readings. The *SensorAggregate* determines how many readings are accumulated prior to sending a data packet to the gateway. For example, if the *SensorAggregate* is set to 2 and the *ReadSensorPeriod* is set to 60 seconds, the data is sent over LoRa 2 x 60 seconds or every 120 seconds. Increasing the *SensorAggregate* value has a positive effect on battery life.

### Parameters

| Byte Index | Name                                     | Description  | Type    | Default Value                     | Example Value | Min Value                                | Max Value |        |        |      |      |
|------------|--|--|---------|-----------------------------------|---------------|--|-----------|--------|--------|------|------|
| [0]        | MsgType                                  | Message type   | 0x03    | 0x03                              | 0x03          | 0x03                                     | 0x03      |        |        |      |      |
| [1]        | Options                                  | See the <a href="#">Message Options</a> section  | enum    | 0x00                              | 0x00          | 0x00                                     | 0x03      |        |        |      |      |
| [2]        | Config_BatteryType                       | <table border="1"> <tr> <td>1</td> <td>Zinc-Manganese Dioxide (Alkaline)</td> </tr> <tr> <td>2</td> <td>Lithium/Iron Disulfide (Primary Lithium)</td> </tr> </table> | 1       | Zinc-Manganese Dioxide (Alkaline) | 2             | Lithium/Iron Disulfide (Primary Lithium) | uint8_t   | 0x01   | 0x01   | 0x01 | 0x02 |
| 1          | Zinc-Manganese Dioxide (Alkaline)        |  |         |                                   |               |  |           |        |        |      |      |
| 2          | Lithium/Iron Disulfide (Primary Lithium) |  |         |                                   |               |  |           |        |        |      |      |
| [3:4]      | Config_ReadSensorPeriod                  | Period in seconds to read the sensor<br><table border="1"> <tr> <td>0</td> <td>Disabled</td> </tr> </table>  | 0       | Disabled                          | uint16_t      | 0x0000                                   | 0x000A    | 0x0000 | 0xFFFF |      |      |
| 0          | Disabled                                 |  |         |                                   |               |  |           |        |        |      |      |
| [5]        | Config_SensorAggregate                   | Number of readings to aggregate before sending on LoRa   | uint8_t | 0x01                              | 0x01          | 0x01                                     | 0x0B      |        |        |      |      |
| [6]        | Config_TempAlarmEnable                   | Enable temperature alarm   | bool    | 0                                 | 1             | 0  | 1         |        |        |      |      |
| [7]        | Config_HumidityAlarmEnable               | Enable humidity alarm  | bool    | 0                                 | 0             | 0  | 1         |        |        |      |      |
| [8]        | Config_TempAlarmLimitLow                 | Temperature alarm limit - low  | int8_t  | 0x00                              | 0x00          | 0xD8                                     | 0x55      |        |        |      |      |
| [9]        | Config_TempAlarmLimitHigh                | Temperature alarm limit - high   | int8_t  | 0x00                              | 0x32          | 0xD8                                     | 0x55      |        |        |      |      |
| [10]       | Config_RHAlarmLimitLow                   | Humidity alarm limit - low   | int8_t  | 0x0A                              | 0x0A          | 0x00                                     | 0x64      |        |        |      |      |
| [11]       | Config_RHAlarmLimitHigh                  | Humidity alarm limit - high  | int8_t  | 0x50                              | 0x50          | 0x00                                     | 0x64      |        |        |      |      |
| [12:13]    | Config_LED_BLE                           | Flash period in seconds when in BLE connection<br><table border="1"> <tr> <td>0</td> <td>No flash</td> </tr> </table>  | 0       | No flash                          | uint16_t      | 0x000A                                   | 0x000A    | 0x0000 | 0xFFFF |      |      |
| 0          | No flash                                 |  |         |                                   |               |  |           |        |        |      |      |

| Byte Index | Name            | Description             | Type     | Default Value | Example Value | Min Value | Max Value                                     |
|------------|-----------------|-------------------------|----------|---------------|---------------|-----------|---|
| [14:15]    | Config_LED_LoRa | Flash period in seconds | uint16_t | 0x000A        | 0x000A        | 0x0000    | 0xFFFF  |
|            |                 | 0                       |          |               |               |           | No flash                                      |
|            |                 | 65535                   |          |               |               |           | Tx/Rx debug mode<br>Tx – green<br>Rx – orange |

### Example Message

Sent MSB First.

03 00 01 00 0A 01 01 00 00 32 0A 50 00 0A 00 0A

### 3.5 Message Name: Heater Control Notification

#### Notification

Method Name: HeaterControl

#### Notes

Message Length = 5 bytes

Server Method: asSetHeater

Direction: Downlink

#### Parameters

| Index | Name                          | Description  | Type     | Default Value | Example Value | Min Value | Max Value |
|-------|-------------------------------|--|----------|---------------|---------------|-----------|-----------|
| [0]   | MsgType                       | Message type   | 0x04     | 0x04          | 0x04          | 0x04      | 0x04      |
| [1]   | Options                       | See the <a href="#">Message Options</a> section  | enum     | 0x00          | 0x00          | 0x00      | 0x03      |
| [2]   | Options_Si702x_Heater_Setting | Setting that controls the power sent to the heating element. See si7021 datasheet for more details | uint8_t  | 0x00          | 0x02          | 0x00      | 0x0F      |
| [3:4] | Options_Si702x_Heater_Time    | Time in milliseconds for the unit to turn on the heating element<br>0 Heater disabled              | uint16_t | 0x0000        | 0x000A        | 0x0000    | 0xFFFF    |

#### Example Message

Sent MSB First.

04 00 02 00 0A

## 3.6 Message Name: Backoff Notification

### Notification

Method Name: Backoff

### Notes

Message Length = 4 bytes

Server Method: *asRequestBackoff*

Direction: Downlink

### Parameters

| Index | Name          | Description                                     | Type     | Default Value | Example Value | Min Value | Max Value |
|-------|---------------|---|----------|---------------|---------------|-----------|-----------|
| [0]   | MsgType       | Message type                                    | uint8_t  | 0x05          | 0x05          | 0x05      | 0x05      |
| [1]   | Options       | See the <a href="#">Message Options</a> section | enum     | 0x00          | 0x00          | 0x00      | 0x03      |
| [2:3] | BackoffPeriod | Amount of time in seconds to backoff the sensor | uint16_t | 0x0000        | 0x003C        | 0x0000    | 0xFFFF    |

### Example Message

Sent MSB First.

```
05 00 00 3C
```

## 3.7 Message Name: FIFO Backlog Retrieval Notification

### Notification

Method Name: BacklogRetrievalFIFO

### Notes

Message Length = 6 bytes

Server Method: asRequestBacklogFIFO

Direction: Downlink

This command instructs the module to upload the stored logs and alarms oldest record first.

### Parameters

| Index | Name                  | Description   | Type     | Default Value | Example Value | Min Value | Max Value |
|-------|-----------------------|---|----------|---------------|---------------|-----------|-----------|
| [0]   | MsgType               | Message type  | uint8_t  | 0x06          | 0x06          | 0x06      | 0x06      |
| [1]   | Options               | See the <a href="#">Message Options</a> section   | enum     | 0x00          | 0x00          | 0x00      | 0x03      |
| [2:3] | Backlog_PullReqNum    | Number of records to send<br><br><b>Note:</b> Alarm messages are sent first.<br><br>0 Cancel pull request | uint16_t | 0x0000        | 0x0001        | 0x0000    | 0xFFFF    |
| [4:5] | Backlog_PullReqPeriod | How often to send a backlog LoRa packet (seconds)   | uint16_t | 0x0000        | 0x000A        | 0x0000    | 0xFFFF    |

### Example Message

```
Sent MSB First.
06 00 00 01 00 0A
```

## 3.8 Message Name: Format Log Flash Notification

### Notification

Method Name: FormatLogFlash.

### Notes

Message Length = 2 bytes

Server Method: asFormatFlash

Direction: Downlink

### Parameters

| Index | Name    | Description                                     | Type | Default Value | Example Value | Min Value |
|-------|---------|---|------|---------------|---------------|-----------|
| [0]   | MsgType | Message type                                    | 0x07 | 0x07          | 0x07          | 0x07      |
| [1]   | Options | See the <a href="#">Message Options</a> section | enum | 0x00          | 0x00          | 0x03      |

### Example Message

```
Sent MSB First.
07 00
```



### 3.9 Message Name: Set Alkaline Thresholds Notification

#### Notification

Method Name: SetAlkalineThresholds.

#### Notes

Message Length = 14 bytes

Server Method: asSetAlkalineThresholds

Direction: Downlink

#### Parameters

| Index   | Name          | Description                                     | Type     | Default Value | Example Value | Min Value | Max Value |
|---------|---------------|---|----------|---------------|---------------|-----------|-----------|
| [0]     | MsgType       | Message type                                    | uint8_t  | 0x08          | 0x08          | 0x08      | 0x08      |
| [1]     | Options       | See the <a href="#">Message Options</a> section | enum     | 0x00          | 0x00          | 0x00      | 0x03      |
| [2:3]   | EightyPercent | 80% threshold in mV                             | uint16_t | 0x0A96        | 0x0A96        | 0x0000    | 0xFFFF    |
| [4:5]   | SixtyPercent  | 60% threshold in mV                             | uint16_t | 0x0A14        | 0x0A14        | 0x0000    | 0xFFFF    |
| [6:7]   | FortyPercent  | 40% threshold in mV                             | uint16_t | 0x099C        | 0x099C        | 0x0000    | 0xFFFF    |
| [8:9]   | TwentyPercent | 20% threshold in mV                             | uint16_t | 0x0898        | 0x0898        | 0x0000    | 0xFFFF    |
| [10:11] | FivePercent   | 5% threshold in mV                              | uint16_t | 0x076C        | 0x076C        | 0x0000    | 0xFFFF    |
| [12:13] | Offset        | Offset from 20 degrees C in mV                  | uint16_t | 0x0064        | 0x0064        | 0x0000    | 0xFFFF    |

#### Example Message

Sent MSB First.

```
08 00 0A 96 0A 14 09 9C 08 98 07 6C 00 64
```

### 3.10 Message Name: Set Lithium Thresholds Notification

#### Notification

Method Name: SetLithiumThresholds

#### Notes

Message Length = 14 bytes

Server Method: asSetLithiumThresholds

Direction: Downlink

#### Parameters

| Index   | Name          | Description                                     | Type     | Default Value | Example Value | Min Value | Max Value |
|---------|---------------|---|----------|---------------|---------------|-----------|-----------|
| [0]     | MsgType       | Message type                                    | 0x09     | 0x09          | 0x09          | 0x09      |           |
| [1]     | Options       | See the <a href="#">Message Options</a> section | 0x00     | 0x00          | 0x00          | 0x03      |           |
| [2:3]   | EightyPercent | 80% threshold in mV                             | uint16_t | 0x0B86        | 0x0B86        | 0x0000    | 0xFFFF    |
| [4:5]   | SixtyPercent  | 60% threshold in mV                             | uint16_t | 0x0B40        | 0x0B40        | 0x0000    | 0xFFFF    |
| [6:7]   | FortyPercent  | 40% threshold in mV                             | uint16_t | 0x0AF0        | 0x0AF0        | 0x0000    | 0xFFFF    |
| [8:9]   | TwentyPercent | 20% threshold in mV                             | uint16_t | 0x0A5A        | 0x0A5A        | 0x0000    | 0xFFFF    |
| [10:11] | FivePercent   | 5% threshold in mV.                             | uint16_t | 0x0834        | 0x0834        | 0x0000    | 0xFFFF    |
| [12:13] | Offset        | Offset from 20 degrees C in mV                  | uint16_t | 0x0096        | 0x0096        | 0x0000    | 0xFFFF    |

#### Example Message

Sent MSB First.

```
09 00 0B 86 0B 40 0A F0 0A 5A 08 34 00 96
```

## 3.11 Message Name: LIFO Backlog Retrieval Notification

### Notification

Method Name: BacklogRetrievalLIFO

### Notes

Message Length = 6 bytes

Server Method: asRequestBacklogLIFO

Direction: Downlink

This command instructs the module to upload the stored logs and alarms oldest record first.

### Parameters

| Index | Name                  | Description  | Type     | Default Value | Example Value | Min Value | Max Value |
|-------|-----------------------|--|----------|---------------|---------------|-----------|-----------|
| [0]   | MsgType               | Message type.  | uint8_t  | 0x0A          | 0x0A          | 0x0A      | 0x0A      |
| [1]   | Options               | See the <a href="#">Message Options</a> section  | enum     | 0x00          | 0x00          | 0x00      | 0x03      |
| [2:3] | Backlog_PullReqNum    | Number of records to send.<br><b>Note:</b> Alarm messages are sent first.<br>0   Cancel pull request | uint16_t | 0x0000        | 0x0001        | 0x0000    | 0xFFFF    |
| [4:5] | Backlog_PullReqPeriod | How often to send a backlog LoRa packet (seconds).   | uint16_t | 0x0000        | 0x000A        | 0x0000    | 0xFFFF    |

### Example Message

```
Sent MSB First.
0A 00 00 01 00 0A
```

## 3.12 Message Name: Cancel Backlog Retrieval

### Notification

Method Name: CancelBacklogRetrieval

### Notes

Message Length = 2 bytes

Server Method: asCancelBacklogRetrieval

Direction: Downlink

This command instructs the module to cancel the uploading of the logs and alarms.

### Parameters

| Index | Name    | Description                                     | Type    | Default Value | Example Value | Min Value | Max Value |
|-------|---------|---|---------|---------------|---------------|-----------|-----------|
| [0]   | MsgType | Message type.                                   | uint8_t | 0x0B          | 0x0B          | 0x0B      | 0x0B      |
| [1]   | Options | See the <a href="#">Message Options</a> section | enum    | 0x00          | 0x00          | 0x00      | 0x03      |

### Example Message

```
Sent MSB First.
0B 00
```

## 4 SENSOR-TO-SERVER MESSAGES

### 4.1 Sensor-to-Server Message Options

All Sensor-to-Server messages have the same options byte format. The options byte is always at byte index 1

| Byte Index | Name    | Description                        | Type     | Default Value | Example Value | Min Value | Max Value |
|------------|---------|------------------------------------|----------|---------------|---------------|-----------|-----------|
| [1]        | Options | BIT POSITION                       | bitfield | 0x00          | 0x00          | 0x00      | 0x0F      |
|            |         | 0   Sensor request for server time |          |               |               |           |           |
|            |         | 1   Sensor configuration error     |          |               |               |           |           |
|            |         | 2   Sensor alarm flag              |          |               |               |           |           |
|            |         | 3   Sensor reset flag              |          |               |               |           |           |

## 4.2 Message Name: Send Temp RH Data Notification

### Notification

Method Name: SendTempRHData

### Notes

Message Length = 11 bytes

Server Method: asTempAndRhData Ni

Direction: Uplink

**Note:** There is no timestamp in this message. The server must assume the time it receives this message is the timestamp and this should almost always be within a few seconds of the actual time.

A *Sensor Configuration Error* in the Options byte means that the sensor is operating in North America at DR0, where the maximum LoRa payload is 11-bytes, but it is configured with an aggregate count > 1, which results in a payload greater than 11-bytes. In this case, the sensor stores the data it can't send to FLASH. The server should change the sensor aggregate count in this case.

### Parameters

| Byte Index | Name               | Description  | Type     | Default Value | Example Value | Min Value | Max Value |        |   |        |   |        |   |         |         |      |      |      |      |
|------------|--------------------|--|----------|---------------|---------------|-----------|-----------|--------|---|--------|---|--------|---|---------|---------|------|------|------|------|
| [0]        | MsgType            | Message type   | uint8_t  | 0x01          | 0x01          | 0x01      | 0x01      |        |   |        |   |        |   |         |         |      |      |      |      |
| [1]        | Options            | See the <a href="#">Sensor-To-Server Message Options</a> section   | bitfield | 0x00          | 0x00          | 0x00      | 0x0F      |        |   |        |   |        |   |         |         |      |      |      |      |
| [2]        | HumidityFractional | Fractional portion of humidity measurement in %  | uint8_t  | 0x1E          | 0x1E          | 0x00      | 0x64      |        |   |        |   |        |   |         |         |      |      |      |      |
| [3]        | HumidityInteger    | Decimal portion of humidity measurement in %   | uint8_t  | 0x01          | 0x01          | 0x00      | 0xFF      |        |   |        |   |        |   |         |         |      |      |      |      |
| [4]        | TempFractional     | Fractional portion of temperature measurement in C   | int8_t   | 0x41          | 0x41          | 0x00      | 0xFF      |        |   |        |   |        |   |         |         |      |      |      |      |
| [5]        | TempInteger        | Integer portion of temperature measurement in C  | int8_t   | 0x19          | 0x19          | 0xD8      | 0x55      |        |   |        |   |        |   |         |         |      |      |      |      |
| [6]        | BatteryCapacity    | Index for percentage of battery capacity remaining<br><table border="1" style="margin-left: 20px;"> <tr><td>0</td><td>0-5%</td></tr> <tr><td>1</td><td>5-20%</td></tr> <tr><td>2</td><td>20-40%</td></tr> <tr><td>3</td><td>40-60%</td></tr> <tr><td>4</td><td>60-80%</td></tr> <tr><td>5</td><td>80-100%</td></tr> </table> | 0        | 0-5%          | 1             | 5-20%     | 2         | 20-40% | 3 | 40-60% | 4 | 60-80% | 5 | 80-100% | uint8_t | 0x05 | 0x02 | 0x00 | 0x05 |
| 0          | 0-5%               |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| 1          | 5-20%              |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| 2          | 20-40%             |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| 3          | 40-60%             |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| 4          | 60-80%             |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| 5          | 80-100%            |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| [7:8]      | AlarmMsgCount      | Number of backlog alarm messages in sensor FLASH   | uint16_t | 0x0000        | 0x0000        | 0x0000    | 0x0FFF    |        |   |        |   |        |   |         |         |      |      |      |      |
| [9:10]     | BacklogMsgCount    | Number of backlog non-alarm messages in sensor FLASH   | uint16_t | 0x0000        | 0x0000        | 0x0000    | 0x0FFF    |        |   |        |   |        |   |         |         |      |      |      |      |

### Example Message

```
Sent MSB First.
01 00 1E 01 41 19 02 00 00 00 00
```

## 4.3 Message Name: Send Temp and RH Aggregated Data Notification

### Notification

Method Name: SendTempRHAggregatedData

### Notes

Server Method: asTempAndRhAggregate

Direction: Uplink

The length of this message is variable.

The MsgType, Options, Msg Counts, NumberReadings and Timestamp = 11 bytes

One Temperature and Humidity Reading = 4 bytes

Message Length = NumReading \* 4 + 11. Since NumReadings max is 10: 10 x 4 = 40. 40 + 11 = 51.

It is important that this packet be 51 bytes or less since the max EU packet across datarates is 51 bytes in length.

A *Sensor Configuration Error* in the Options byte means that the sensor is operating in North America at DR0, where the maximum LoRa payload is 11-bytes, but it is configured with an aggregate count > 1, which results in a payload greater than 11-bytes. In this case, the sensor stores the data it can't send to FLASH. The server should change the sensor configuration.

The timestamp is the time of the last sensor reading. The server must use the sensor read period parameter (part of the device configuration) to calculate the timestamps of the remaining data.

### Parameters

| Byte Index | Name            | Description  | Type     | Default Value | Example Value | Min Value | Max Value |        |   |        |   |        |   |         |         |      |      |      |      |
|------------|-----------------|--|----------|---------------|---------------|-----------|-----------|--------|---|--------|---|--------|---|---------|---------|------|------|------|------|
| [0]        | MsgType         | Message type.  | uint8_t  | 0x02          | 0x02          | 0x02      | 0x02      |        |   |        |   |        |   |         |         |      |      |      |      |
| [1]        | Options         | See the <a href="#">Sensor-To-Server Message Options</a> section   | bitfield | 0x00          | 0x00          | 0x00      | 0x0F      |        |   |        |   |        |   |         |         |      |      |      |      |
| [2]        | AlarmMsgCount   | Number of backlog alarm messages in sensor FLASH.  | uint8_t  | 0x00          | 0x00          | 0x00      | 0xFF      |        |   |        |   |        |   |         |         |      |      |      |      |
| [3:4]      | BacklogMsgCount | Number of backlog non-alarm messages in sensor FLASH   | uint16_t | 0x0000        | 0x0000        | 0x0000    | 0x0FFF    |        |   |        |   |        |   |         |         |      |      |      |      |
| [5]        | BatteryCapacity | Index for percentage of battery capacity remaining<br><table border="1" style="margin-left: 20px;"> <tr><td>0</td><td>0-5%</td></tr> <tr><td>1</td><td>5-20%</td></tr> <tr><td>2</td><td>20-40%</td></tr> <tr><td>3</td><td>40-60%</td></tr> <tr><td>4</td><td>60-80%</td></tr> <tr><td>5</td><td>80-100%</td></tr> </table> | 0        | 0-5%          | 1             | 5-20%     | 2         | 20-40% | 3 | 40-60% | 4 | 60-80% | 5 | 80-100% | uint8_t | 0x05 | 0x02 | 0x00 | 0x05 |
| 0          | 0-5%            |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| 1          | 5-20%           |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| 2          | 20-40%          |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| 3          | 40-60%          |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| 4          | 60-80%          |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| 5          | 80-100%         |  |          |               |               |           |           |        |   |        |   |        |   |         |         |      |      |      |      |
| [6]        | NumberReadings  | Number of sensor readings in packet.   | uint8_t  | -             | 0x02          | 0x01      | 0x0B      |        |   |        |   |        |   |         |         |      |      |      |      |

| Byte Index | Name               | Description  | Type     | Default Value | Example Value | Min Value  | Max Value  |
|------------|--------------------|--|----------|---------------|---------------|------------|------------|
| [7:10]     | Timestamp          | Seconds since Jan 1 2015                           | uint32_t | -             | 0x00000064    | 0x00000000 | 0xFFFFFFFF |
| [11]       | HumidityFractional | Fractional portion of humidity measurement in %    | uint8_t  | -             | 0x01          | 0x9D       | 0x64       |
| [12]       | HumidityInteger    | Decimal portion of humidity measurement in %       | uint8_t  | -             | 0x1E          | 0x00       | 0x64       |
| [13]       | TempFractional     | Fractional portion of temperature measurement in C | int8_t   | -             | 0x19          | 0x9D       | 0x63       |
| [14]       | TempInteger        | Integer portion of temperature measurement in C    | int8_t   | -             | 0x41          | 0xD8       | 0x55       |
| [15]       | HumidityFractional | Fractional portion of humidity measurement in %    | uint8_t  | -             | 0x03          | 0x1E       | 0x64       |
| [16]       | HumidityInteger    | Decimal portion of humidity measurement in %       | uint8_t  | -             | 0x1D          | 0x01       | 0xFF       |
| [17]       | TempFractional     | Fractional portion of temperature measurement in C | int8_t   | -             | 0x19          | 0x19       | 0xFF       |
| [18]       | TempInteger        | Integer portion of temperature measurement in C    | int8_t   | -             | 0x41          | 0xD8       | 0x55       |

### Example Message

Sent MSB First.

```
02 00 00 00 00 00 05 02 00 00 00 64 01 1E 19 41 03 1D 19 41
```

## 4.4 Message Name: Send BackLog Message Notification

### Notification

Method Name: SendBackLogMessage

### Notes

Message Length = 10 bytes.

Server Method: asBacklogTempAndRh

Direction: Uplink

The timestamp is the time the sensor data was sampled.

### Parameters

| Byte Index | Name               | Description  | Type     | Default Value | Example Value | Min Value  | Max Value  |
|------------|--------------------|--|----------|---------------|---------------|------------|------------|
| [0]        | MsgType            | Message type   | uint8_t  | 0x03          | 0x03          | 0x03       | 0x03       |
| [1]        | Options            | See the <a href="#">Sensor-To-Server Message Options</a> section | bitfield | 0x00          | 0x00          | 0x00       | 0x0F       |
| [2:5]      | Timestamp          | Seconds since Jan 1 2015   | uint32_t | -             | 0x00000064    | 0x00000000 | 0xFFFFFFFF |
| [6]        | HumidityFractional | Integer value of humidity measurement in %                       | uint8_t  | -             | 0x1E          | 0x00       | 0x63       |
| [7]        | HumidityInteger    | Decimal value of humidity measurement in %                       | uint8_t  | -             | 0x01          | 0x00       | 0x64       |
| [8]        | TempFractional     | Decimal value of temperature measurement in C                    | int8_t   | -             | 0x00          | 9D         | 0x63       |
| [9]        | TempInteger        | Integer value of temperature measurement in C                    | int8_t   | -             | 0x19          | 0xD8       | 0x55       |

### Example Message

Sent MSB First.

```
03 00 00 00 00 64 1E 01 41 19
```



## 4.5 Message Name: Send BackLog Messages Notification

### Notification

Method Name: SendBackLogMessages

### Notes

Server Method: *asBacklogTempAndRhAggregate*

Direction: Uplink

The length of this message is variable.

The MsgType, NumberReadings, and Options = 3 bytes.

The Timestamp, Temperature, and Humidity = 8 bytes.

Message Length = NumReading \* 8 + 3. Since NumReadings max is 6:  $6 \times 8 = 48$ .  $48 + 3 = 51$ .

It is important that this packet be 51 bytes or less since the max EU packet across datarates is 51 bytes in length.

A *Sensor Configuration Error* in the Options byte means that the sensor is operating in North America at DR0, where the maximum LoRa payload is 11-bytes, but it is configured with an aggregate count > 1, which results in a payload greater than 11-bytes. In this case, the sensor stores the data it can't send to FLASH. The server should change the sensor configuration.

The *Timestamp* is the time the sensor data was sampled.

### Parameters

| Byte Index | Name               | Description  | Type     | Default Value | Example Value | Min Value  | Max Value  |
|------------|--------------------|--|----------|---------------|---------------|------------|------------|
| [0]        | MsgType            | Message type   | uint8_t  | 0x04          | 0x04          | 0x04       | 0x04       |
| [1]        | Options            | See the <a href="#">Sensor-To-Server Message Options</a> section | bitfield | 0x00          | 0x00          | 0x00       | 0x0F       |
| [2]        | NumReading         | Number of sensor reading backlogs in packet                      | uint8_t  | 0x01          | 0x01          | 0x01       | 0x06       |
| [3:6]      | Timestamp          | Seconds since Jan 1 2015   | uint32_t | -             | 0x00000064    | 0x00000000 | 0xFFFFFFFF |
| [7]        | HumidityFractional | Integer value of humidity measurement in %                       | uint8_t  | -             | 0x1E          | 0x00       | 0x64       |
| [8]        | HumidityInteger    | Decimal value of humidity measurement in %                       | uint8_t  | -             | 0x01          | 0x00       | 0x64       |
| [9]        | TempFractional     | Decimal value of temperature measurement in C                    | int8_t   | -             | 0x41          | 0x9D       | 0x64       |
| [10]       | TempInteger        | Integer value of temperature measurement in C                    | int8_t   | -             | 0x19          | 0xD8       | 0x7D       |

### Example Message

```
Sent MSB First.
04 00 01 00 00 00 64 1E 01 41 19
```

## 4.6 Message Name: Send Sensor Config Simple Notification

### Notification

Method Name: SendSensorConfigSimple

### Notes

Message Length = 8 bytes.

Server Method: asSensorConfigSimple

Direction: Uplink

A *Sensor Configuration Error* in the Options byte means that the sensor is operating in North America at DR0, where the maximum LoRa payload is 11-bytes, but it is configured with an aggregate count > 1, which results in a payload greater than 11-bytes. In this case, the sensor stores the data it can't send to FLASH. The server should change the sensor configuration.

The *SensorAggregate* setting aggregates or collects sensor data every *ReadSensorPeriod* and only sends them over LoRa when the *SensorAggregate* number is reached. For example, if the *SensorAggregate* is set to 2 and the *ReadSensorPeriod* is set to 60 seconds, the data is sent over LoRa 2 x 60 seconds or every 120 seconds. This has a positive effect on battery life.

### Parameters

| Byte Index | Name                 | Description  | Type                                     | Default Value | Example Value | Min Value | Max Value |
|------------|----------------------|--|--|---------------|---------------|-----------|-----------|
| [0]        | MsgType              | Message type   | uint8_t                                  | 0x05          | 0x05          | 0x05      | 0x05      |
| [1]        | Options              | See the <a href="#">Sensor-To-Server Message Options</a> section | bitfield                                 | 0x00          | 0x00          | 0x00      | 0x0F      |
| [2]        | BatteryType          | 1  | Zinc-Manganese Dioxide (Alkaline)        | uint8_t       | 0x01          | 0x01      | 0x01      |
|            |                      | 2  | Lithium/Iron Disulfide (Primary Lithium) |               |               |           |           |
| [3:4]      | ReadSensorPeriod     | Period in seconds to read the sensor<br>0 Disabled               | uint16_t                                 | 0x003C        | 0x003C        | 0x001E    | 0xFFFF    |
| [5]        | SensorAggregate      | Number of readings to aggregate before sending on LoRa           | uint8_t                                  | 0x01          | 0x01          | 0x01      | 0x0B      |
| [6]        | TempAlarmEnabled     | False  | Disabled                                 | bool          | 0             | 0         | 0         |
|            |                      | True   | Enabled                                  |               |               |           |           |
| [7]        | HumidityAlarmEnabled | False  | Disabled                                 | bool          | 0             | 0         | 0         |
|            |                      | True   | Enabled                                  |               |               |           |           |

### Example Message

```
Sent MSB First.
05 00 01 00 00 01 00 00
```

## 4.7 Message Name: Send Sensor Config Advanced Notification

### Notification

Method Name: SendSensorConfigAdvanced

### Notes

Message Length = 16 bytes.

Server Method: asSensorConfigAdvanced

Direction: Uplink

A *Sensor Configuration Error* in the Options byte means that the sensor is operating in North America at DR0, where the maximum LoRa payload is 11-bytes, but it is configured with an aggregate count > 1, which results in a payload greater than 11-bytes. In this case, the sensor stores the data it can't send to FLASH. The server should change the sensor configuration.

The *SensorAggregate* setting aggregates or collects sensor data every *ReadSensorPeriod* and only sends them over LoRa when the *SensorAggregate* number is reached. For example, if the *SensorAggregate* is set to 2 and the *ReadSensorPeriod* is set to 60 seconds, the data is sent over LoRa 2 x 60 seconds or every 120 seconds. This has a positive effect on battery life.

### Parameters

| Byte Index                           | Name                                     | Description  | Type                                 | Default Value                     | Example Value | Min Value                                | Max Value |        |        |      |      |
|--------------------------------------|--|--|--------------------------------------|-----------------------------------|---------------|--|-----------|--------|--------|------|------|
| [0]                                  | MsgType                                  | Message type   | uint8_t                              | 0x06                              | 0x06          | 0x06                                     | 0x06      |        |        |      |      |
| [1]                                  | Options                                  | See the <a href="#">Sensor-To-Server Message Options</a> section   | bitfield                             | 0x00                              | 0x00          | 0x00                                     | 0x0F      |        |        |      |      |
| [2]                                  | BatteryType                              | <table border="1"> <tr> <td>1</td> <td>Zinc-Manganese Dioxide (Alkaline)</td> </tr> <tr> <td>2</td> <td>Lithium/Iron Disulfide (Primary Lithium)</td> </tr> </table> | 1                                    | Zinc-Manganese Dioxide (Alkaline) | 2             | Lithium/Iron Disulfide (Primary Lithium) | uint8_t   | 0x01   | 0x01   | 0x01 | 0x02 |
| 1                                    | Zinc-Manganese Dioxide (Alkaline)        |  |                                      |                                   |               |  |           |        |        |      |      |
| 2                                    | Lithium/Iron Disulfide (Primary Lithium) |  |                                      |                                   |               |  |           |        |        |      |      |
| [3:4]                                | ReadSensorPeriod                         | <table border="1"> <tr> <td>Period in seconds to read the sensor</td> </tr> <tr> <td>0 Disabled, sensor is not read</td> </tr> </table>                              | Period in seconds to read the sensor | 0 Disabled, sensor is not read    | uint16_t      | 0x003C                                   | 0x003C    | 0x0000 | 0xFFFF |      |      |
| Period in seconds to read the sensor |  |  |                                      |                                   |               |  |           |        |        |      |      |
| 0 Disabled, sensor is not read       |  |  |                                      |                                   |               |  |           |        |        |      |      |
| [5]                                  | SensorAggregate                          | Number of readings to aggregate before sending on LoRa   | uint8_t                              | 0x01                              | 0x01          | 0x01                                     | 0x0B      |        |        |      |      |
| [6]                                  | TempAlarmsEnabled                        | <table border="1"> <tr> <td>True</td> <td>Temperature alarms enabled</td> </tr> <tr> <td>False</td> <td>Temperature alarms disabled</td> </tr> </table>              | True                                 | Temperature alarms enabled        | False         | Temperature alarms disabled              | bool      | 0      | 0      | 0    | 1    |
| True                                 | Temperature alarms enabled               |  |                                      |                                   |               |  |           |        |        |      |      |
| False                                | Temperature alarms disabled              |  |                                      |                                   |               |  |           |        |        |      |      |
| [7]                                  | HumidityAlarmsEnabled                    | <table border="1"> <tr> <td>True</td> <td>Humidity alarms enabled</td> </tr> <tr> <td>False</td> <td>Humidity alarms disabled</td> </tr> </table>                    | True                                 | Humidity alarms enabled           | False         | Humidity alarms disabled                 | bool      | 0      | 0      | 0    | 1    |
| True                                 | Humidity alarms enabled                  |  |                                      |                                   |               |  |           |        |        |      |      |
| False                                | Humidity alarms disabled                 |  |                                      |                                   |               |  |           |        |        |      |      |
| [8]                                  | TempAlarmLimitLow                        | Temperature alarm limit - low in C   | int8_t                               | 0x00                              | 0x00          | 0xD8                                     | 0x55      |        |        |      |      |
| [9]                                  | TempAlarmLimitHigh                       | Temperature alarm limit - high in C  | int8_t                               | 0x00                              | 0x32          | 0xD8                                     | 0x55      |        |        |      |      |
| [10]                                 | HumidityAlarmLimitLow                    | Humidity alarm limit - low in %  | int8_t                               | 0x0A                              | 0x0A          | 0x00                                     | 0x64      |        |        |      |      |
| [11]                                 | HumidityAlarmLimitHigh                   | Humidity alarm limit - high in %   | int8_t                               | 0x50                              | 0x50          | 0x00                                     | 0x64      |        |        |      |      |

| Byte Index | Name          | Description  | Type     | Default Value | Example Value | Min Value | Max Value |
|------------|---------------|--|----------|---------------|---------------|-----------|-----------|
| [12:13]    | LED_BLE       | Flash period in seconds when in BLE connection<br>0   No flash   | uint16_t | 0x000A        | 0x000A        | 0x0000    | 0xFFFF    |
| [14:15]    | LED_Heartbeat | Flash period in seconds<br>0   No flash<br>65535   LED in LoRa Tx/Rx mode<br>Tx – green<br>Rx – orange | uint16_t | 0x000A        | 0x000A        | 0x0000    | 0xFFFF    |

### Example Message

Sent MSB First.

```
06 00 01 00 00 01 00 00 00 32 0A 50 00 0A 00 0A
```

## 4.8 Message Name: Send FW Version Notification

### Notification

Method Name: SendFWVersion

### Notes

Message Length = 11 bytes.

Server Method: asFwVersion

Direction: Uplink

A *Sensor Configuration Error* in the Options byte means that the sensor is operating in North America at DR0, where the maximum LoRa payload is 11-bytes, but it is configured with an aggregate count > 1, which results in a payload greater than 11-bytes. In this case, the sensor stores the data it can't send to FLASH. The server should change the sensor configuration.

### Parameters

| Byte Index | Name         | Description  | Type     | Default Value | Example Value | Min Value  | Max Value  |
|------------|--------------|--|----------|---------------|---------------|------------|------------|
| [0]        | MsgType      | Message type   | uint8_t  | 0x07          | 0x07          | 0x07       | 0x07       |
| [1]        | Options      | See the <a href="#">Sensor-To-Server Message Options</a> section | bitfield | 0x00          | 0x00          | 0x00       | 0x0F       |
| [2]        | Year         | Version year   | 0x00     | 0x11          | 0x00          | 0x00       | 0xFF       |
| [3]        | Month        | Version month  | 0x00     | 0x01          | 0x01          | 0x01       | 0x0C       |
| [4]        | Day          | Version day  | 0x00     | 0x01          | 0x01          | 0x01       | 0x1F       |
| [5]        | VersionMajor | Version major  | 0x00     | 0x01          | 0x00          | 0x01       | 0xFF       |
| [6]        | VersionMinor | Version minor  | 0x00     | 0x00          | 0x00          | 0x00       | 0xFF       |
| [7:10]     | PartNumber   | Part number of firmware  | uint32_t | 0x00000000    | 0x00493E6F    | 0x00000000 | 0xFFFFFFFF |

### Example Message

Sent MSB First.

```
07 00 11 01 01 01 00 00 49 3E 6F
```

## 5 REVISION HISTORY

| Version | Date             | Notes   | Contributor(s) | Approver         |
|---------|------------------|---|----------------|------------------|
| 1.0     | 17 Dec 2017      | Initial Release   |                | Jonathan Kaye    |
| 2.0     | 16 March 2018    | Updated to latest messages  |                | Chris Hofmeister |
| 2.1     | 11 April 2018    | Corrected message type values, put them in numeric order  |                | Mark Monson      |
| 2.2     | 3 August 2018    | Updated to new template. Updates to BatteryCapacity.  |                | Seokwoo Yoon     |
| 2.3     | 5 September 2018 | Updated options byte in uplink messages for version 4.2 FW  |                | Chris Hofmeister |
| 2.4     | 25 Jan 2019      | Updated <i>seconds</i> to <i>milliseconds</i> for <b>Options_Si702x</b>                                     | Mark Monson    | Jonathan Kaye    |
| 2.5     | 15 Aug 2019      | Updated formatting and aggregate uplink   | Mark Monson    | Jonathan Kaye    |
| 2.6     | 9 October 2019   | Updated content of Send Temp and RH Aggregated Data Notification and Send Sensor Config Simple Notification | Greg Leach     | Chris Boorman    |