LoRaWAN Network Server Demonstration: High Level Description

1 History 2
2 Introduction 3
2.1 Main features 3
3 LoRa communication paths 6
3.1 Message protocol layers 7
3.1.1 LoRa gateway message protocol (GWMP) overview 7
3.1.2 JSON string 8
4 Adding a mote to the LoRa network 9
5 Service type 11
6 Compliance to the LoRa WAN protocol [1] 11
7 Glossary 12
8 References 16

1 History

Revision	Modification / Remarks / Motive	Author
1.0	Document created	DRo

2 Introduction

The document describes the general operation and interfaces of the LoRa™ servers (including the network controller).

The LoRa network server (NS), application server (AS) and network controller (NC) are licensed as part of the Semtech 'LoRa IoT Reference Network Software Solution'.

The LoRa customer server (CS) is licensed in the same way. The CS simply receives data from the AS and either stores it in a relational database or appends it to a text file. It is expected that the CS will be largely replaced or completely replaced in any operational LoRa system.

2.1 Main features

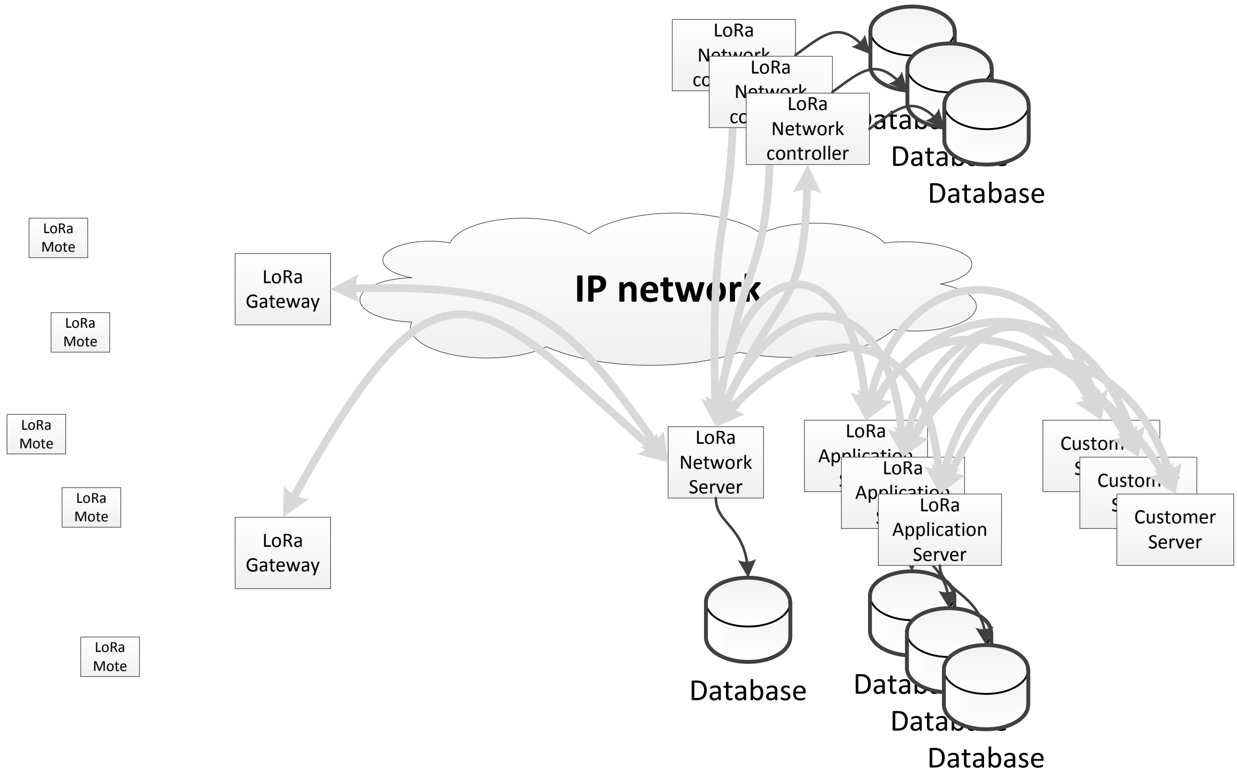

Figure 1 Context diagram

Mote

The Mote is the end-device of a LoRa network. Each mote communicates with one or more LoRa gateways. The communication protocol complies with the LoRaWAN specification [1] defined by the LoRa Alliance, with the exceptions defined in Section 6 of this document. The communication may be bidirectional or unidirectional (from the mote to the gateway).

A mote is admitted to the network via one of three mechanisms:

Personalization: The mote must be configured in both the network server and the application server.

Auto-creation: If auto-creation is enabled, any mote using the configured default authentication key is admitted to the network and will use the configured default encryption key

Over the air (OTA): The mote must be configured in the application server, with its EUI, the EUI of the application to which it is assigned and a fixed 'appKey', unique to the mote. When the mote enacts the OTA join protocol, the network server (NS) allocates a network address to it. The application server (AS) generates the mote's authentication session key and its encryption session key and forwards the authentication session key to the NS.

Gateway

Each LoRa gateway forwards data between motes and a single LoRa network server. The communication protocol is defined by [2].

Network Server (NS)

The LoRa network server maintains a record for each mote. The record contains the following information:

- The EUI of the mote
- The EUI of the application to which the mote is assigned
- The LoRa network address assigned to the mote
- The sequence number of the next LoRa frame expected from the mote
- The sequence number of the next LoRa frame to be transmitted to the mote
- The authentication session key assigned to the mote

The network server authenticates the received frame and forwards user data to an application server. The received frame is transported from the Gateway to the NS using JSON/GWMP/UDP/IP (defined in [2]). The frame is forwarded to an AS using JSON/TCP/IP (defined in [3]).

The network server adds a cryptographic hash to all LoRa frames transmitted to the LoRa mote. The hash algorithm is defined by the LoRa Alliance publication ‘LoRaWAN Specification’ [1].

A single network server may be connected to many application servers and network controllers. The remote server or controller used for a given mote is determined by the application to which the mote is assigned.

Application Server (AS)

The LoRa application server is responsible for admitting OTA motes to the network and for encrypting user data sent to, and decrypting user data received from, the mote.

A single application server may be connected to many network and customer servers. The remote server or controller used for a given mote is determined by the application to which the mote is assigned.

The LoRa application server maintains a record for each mote. The record contains the following information:

- The EUI of the mote
- The EUI of the application to which the mote is assigned
- The encryption session key assigned to the mote

The LoRa application server decrypts the received user data and forwards it to a customer server. It also encrypts downstream user data before forwarding it to the NS. The encryption algorithm is defined by the LoRa Alliance publication ‘LoRaWAN Specification’ [1].

Network Controller (NC)

The network controller receives the transmission parameters used by the mote and characteristics of the signal received by the gateway for each frame. It may perform operations using that data. The NC supplied by Semtech may be set to control mote data rate (ADR); it performs no other operation at present.

A single network controller may be connected to many network servers. The remote server or controller used for a given mote is determined by the application to which the mote is assigned.

Customer Server

The customer server is a trivial implementation of the program used by the data owner to receive mote data. It is expected that a user would substantially replace the customer server in order to pass data into his network in the required format.

The provided customer server can be set to store the received information in its relational database, append it to an ASCII file or both.

A single customer server may be connected to many application servers. The remote server or controller used for a given mote is determined by the application to which the mote is assigned.

3 LoRa communication paths

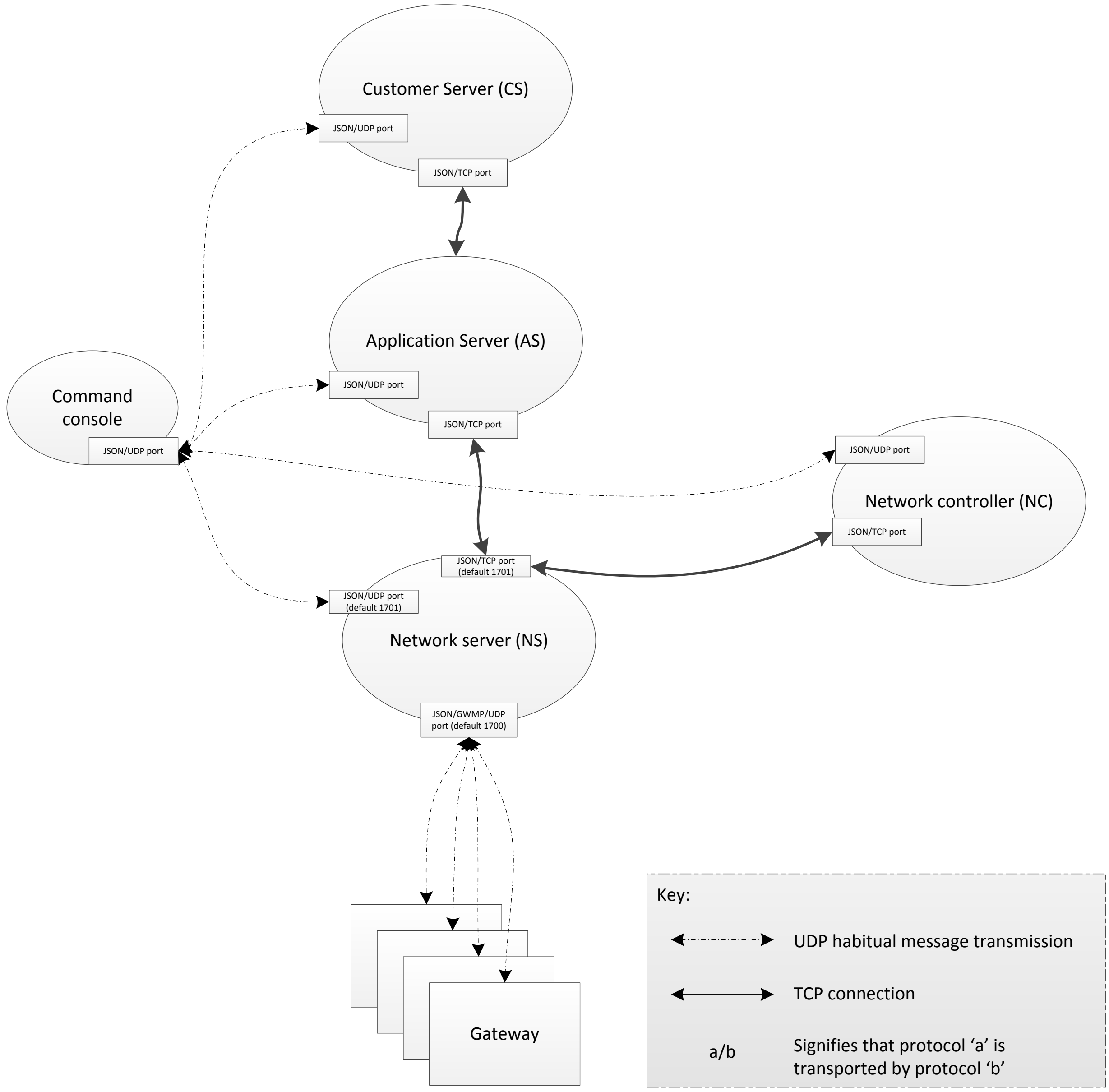

Figure 2 LoRa system communication diagram

Communication between the Gateways and the LoRa network server is via JSON/GWMP/UDP/IP (defined by [2]).

Communication to and from the LoRa command console is via JSON/UDP/IP (defined by [3]).

Communication between any pair of LoRa servers (including the LoRa Network Controller) is via JSON over TCP over IP (defined by [3]).

The network server maintains two UDP ports. By default, it receives JSON/GWMP/UDP/IP on UDP Port 1700. It always receives JSON/UDP/IP and JSON/TCP/IP on a port number that is one greater than the port on which it receives JSON/GWMP/UDP/IP; this means that the port number's default is 1701.

3.1 Message protocol layers

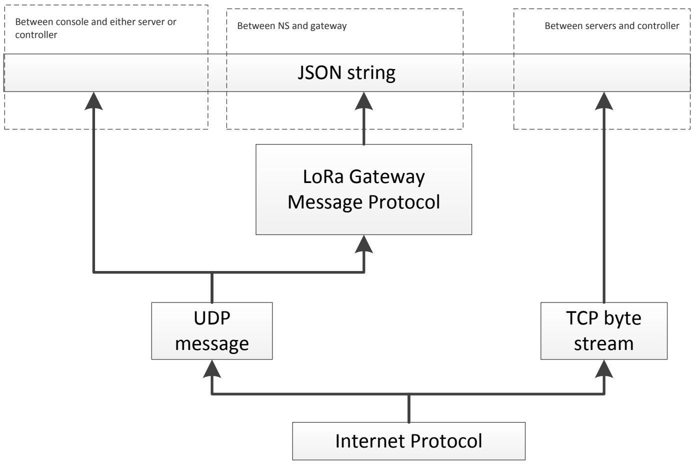
Between console and either server or controller
Between NS and gateway
Between servers and controller
JSON string
LoRa Gateway Message Protocol
UDP message
TCP byte stream
Internet Protocol

Figure 3: LoRa server protocol stacks

3.1.1 LoRa gateway message protocol (GWMP) overview

The gateway message protocol layer is defined by [2].

In the upstream direction, the gateway sends a PUSH_DATA message. The PUSH_DATA message contains a single JSON top-level object.

The server immediately acknowledges receipt of a PUSH_DATA message with a PUSH_ACK message, transmitted to the source port of the PUSH_DATA message.

The gateway periodically transmits a PULL_DATA message. The destination port of the message is always the GWMP port of the NS (default value 1700). The message is periodically transmitted by the gateway to the NS in order to keep intervening IP firewalls open.

The server immediately acknowledges receipt of a PULL_DATA message with a PULL_ACK message.

The NS transmits a PULL_RESP message when it wishes to command a gateway to transmit a frame to a mote. A PULL_RESP message contains a single JSON top-level object and is sent from the server to the gateway. The destination UDP port is the source port of the most recently received PULL_DATA message.

3.1.2 JSON string

The JSON data interchange format is defined by [4]. The JSON objects used by the server are defined in [2] and [3].

A LoRa JSON top level object is encoded into an ASCII text string.

4 Adding a mote to the LoRa network

A mote may be added to the LoRa network in three ways:

Personalized: The mote details are configured on NS, AS, NC and the CS. This method provides little protection from attack as the authentication and encryption session keys cannot be changed without reconfiguring the mote. It is designed for use in a low threat environment.

Auto creation: All motes use default session keys that are configured in the NS (authentication session key) and AS (application session key). This method provides little protection against malice but does, if other nearby LoRa systems use different keys, provide protection against accidental interference.

Over the air (OTA): The mote and the AS are configured with the mote EUI, mote application EUI and the application key (appKey). The mote authentication session key and mote encryption session key are independently generated by the mote and the AS. This method provides significant protection against attack.

Attribute	Bits	Purpose	Mote type		
			Personalized	Auto discovered	Over the air
Mote EUI	**64**	**Identifies mote globally**	**Always equal to the network address of the mote**[1]	**Always equal to the network address of the mote**	**Received from mote and also configured in AS and NS**
Application EUI	64	Identifies client application	Always zero	Always zero	Received from mote and also configured in AS and NS
Mote network address	32	Identifies mote within its network server's geographic area	Configured	Received from mote	Assigned by NS
App key	128	Shared secret, in OTA, the App key is used by the AS and the mote to generate identical encryption and authentication session keys	Not used	Not used	Configured
Encryption session key	128	Shared secret, used to encrypt and decrypt application data	Configured	A single value configured in the AS and common to all motes, is used	Generated by AS from App nonce, Device nonce, App Key and Network Address
Authentication session key	128	Shared secret, used to authenticate frames transmitted to and received from the mote	Configured	A single value configured in the NS and common to all motes, is used	Generated by NS from App nonce, Device nonce, App Key and Network Address by AS Passed by AS to NS.
App nonce	24	Random value, used in the generation of OTA session keys	Not used	Not used	Generated by AS
Device nonce	16	Random value, used in the generation of OTA session keys	Not used	Not used	Generated by mote

Table 1: Table showing use and source of data when a mote joins a network

[1] This requires that the more significant 32 bits of the Mote EUI are zero.

5 Service type

A server provides one or more 'services' to an application.

The recognized services are:

user: The remote server should receive upstream user (payload) data

motetx: The remote server should receive metadata concerning the characteristics (for example frequency and coding rate) of transmissions emitted by each mote.

gwrx: The remote server should receive metadata concerning the characteristics (for example signal strength and signal to noise ratio) of the frames received from each mote.

joinserver: The remote server will provide the 'application server' part of the OTA (join) protocol for admitting a mote to the network.

joinmonitor: The remote server does not participate in the OTA join protocol which admits a mote to the network but wishes to be informed of motes joining the network.

downstream: The remote server is closer, topologically, to the motes than is this server. This allows downstream data that this server must sent to a mote to be sent to the remote server for forwarding to the mote.

maccmd: The remove server should receive the content of any header extension fields

gwst: The remote server should receive gateway status information

6 Compliance to the LoRa WAN protocol [1]

The current version of the LoRa server complies only to the EU 863-870MHz ISM physical layer of the LoRa WAN protocol.

The spreading factor used by the server when commanding the gateway to transmit to the 2nd mote reception window is SF9, while the EU 863-870MHz ISM physical layer of the LoRa WAN protocol specifies SF12.

The NS requires that RECEIVE_DELAY1 (the delay from the end of a mote's transmission of a data frame until the opening of the mote's first receive window) to be 1 second.

The NS requires that RECEIVE_DELAY2 (the delay from the end of a mote's transmission of a data frame until the opening of the mote's second receive window) to be 2 seconds.

The NS requires that JOIN_ACCEPT_DELAY1 (the delay from the end of a mote's transmission of a 'join request' frame until the opening of the mote's second receive window) to be 5 seconds.

7 Glossary

'/': The construct 'a/b' is used when Protocol 'a' is transported by Protocol 'b'.

ADR: Adaptive Data Rate. ADR observes the quality of the signal received by the mote and changes the mote's spreading factor and transmit power in order to optimise the time and energy required for the mote to transmit a frame.

Application: An application is identified by an 'application EUI'. Each mote is assigned to a single application. The remote server or servers to which information is forwarded (for example the AS to which an NS forwards are received frame) are configured for each application.

AS: The LoRa application server

ASCII: American Standard Code for Information Interchange. A widely used standard for representing Latin text, Arabic numerals and punctuation as binary values.

Base64: A method of encoding binary data into ASCII text. The LoRa system uses Base64 to transport LoRa frames in JSON objects. Base64 is defined by IETF RFC 4648 [5].

Command Console: The LoRa command console is a program that allows a user to configure LoRa servers.

Cryptographic hash: The generation of a hash code using a key which is known only to the sender and receiver or receivers. The transmission and recalculation of a cryptographic hash can be used to verify that the message content has not changed.

CS: The LoRa customer Server

dB: decibel; a logarithmic ratio of power. Defined by Bell Laboratories

dBm A logarithmic measure of power, decibel, relative to 1mW

Downstream: Toward the mote

End-device: Synonymous with 'mote'

EUI: Extended Unique Identifier. In this document 'EUI' refers to a value from the 'EUI-64' number space managed by the IEEE.

Firewall: A firewall is a network security system that controls the incoming and outgoing network traffic based on an applied rule set. A firewall establishes a barrier between a trusted, secure, internal network and another network (e.g., the Internet) that is assumed not to be secure and trusted.

Gateway: A LoRa gateway is transmits LoRa frames to, and receives LoRa frames from, LoRa motes

GWMP: Gateway message protocol. The protocol used the transport JSON objects between the network server and the gateways. Defined by [2].

IEEE: Institution of Electrical and Electronic Engineers (www.ieee.org).

IETF: Internet Engineering Task Force (www.ietf.org).

IP: Internet Protocol

IP port address An IP address or host name and either a UDP or a TCP port number. This document represents a port address in the form <IP address>:<port number> or <host name>:<port number>. E.g. 1.2.3.4:4500 or a.com:4500.

Join: A colloquial name for 'Over The Air' activation.

Join request frame: A LoRa frame sent as the initial part of the OTA activation protocol. The frame contains the mote's EUI, its application's EUI and a nonce (a 16 bit random number).

Join accept frame A LoRa frame sent as the concluding part of the OTA activation protocol. The frame contains the mote's LoRa network address, its network Id and the application nonce (a 24 bit random number generated by a server).

JSON: JavaScript Object Notation. JSON is a textual based method of representing name, value pairs. The value of an object may itself be a JSON object. Within LoRa, JSON objects contain only ASCII characters. It is defined by [4].

JSON object A JSON name, value pair

Key: In cryptography, a key is a piece of information (a parameter) that determines the functional output of a cryptographic algorithm or cipher. Without a key, the algorithm would produce no useful result.

LoRa: Long Range. Defined by the LoRa Alliance

LoRa Alliance: The industry body that defines the LoRaWAN protocol. (http://lora-alliance.org/)

LoRa port: Any user data transmitted to or received from the mote is associated with a 'port' number. User data to or from LoRa Port 0 is MAC command or MAC status data. The remaining 255 LoRa port values are available to the mote user.

LoRaWAN: The protocol by which a LoRa mote communicates with a LoRa gateway. LoRaWAN is defined by the LoRa Alliance [1].

MAC: Media access control

MAC command: A command transmitted to the mote. A MAC command is transmitted to the mote either in the LoRa frame 'header option' area or as user data to LoRa Port 0. Multiple commands may be transmitted in a single frame.

MAC status: Status information received from the mote. A MAC status message is transmitted by the mote either in the LoRa frame 'header option' area or as user data from LoRa Port 0. Multiple status messages may be transmitted in a single frame.

Metadata: LoRa Metadata refers to information about the transmission or reception of a LoRa frame.

Mote: A LoRa end device. A LoRa mote communicates with a LoRa Gateway using the LoRa MAC or LoRa WAN protocol.

NC: The LoRa network controller

Network id: The 'network id' of a mote is its 'network address' shifted right by 25 bits, leaving 7 bit value.

Network address: The LoRa network address is a 32 bit value contained in the LoRa frame that identifies its source or destination mote. The network address need be unique only within the transmission range of a mote or gateway and is distinct from the mote EUI.

NS: The LoRa network server

OTA: Over the Air

Over the air: One of two methods of adding a LoRa mote to a LoRa network. In the OTA method, the mote is configured with a mote EUI, an application EUI and a 128 bit cypher key ('appKey'). Handshaking between the mote and the LoRa servers causes a 32 bit LoRa network address and two 128 bit session keys to be generated. One session key (the 'authentication' key) is known to the mote and the NS. The other (the 'encryption' key) is known to the mote and the AS.

Process: A running computer program. A process cannot access the memory used by another. Processes are started and stopped independently of others.

Personalization: One of two methods of adding a LoRa mote to a LoRa network. The mote is configured with its network address and its authentication and encryption keys. The mote's EUI is always equal to its network address and the application EUI is always zero.

Provisioning: A synonym for 'personalization'

RSSI: Received Signal Strength Indication. The power of the received signal, normally measured in dBm.

Rx: Receive

Signal quality: The signal quality is normally measured in dBm and is the sum of the SNR (measured in dB) and the RSSI (measured in dBm).

SNR: Ratio of signal power to noise power, normally measured in dB.

Spreading factor: A parameter of a LoRa transmission. Two to the power of 'spreading factor' 'on the air' bits are transmitted to represent each frame bit.

TCP: Transmission Control Protocol. A connection based protocol for transporting a sequence of bytes. While the connection exists, the content is guaranteed to be delivered in order and without loss or corruption.

Transform:	An element of a data flow diagram that transforms its inputs to generate one or more outputs (http://en.wikipedia.org/wiki/Data_flow_diagram)
Tx:	Transmit
UDP:	User Datagram protocol: a simple protocol for transporting data packets. Delivery is not guaranteed. In addition the order of receipt is not necessarily the same as the order of transmission.
upstream:	Away from the mote
UTC	Co-ordinated Universal Time; also known as Greenwich Mean Time and Zulu

8 References

Each trademark is the property of its owner.

[1] LoRa Alliance, “LoRaWAN Specification,” LoRa Alliance, 2015.

[2] Semtech Ltd, “LoRaWAN Network Server Demonstration: Gateway to Server Interface Definition,” 2015.

[3] Semtech Ltd, “LoRaWAN Network Server Demonstration: Inter-Server interface definition,” 2015.

[4] ECMA International, The JSON Data Interchange Format, 2013.

[5] IETF, “The Base16, Base32, and Base64 Data Encodings,” October 2006. [Online]. Available: https://www.ietf.org/rfc/rfc4648.txt.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent or other industrial or intellectual property rights. Semtech assumes no responsibility or liability whatsoever for any failure or unexpected operation resulting from misuse, neglect improper installation, repair or improper handling or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified range.

SEMTECH PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF SEMTECH PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE UNDERTAKEN SOLELY AT THE CUSTOMER'S OWN RISK. Should a customer purchase or use Semtech products for any such unauthorized application, the customer shall indemnify and hold Semtech and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs damages and attorney fees which could arise.

Contact Information

Semtech Corporation
Wireless Sensing and Timing Products Division
200 Flynn Road, Camarillo, CA 93012
Phone: (805) 498-2111 Fax: (805) 498-3804
E-mail: support_rf_na@semtech.com
Internet: http://www.semtech.com