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

LoRa IoT Industrial Transceiver

Technical Reference Manual

Document Type:	Technical Reference Manual
Document Number:	T0005485_TRM
Document Version:	2.1
Product Name:	LoRa IoT Industrial Transceiver
Product Codes & Revision:	T0005322 (EU Module) Rev E
	T0005500 (NA/DN Module) Rev E
	T0005633 (CN Module) Rev C
Release Date:	December 2, 2021

TEKTELIC Communications Inc.
7657 10th Street NE
Calgary, AB, Canada T2E 8X2
Phone: (403) 338-6900

Revision History

Version	Date	Editor	Comments
1.8	January 4, 2021	Mark Oevering	• Edit to Section 2
1.9	February 1, 2021	Mark Oevering	• Bold note added to Section 4.3.3.5.7 • Change to description at the beginning of Section 4.3.3.5 • Change to naming of registers 6A and 6B in Section 4.3.3.5
2.0	April 22, 2021	Mark Oevering	• Removed register 62, and added note in section 4.3.3.5 • Changed size of register 68 and 69 in section 4.3.3.8 to 2 bytes each • Added details on larger serial payloads in Section 4.3.3.6 • Added details on continuous serial receive mode in Section 4.3.3.7
2.1	December 2, 2021	Mark Oevering	• Corrected example in section 4.3.3.8.5

Table of Contents

List of Tables

List of Figures

Acronyms and Glossary

ABP...... .... Activation By Personalization
ADR .... Adaptive Data Rate
bps....... .... bits per second
CN........ .... China as an RF region for LoRaWAN
CRC...... .... Cyclic Redundancy Check
DL .... Downlink
DN .... a special LoRaWAN RF region considered for Dish Network Corporation
DR........ .... Data Rate
EIRP .... Effective Isotropic Radiated Power
EU........ .... European Union as an RF region for LoRaWAN
Flash memory .. .. Non-volatile memory containing application and configuration settings
FW .... Firmware
Industrial Transceiver... .... a LoRa IoT Industrial Transceiver module
ID......... .... Identity / Identifier
IoT .... Internet of Things
ISM .... Industrial, Scientific, and Medical
LoRa .... a patented "Long-Range" IoT technology acquired by Semtech
LoRaMAC .... LoRaWAN MAC
LoRaWAN......... .... LoRa wide area network (a network protocol based on LoRa)
LoRaWAN Commissioning**.........** the unique device identifiers and encryption keys used for LoRaWAN communication (see LoRaWAN Specification [1] for more details)
LSB....... .... Least Significant Bit
LTC....... .... Lithium Thionyl Chloride (the chemistry of LTC batteries)
MAC..... .... Medium Access Control
MCU .... Microcontroller Unit
ms........ .... millisecond(s)
MSB..... .... Most Significant Bit
NA .... North America as an RF region for LoRaWAN
NS........ .... Network Server
OTA .... Over-The-Air
OTAA... .... OTA Activation
Reg .... Register
RF .... Radio Frequency
RFU...... .... Reserved for Future Use
RO........ .... Read-Only
RTU...... .... Remote Terminal Unit
R/W..... .... Read/Write
Rx .... Receiver
SW....... .... Software
Transceiver....... .... a LoRa IoT Industrial Transceiver module

Transducer .... a sensing element attached to the Industrial Transceiver (e.g. the temperature transducer)

TRM..... .... Technical Reference Manual (this document)

Tx......... .... Transmitter

UL........ .... Uplink

1 Overview

This TRM describes the configuration options supported by the Industrial Transceiver. This document is intended for a technical audience, such as application developers, with an understanding of the NS and its command interfaces.

This TRM is only applicable to the Industrial Transceiver modules listed in Table 1-1 (all of which have ***PCBA Rev C*** and use ***FW version 1.x***).

The Industrial Transceiver is an all-purpose LoRaWAN IoT sensor powered by an LTC battery and built into a small IP67 rated casing. The Industrial Transceiver features two analog inputs, one digital input, two digital outputs, and a serial port that supports Modbus RTU over RS-232, RS-422, or RS-485 in half-duplex or full-duplex mode. The Industrial Transceiver is also equipped with an on-board temperature transducer. Also, the MCU on the board can measure and provide the MCU temperature and battery voltage. The battery lifetime of the Industrial Transceiver is estimated to be 25 years.[1] Table 1-1 presents the currently available Industrial Transceiver models.

Table 1-1: Industrial Transceiver Models

Product Code & Revision	Description	RF Region
T0005322 Rev E	Industrial Transceiver Module, EU	EU 863-870 MHz (ISM band)
T0005500 Rev E	Industrial Transceiver Module, NA/DN	NA: 902-928 MHz (ISM band) DN: 902-915 MHz UL, 722-728 MHz DL
T0005633 Rev C	Industrial Transceiver Module, CN	CN 470-510 MHz

Information streams currently supported by the SW are as follows:

- UL stream (i.e. data from the Transceiver)
  - Readings obtained from on-board transducers; ***sent on LoRaWAN port 10***
  - Data obtained from the Transceiver's serial port; ***sent on LoRaWAN port 20***
  - Data obtained from a connected Modbus RTU device, when periodically polled through registers 0x6A, 6B, 6C, 6D, 6E, 6F, ***sent on LoRaWAN ports 21, 22, 23, 24, 25 and 26 respectively***
  - Data obtained from the Transceiver's serial port, ***if extended serial format enabled; sent on LoRaWAN port 40***
  - Response to configuration and control commands from the NS; ***sent on LoRaWAN port 100***
- DL stream (i.e. data from the NS)
  - Data intended for the Transceiver's serial port; ***sent on LoRaWAN port 20***
  - Changing the state of the Transceiver's (digital) outputs, i.e. open/close them; ***sent on LoRaWAN port 10***
  - Configuration and control commands used to change the Transceiver's behavior; ***sent on LoRaWAN port 100***

[1] This is for transmission at maximum power every 15 minutes at room temperature, with an LTC battery having a nominal capacity of 19 Ah and self-discharge rate of 0.7%. Significant variations to this estimate can occur depending on the ambient temperature, amount of usage, battery capacity, and battery self-discharge rate. For example, continuously being at -30°C and transmitting at maximum power every 30 seconds, the same battery may not last above a year.

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

- Report the battery voltage every hour.
- Report ambient temperature every hour.
- Report actuation of Input 1 (digital) every 1 (one) actuation.
- Report Input 1 (digital) reading every 15 (fifteen) minutes.
- Report Input 2 (current) readings every 15 (fifteen) minutes.
- Report Input 3 (voltage) readings every 15 (fifteen) minutes.

The default configuration for the Industrial Transceiver serial port is as follows:

- Protocol RS232
- 115,200 bps baud rate, 8 data bits, no parity bits, 1 stop bit

In the following sections, the UL (departing from the Transceiver) and DL (destined to the Transceiver) payload formats are explained. Refer to the *Sensor Configuration Tool* [2] for an online application to decode any UL frame payload, as well as encode any DL frame payload by varying parameter values, toggling read/write actions, and enabling/disabling different fields as desired.

2 Serial Port Operation

As mentioned above, the Industrial Transceiver has a serial transceiver that supports connection to a Modbus RTU device over RS-232, RS-422, or RS-485 in half- or full-duplex mode. The Industrial Transceiver is fully transparent from the Application Server to the connected Modbus device, in the sense that when the Application Server needs to send a message to the connected device, it sends it on LoRaWAN port 20, and the Industrial Transceiver forwards the whole message payload to the connected device as soon as it receives the message (see Section 4.1).

The default configuration for the Industrial Transceiver serial port is as follows:

- Protocol RS232
- 115,200bps baud rate, 8 data bits, no parity bits, 1 stop bit

IMPORTANT NOTES:

- **Maximum serial payload size is 512. However, please note that the extended serial uplink header might need to be enabled at lower data rates to avoid losing framing information. See Section 4.3.3.6 for more information.**
- **The serial port on Industrial Transceiver can be set to continuously receive, which will enable the transceiver to receive and uplink data from the serial port without any application or network server intervention. See Section 4.3.3.7 for more information.**

3 UL Payload Formats

The UL streams (from the Transceiver to the NS) supported by the SW include the following:

- Readings obtained from on-board transducers; sent on ***LoRaWAN port 10***
- Data received form the Transceiver's serial port; sent on ***LoRaWAN port 20***
- Data obtained from a connected Modbus RTU device, when periodically polled through registers 0x6A, 6B, 6C, 6D, 6E, 6F; ***sent on LoRaWAN ports 21, 22, 23, 24, 25 and 26 respectively***
- Data obtained from the Transceiver's serial port, if extended serial format enabled; ***sent on LoRaWAN port 40***
- Responses to configuration and control commands from the NS; sent on ***LoRaWAN port 100***

These topics are explained in Sections 3.1, 3.2, and 3.3, respectively. Refer to [2] for a comprehensive application to decode Industrial Transceiver UL frame payloads.

3.1 Frame Payload to Report Transducers Data

Each data field from the Transceiver is encoded in a frame format shown in Figure 3-1. A big-endian format (MSB first) is always followed.

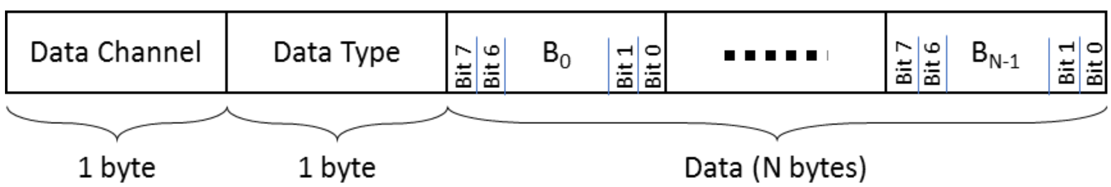


Figure 3-1: The UL frame payload format.

A Transceiver message payload can include multiple transducer data frames. Frames can be arranged in any order. A single payload may include data from any given transducer. The Industrial Transceiver frame payload values for transducers data are shown in Table 3-1. In this table, B_i refers to data byte indexed *i* as shown in Figure 3-1. Transducers data in the UL are sent through ***LoRaWAN port 10***.

Table 3-1: UL Frame Payload Values for Transducers Data

Information Type	Data Channel ID	Data Type ID	# Bytes	Data Type	Data Format	JSON Variable (Type/Unit)
Battery Voltage	0x00	0xFF	2	Analog input: Signed	• 0.01 V / LSB	*battery_voltage: <value> (signed/volt)*
Output 1	0x01	0x01	1	Digital output: Boolean	• 0x00 = Open • 0xFF = Closed	*output_1: <value> (unsigned/boolean)*
Output 2	0x02	0x01	1	Digital Output: Boolean	• 0x00 = Open • 0xFF = Closed	*output_2: <value> (unsigned/boolean)*
Temperature	0x03	0x67	2	Temperature: Signed	• 0.1°C / LSB	*temperature: <value> (signed/celsius)*
Input 1 State	0x05	0x00	1	Digital Input: Boolean	• 0x00 = false	*input_1: <value> (unsigned/boolean)*

					• 0x01 = true[2]	
Input 2	0x06	0x02	2	Analog Input: Unsigned	• 1 μA / LSB	*input_2: <value> (unsigned/ampere)*
Input 3	0x07	0x02	2	Analog Input: Unsigned	• 1 mV / LSB	*input_3: <value> (unsigned/volt)*
Input 1 Count	0x08	0x04	2	Counter Input: Unsigned	• 1 count / LSB	*input_1_count: <value> (unsigned/no unit)*
MCU Temperature	0x09	0x67	2	Temperature: Signed	• 0.1°C / LSB	*mcu_temperature: <value> (signed/celsius)*

3.1.1 Example Uplink Payloads

- 0x 03 67 00 0A
  - Temperature = 1°C
- 0x 05 00 01 08 04 00 05
  - Input 1 = Closed
  - Input 1 Count = 5
- 0x 03 67 FF 00 FF 01 2C
  - Temperature = -0.1°C
  - Battery Voltage = 3.00 V

3.2 Serial Payload

The Transceiver sends the serial data that it receives from the connected Modbus device to the NS. Such payloads have the format as shown in Figure 3-2, and are sent on ***LoRaWAN port 20***. A single payload will only include data for one serial message.

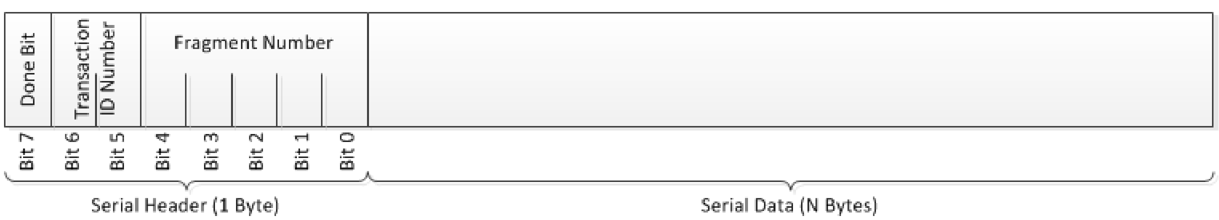


Figure 3-2: The UL serial payload.

In Figure 3-2, the "Serial Data" is in a big-endian format (MSB first), and is the exact serial data received from the connected Modbus device. In the event that the serial data is too large to fit into a single LoRaWAN UL message, it will automatically be fragmented, and the fragments, with the numbers given by "Fragment Numbers", will be transmitted up when the Transceiver is able. The "Fragment Number" is used to rebuild the packet at the user application in the proper order. The "Transaction ID Number" is an ID given to a serial data transaction. All fragments of a serial data

[2] The Input 1 "true" state is when Input 1 is open circuited, or a voltage of 1.8 V to 60 is applied to it. The "false" state is when a voltage of 0 V (equivalent to a short circuit) to 0.8 V is applied to Input 1.

request from the connected Modbus device will have the same Transaction ID Number. The “Done Bit” is only set if this is the last fragment of the transmission.

IMPORTANT NOTES:

- **Serial payloads of greater than 255 bytes can be enabled with a configuration register. See Section 4.3.3.6 for more information.**
- **The serial port on Industrial Transceiver can be set to continuously receive, which will enable the transceiver to receive and uplink data from (client facing) serial without any application or network server intervention. See Section 4.3.3.7 for more information.**

3.3 Response to Configuration and Control Commands

Transceiver responses to DL configuration and control commands (which are sent on LoRaWAN port 100; see Section 4.3) 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 Transceiver 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 Transceiver 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 Transceiver response in the former case).

4 DL Payload Formats

The DL streams (from the NS to the Transceiver) supported by the SW include the following:

- The data intended for the Transceiver's serial port (sent on ***LoRaWAN port 20***)
- Change requests for the state of the Transceiver's (digital) outputs, i.e. opened/closed (sent on ***LoRaWAN port 10***)
- Configuration and control commands used to change the Transceiver's behavior (sent on ***LoRaWAN port 100***)

These topics are explained in Sections 4.1, 4.2, and 4.3, respectively. Refer to [2] for a comprehensive tool to encode DL messages into DL frame payloads.

4.1 Serial Payload

Applications may need to send data in the DL intended for the Transceiver's serial port, i.e. the serial data (e.g. requests, commands, etc.) to be received by the Modbus device connected to the Transceiver. Such data are sent on ***LoRaWAN port 20*** and should be the exact message required to be sent to the Modbus device; i.e. no additional formatting is required. In fact, whenever the Transceiver receives data on port 20, it directs the full payload to its serial port.

4.2 Request to Change Output States

Requests to change the states of the Transceiver's Output 1 and Output 2 are sent on ***LoRaWAN port 10***, with a payload as shown in Table 4-1.

Table 4-1: DL Frame Payload Values to Change Output States

Information Type	Data Channel ID	Data Type ID	# Bytes	Data Type	Data Format	JSON Variable (Type/Unit)
Output 1	0x01	0x01	1	Boolean	0x00 = Open 0xFF = Closed	*output_1: <value>* *(unsigned/boolean)*
Output 2	0x02	0x01	1	Boolean	0x00 = Open 0xFF = Closed	*output_2: <value>* *(unsigned/boolean)*

In other words, on port 10, we send the following:

- 0x 01 01 00 to open Output 1
- 0x 01 01 FF to close Output 1
- 0x 02 01 00 to open Output 2
- 0x 02 01 FF to close Output 2

Note 1: The state of Output 1 or Output 2 can be changed by a request sent in the DL only when the Output is configured to be manually controllable over the DL by Application (see Section 4.3.3.4 for the configuration of the Outputs).

Note 2: In the current version of the SW, it is not possible to set the states of both Outputs in one message. For example, closing Output 1 and Output 2 cannot be done by sending the single message of 0x 01 01 FF 02 01 FF on port 10; it has to be done by two separate messages.

4.3 Configuration and Control Commands

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 4-1. A big-endian format (MSB first) is always followed.

The Command Field has a "register" address that is used to access various configuration parameters. These addresses are bound between 0x00 and 0x7F.

Bit 7 of the Command Field determines whether a read or write action is being performed. To write to a register, this bit must be set to 1 (one), but to read a register, it must be set to 0 (zero). 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.

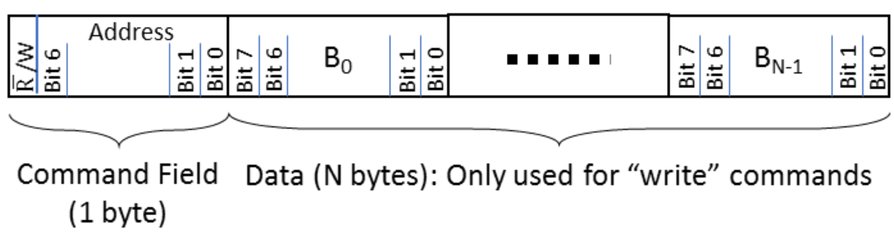


Figure 4-1: The format of a DL configuration and control message block.

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

Examples:

In the following examples, the Command Field is boldfaced:

- Read Reg 0x00, 0x01, 0x02:
  - DL command: { 0x **00 01 02** }
- Read Reg 0x05 and write value 0x8000 to Reg 0x10:
  - DL command: { 0x **05 90** 80 00 }

When a write command is sent to the Transceiver, it 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 3.3).

DL configuration and control commands fall into one of the following 4 (four) categories and are discussed in Sections 4.3.1, 4.3.2, 4.3.3, and 4.3.4, respectively:

- LoRaWAN Commissioning
- LoRaMAC Configuration
- Application Configuration
- Command and Control

4.3.1 LoRaWAN Commissioning

LoRaWAN commissioning values can be read back from the Transceiver using DL commands. These registers are RO. See LoRaWAN 1.0.3 specification [1] for description of the values. Table 4-2 shows a list of these registers.

Table 4-2: LoRaWAN Commissioning Registers

Address	Access	Value	# Bytes
0x00	R	DevEUI	8
0x01	R	AppEUI	8
0x02	R	AppKey	16
0x03	R	DevAddr	4
0x04	R	NwkSKey	16
0x05	R	AppSKey	16

Note 1: Commissioning values need to be kept secure at all times.
Note 2: Registers 0x02, 0x04, 0x05 cannot be read back in some regions if the DR number is too small. For example, in the NA region, the maximum frame payload size with DR0 is 11 bytes.

4.3.2 LoRaMAC Configuration

LoRaMAC options can be configured using DL commands. These configuration options change the default MAC configuration that the Transceiver loads on start-up. They can also change certain run-time parameters.
Table 4-3 shows the MAC configuration registers. In this table, B_i refers to data byte indexed *i* as defined Figure 4-1.

Table 4-3: LoRaMAC Configuration Registers

Address	Access	Value	# Bytes	Description	JSON Variable (Type/Unit)
0x10	R/W	Join Mode	2	• B_0-bits 0-6, B_1: Ignored • B_0-bit 7: ➢ 0 = ABP ➢ 1 = OTAA	*loramac_join_mode: <value>* *(unsigned/no unit)*
0x11	R/W	• Unconfirmed / Confirmed UL • Disable / Enable Duty Cycle • Disable / Enable ADR	2	• B_0-bits 0–3: Ignored • B_0-bits 4–7: ➢ 0x0 = Class A ➢ 0xC = Class C ➢ 1-11, 13-15: Invalid and Ignored • B_1-bit 0: ➢ 0 = Unconfirmed UL ➢ 1 = Confirmed UL • B_1-bit 2: ➢ 0 = Disable duty cycle ➢ 1 = Enable duty cycle • B_1-bit 3: ➢ 0 = Disable ADR ➢ 1 = Enable ADR • B_1-bits 1, 4-7: Ignored	*loramac_opts {* *lora_class: <value> (unsigned/A, C, or F if invalid)* *confirm_mode: <value>, (unsigned/no unit)* *duty_cycle: <value>, (unsigned/no unit)* *adr: <value> (unsigned/no unit)* *}*

0x12	R/W	• Default DR number • Default Tx Power number	2	B_0-bits 3–0: Default DR number [3] B_1-bits 3–0: Default Tx power number [3]	*loramac_dr_tx { dr_number: <value>, (unsigned/no unit) tx_power_number: <value>, (unsigned/no unit) }*
0x13	R/W	• Rx2 window DR number • Rx2 window channel frequency	5	B_0-B_1-B_2-B_3: Channel frequency in Hz for Rx2 B_4: DR for Rx2	*loramac_rx2 { frequency: <value>, (unsigned/Hertz) dr_number: <value> (unsigned/no unit) }*
0x19	R/W	Net ID MSBs	2	Bytes B_0-B_1 in the Net ID (B_0-B_1-B_2-B_3)	*netid_msb: <value> (unsigned/no unit)*
0x1A	R/W	Net ID LSBs	2	Bytes B_2-B_3 in the Net ID (B_0-B_1-B_2-B_3)	*netid_lsb: <value> (unsigned/no unit)*

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

Examples:

In the following example payloads, the Command Field is boldfaced:

- Switch Transceiver to ABP Mode:
  - DL payload: { 0x **90** 00 00 }
- Set ADR On, No Duty Cycle, and Confirmed UL Payloads:
  - DL payload: { 0x **91** 00 09 }
- Set default DR number to 1 and default Tx Power number to 2:
  - DL payload: { 0x **92** 01 02 }

4.3.2.1 Default Configuration

Table 4-4 and Table 4-6 list the default values for the LoRaMAC configuration registers (cf. [1], [3]).

Table 4-4: Default Values of LoRaMAC Configuration Registers

Address	Default Value
0x10	0x 80 00 (OTAA mode)
0x11	0x 00 0E (Class A, Unconfirmed UL, enabled duty cycle, enabled ADR)
0x12	0x 00 00 (DR0, Tx Power 0—max power, see Table 4-5)
0x13	As per Table 4-6.
0x19	0x 00 00
0x1A	0x 00 00

Table 4-5: Maximum Tx Power in Different Regions by Default

RF Region	Max Tx EIRP [dBm]
EU868	16
US915	30
AS923	16
AU915	30
IN865	30
CN470	19.15
KR920	14
RU864	16
DN915	30

Table 4-6: Default Values of Rx2 Channel Frequency and DR Number in Different Regions

RF Region	Default Value	Channel Frequency	DR Number
EU868	0x 33 D3 E6 08 00	869.525 MHz	DR0
US915	0x 37 08 70 A0 08	923.3 MHz	DR8
AS923	0x 37 06 EA 00 02	923.2 MHz	DR2
AU915	0x 37 08 70 A0 08	923.3 MHz	DR8
IN865	0x 33 A6 80 F0 02	866.55 MHz	DR2
CN470	0x 1E 1E 44 20 00	505.3 MHz	DR0
KR920	0x 36 F3 13 E0 00	921.9 MHz	DR0
RU864	0x 33 CD 69 E0 00	869.1 MHz	DR0
DN915	0x 2B 44 5A E0 08	725.9 MHz	DR8

4.3.3 Application Configuration

This section lists all possible Transceiver application configurations (as part of DL configuration and control commands), including periodic Tx configuration, Input 1 configuration, threshold configuration, Output 1 and Output 2 configuration, serial interface configuration, and Modbus RTU configuration.

Note: Care must be taken to avoid stranding the Transceiver during reconfiguration. If all sensing inputs are disabled, the device will not be able to be reconfigured.

4.3.3.1 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 4-7.

Table 4-7: Periodic Transmission Configuration

Address	Access	Value	# Bytes	Description	JSON Variable (Type/Unit)
0x20	R/W	Seconds per Core *Tick*	4	Sets the core *tick* for periodic events. A value of 0 disables all periodic transmissions.	*seconds_per_core_tick: <value> (unsigned/second)*

0x21	R/W	*Ticks* per Battery	2	*Ticks* between battery voltage reports. A value of 0 disables periodic battery reports.	*tick_per_battery: <value> (unsigned/no unit)*
0x22	R/W	*Ticks* per Ambient Temp	2	*Ticks* between temperature reports (0 disables)	*tick_per_ambient_temperature: <value> (unsigned/no unit)*
0x24	R/W	*Ticks* per Input 1	2	*Ticks* between Input 1 reports (0 disables)	*tick_input1: <value> (unsigned/no unit)*
0x25	R/W	*Ticks* per Input 2	2	*Ticks* between Input 2 reports (0 disables)	*tick_input2: <value> (unsigned/no unit)*
0x26	R/W	*Ticks* per Input 3	2	*Ticks* between Input 3 reports (0 disables)	*tick_input3: <value> (unsigned/no unit)*
0x27	R/W	*Ticks* per MCU Temp	2	*Ticks* between MCU temperature reports (0 disables)	*tick_per_mcu_temperature: <value> (unsigned/no unit)*
0x28	R/W	*Ticks* per Output 1	2	*Ticks* between Output 1 reports (0 disables)	*tick_output1: <value> (unsigned/no unit)*
0x29	R/W	*Ticks* per Output 2	2	*Ticks* between Output 2 reports (0 disables)	*tick_output2: <value> (unsigned/no unit)*

4.3.3.1.1 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 Transceiver data. The minimum seconds per *tick* is 30 seconds and the maximum is 86,400 seconds (one day). Values from 1 to 29 or above 86,400 are invalid and ignored. A value of 0 (zero) disabled all periodic reporting.

4.3.3.1.2 *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 Transceiver polls the specified transducer and reports the data in an UL message. A setting of 0 (zero) disables periodic reporting for the specified transducer.

4.3.3.1.3 Default Configuration

Table 4-8: Periodic Transmission Default Configuration

Seconds per Core *Tick*	900 seconds (15 min)
***Ticks* per Battery**	4 (1 hour)
***Ticks* per Ambient Temp**	4 (1 hour)
***Ticks* per Input 1**	1 (15 min)
***Ticks* per Input 2**	1 (15 min)
***Ticks* per Input 3**	1 (15 min)
***Ticks* per MCU Temp**	0 (disabled)
***Ticks* per Output 1**	0 (disabled)
***Ticks* per Output 2**	0 (disabled)

4.3.3.1.4 Example DL Messages

- Disable all periodic events:
  - 0x A0 00 00 00 00 (Reg 20, write bit set to true) – Seconds in a Tick = 0 (disabled)
- Read the current "Seconds in a Tick" value:
  - 0x 20 (Reg 20, write bit set to false)
- Write "Tick per Temperature Tx":
  - 0x A2 00 01 (Reg 22, write bit set to true) – set "Ticks per Temperature Tx" to 1 (one)

4.3.3.2 Input 1 Configuration

Input 1 provides digital on/off sensing of the input signal. It can never be disabled; however, the (periodic or event-based) reporting can be disabled if care is not taken during configuration.

Table 4-9: Input 1 Configuration

Address	Access	Value	# Bytes	Description	JSON Variable (Type/Unit)
0x2A	R/W	Mode	1	Bit 0: Rising edge (0 = disable, 1 = enable) Bit 1: Falling edge (0 = disable, 1 = enable)	*input1_mode {* *rising_edge_enabled: <value>,* *(unsigned/no unit)* *falling_edge_enabled: <value>* *(unsigned/no unit)* *}*
0x2B	R/W	Count Threshold	2	Number of triggers for event Tx (0 disables event Tx)	*input1_count_threshold: <value>* *(unsigned/no unit)*
0x2C	R/W	Value to Tx	1	Bit 0: Input State (0 = disable, 1 = enable) Bit 1: Counter Value (0 = disable, 1 = enable)	*input1_tx {* *report_state_enabled: <value>,* *(unsigned/no unit)* *report_count_enabled: <value>* *(unsigned/no unit)* *}*

4.3.3.2.1 Mode

Input 1 is edge-triggered, and can be set to trigger to rising-edge trigger (closed to open), falling-edge triggered (open to closed) or both.

An attempt to set Input 1 Mode to 0x00 (i.e. disable both "Falling Edge" and "Rising Edge") will be ignored by the Transceiver.

Application Examples:

- Pulse counting from a water meter would use a single edge trigger, depending on the resting state of the connected device (positive pulse would use a rising edge, negative pulse would use a falling edge).

4.3.3.2.2 Count Threshold

The Count Threshold determines when the Transceiver transmits after seeing an event on Input 1. A value of 0 (zero) disables the event driven transmission, while a value of 1 or greater will trigger an event-based transmission after the configured number of events has occurred. In fact, Input 1 has two counters:

1) Counter 1: that keeps a total number of times Input 1 is triggered since the Transceiver has joined the network, and keeps the actual "Counter Value" that can be reported (see Section 4.3.3.2.3).
2) Counter 2: that increments like Counter 1 each time Input 1 is triggered, but that resets to 0 and triggers a transmission whenever it reaches the Count Threshold.

Application Example:

- Pulse counting from a high-volume water meter. The Transceiver owner may disable event-based transmission in favor of getting hourly reports of pulse count from the device.

4.3.3.2.3 Value to Tx

The Input 1 Value to Tx determines what information is transmitted whenever an event or periodic Input 1 Tx is required. "Input State" will transmit the current Input 1 state (open/closed). "Counter Value" will contain the total number of times Input 1 has been triggered since the Transceiver has joined the Network (see Section 4.3.3.2.2). Not both Input State and Counter Value can be disabled. An attempt to do so is ignored by the SW.

4.3.3.2.4 Default Configuration

Table 4-10: Input 1 Default Configuration

Mode	0x03 ("Rising Edge" and "Falling Edge" are both enabled)
Count Threshold	1
Value to Tx	0x03 ("Input State" and "Counter Value" are both enabled)

4.3.3.2.5 Example DL Messages

- Set "Mode" to single rising edge:
  - 0x AA 01 (Reg 2A, write bit set to true) – rising edge only enabled
- Read "Count Threshold":
  - 0x 2B (Reg 2B, write bit set to false)
- Write to "Count Threshold" and "Value to Tx":
- 0x AB 00 0A AC 02 (Reg 2B and Reg 2C, write bit set to true) – set "Count Threshold" to 10 (ten) and "Value to Tx" to "Counter Value".

4.3.3.3 Threshold-Based Configuration

The Industrial Transceiver supports threshold-based transmission on 4 (four) different transducers:

- Input 2
- Input 3
- Temperature
- MCU Temperature

When a threshold is enabled, the Industrial Transceiver will report the transducer value when it leaves the configured threshold window, and once again when the transducer value re-enters the threshold window. Inside the configured threshold window is called the Idle State. Outside the window is the Active Sate.

The threshold mode can be enabled concurrently with periodic reporting. The transducer will be reported at its scheduled periodic interval, and also if the threshold is triggered. Table 4-11 shows configuration parameters for the threshold-based operation of the Transceiver.

Table 4-11: Threshold-Based Transmission Configuration

Address	Access	Value	# Bytes	Description	JSON Variable (Type/Unit)
0x30	R/W	Input 2/Input 3 Sample Period: Idle State	4	Sample period of Input 2/Input 3 in seconds in Idle State	*input_sample _period_idle: <value>* *(unsigned/seconds)*
0x31	R/W	Input 2/Input 3 Sample Period: Active State	4	Sample period of Input 2/Input 3 in seconds in Active State	*input_sample _period_active: <value>* *(unsigned/seconds)*
0x32	R/W	Input 2 Thresholds	2	• MSB byte: High current threshold (unsigned, 100 µA/LSB) • LSB byte: Low current threshold (unsigned, 100 µA/LSB)	*input2_threshold {* *high: <value>, (unsigned/ampere)* *low: <value> (unsigned/ampere)* *}*
0x33	R/W	Input 3 Thresholds	2	• MSB byte: High voltage threshold (unsigned, 50 mV/LSB) • LSB byte: Low voltage threshold (unsigned, 50 mV/LSB)	*input3_threshold {* *high: <value>, (unsigned/volt)* *low: <value> (unsigned/volt)* *}*
0x34	R/W	Input Threshold Enable	1	Bit 0: Input 2 Threshold enabled Bit 4: Input 3 Threshold enabled	*threshold_enabled {* *input2: <value>, (unsigned/boolean)* *input3: <value> (unsigned/boolean)* *}*
0x39	R/W	Temperature Sample Period: Idle State	4	Sample period of Temperature in seconds in Idle State	*temperature_sample _period_idle: <value>* *(unsigned/seconds)*
0x3A	R/W	Temperature Sample Period: Active State	4	Sample period of Temperature in seconds in Active State	*temperature_sample _period_active: <value>* *(unsigned/seconds)*

0x3B	R/W	Temperature Thresholds	2	• MSB byte: High temperature threshold (signed, 1°C/LSB) • LSB byte: Low temperature threshold (signed, 1°C/LSB)	*temperature_threshold { high: <value>, (signed/celsius) low: <value> (signed/celsius) }*
0x3C	R/W	Temperature Threshold Enable	1	Bit 0: 0 = Off 1 = On	*temperature_theshold _enabled: <value> (unsigned/boolean)*
0x40	R/W	MCU Temperature Sample Period: Idle State	4	Sample rate of MCU Temperature in seconds in Idle State	*mcu_temperature_sample _period_idle: <value> (unsigned/seconds)*
0x41	R/W	MCU Temperature Sample Period: Active State	4	Sample rate of MCU Temperature in seconds in Active State	*mcu_temperature_sample _period_active: <value> (unsigned/seconds)*
0x42	R/W	MCU Temperature Thresholds	2	• MSB byte: High MCU temperature threshold (signed, 1°C/LSB) • LSB byte: Low MCU temperature threshold (signed, 1°C/LSB)	*mcu_temperature_threshold { high: <value>, (signed/celsius) low: <value> (signed/celsius) }*
0x43	R/W	MCU Temperature Threshold Enable	1	Bit 0: 0 = Off 1 = On	*mcu_temperature_theshold _enabled: <value> (unsigned/boolean)*

4.3.3.3.1 Input 2/Input 3/Temperature/MCU Temperature Sample Period: Idle State

The Idle State sample period determines how often a transducer is checked when the reported value is within the threshold window. This value is given in seconds, with a minimum of 10 and a maximum of 86400. Values smaller than 10 or larger than 86400 are ignored by the SW.

Note: When the threshold-based reporting is enabled first, the Transceiver will start in the Idle State.

4.3.3.3.2 Input 2/Input 3/Temperature/MCU Temperature Sample Period: Active State

The Active State sample period determines how often a transducer is checked when the reported value is outside the threshold window. This value is given in seconds, with a minimum of 10 and a maximum of 86400. Values smaller than 10 are changed to 10, and values larger than 86400 are changed to 86400, automatically.

4.3.3.3.3 Input 2/Input 3 Thresholds

Input Thresholds are stored in a single 2-byte register, with the MSB byte storing the high threshold, and the LSB byte storing the low threshold. The high threshold must be greater than the low threshold.

4.3.3.3.4 Temperature/MCU Temperature Thresholds

Temperature thresholds are stored in a single 2-byte register, with the MSB byte storing the high threshold, and the LSB byte storing the low threshold. The high threshold must be greater than the low threshold.

4.3.3.3.5 Input/Temperature/MCU Temperature Threshold Enabled

The <Transducer> Threshold Enabled register enables or disables the threshold reporting on the specified transducer. "Thresholds" and "Sample Period" values can be configured, but will not be activated unless the "Threshold Enabled" bit is set.

Note: Input 2 and Input 3 "Threshold Enabled" is configured within the same register.

4.3.3.3.6 Default Configuration

Table 4-12: Threshold-Based Transmission Default Configuration

Input 2/Input 3 Sample Period: Idle State	60 s
Input 2/Input 3 Sample Period: Active State	60 s
Input 2 Thresholds (High/Low)	10mA/0.3mA
Input 3 Thresholds (High/Low)	2V/0.5V
Input Threshold Enabled (Input 2/Input 3)	Off/Off
Temperature Sample Periods: Idle State	60 s
Temperature Sample Periods: Active State	60 s
Temperature Thresholds (High/Low)	30°C/15°C
Temperature Threshold Enabled	Off
MCU Temperature Sample Period: Idle State	60 s
MCU Temperature Sample Period: Active State	60 s
MCU Temperature Thresholds (High/Low)	30°C/15°C
MCU Temperature Threshold Enabled	Off

4.3.3.3.7 Example DL Messages

- Write Temperature thresholds:
  - 0x BB 19 F1 (Reg 3B, write bit set to true) – high 25°C/low -15°C.
- Read Temperature sample periods:
  - 0x 39 3A (Reg 39 and Reg 3A, write bit set to false)

4.3.3.4 Output Configuration

The Outputs of the Industrial Transceiver can be configured to be either controllable via the Application (i.e. from the NS with a DL command—see Section 4.2), or tied to a configurable input (i.e. Input 1, Input 2, or Input 3) to automatically toggle on and off when the configured input is sampled. This allows for power hungry input sources to be powered down in order to save power when not actively being sampled.

Table 4-13: Output 1 and Output 2 Configuration

Address	Access	Value	# Bytes	Description	JSON Variable (Type/Unit)
0x50	R/W	Output 1 Config	1	Bits 1-0: 00 = Manual control (over DL by Application) 01 = Tied to Input 1 (automatically toggled before and after measuring Input 1) 10 = Tied to Input 2 (automatically toggled before and after measuring Input 2) 11 = Tied to Input 3 (automatically toggled before and after measuring Input 3)	*output1_control: <value> (unsigned/no unit)*
0x51	R/W	Output 1 Delay	2	Time in ms to close Output 1 before sampling the tied Input and to open Output 1 after sampling the tied Input (N/A in the Application Control mode)	*output1_delay: <value> (unsigned/seconds)*
0x52	R/W	Output 2 Config	1	Bits 1-0: 00 = Manual control (over DL by Application) 01 = Tied to Input 1 (automatically toggled before and after measuring Input 1) 10 = Tied to Input 2 (automatically toggled before and after measuring Input 2) 11 = Tied to Input 3 (automatically toggled before and after measuring Input 3)	*output2_control: <value> (unsigned/no unit)*
0x53	R/W	Output 2 Delay	2	Time in ms to close Output 2 before sampling the tied Input and to open Output 2 after sampling the tied Input (N/A in the Application Control mode)	*output2_delay: <value> (unsigned/seconds)*

4.3.3.4.1 Output 1/Output 2 Config

The Industrial Transceiver outputs can be configured to automatically toggle on and off when an input is sampled. This affects both periodic and threshold-based reporting.

4.3.3.4.2 Output 1/Output 2 Delay

The Output Delay tells the Transceiver to toggle the Output how long before and how long after sampling the Input. For example, a value of 1000 ms, closes the Output, 1 second before and opens the Output, 1 second after sampling the Input. This value can range from 0 ms to 65535 ms.

The interaction between inputs and outputs are best explained in the diagrams of Figure 4-2, where the whole periodic reporting process is illustrated. In this figure, the "wakeup schedule" for the MCU is every "core tick" (see Section 4.3.3.1). For example, if the core tick is set to a minute, the MCU wakes up every minute to take care of periodic reporting. Also, "In" and "Out" in the figure stand for Input and Output. Moreover, Δ_1 and Δ_2 denote the Output 1 and Output 2 delays, respectively.

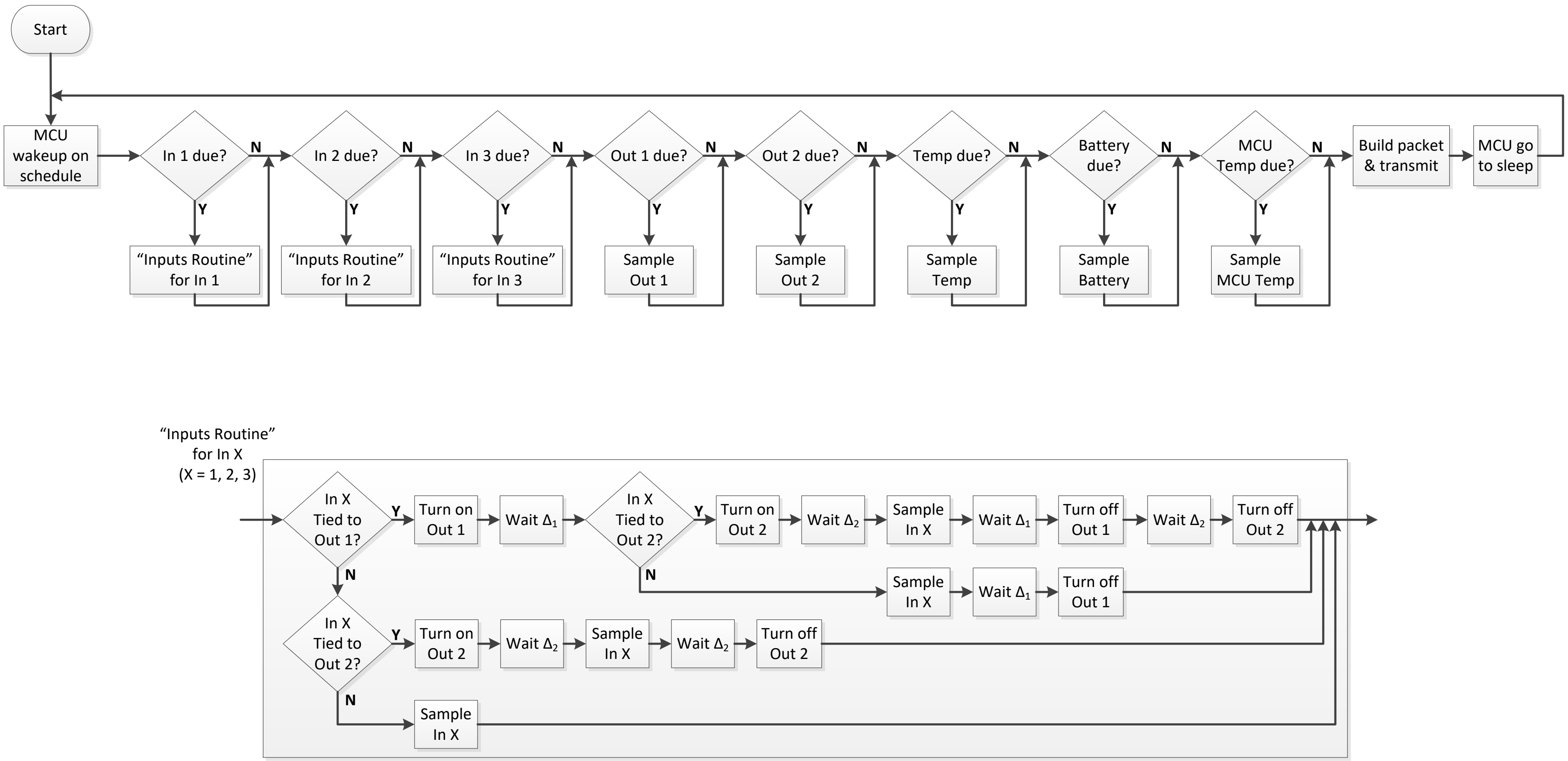


Figure 4-2: The periodic reporting flow diagram showing the input-output interactions

Application Example:

- The current source attached to Input 2 needs to be turned on prior to taking a measurement. Proper configuration allows for the Industrial Transceiver to automatically enable the current source prior to a measurement being taken, and then automatically disable the source after the measurement is complete.

4.3.3.4.3 Default Configuration

Table 4-14: Output 1 and Output 2 Default Configuration

Output 1 Config	Manual Control
Output 1 Delay	100 ms
Output 2 Config	Manual Control
Output 2 Delay	100 ms

4.3.3.5 Serial Interface Configuration

The serial interface provides the ability to send/receive serial messages to/from a connected Modbus device using the Industrial Transceiver as a relay. Table 4-15 shows the list of configuration parameters for the serial interface.

Table 4-15: Serial Interface Configuration

Address	Access	Value	# Bytes	Description	JSON Variable (Type/Unit)
0x60	R/W	Interface Type	1	Bit 0: 1 = RS232 0 = RS485/RS422 Other bits: RFU.	*serial_interface_type: <value> (unsigned/no unit)*
0x61	R/W	Baud Rate	4	Baud rate for the RS232/RS485/RS422 interfaces in bps	*serial_baud_rate: <value> (unsigned/bits per second)*
0x63	R/W	Parity Bits	1	Parity bits: 0x00 = no parity 0x01 = odd parity 0x02 = even parity	*serial_parity_bits: <value> (unsigned/no unit)*
0x64	R/W	Stop Bits	1	Stop bits: 0x05 = 0.5 bits 0x0A = 1 bit 0x0F = 1.5 bits 0x14 = 2 bits	*serial_stop_bits: <value> (unsigned/bits)*
0x65	R/W	Duplex Mode	1	Bit 0: 0 = half duplex 1 = full duplex Other bits: RFU.	*serial_duplex_mode: <value> (unsigned/no units)*

NOTE: Register 0x62 for data bit configuration of either 8 or 9 bits has been REMOVED. TEKTELIC only supports the use of 8 data bits. Should there be any issue, please contact TEKTELIC Customer Support.

4.3.3.5.1 Interface Type

Selects the base protocol of the serial interface, either RS232 or RS485/422 (differential).

4.3.3.5.2 Baud Rate

The baud rate of the serial interface in bps.

4.3.3.5.3 Parity Bits

Sets the parity bit option of the serial interface.

4.3.3.5.4 Stop Bits

Sets the stop bits of the serial interface.

4.3.3.5.5 Duplex Mode

Sets the duplex mode of the serial interface.

4.3.3.5.6 Default Configuration

Table 4-16: Serial Interface Default Configuration

Type	RS232
Baud Rate	115200
Parity Bits	0 (no parity)
Stop Bits	10 (1 stop bit)
Duplex Mode	1 (full duplex)

4.3.3.5.7 Example Configuration

- Write Baud Rate:
  - 0x E1 00 00 25 80 (Reg. 61, write bit set to true): Baud Rate = 9600 bps
  - 0x E1 00 00 4B 00 (Reg. 61, write bit set to true): Baud Rate = 19200 bps
- Read serial interface configuration registers:
  - 0x 60 61 62 63 64 65

NOTE: When writing new configuration to the serial interface, the configuration must be SAVED and the sensor RESTARTED. This can be accomplished by sending 0x F0 60 01 to the Industrial Transceiver. This will save both LoRa and application configuration, then restart the sensor.

4.3.3.6 Extended Serial Payload Capability[3]

The industrial transceiver is capable of handling serial payload sizes greater than 255 bytes. The original serial uplink payload format (shown in Figure 4-3 below) limits the fragment number to 32. In some use cases, this is simply not enough, nor can a smaller fragment number handle edge cases such as lowest data rate. Along with the increase in allocated serial response data size from 255 bytes to 512, the extended serial uplink payload format (shown in Figure 4-4 below) solves this issue. As the name suggests, this extends the functionality of the original format and does not replace the original format. The extended serial uplink format can be enabled by configuring register 0x66. Once enabled, serial uplinks – except for periodic Modbus responses – will be sent through **port 40** following the format outlined in Figure 4-4.

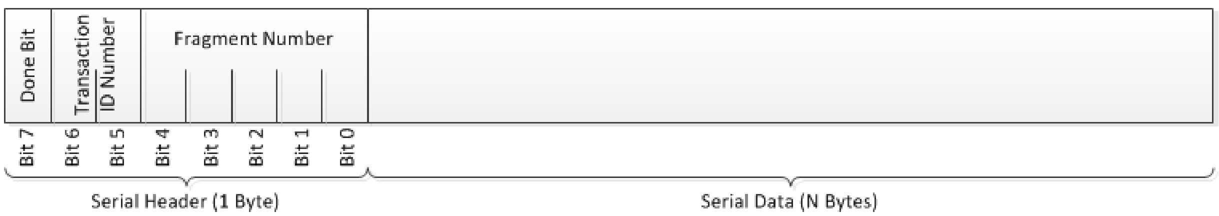


Figure 4-3: Original Serial Uplink Payload Format

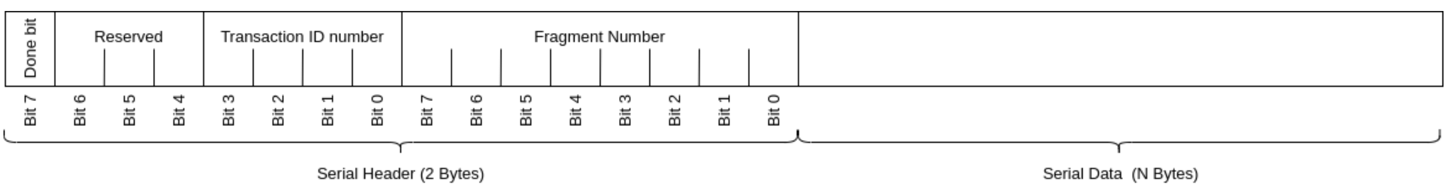


Figure 4-4: Extended Serial Uplink Payload Format

A configuration register will be used to enable extended serial uplink as outlined on Table 4-17 below.

Table 4-17: Extended Serial Uplink Register

Address	Access	Value	# Bytes	Description	JSON Variable (Type/Unit)
0x66	R/W	Serial Uplink Format	1	Extended Serial Uplink Format: 0x00 = short (original) format 0x01 = extended format	*serial_uplink_format: <value> (unsigned/no unit)*

4.3.3.6.1 Default Configuration

Table 4-18: Extended Serial Uplink Default Setting

Extended Serial Uplink Format	0x00 (disabled)

[3] This feature is only available in SW Version 1.1.2.

4.3.3.6.2 Operation

With the exception of periodic Modbus commands, the extended serial uplink format can only be sent through LoRaWAN **port 40**. This ensures backwards compatibility for user applications using the original format. Periodic Modbus response (i.e. serial uplinks) will be sent through their assigned ports using the format enabled by register 0x66.

The same framing procedure as the original format will be followed by the extended format. However, the total number of serial data bytes that can be sent in one transaction is increased to 512 bytes. It is not advised, however, to use all 512 bytes at lower data rates. Although the extended format can handle this edge case, the sensor's performance, and battery life, will diminish.

4.3.3.7 Continuous Serial Receive[4]

Continuous serial receive will enable the transceiver to receive and uplink data from the serial port without any application or network server intervention. This feature is available for both Class A and Class C industrial transceiver devices, and can be enabled by configuring register 0x67.

Register 0x67 can be configured as outlined in Table 4-19 below.

Table 4-19: Continuous Serial Receive Register

Address	Access	Value	# Bytes	Description	JSON Variable (Type/Unit)
0x67	R/W	Continuous Serial Receive	1	Continuous Serial Receive: 0x00 = disabled 0x01 = enabled	*Continuous_serial_receive: <value>* *(unsigned/no unit)*

4.3.3.7.1 Default Configuration

Table 4-20: Continuous Serial Receive Default Setting

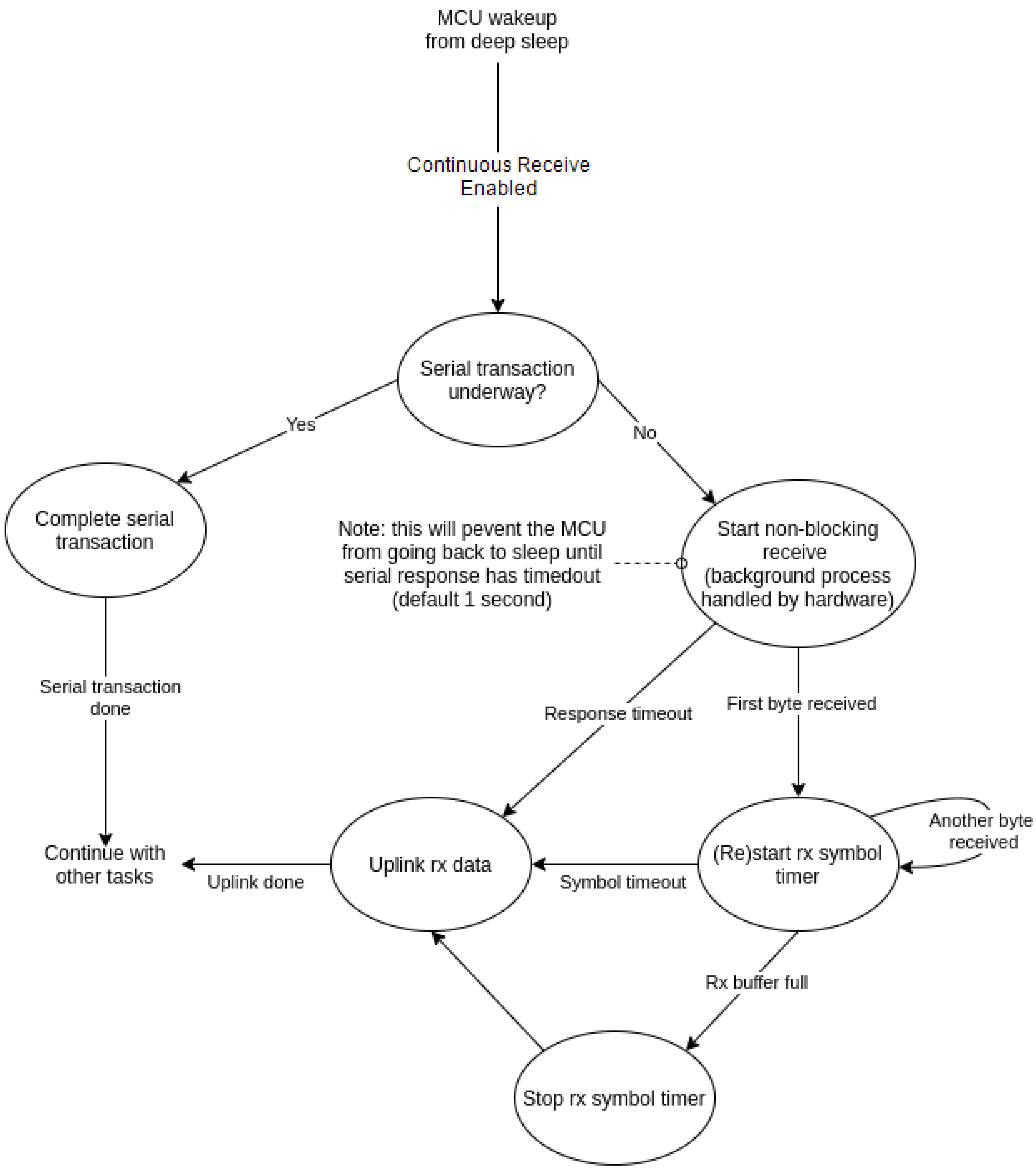
Continuous Serial Receive	0x00 (disabled)

4.3.3.7.2 Operation

Since this feature leaves the serial receive window open, an increase in power consumption, and in turn a shorter battery life, is to be expected. This increased power consumption is attributed to the UART peripheral continuously sampling its' receive (RX) line. Hence, to conserve power, Class A devices will have the serial receive window open only after waking up, and will remain open for the specified response timeout in register 0x69. The complete flow of operation for Class A is shown in Figure 4-5 below.

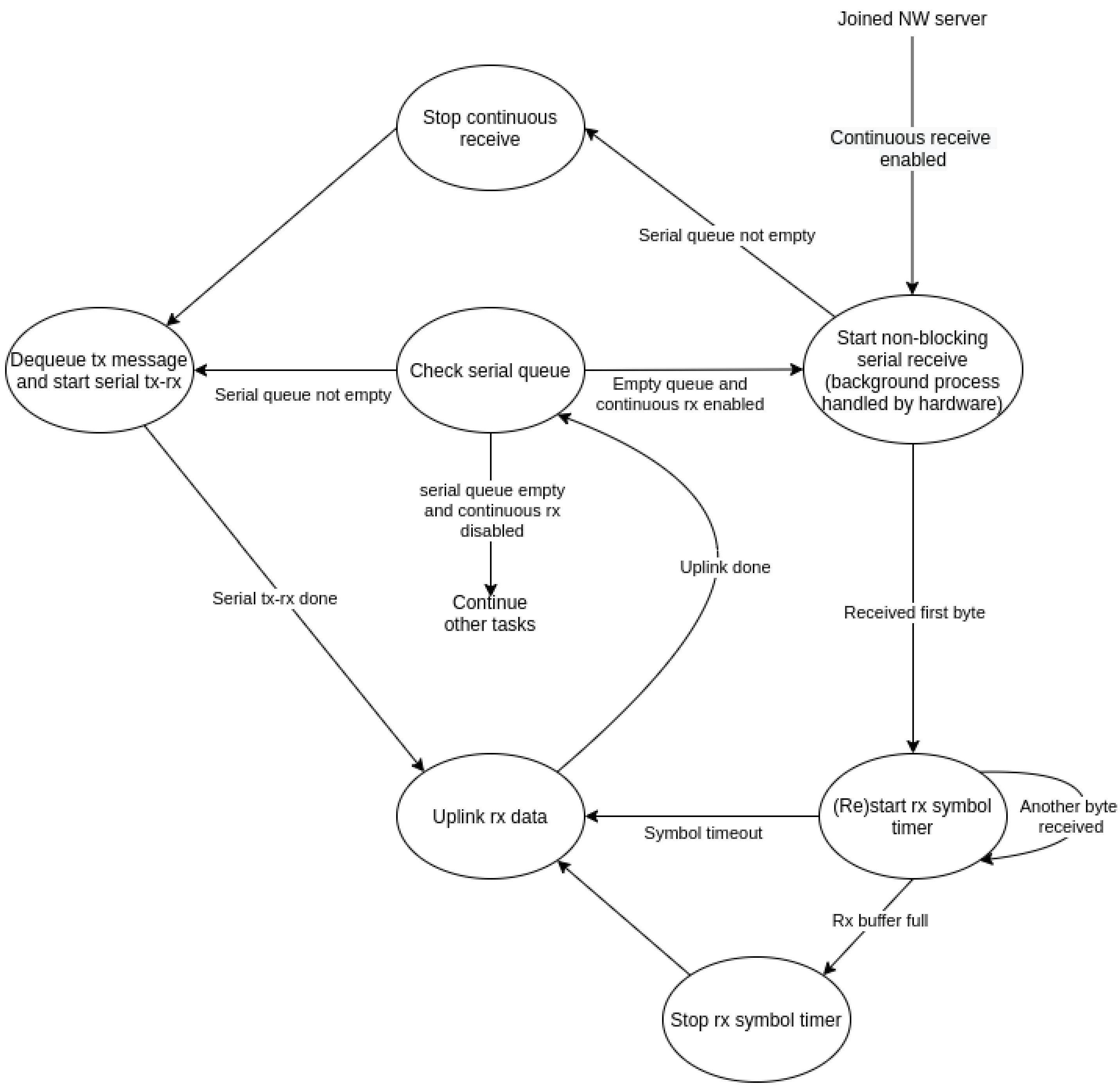
[4] This feature is only available in SW Version 1.1.2.

Figure 4-5: Class A Continuous Receive Operation



Historically, Class C devices have been externally powered and the increase in power consumption due to continuous UART receive sampling is negligible. Therefore, if continuous serial receive is enabled, the serial receive window will be open indefinitely until there are other pending serial transactions (i.e. periodic Modbus or serial downlink). The complete flow of operation is shown in Figure 4-6 below.

Figure 4-6: Class C Continuous Receive Operation



4.3.3.8 Serial Timeouts & Modbus RTU Configuration

Timeouts for a serial device can be set with registers 0x68 and 0x69. See section 4.3.3.8.1 and 4.3.3.8.2 for a description.

The Industrial Transceiver can communicate with a connected Modbus RTU device over the serial interface to poll the Modbus device at periodic intervals. This is accomplished in registers 0x6A to 6F. Responses for a Modbus RTU device will come in uplinks on the ports shown in Table 4-21 below:

Table 4-21: Periodic Modbus Ports

Periodic Modbus Register	Port
0x6A	21
0x6B	22
0x6C	23
0x6D	24
0x6E	25
0x6F	26

Table 4-22 shows a list of Modbus RTU configuration parameters.

Table 4-22: Serial Timeouts & Modbus RTU Configuration

Address	Access	Value	# Bytes	Description	JSON Variable (Type/Unit)
0x68	R/W	Serial Interface Symbol Timeout	2	Minimum symbol timeout between serial frames.	*serial_interface_symbol_timeout: <value>* *(unsigned/no units)*
0x69	R/W	Serial Interface Rx Timeout	2	Modbus Rx timeout in ms.	*serial_interface_rx_timeout: <value>* *(unsigned/milliseconds)*
0x6A[5]	R/W	Modbus RTU Polling Period and Command	10	B_0-B_1: Polling for autonomous polling in multiples of 10s. B_2-B_9: Modbus command	*modbus_rtu_polling_command: <value>* *(unsigned/10s)*
0x6B	R/W	Modbus RTU Polling Period and Command	10	B_0-B_1: Polling for autonomous polling in multiples of 10s. B_2-B_9: Modbus command	*modbus_rtu_polling_command: <value>* *(unsigned/10s)*

[5] **PLEASE NOTE: Register 6A has changed in functionality in SW versions succeeding version 1.0.2. This register now allows for a Modbus command in addition to setting a polling period. Prior functionality of this register was strictly polling period in ms.**

0x6C	R/W	Modbus RTU Polling Period and Command	10	B_0-B_1: Polling for autonomous polling in multiples of 10s. B_2-B_9: Modbus command	*modbus_rtu_polling_command: <value> (unsigned/10s)*
0x6D	R/W	Modbus RTU Polling Period and Command	10	B_0-B_1: Polling for autonomous polling in multiples of 10s. B_2-B_9: Modbus command	*modbus_rtu_polling_command: <value> (unsigned/10s)*
0x6E	R/W	Modbus RTU Polling Period and Command	10	B_0-B_1: Polling for autonomous polling in multiples of 10s. B_2-B_9: Modbus command	*modbus_rtu_polling_command: <value> (unsigned/10s)*
0x6F	R/W	Modbus RTU Polling Period and Command	10	B_0-B_1: Polling for autonomous polling in multiples of 10s. B_2-B_9: Modbus command	*modbus_rtu_polling_command: <value> (unsigned/10s)*

4.3.3.8.1 Symbol Timeout

The serial link must be quiet for a minimum number of symbols before a frame is considered complete and a new frame may be sent.

4.3.3.8.2 Rx Timeout

Time to wait for a response after transmitting a serial frame.

4.3.3.8.3 Polling Period and Command, Registers

These registers are used when polling a Modbus device at periodic intervals. This polling rate is defined in multiples of 10s, and is contained in bytes 0 to 1. Setting this polling rate to 0 (zero) disables periodic polling. The Modbus command to query the connected device is contained in bytes 2 through 9. For devices operating with duty cycle limitations care should be taken when setting the polling period.

4.3.3.8.4 Default Configuration

Table 4-23: Modbus RTU Default Configuration

Register	Description	Default Configuration
0x68	Symbol Timeout	28
0x69	Rx Timeout	1000 ms
0x6A to 6F	Polling Period (2 bytes)	0x 00 00 (Disabled)
0x6A to 6F	Modbus command (8 bytes)	0x00 00 00 00 00 00 00 00 (Disabled)

4.3.3.8.5 Example Configuration

- Write Rx Timeout:
  - 0x E9 0B B8 (Reg 69, write bit set to true) – Rx Timeout = 3000 ms
- Read current Modbus RTU configuration:
  - 0x 68 69 6A 6B 6C 6D 6E 6F

4.3.4 Command and Control

Configuration changes are not retained after a power cycle unless they are saved in the flash memory. Table 4-24 shows the structure of the Command and Control registers. In this table, B_i refers to data byte indexed *i* as defined in Figure 4-1.

Table 4-24: Command Control Registers

Address	Access	Name	# Bytes	Description	JSON Variable (Type/Unit)
0x70	W	Flash Memory Write Command	2	B_0, bit 5: Write App Config B_0, bit 6: Write LoRa Config B_1, bit 0: Restart Sensor In all cases: 0 = De-asserted, 1 = Asserted Other bits are ignored.	*write_to_flash {* *app_configuration:* <value>, *(unsigned/no unit)* *lora_configuration:* <value>, *(unsigned/no unit)* *restart_sensor:* <value> *(unsigned/no unit)* }
0x71	R	FW Version	7	B_0: App version major B_1: App version minor B_2: App version revision B_3: LoRaMAC version major B_4: LoRaMAC version minor B_5: LoRaMAC version revision B_6: LoRaMAC region number	*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[6]	1	0x0A: Reset App Config 0xB0: Reset LoRa Config 0xBA: Reset both App and LoRa Configs Any other value is ignored.	*configuration_factory_reset:* <value> *(unsigned/no unit)*

Note: The Flash Memory Write Command 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 the NS to continually reboot the Transceiver.

4.3.4.1 LoRaMAC Region

The LoRaMAC region is indicated by B_6 in the FW Version register (Reg 0x71). Current LoRaMAC regions and corresponding region numbers are listed in Table 4-25.

Table 4-25: LoRaMAC Regions and Region Numbers

LoRaMAC Region	Region Number
EU868	0
NA915	1
AS923	2
AU915	3
IN865	4
CN470	5
KR920	6
RU864	7
DN915	8

4.3.4.2 Command Examples

In the following examples, the Command Field is boldfaced:

- 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 Transceiver:
  - DL payload: { 0x **F0** 00 01 }
- Read FW versions, and reset application configuration to factory defaults:
  - DL payload: { 0x **71 F2** 0A }

[6] Resetting to factory defaults takes effect on the next power cycle.

4.3.5 Bricking Prevention

Care has been taken to avoid stranding (hard or soft bricking) the Transceiver during reconfiguration. Hard bricking refers to the condition that the Transceiver does not transmit any more 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 Transceiver 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. Therefore, transmissions from a soft-bricked Transceiver cannot be smaller than a week apart.

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

After the reconfiguration is applied, if all event-based reporting (see Sections 4.3.3.2 and 4.3.3.3 for event-based reporting) is disabled, then periodic reporting is checked (see Section 4.3.3.1 for periodic reporting). If all periodic reporting is disabled or the minimum non-zero period is greater than a week, then to avoid bricking the Transceiver, the core tick is set to 86400 (i.e. one day), and the battery voltage tick is set to 1 (one).

References

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

[2] TEKTELIC Communications Inc., "Sensor Configuration Tool": Sensor Config App (tektelic-dev.com)

[3] LoRa Alliance, "LoRaWAN Regional Parameters," ver. 1.1, rev. B, Jan 2018.

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