TEKTELIC Communications Inc.
7657 10th Street NE Calgary, Alberta
Canada, T2E 8X2

Tundra Sensor

Technical Reference Manual

DOCUMENT TYPE:	Technical Reference Manual
DOCUMENT NUMBER:	T0006940_Tundra-Sensor_TRM
DOCUMENT VERSION:	0.2
PRODUCT NAME:	Tundra Sensor
PRODUCT CODE:	T0006778 Base Model T0007334 Base Model + Wall Mounting T0007380 Probe Model T0006779 Probe Model + Wall Mounting
FIRMWARE VERSION:	1.0.1
RELEASE DATE:	**FEBRUARY 16, 2021**

Revision History

Version	Date	Editor	Comments
0.1	November 10, 2020	Carter Mudryk	Initial draft based on Smart Room Sensors TRM v1.1.
0.2	February 16, 2021	Carter Mudryk	• Updated to include Store & Forward system UL and DL format. • Updated to include accelerometer configuration registers.

Table of Contents

List of Tables

List of Figures

Acronyms and Glossary

ABP	Activation By Personalization
ADR	Adaptive Data Rate
CRC	Cyclic Redundancy Check
DL	DownLink
DR	Data Rate
EIRP	Effective Isotropic Radiated Power
g	gravity (unit of acceleration ≈ 9.8 m/s^2)
ID	IDentity
IoT	Internet of Things
LED	Light-Emitting Diode
LoRa	a patented “Long-Range” IoT technology acquired by Semtech
LoRaMAC	LoRaWAN MAC
LoRaWAN	LoRa Wide Area Network (a network protocol based on LoRa)
LSb	Least Significant bit
MAC	Medium Access Control
MCU	Microcontroller Unit
min	minute(s)
MSb	Most Significant bit
NS	Network Server
NVM	Non-Volatile Memory
OTA	Over-The-Air
OTAA	OTA Activation
RH	Relative Humidity
RLE	Run-Length Encoding
RO	Read-Only
R/W	Read/Write
Rx	Receive
sec	second(s)
Sensor	Tundra Sensor
S&F	Store & Forward
TRM	Technical Reference Manual (this document)
Tx	Transmit
UL	UpLink
WO	Write-Only

1 Overview

This Technical Reference Manual (TRM) describes the user accessible configuration settings (pseudo-registers) supported by the Tundra Sensor, previously known as the Cold Room Sensor, and referred to as the Sensor henceforth. This document is intended for a technical audience with an understanding of the NS and its command interfaces, such as application developers or engineers.

The Tundra Sensor is a multi-purpose LoRaWAN IoT sensor packed into a small form factor. The Sensor is ideal for monitoring and reporting temperature (ambient, remote through a probe, or MCU) and ambient relative humidity. The ambient temperature and ambient relative humidity readings are saved on the device using the Store & Forward (S&F) system to ensure data is not lost if network connectivity is lost. Additionally, the Sensor is equipped with an accelerometer and magnetic reed switch. Table 1-1 lists the Sensor variants for regions identified by the LoRa Alliance [2]—see [2] for the Tx and Rx bands in each LoRaWAN region.

Table 1-1: Tundra Sensor Region Specific Variants

LoRaWAN RF Variant	Functional HW Variant	Module T-Code
AS923 AU915 EU868 IN865 US915	Base	T0006778
	Base + wall mounting	T0007334
	Probe	T0007380
	Probe + wall mounting	T0006779

Regarding communication direction (UL or DL) and LoRaWAN ports, all information streams currently supported by the SW are as follows:

- Readings obtained from the accelerometer, battery gauge, and MCU thermometer (***sent in UL, LoRaWAN port 10***). This data is not tagged nor stored in the sensor memory.
- Readings obtained from the ambient temperature and relative humidity transducer (***sent in UL, LoRaWAN port 32***). This data is tagged and stored in the sensor memory.
- Forwarding of tagged, stored data after a DL request for retrieval (***sent in UL, LoRaWAN port 33***).
- Requests for retrieval of stored data (***sent in DL, LoRaWAN ports 112-122***).
- Configuration and control commands from the NS used to change the Sensor's behavior or inquire the Sensor for the values of registers (***sent in DL, LoRaWAN port 100***)
- Response to configuration and control commands from the NS (***sent in UL, LoRaWAN port 100***)

The default configuration of the Sensor for reporting transducer readings includes the following:

- Report the battery voltage every hour.
- Report and store the ambient temperature every hour.
- Report and store the ambient RH every hour.

In the Sections 2 and 3, respectively, the UL (departing from the Sensor) and DL (destined to the Sensor) payload formats are explained.

1.1 Reed Switch Operation

The Sensor is equipped with a magnetic reed switch. The reed switch can be operated by the provided magnet, and is used for the following purposes:

1) MCU reset upon observing a specified magnetic pattern:

   This is mainly used to wake up the Sensor from DEEP SLEEP and having it try to join the network. When the Sensor comes out of the factory it is in the DEEP SLEEP mode and can be activated using the specified magnetic pattern. The same magnetic pattern can just be used to reset the Sensor during normal operation, causing it to try to rejoin the network.

   The magnetic pattern in this application is illustrated in Figure 1-1. A "magnet presence" is achieved by attaching the magnet to the enclosure at the magnet symbol. A "magnet absence" is achieved by taking the magnet away from the enclosure. Figure 1-1 shows that the pattern involves sustaining a "magnet presence" continuously for at least 3 sec but less than 10 sec.

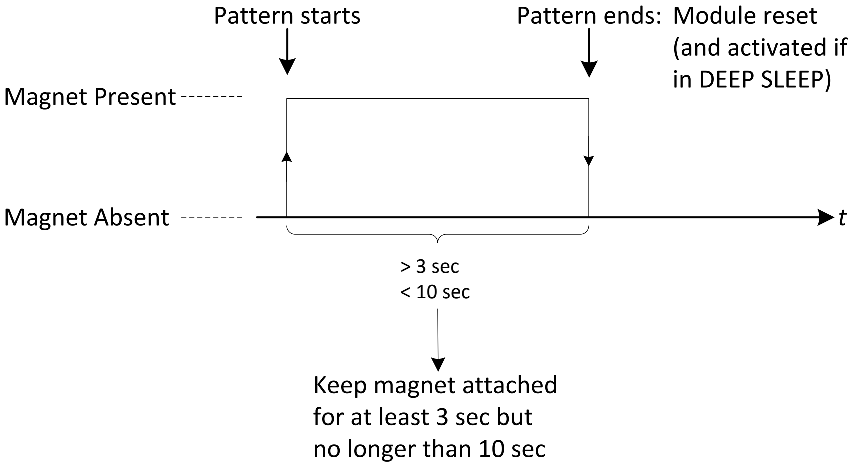


Figure 1-1: The Sensor magnetic reset/wake-up pattern.

   When the specified magnetic pattern is applied to the Sensor, the Sensor displays an LED indication, described in Section 1.2.2, to show that it has accepted the magnetic pattern. Then the Sensor is reset and tries to rejoin the network.

   The same magnetic pattern can just be used to reset the Sensor during normal operation, causing it to try to rejoin the network. If the magnetic pattern is applied while the Sensor is trying to join the network, but before it receives the JOIN ACCEPT, the Sensor will go back into DEEP SLEEP mode. Applying the magnetic pattern again will cause it to wake up.

2) Triggering the Sensor to uplink something upon observing a magnetic pattern:

This is used to get the LoRaWAN Class-A Sensor to open a receive window so it can receive DL commands from the NS, or simply to trigger the Sensor to uplink some desired transducer data.

The magnetic pattern in this case is not user configurable, and involves attaching and taking away the magnet to and from the magnet sign at the bottom of the enclosure once, all in less than 2 sec, as shown in Figure 1-2. *It is important to note here that mistakenly holding the magnet attached to the module for more than 3 sec may trigger a module reset, as explained in item 1.*

The magnetic-forced UL does not contain a payload (empty payload) and is sent on ***LoRaWAN port 0.***

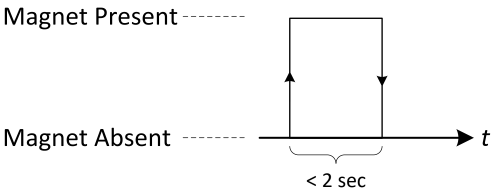


Figure 1-2: The Sensor magnetic UL-triggering pattern.

1.2 LEDs Behavior

The Tundra Sensor is equipped with two on-board LEDs: GREEN and RED. The LED behavior in the Sensor is specified in Sections 1.2.1 and 1.2.2.

1.2.1 Normal Operation

The LED behavior under normal operation is described as follows:

1. Both **GREEN** and **RED** are turned off when the Sensor is powered on.
2. Both **GREEN** and **RED** are turned on when the POST begins.
3. When the POST ends, depending on the POST result:
    a. If the POST passes, **GREEN** is toggled ON and OFF every 50 ms for 0.5 sec, as shown in Figure 1-3.
    b. If the POST fails, **RED** is toggled ON and OFF every 50 ms for 0.5 sec, as shown in Figure 1-3.
4. Both **GREEN** and **RED** are turned off when the POST and the subsequent LED flashing specified in item 3 end.
5. While the Sensor is attempting to join:
    a. **GREEN** is toggled ON and OFF every 50 ms for the first hour. But after that, it just flashes 5 times (ON time: 50 ms, OFF time: 50 ms) every 5 sec. This scheme has been shown in Figure 1-4.
    b. **RED** flashes just once:
        i. with a pulse duration of 25 ms right after transmitting a JOIN REQUEST
        ii. with a pulse duration of 100 ms right after receiving a JOIN ACCEPT

6. After the Sensor joined:
    a. **RED** flashes just once with a pulse duration of 25 ms right after transmitting an uplink
    b. **GREEN** flashes just once with a pulse duration of 25 ms right after receiving a downlink

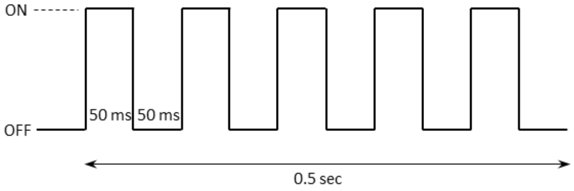


Figure 1-3: The LEDs pattern for GREEN or RED after the POST.

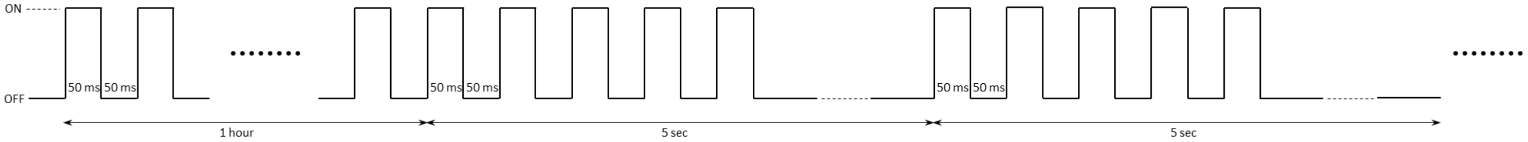


Figure 1-4: The GREEN pattern during join.

1.2.2 DEEP SLEEP

The Sensor displays an LED indication when it is put into DEEP SLEEP by pressing the sleep button on the PCBA. The LED indication is displayed about 5 sec after the button is pressed. The LED pattern is run by **RED**, as shown in Figure 1-5.

There is another similar LED pattern, shown in Figure 1-6, that is played with **GREEN** after the magnetic wake-up/reset pattern is applied to the Sensor.

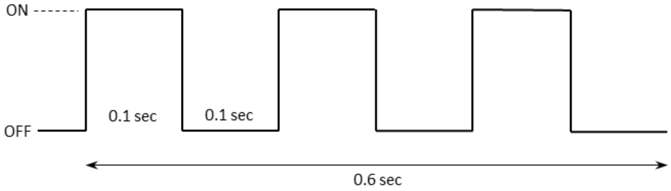


Figure 1-5: The LED pattern with RED before going to DEEP SLEEP.

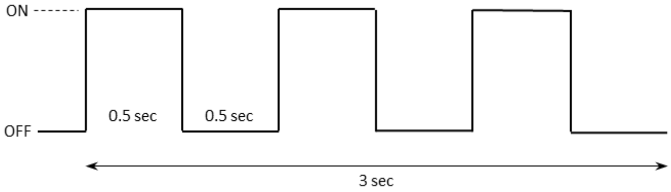


Figure 1-6: The LED pattern with GREEN after the magnetic wake-up/reset pattern is observed.

1.3 Store & Forward

Store & forward (S&F) is a system design feature intended to support the storing of Tundra Sensor transducer data such that it can be accessed and forwarded to the Network Server and Application layer at a later time. This is to prevent data loss in the case of network connection loss, dropped packets, dead battery, or otherwise.

The S&F system implemented for the Tundra Sensor is capable of storing 2 types of transducer data: the ambient temperature and the ambient relative humidity. All other transducer data is not stored and cannot be retrieved at a later time.

1.3.1 Tagging

The Sensor tags all the data packets that are intended to be uplinked, and that can be retransmitted later if requested, with a 2-byte *tag header*. The tag header is the first 2 bytes in a tagged frame payload, and is a wraparound counter indexing the data packets that are uplinked, and stored in the non-volatile memory (NVM), so they can be retrieved and retransmitted later if requested by the application. Figure 1-7 shows this scheme.

Note: Tags are incremented for all sensed data destined to go over the air, whether or not the sensor succeeds at sending them over the air. Also, all tagged sensed data are saved to the flash regardless of the sensor being successful at uplinking them. Failure at uplinking can be due to LoRaMAC rejecting the packet because of duty cycle limitations in some regions.

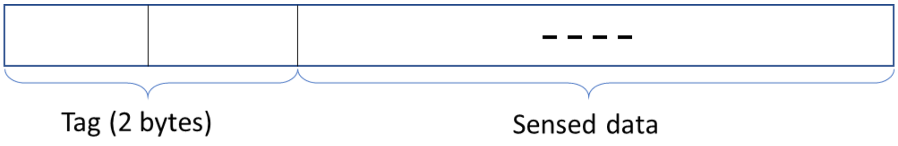


Figure 1-7: Tagged LoRaWAN frame payload.

1.3.2 Storing

The Sensor stores every tagged frame payload in its NVM. This includes the tag header and corresponding data sample(s) in the payload.

The Sensor can store up to 3000 tagged entries. Each entry can include ambient temperature data, ambient RH data, or both. For example, if the Sensor is configured to sample and UL temperature twice as frequently as RH, entry 0001 would contain both temperature and RH, entry 0002 would only contain temperature, entry 0003 would include both, and so on. Once tag 3000 has data written to it, the next entry will overwrite tag 0001. This way, the oldest entries are overwritten with newest data once every tag is full.

Stored data persists through resets, loss of Sensor power, and loss of network coverage.

1.3.3 Forwarding

The sensor can retrieve the data identified by their tag headers from its NVM. The requested tag headers are communicated to the sensor by the user or application via a DL. The Sensor responds with the retrieved data by forwarding it via a UL. Please see Sections 2.3 and 3.4 for the UL and DL formats for forwarding and requesting stored data.

2 UL Payload Formats

The UL streams (from the Sensor to the NS) include the following:

- The non-tagged readings obtained from on-board transducers (***sent on LoRaWAN port 10***).
- The tagged readings obtained from on-board transducers (***sent on LoRaWAN port 32***).
- The forwarded tagged readings that were stored in memory (***sent on LoRaWAN port 33***).
- Response to configuration and control commands from the NS (***sent on LoRaWAN port 100***)

These topics are explained in the following subsections.

2.1 Frame Payload to Report Non-Tagged Transducer Data

Each data field from the Sensor is encoded in a frame format shown in Figure 2-1. A big-endian format (MSB first) is always followed.

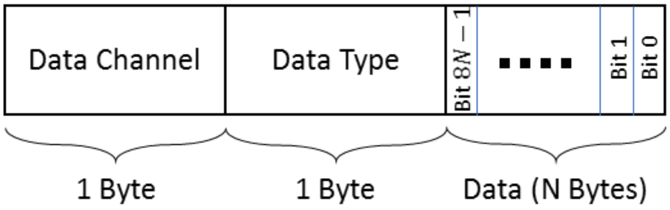


Figure 2-1: The UL frame payload format.

A Sensor message payload can include multiple transducer data frames. The ordering of frames is not guaranteed (they can be in any order). A single payload may include data from any given transducer. The Sensor payload frame values are shown in Table 2-1. In this table, the bit indexing scheme is as shown in Figure 2-1. Payload frame values in Table 2-1 has been grouped by bolded boundaries. This grouping is only to indicate which payloads are related to the same physical transducer. The grouping *does not imply* that the payloads within the same group are uplinked together.

Transducer data in the UL are sent through ***LoRaWAN port 10***.

Table 2-1: UL Frame Payload Values for Transducer Data

Information Type	Channel ID	Type ID	Size	Data Type	Data Format	JSON Variable (Type/Unit)
Battery Voltage	0x00	0xFF	2 B	Analog	• 10 mV / LSB (signed)	*battery_voltage: <value> (signed/volt)*
MCU Temperature	0x0B	0x67	2 B	Temperature	• 0.1°C / LSB (signed)	*mcu_temperature: <value> (signed/°C)*
Impact Alarm	0x0C	0x00	1 B	Digital	• 0x00 = Impact alarm inactive • 0xFF = Impact alarm active	*impact_alarm: <value> (unsigned/no unit)*

Acceleration Magnitude	0x05	0x02	2 B	Analog	• 1 milli-g/LSB (unsigned)	*impact_magnitude: <value>* *(unsigned/g)*
Acceleration Vector	0x07	0x71	6 B	Acceleration	• 1 milli-g/LSB (signed) • Bits 32-47: X-axis acceleration • Bits 16-31: Y-axis acceleration • Bits 0-15: Z-axis acceleration	*acceleration {* *xaxis: <value>, (signed/g)* *yaxis: <value>, (signed/g)* *zaxis: <value> (signed/g)* *}*

Examples:

In the following example payloads, the data channel ID and data type ID are boldfaced:

- 0x **0B 67** 00 0A
    - $0\text{x }\mathbf{0B\ 67}\ (\text{MCU Temperature}) = (0\text{x }00\ 0\text{A}) \times 0.1°\text{C} = 1°\text{C}$
- 0x **0C 00** FF **05 02** 07 D0
    - $0\text{x }\mathbf{0C\ 00}\ (\text{Impact Alarm}) = (0\text{x }FF) = \text{Alarm active}$
    - $0\text{x }\mathbf{05\ 02}\ (\text{Acceleration Magnitude}) = (0\text{x }07\ \text{D}0) \times 0.001\ \text{g} = 2000\ \text{g}$
- 0x **00 FF** 01 2C
    - $0\text{x }\mathbf{00\ FF}\ (\text{Battery Voltage}) = (0\text{x }01\ 2\text{C}) \times 0.01\ \text{V} = 3.00\ \text{V}$

2.2 Frame Payload to Report Tagged Transducer Data

All tagged data are sent on ***LoRaWAN port 32 (0x20)*** when first sampled. The application can easily identify a tagged payload from the port number. After the UL is sent, the tagged data is stored in the NVM for later retrieval.

Each tagged data field from the Sensor is encoded in a frame format shown in Figure 2-2. A big-endian format (MSB first) is always followed. The tagged data included in the UL can be the ambient temperature, ambient RH, or both.

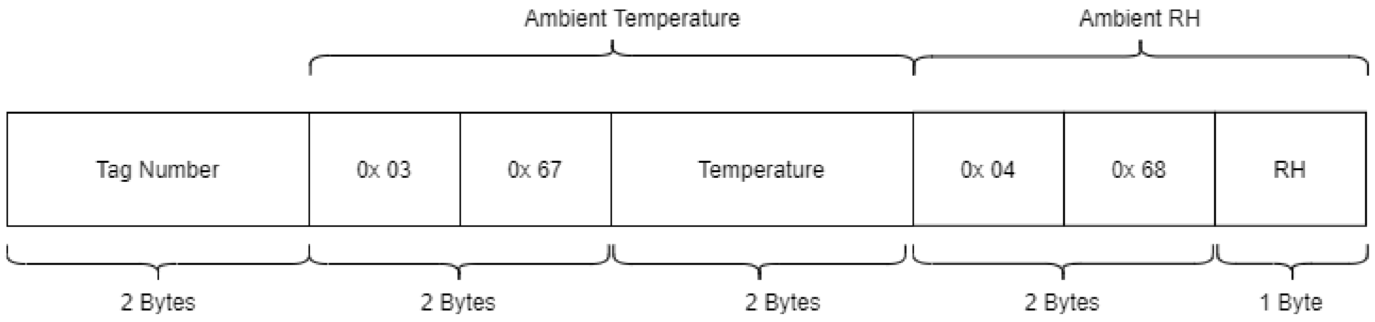


Figure 2-2: The tagged UL frame payload format.

A Sensor message payload can include one or multiple transducer data frames. The ordering of frames is not guaranteed (they can be in any order). A single payload may include data from any given transducer. The Sensor tagged payload frame values are shown in Table 2-2. In this table, the bit indexing scheme is as shown in Figure 2-2.

Table 2-2: UL Frame Payload Values for Tagged Transducer Data

Information Type	Channel ID	Type ID	Size	Data Type	Data Format	JSON Variable (Type/Unit)
Ambient Temperature	0x03	0x67	2 B	Temperature	• 0.1°C / LSB (signed)	*ambient_temperature: <value> (signed/°C)*
Ambient RH	0x04	0x68	1 B	RH	• 0.5% / LSB	*relative_humidity: <value> (unsigned/1%)*

Examples:

In the following example payloads, the data channel ID and data type ID are boldfaced:

- 0x 0A 46 **03 67** 00 0A
    - $0\mathrm{x}\ 0\mathrm{A}\ 46 = \text{Tag entry }\#2630$
    - $0\mathrm{x}\ \mathbf{03\ 67}\ (\text{Ambient Temperature}) = (0\mathrm{x}\ 00\ 0\mathrm{A}) \times 0.1^{\circ}\mathrm{C} = 1^{\circ}\mathrm{C}$
- 0x 00 05 **04 68** 2A **03 67** FF FF
    - $0\mathrm{x}\ 00\ 05 = \text{Tag entry }\#5$
    - $0\mathrm{x}\ \mathbf{04\ 68}\ (\text{Ambient RH}) = (0\mathrm{x}\ 2\mathrm{A}) \times 0.5\% = 21\%$
    - $0\mathrm{x}\ \mathbf{03\ 67}\ (\text{Ambient Temperature}) = (0\mathrm{x}\ \mathrm{FF}\ \mathrm{FF}) \times 0.1^{\circ}\mathrm{C} = -0.1^{\circ}\mathrm{C}$

2.3 Frame Payload to Retransmit Stored Transducer Data

Upon receiving a DL request to retransmit some stored data, the Sensor will retrieve the data from flash and sent on ***LoRaWAN port 33 (0x21)***. The application can easily identify a retransmitted tagged payload from the port number.

Each tagged data field from the Sensor is encoded in a frame format shown in Figure 2-1. A big-endian format (MSB first) is always followed. The tagged data included in the UL can be the ambient temperature, ambient RH, or both. If more than one tag entry is requested, the Sensor will fit as many tagged data in the UL as possible, simply be repeating the tag-header-data frame format as shown in Figure 2-2. If all tagged data cannot fit in a single UL, subsequent ULs containing the remaining requested packets will be sent.

Up to 100 tagged entries can be requested/sent at one time. If more than 100 entries are required, multiple DL requests are necessary. Please see Section 3.4 for information about how to request packet forwarding.

Table 2-3 shows the "error" frame payload values sent when data is requested from a tag entry that has yet to be written to (i.e. the tag entry is empty).

Table 2-3: Invalid UL Frame Payload Values for Empty Tag Entries

Information Type	Channel ID	Type ID	Size	Data Type	Invalid Data
Ambient Temperature	0x03	0x67	2 B	Temperature	• 0x F4 48 • -300°C
Ambient RH	0x04	0x68	1 B	RH	• 0x F0 • 120% RH

2.4 Response to Configuration and Control Commands

Sensor responses to DL configuration and control commands (which are sent on ***LoRaWAN port 100***; see Section 3) are sent in the UL on ***LoRaWAN port 100***. These responses include the following:

- Returning the value of a configuration register in response to an inquiry from the NS.
- Writing to a configuration register.

In the former case, the Sensor responds by the address and value of each of the registers under inquiry (this can be in one or more consecutive UL packets depending on the maximum frame payload size allowed). In the latter case, the Sensor responds with a CRC32 of the entire DL payload (which may be a combination of read and write commands) as the first 4 bytes of the UL frame. If the DL payload has also had read commands, the 4 CRC32 bytes are followed by the address and value of each of the registers under inquiry (similar to the Sensor response in the former case).

3 DL Payload Formats

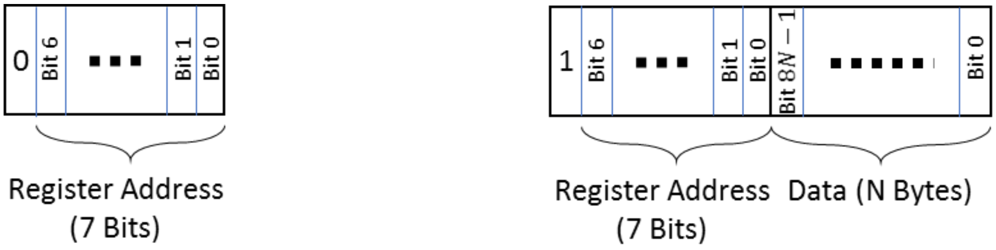
The only DL message (from the NS to the Sensor) supported by the SW includes the following:

- Configuration and control commands used to change the Sensor's behavior or inquire the Sensor for the values of registers (***sent on LoRaWAN port 100***).
- Requests for retrieval of stored data (***sent in DL, LoRaWAN ports 112-122***).

A single DL configuration and control message can contain multiple command blocks, with a possible mix of read and write commands. Each message block is formatted as shown in Figure 3-1. A big-endian format (MSB first) is always followed.

For port 100 commands, the Register Address is used to access various configuration parameters. These addresses are bound between 0x00 and 0x7F. For stored data requests on ports 112-122, please see Section 3.4.

Bit 7 of the first byte determines whether a read or write action is being performed, as shown in Figure 3-1. All read commands are one-byte long. Data following a read access command will be interpreted as a new command block. Read commands are processed last. For example, in a single DL message, if there is a read command from a register and a write command to the same register, the write command is executed first.



(a) The read command. (b) The write command.

Figure 3-1: The format of a DL configuration and control message block.

All DL configuration and control commands are sent on ***LoRaWAN port 100***.

When a write command is sent to the Sensor, the Sensor immediately responds with a CRC32 of the entire DL payload as the first 4 bytes of the UL frame on ***LoRaWAN port 100*** (also see Section 2.4).

DL configuration and control commands fall into one of the following four categories and are discussed in Sections 3.1, 3.2, and 3.3, respectively:

- LoRaMAC Configuration
- Application Configuration
- Command and Control

3.1 LoRaMAC Configuration

LoRaMAC options can be configured using DL commands sent on port 100. These configuration options change the default MAC configuration that the Sensor loads on start-up. They can also change certain run-time parameters. Table 3-1 shows the MAC configuration registers. All the registers have R/W access. In this table, the bit indexing scheme is as shown in Figure 3-1.

Table 3-1: LoRaMAC Configuration Registers

Address	Value	Size	Description	JSON Variable (Type/Unit)
0x10	Join Mode	2 B	• Bit 15: 0/1 = ABP/OTAA mode • Bits 0-14: Ignored	*loramac_join_mode: <value>* *(unsigned/no unit)*
0x11	Options	2 B	• Bit 0: 0/1 = Unconfirmed/Confirmed UL • Bit 1 = 1 (RO): 0/1 = Private/Public Sync Word • Bit 2: 0/1 = Disable/Enable Duty Cycle • Bit 3: 0/1 = Disable/Enable ADR • Bits 4-15: Ignored	*loramac_opts {* *confirm_mode: <value>,* *(unsigned/no unit)* *sync_word: <value>,* *(unsigned/no unit)* *duty_cycle: <value>,* *(unsigned/no unit)* *adr: <value>* *(unsigned/no unit)* *}*
0x12	DR and Tx Power[1]	2 B	• Bits 8-11: Default DR number • Bits 0-3: Default Tx power number • Bits 4-7, 12-15: Ignored	*loramac_dr_tx {* *dr_number: <value>,* *(unsigned/no unit)* *tx_power_number: <value>,* *(unsigned/no unit)* *}*
0x13	Rx2 Window	5 B	• Bits 8-39: Channel frequency in Hz for Rx2 • Bits 0-7: DR for Rx2	*loramac_rx2 {* *frequency: <value>,* *(unsigned/Hz)* *dr_number: <value>* *(unsigned/no unit)* *}*

---

[1] Tx power number m translates to the maximum Tx power, which is a function of the LoRaWAN RF region, minus $2 \times m$ dB.

Note: Modifying these values only changes them in the Sensor. Options for the Sensor in the NS also need to be changed in order to not strand a Sensor. Modifying configuration parameters in the NS is outside the scope of this document.

Examples:

- Switch Sensor to ABP Mode:
  - DL payload: { 0x **90** 00 00 }
- Set ADR enabled, no duty cycle, and confirmed UL payloads:
  - DL payload: { 0x **91** 00 0B }
- Set default DR number to 1 and default Tx power number to 2:
  - DL payload: { 0x **92** 01 02 }

3.1.1 Default Configuration

Table 3-2 and Table 3-4 show the default values for the LoRaMAC configuration registers (cf. [1]).

Table 3-2: Default Values of LoRaMAC Configuration Registers

Address	Default Value
0x10	• OTAA mode
0x11	• Unconfirmed UL • Duty cycle enabled[2] • ADR enabled
0x12	• DR0 • Tx Power 0 (max power; see Table 3-3)
0x13	• As per Table 3-4

Table 3-3: Default Maximum Tx Power in Different Regions

RF Region	Max Tx EIRP [dBm]
EU868	16
US915	30
AS923	16
AU915	30
IN865	30
CN470	19.15
KR920	14
RU864	16

---

[2] In the LoRa RF regions where there is no duty cycle limitation, such as US915, the "enabled duty cycle" configuration of the Sensor is ignored.

DN915	30

Table 3-4: Default Values of Rx2 Channel Frequency and DR Number in Different Regions

RF Region	Channel Frequency [Hz]	DR Number
EU868	869525000	0
US915	923300000	8
AS923	923200000	2
AU915	923300000	8
IN865	866550000	2
CN470	505300000	0
KR920	921900000	0
RU864	869100000	0
DN915	725900000	8

3.2 Application Configuration

This section lists all possible application configurations (as part of DL configuration and control commands), including periodic Tx configuration and configurations of the different transducers.

3.2.1 Periodic Tx Configuration

All periodic transducer reporting is synchronized around *ticks*. A *tick* is simply a user configurable time-base that is used to schedule transducer measurements. For each transducer, the number of elapsed *ticks* before transmitting can be defined as shown in Table 3-5. All the registers in this table have R/W access.

Note: Certain transducer types, such as accelerometer and light, need to be enabled for periodic reporting. Details are available in each transducer's respective section.

Table 3-5: Periodic Transmission Configuration Registers

Address	Value	Size	Description	JSON Variable (Type/Unit)
0x20	Seconds per Core *Tick*	4 B	• *Tick* value for periodic events • Acceptable values: 0, 60, 61, …, 86400 • 0 disables all periodic transmissions • Other values: Invalid and ignored	*seconds_per_core_tick: <value>* *(unsigned/sec)*
0x21	*Ticks* per Battery	2 B	• *Ticks* between battery reports • 0 disables periodic battery reports	*tick_per_battery: <value>* *(unsigned/no unit)*
0x22	*Ticks* per Ambient Temperature	2 B	• *Ticks* between ambient temperature reports • 0 disables periodic ambient temperature reports	*tick_per_ambient_temperature: <value>* *(unsigned/no unit)*

0x23	*Ticks* per Ambient RH	2 B	• *Ticks* between ambient RH reports • 0 disabled periodic ambient RH reports	*tick_per_relative_humidity: <value> (unsigned/no unit)*
0x26	*Ticks* per Accelerometer (both Acceleration and Impact Alarm)	2 B	• *Ticks* between accelerometer reports • 0 disables periodic accelerometer reports	*tick_per_accelerometer: <value> (unsigned/no unit)*
0x27	*Ticks* per MCU Temperature	2 B	• *Ticks* between MCU temperature reports • 0 disables periodic MCU temperature reports	*tick_per_mcu_temperature: <value> (unsigned/no unit)*

3.2.1.1 Seconds per Core Tick

All periodic Tx events are scheduled in *ticks*. This allows for transducer reads to be synchronized, reducing the total number of ULs required to transmit Sensor data. The minimum seconds per *tick* is 60 sec, and the maximum is 86,400 sec (one day). Values from 1 sec to 59 sec and values above 86,400 sec are invalid and ignored. A value of 0 (zero) disables all periodic reporting.

3.2.1.2 Ticks per <Transducer>

This register sets the reporting period for a transducer in terms of *ticks*. Once the configured number of *ticks* has expired, the Sensor polls the specified transducer and reports the data in an UL message. A setting of 0 (zero) disables periodic reporting for the specified transducer.

3.2.1.3 Default Configuration

Table 3-6 shows the default values for the periodic transmission configuration registers.

Table 3-6: Default Values of Periodic Transmission Configuration Registers

Seconds per Core *tick*	3600 (1 hour)
***Ticks* per Battery**	1 (thus 1-hour period)
***Ticks* per Ambient Temperature**	1 (thus 1-hour period)
***Ticks* per Ambient RH**	1 (thus 1-hour period)
***Ticks* per Accelerometer**	0 (periodic Tx disabled)
***Ticks* per MCU Temperature**	0 (periodic Tx disabled)

Examples:

- Disable all periodic events:
  - DL payload: { 0x **A0** 00 00 00 00 }
    - Register 0x20 with the write bit set to true
    - Seconds per *Tick* set to 0 (zero)—i.e. disable periodic transmissions

- Read current value of Seconds per *Tick*:
  - DL payload: { 0x **20** }
    - Register 0x20 with the write bit set to false
- Report Temperature every *tick* and RH every two *ticks*:
  - DL payload: { 0x **A2** 00 01 **A3** 00 02 }
    - Registers 0x22 and 0x23 with their write bits set to true
    - Temperature *Ticks* set to 1 (one)
    - RH *Ticks* set to 2 (two)

3.2.1.4 Anti-Bricking Strategy

Care has been taken to avoid stranding (hard or soft bricking) the Sensor during reconfiguration. Hard bricking refers to the condition that the Sensor does not transmit anymore as all periodic and event-based reporting (see subsequent sections) have been disabled and the configuration has been saved to the Flash memory. Soft bricking refers to the condition where the Sensor has been configured such that all event-based reporting is disabled and any periodic reporting is either disabled or has a period of larger than a week.

To avoid these situations, for any reconfiguration command sent to the Sensor, the following algorithm is executed:

After the reconfiguration is applied, if all event-based reporting (as explained in subsequent sections) is disabled, then periodic reporting is checked. If all periodic reporting is disabled or the minimum non-zero period is greater than a week, then to avoid bricking the Sensor, the core *tick* is set to 86,400 (i.e. one day), and the battery *tick* is set to 1 (one).

3.2.2 Accelerometer Configuration

The accelerometer transducer offers a threshold for an “impact alarm event”[3], and a threshold for an “acceleration event”. It can also be polled periodically for applications where the Sensor orientation may be of interest. Table 3-7 shows a list of accelerometer configuration registers. All registers have R/W access. In this table, the bit indexing scheme is as shown in Figure 3-1.

Some terminology in this section is as follows:

- Accelerometer (transducer) refers to the accelerometer transducer component.
- Impact alarm (event) refers to an accelerometer event based on exceeding an impact alarm event threshold. Impact alarm events are reported with an impact alarm.
- Acceleration (event) refers to an accelerometer event, independent of the impact alarm event, and based on exceeding an acceleration event threshold. Acceleration events are reported with the acceleration magnitude, acceleration vector, or both.

---

[3] Here “impact” generally refers to a Sensor motion event (i.e. not necessarily an *impact* to the Sensor).

Table 3-7: Accelerometer Configuration Registers

Address	Name	Size	Description	JSON Variable (Type/Unit)
0x30	Impact Alarm Event Threshold	2 B	• Unsigned, 1 milli-g/LSB	*impact_event_threshold* *(unsigned/g)*
0x31	Acceleration Event Threshold	2 B	• Unsigned, 1 milli-g/LSB	*acceleration_event_threshold:* *<value>* *(unsigned/g)*
0x32	Value to Tx	1 B	• Bit 0 (applicable to accelerometer periodic reporting only[4]): 0/1 = Impact alarm not reported/reported • Bit 4 (applicable to both accelerometer periodic reporting and acceleration event reporting): 0/1 = Acceleration magnitude not reported/reported • Bit 5 (applicable to both accelerometer periodic reporting and acceleration event reporting): 0/1 = Acceleration vector not reported/reported • Bits 1-3, 6, 7: Ignored	*accelerometer_tx {* *report_alarm_enabled: <value>,* *(unsigned/no unit)* *report_magnitude_enabled:* *<value>,* *(unsigned/no unit)* *report_vector_enabled: <value>* *(unsigned/no unit)* *}*
0x33	Acceleration Event Debounce Time	2 B	• Seconds to wait before possibly reporting an acceleration event again • Acceptable values: 1, 2, …, 65535 • 0: Invalid and ignored	*acceleration_impact_grace_period:* *<value>* *(unsigned/seconds)*
0x34	Mode	1 B	• Bit 0: 0/1 = Impact alarm event threshold disabled/enabled • Bit 1: 0/1 = Acceleration event threshold disabled/enabled • Bits 2, 3: Ignored • Bit 4/5/6:	*accelerometer {* *impact_threshold_enabled:* *<value>,* *(unsigned/no unit)* *acceleration_threshold_enabled:* *<value>,* *(unsigned/no unit)*

---

[4] This bit only controls whether the impact alarm status (i.e. raised or cleared) will be present in periodic reporting when such accelerometer periodic reporting is enabled (see Section 3.2.1). This bit does not control reporting of the impact alarm status for impact alarm events. If the impact alarm event threshold is enabled (register 0x34, bit 0), an impact alarm is always raised (reported) when the impact alarm event threshold (register 0x30) is exceeded, and is cleared after an impact alarm event grace period (register 0x36) elapses without any impact alarm events (see Section 3.2.2.7).

			0/1 = X/Y/Z-axis disabled/enabled • Bit 7: 0/1 = Accelerometer power off/on	*xaxis_enabled: <value>,* *(unsigned/no unit)* *yaxis_enabled: <value>,* *(unsigned/no unit)* *zaxis_enabled: <value>,* *(unsigned/no unit)* *poweron: <value>* *(unsigned/no unit)* *}*
0x35	Sensitivity	1 B	• Bits 0-2 (Sample Rate): 0: Invalid and ignored 1/2/3/4/5/6/7 = 1/10/25/50/100/200/400 Hz • Bits 4-5 (Measurement Range[5]): 0/1/2/3 = ±2 *g*/±4 *g*/±8 *g*/±16 *g* • Bits 3, 6, 7: Ignored	*sensitivity {* *accelerometer_sample_rate: <value>,* *(unsigned/Hz)* *accelerometer_measurement_range: <value>,* *(unsigned/g)* *}*
0x36	Impact Alarm Event Grace Period	2 B	• Impact alarm grace period in sec (time to pass after the last impact alarm before the alarm can be cleared) • Acceptable values: 15, 16, …, 65535 • Other values: Invalid and ignored	*impact_alarm_grace_period: <value>* *(unsigned/seconds)*
0x37	Impact Alarm Event Threshold Count	2 B	• Number of impact alarm events before an impact alarm is raised • Acceptable values: 1, 2, …, 65535 • 0: Invalid and ignored	*impact_alarm_threshold_count: <value>* *(unsigned/no unit)*
0x38	Impact Alarm Event Threshold Period	2 B	• Period in sec over which impact alarm events are counted for threshold detection • Acceptable values: 5, 6, …, 65535 • Other values: Invalid and ignored	*impact_alarm_threshold_period: <value>* *(unsigned/seconds)*

---

[5] Measurement ranges ±2 *g*, ±4 *g*, ±8 *g*, ±16 *g* correspond to typical transducer output precisions of 16 m*g*, 32 m*g*, 64 m*g*, 192 m*g*, respectively. Note that if a threshold configured in register 0x30 or register 0x31 is equal to or greater than the configured measurement full scale (2 *g*, 4 *g*, 8 *g*, 16 *g*), then the corresponding event (impact alarm or acceleration event) will never be triggered.

3.2.2.1 Impact Alarm Event Threshold

This parameter is the g-threshold for an impact alarm event. Impact alarm events are reported only if,

- the impact alarm event threshold (bit 0 of register 0x34) is enabled; and
- the impact alarm event threshold is exceeded on at least one of the enabled axes (X, Y, Z) within a period (Impact Alarm Event Threshold Period—register 0x38) for more than a number of times (Impact Alarm Event Threshold Count—register 0x37).

3.2.2.2 Acceleration Event Threshold

This parameter is the g-threshold for an acceleration event. Provided that the acceleration threshold is enabled (bit 1 of register 0x34), acceleration events are reported as soon as the Acceleration Event Threshold is exceeded on at least one of the enabled axes (X, Y, Z). However, acceleration event interrupts are totally ignored (not registered) for a time period equal to the Acceleration Event Debounce Time (register 0x33) after a registered (and thus reported) acceleration event.

3.2.2.3 Value to Tx

Determines what is reported (transmitted) in the case of an acceleration event or accelerometer periodic transmission. The parameters to report include the status of the impact alarm (alarm on/off), the acceleration magnitude $\|\langle x, y, z\rangle\| = \sqrt{x^2 + y^2 + z^2}$, and the acceleration vector $\langle x, y, z\rangle$.

3.2.2.4 Acceleration Event Debounce Time

Interrupts due to acceleration events are disabled for a configurable time frame, called the Acceleration Event Debounce Time, after an acceleration event is registered. This is done to prevent a single acceleration event from being transmitted as multiple events. The minimum debounce time is 1 (one) sec. Value 0 is invalid and ignored.

3.2.2.5 Mode

When not being used in an end-user application, the accelerometer transducer can be put in the power-down mode to save battery life. Otherwise, the accelerometer is put in the low-power mode, which is an active and operational, but a low consumption, mode for the accelerometer.

Additionally, impact alarm and acceleration event thresholds can be enabled/disabled. Disabling a threshold prevents the Sensor from generating the corresponding event. It is also possible to enable/disable X, Y, Z axes independently. When an axis is disabled, it is not considered in monitoring impact alarm or acceleration events.

3.2.2.6 Sensitivity

When powered on, the accelerometer always samples the transducer element at a fixed rate, called the Sample Rate. To capture an impact alarm or acceleration event, the physical event needs to last longer than the sample period. Larger sample rates have a shorter period and can therefore resolve shorter impacts. However, sampling the transducer at a larger rate increases the power usage, impacting the battery life. Table 3-8 shows how much continuous current draw is expectable to be drawn from a 3.2-V battery for the different sample rates when the accelerometer is powered on. For example, the sample rate of 1 Hz would translate to about 15 mAh/year battery usage, while a sample rate of 50 Hz would triple that usage.

Table 3-8: Typical Current Draws at 3.2 V for Different Accelerometer Sample Rates

Sample Rate [Hz]	1	10	25	50	100	200	400
Current Draw [μA]	1.6	2.3	3.1	4.7	7.8	14.1	28.1

Furthermore, the Sensitivity register sets the measurement range or full scale, which shows the dynamic range of accelerations that can be monitored on any enabled axis. Note that when active, the accelerometer is always put in its low power mode, which means the output acceleration values on any given axis (X, Y, or Z), is an 8-bit signed number. Therefore, a measurement range of ±2 g implies a precision of 4/256 g/LSB.

3.2.2.7 Impact Alarm Event Grace Period

The Grace Period determines how long the Room Sensor waits before the previously reported impact alarm event is considered clear. For example, a Grace Period of 5 (five) min results in the sensor transmitting “Impact Detected” when there is movement, and “Impact Alarm Cleared” 5 (five) min after the Sensor has been still.

The minimum acceptable value for this register is 15. Values smaller than 15 are invalid and ignored.

3.2.2.8 Impact Alarm Event Threshold Count

The accelerometer generates an impact alarm event each time it detects movement. Depending on the customer use case, it may be desirable to increase the threshold count to reduce sensitivity. This feature is to allow customers to filter out short impact events, while still allowing longer impact events to be reported.

The minimum acceptable value for this register is 1. Value 0 is invalid and ignored.

3.2.2.9 Impact Alarm Event Threshold Period

The Impact Alarm Event Threshold Period is the amount of time that impact alarm events are accumulated for threshold detection. For example, an Impact Alarm Event Threshold Period of 10 (ten) sec accumulates impact alarm events over a 10 (ten)-sec period from the time of first detection. If the Impact Alarm Event Threshold Count is reached before the time expires, the sensor reports “Impact Detected”, otherwise it does not report.

The minimum acceptable value for this register is 5. Values smaller than 5 are invalid and ignored.

3.2.2.10 Default Configuration

Table 3-9 shows the default values for the accelerometer configuration registers.

Table 3-9: Default Values of Accelerometer Configuration Registers

Impact Alarm Event Threshold	1500 milli-g

Acceleration Event Threshold	3000 milli-g
Value to Tx	Acceleration vector
Acceleration Event Debounce Time	2 sec
Mode	• Impact alarm threshold disabled • Acceleration threshold disabled • X-axis, Y-axis, and Z-axis enabled • Accelerometer power off
Sensitivity	• Sample rate 1 Hz • Measurement range ±8 g
Impact Alarm Event Grace Period	300 sec (5 min)
Impact Alarm Event Threshold Count	1
Impact Alarm Event Threshold Period	15 sec

3.2.3 Temperature/RH Threshold Configuration

The Sensor supports threshold transmission on four different transducer values:

- Ambient temperature: Measured by the Temperature/RH transducer
- Ambient RH: Measured by the Temperature/RH transducer
- MCU Temperature: Measured by the MCU (with lower accuracy compared to the Ambient Temperature)

When a threshold on a transducer is enabled, the Sensor reports the transducer value when it leaves the configured threshold window[6]. The Threshold mode is compatible with periodic reporting. Table 3-10 shows a list of configuration registers for the temperature/RH/Analog Input threshold setting. All the registers have R/W access. In this table, the bit indexing scheme is as shown in Figure 3-1.

Table 3-10: Temperature/RH Threshold Configuration Registers

Address	Name	Size	Description	JSON Variable (Type/Unit)
0x39	Ambient Temperature/RH Sample Period: Idle	4 B	• Sample period of Ambient Temperature/RH transducer: Idle state (sec) • Acceptable values: 30, 31, …, 86400 • Other values: Invalid and ignored	*temperature_relative_humidity_sample_period_idle: <value> (unsigned/sec)*
0x3A	Ambient Temperature/RH	4 B	• Sample period of Ambient Temperature/RH transducer: Active state (sec)	*temperature_relative_humidity_sample_period_active: <value> (unsigned/sec)*

---

[6] Note that the threshold window here is defined as the open interval "(Low Threshold, High Threshold)", not e.g. the closed interval "[Low Threshold, High Threshold]"; i.e. even if the transducer value is equal to Low Threshold or High Threshold, the Sensor is considered to have left the threshold window.

	Sample Period: Active		- Acceptable values: 30, 31, …, 86400 - Other values: Invalid and ignored	
0x3B	Low/High Ambient Temperature Thresholds	2 B	- Bits 8-15: High temperature threshold (signed, 1°C / LSB) - Bits 0-7: Low temperature threshold (signed, 1°C / LSB) - High threshold ≤ Low threshold: Invalid and ignored	*ambient_temperature_threshold {* *high: <value>* *(signed/°C)* *low: <value>* *(signed/°C)* *}*
0x3C	Ambient Temperature Thresholds Enabled	1 B	- Bit 0: 0/1 = Thresholds disabled/enabled - Bits 1-7: Ignored	*ambient_temperature_threshold_enabled: <value>* *(unsigned/no unit)*
0x3D	Low/High Ambient RH Thresholds	2 B	- Bits 8-15: High RH threshold (unsigned, 1% RH / LSB) - Bits 0-7: Low RH threshold (unsigned, 1% RH / LSB) - High threshold ≤ Low threshold: Invalid and ignored	*relative_humidity_threshold {* *high: <value>,* *(unsigned/%)* *low: <value>* *(unsigned/%)* *}*
0x3E	Ambient RH Thresholds Enabled	1 B	- Bit 0: 0/1 = Thresholds disabled/enabled - Bits 1-7: Ignored	*relative_humidity_threshold_enabled: <value>* *(unsigned/no unit)*
0x40	MCU Temperature Sample Period: Idle	4 B	- Sample period of MCU temperature transducer: Idle state (sec) - Acceptable values: 30, 31, …, 86400 - Other values: Invalid and ignored	*mcu_temperature_sample_period_idle: <value>* *(unsigned/sec)*
0x41	MCU Temperature Sample Period: Active	4 B	- Sample period of MCU temperature transducer: Active state (sec) - Acceptable values: 30, 31, …, 86400 - Other values: Invalid and ignored	*mcu_temperature_sample_period_active: <value>* *(unsigned/sec)*
0x42	Low/High MCU Temperature Thresholds	2 B	- Bits 8-15: High MCU temperature threshold (signed, 1°C / LSB) - Bits 0-7: Low MCU temperature threshold (signed, 1°C / LSB) - High threshold ≤ Low threshold: Invalid and ignored	*mcu_temperature_threshold {* *high: <value>,* *(signed/°C)* *low: <value>* *(signed/°C)* *}*
0x43	MCU Temperature Thresholds Enabled	1 B	- Bit 0: 0/1 = Thresholds disabled/enabled - Bits 1-7: Ignored	*mcu_temperature_threshold_enabled: <value>* *(unsigned/no unit)*

3.2.3.1 Temperature/RH Sample Period: Idle

The idle sample period determines how often the transducer is checked when the reported value is within the threshold window. When first enabled, the transducer starts in the Idle state.

The minimum Sample Period in the Idle state is 30 sec, and the maximum is 86,400 sec (one day). Values smaller than 30 for this register are invalid and ignored.

3.2.3.2 Temperature/RH Sample Period: Active

The active sample period determines how often the transducer is checked when the reported value is outside the threshold window.

The minimum Sample Period in the Active state is 30 sec, and the maximum is 86,400 sec (one day). Values smaller than 30 for this register are invalid and ignored.

3.2.3.3 Temperature/RH Thresholds

The thresholds are stored in a single 2-byte register, with the MSB storing the upper threshold, and the LSB storing the lower threshold. Ambient or MCU Temperature thresholds have a precision of 1°C per bit, and are stored/transmitted as 2-s complement numbers. The RH thresholds have a precision of 1% per bit, and are stored/transmitted as unsigned numbers. The Analog Input thresholds are also unsigned numbers, and have a precision of 1 mV per bit.

In all cases, the upper threshold must be greater than the lower threshold. Otherwise, the configuration is considered invalid and ignored.

3.2.3.4 Temperature/RH Thresholds Enabled

The Thresholds Enabled registers enable and disable the threshold reporting on the specified transducer. Thresholds and Sample Periods can be configured but are not activated unless the Thresholds Enabled bit is set.

3.2.3.5 Default Configuration

Table 3-11 shows the default values for the threshold configuration registers.

Table 3-11: Default Values of Threshold Configuration Registers

Ambient Temperature/RH Sample Period: Idle	• 60 sec
Ambient Temperature/RH Sample Period: Active	• 30 sec
Ambient Temperature Threshold: High	• 30°C
Ambient Temperature Threshold: Low	• 15°C
Ambient Temperature Thresholds Enabled	• Disabled
Ambient RH Threshold: High	• 80%
Ambient RH Threshold: Low	• 15%
Ambient RH Thresholds Enabled	• Disabled

MCU Temperature Sample Period: Idle	• 300 sec
MCU Temperature Sample Period: Active	• 60 sec
MCU Temperature Threshold: High	• 30°C
MCU Temperature Threshold: Low	• 15°C
MCU Temperature Thresholds Enabled	• Disabled

Examples:

- Set Ambient Temperature Thresholds:
  - DL payload: { 0x **BB** 19 0A }
    - Register 0x3B with write bit set to true
    - High threshold set to 25°C
    - Low threshold set to 10°C
- Read Ambient Temperature/RH Sample Periods:
  - DL payload: { 0x **39 3A** }
    - Registers 0x39 and 0x3A with their write bits set to false
- Set and enable Ambient RH thresholds:
  - DL payload: { 0x **BD** 3C 14 **BE** 01 }
    - Registers 0x3D and 0x3E with their write bits set to true
    - High RH thresholds set to 60% RH
    - Low RH threshold set to 20% RH
    - RH thresholds enabled

3.3 Command and Control

Configuration changes are not retained after a power cycle unless they are saved in the Flash memory. Table 3-12 shows the structure of the Command & Control Register. In this table, the bit indexing scheme is as shown in Figure 3-1.

Table 3-12: Sensor Command & Control Register

Address	Access	Name	Size	Description	JSON Variable (Type/Unit)
0x70	W	Flash Write Command	2 B	• Bit 14: • 0/1 = Do not write/Write LoRaMAC Config • Bit 13: • 0/1 = Do not write/Write App Config • Bit 0: • 0/1 = Do not restart/Restart Sensor	*write_to_flash {* *app_configuration: <value>, (unsigned/no unit)* *lora_configuration: <value>, (unsigned/no unit)* *restart_sensor: <value> (unsigned/no unit)* *}*

Address	Access	Name	Size	Description	JSON Variable (Type/Unit)
				• Bits 1-12, 15: Ignored	
0x71	R	FW Version	7 B	• Bits 48-55: App version major • Bits 40-47: App version minor • Bits 32-39: App version revision • Bits 24-31: LoRaMAC version major • Bits 16-23: LoRaMAC version minor • Bits 8-15: LoRaMAC version revision • Bits 0-7: LoRaMAC region number (see Section 3.3.1)	*firmware_version {* *app_major_version: <value>, (unsigned/no unit)* *app_minor_version: <value>, (unsigned/no unit)* *app_revision: <value>, (unsigned/no unit)* *loramac_major_version: <value>, (unsigned/no unit)* *loramac_minor_version: <value>, (unsigned/no unit)* *loramac_revision: <value>, (unsigned/no unit)* *region: <value> (unsigned/no unit)* *}*
0x72	W	Reset Config Registers to Factory Defaults[7]	1 B	• 0x0A = Reset App Config • 0xB0 = Reset LoRa Config • 0xBA = Reset both App and LoRa Configs • Any other value: Invalid and ignored	*configuration_factory_reset: <value> (unsigned/no unit)*

Note: The Command & Control Register is always executed after the full DL configuration message has been decoded. The reset command should always be sent as an "unconfirmed" DL message. Failure to do so may cause a poorly designed NS to continually reboot the Sensor.

---

[7] After sending the reset-to-factory-defaults command, the Sensor is automatically reset with corresponding default configuration values.

3.3.1 LoRaMAC Region

The LoRaMAC region is indicated by B_6 in the FW Version register (register 0x71). Current LoRaMAC regions and corresponding region numbers are listed in Table 3-13.

Table 3-13: LoRaMAC Regions and Region Numbers

LoRaMAC Region	Region Number
EU868	0
US915	1
AS923	2
AU915	3
IN865	4
CN470	5
KR920	6
RU864	7
DN915	8

Examples:

- Write Application Configuration to Flash memory
    - DL payload: { 0x **F0** 20 00 }
- Write Application and LoRa Configurations to Flash memory
    - DL payload: { 0x **F0** 60 00 }
- Reboot Device
    - DL payload: { 0x **F0** 00 01 }
- Get FW version, and reset App Config to factory defaults
    - DL payload: { 0x **71 F2** 0A}

3.4 Requests to Forward Stored Data

To retrieve and retransmit temperature or RH data that has been stored in the Sensor’s memory, a DL request must be sent from the user or application. This DL request tells the sensor which tag entries to retransmit. The format and port of the request DL depends on whether one tagged entry is required or many tag entries are required.

3.4.1 Requesting a Single Tagged Entry

To request a single stored data entry, the 2-Byte tag number is sent on ***LoRaWAN port 112 (0x70)***.

For example, to request tag entry #1725, **0x 06 BD** is sent on port 112. The Sensor will respond with the data stored under that tag entry in the format described in Section 2.3.

3.4.2 Requesting Multiple Tagged Entries

To request more than a single stored data entry, a *Run-Length-Encoding* (RLE) compression scheme is used. Up to 100 tag entries can be requested at once using this scheme. The following are definitions of the variables used in this section:

- $w = \text{word size}$: The number of bits in a repeating RLE block.
- $m = \text{1st tag \#}$: The first missing tag entry that is requested.
- $b_n = 1/0 =$ MSb of the n^{th} word: defines whether the word denotes a run of missing/requested tags (1) or non-missing/non-requested tags (0).
- $L_n = \text{run length}$: The $w - 1$ LSbs in each word that define the run length of missing or non-missing tags (as specified by the value of b_n).
- $q = \text{number of words}$.
- $p = \text{number of placeholder 0s at the end of the payload}$.

LoRaWAN ports 112 (0x70) to 122 (0x7A) are used to request data forwarding, where the different ports indicate the word size in bits, w, of the scheme:

Port 112 (0x70): RLE with $\boldsymbol{w} = \mathbf{0}$ (sending only a 2-byte tag header)

Port 112 (0x70): RLE with $\boldsymbol{w} = \mathbf{1}$ (equivalent to bitmap encoding)

Port 113 (0x71): RLE with $\boldsymbol{w} = \mathbf{3}$

.

.

.

Port 122 (0x7A): RLE with $\boldsymbol{w} = \mathbf{12}$

The cases of $w = 0$ and $w = 1$ use the same port and are distinguished at the sensor simply by the number of bytes in the payload. The case of $w = 2$ has been eliminated as it is always inferior to the case of $w = 1$ in terms of compression.

The general RLE scheme payload has been illustrated in Figure 3-2. The Sensor is able to understand and decode the RLE requests to derive the requested tag numbers. If the payload has only 2 bytes ($w = 0$), it is decoded as pointing to only one missing tag, Tag m. However, for payloads longer than 2 bytes, the **$\boldsymbol{w}$-bit RLE words** successively determine run lengths of 0's and 1's after Tag m, where a 1 indicates a missing/requested tag.

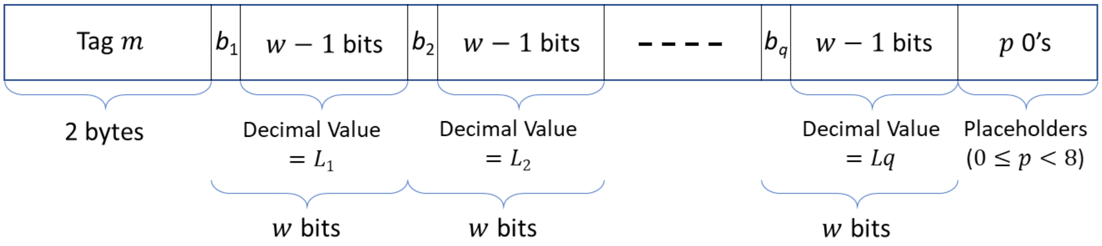


Figure 3-2: The RLE packet scheme.

The MSb in each word is the first bit of a run. The $w-1$ LSbs define the length of the run after the MSb (from 0 to $2^{w-1}-1$). For example, with a word length of 4 bits ($w=4$), a run of 1111111 is encoded as 1110, a run of only one 1 is encoded as 1000, a run of only one 0 is encoded as 0000, etc. Therefore, the payload in Figure 3-2 is decoded as follows:

Tag m: labeled 1 (i.e. missing)

Tags $m+1$ to $m+1+L_1$: labeled b_1 (if $b_1=0$, none is missing, if $b_1=1$, all are missing)

Tags $m+2+L_1$ to $m+2+L_1+L_2$: labeled b_2

.

.

.

Tags $m+q+L_1+\cdots+L_{q-1}$ to $m+q+L_1+\cdots+L_{q-1}+L_q$: labeled b_q

Note that in Figure 3-2, p is such that $q\,w+p$ is an integer multiple of 8 bits, so the length of the whole payload becomes $2+\frac{q\,w+p}{8}$ bytes.

Also, note that the RLE scheme with $w=1$ (and thus, $L_1=L_2=\cdots=L_q=p=0$) degrades to a bitmap encoding scheme where successive tags after Tag m are simply encoded as 0 or 1 depending on being received or lost:

Tag m: labeled 1

Tag $m+1$: labeled b_1

Tag $m+2$: labeled b_2

.

.

.

Tag $m+q$: labeled b_q

References

[1] LoRa Alliance, "LoRaWAN Specification," ver. 1.0.2, Jul 2016.

[2] LoRa Alliance, "LoRaWAN Regional Parameters," ver 1.0.2, Rev B, Feb 2017.

[3] TEKTELIC Communications Inc., "Industrial GPS Asset Tracker User Guide," Dec 2019.

[4] TEKTELIC Communications Inc., "Smart Room Sensor Uplink and Downlink Frame Payloads," ver 0.1, Aug 2019.

[5] TEKTELIC Communications Inc., "Smart Room Sensor User Guide," ver 1.5, Jan 2020.