
LoRaWAN Network Server Demonstration: Installation Guide

1	History.....	3
2	Introduction.....	3
3	Preparation	3
4	Installing MySQL and PHP server.....	3
4.1	Linux Platforms.....	3
4.2	Microsoft Windows Platforms.....	3
5	Configuring the database.....	4
5.1	Creating database structure.....	4
5.2	Creating the database users required by the LoRa server	5
5.2.1	Delete conflicting users	5
5.2.2	Create users required by LoRa servers.....	5
5.2.3	Create users required by LoRa web server	6
6	Configuration.....	7
7	Assumed LoRa server configuration	8
8	Starting the LoRa servers	8
8.1	Parameters	8
8.2	Linux	9
8.3	Windows.....	9
8.4	Starting the command console	10
8.4.1	Linux	11
8.4.2	Windows.....	11
8.5	Stopping the command console.....	11
9	Example configuration.....	12
9.1	General configuration	12
9.2	Creation of applications	12
9.3	Connection of application servers.....	13
9.4	Configuration of non-application servers.....	14
9.5	Create personalised or provisioned motes	14
9.6	Create over the air motes	14
10	Customising the Customer Server.....	15
11	Glossary	16
12	References	19

1 History

Revision	Modification / Remarks / Motive	Author
1.0	Document created	DRo

2 Introduction

The document guides the user to install and configure the Semtech LoRa servers, Version 2.1.x and configure a (separately obtained) MySQL database server.

3 Preparation

3.1 Linux

Find the executable files loraNS loraNC loraAS loraCS loracmd and copy them to the directory that will be used as the working directory of the LoRa servers.

3.2 Windows

Find the executable files ApplicationServer.exe, CustomerServer.exe, Console.exe, NetworkController.exe, NetworkServer.exe and copy them to the directory that will be used as the working directory of the LoRa servers.

4 Installing MySQL and PHP server

Download the XAMPP MySQL and PHP server from <https://www.apachefriends.org/index.html>

4.1 Linux Platforms

Install XAMPP

4.2 Microsoft Windows Platforms

Disable Windows UAC using the Windows 7 control panel, using the instructions in <http://windows.microsoft.com/en-us/windows/turn-user-account-control-on-off#1TC=windows-7>

Install XAMPP

Re-enable Windows UAC using the Windows 7 control panel.

Copy the libmysql.dll file from the location in which it was installed by the XAMPP installation package to the directories in which the LoRa server programs will be stored.

If the server programs were not compiled and linked on the computer on which they are to run, download and install the 'Microsoft Visual C++ 2010 Redistributable Package (x86)' <http://www.microsoft.com/en-us/download/details.aspx?id=5555>

5 Configuring the database

5.1 Creating database structure

Import the database file 'DatabaseStructure.sql'.

The import may be done using the phpMyAdmin web page (<http://localhost/phpmyadmin>), if it is installed.

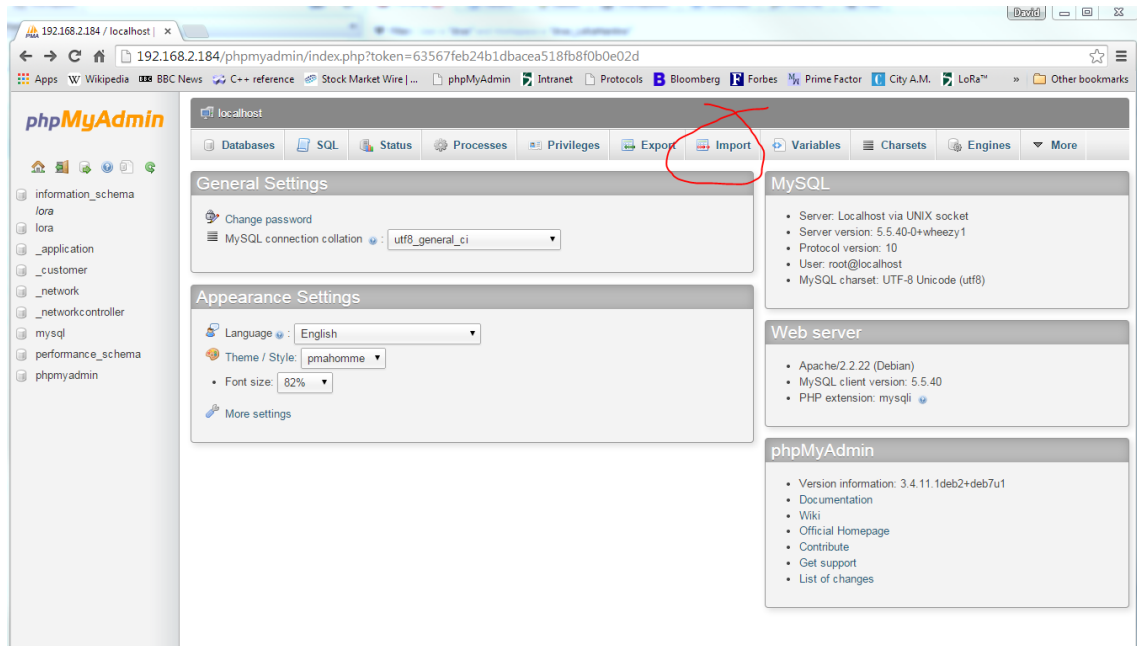


Figure 1 php MyAdmin web page, showing import command

The import can also be done using the command line command

```
mysql -u <USERNAME> -p < DatabaseStructure.sql
```

The '-p' parameter informs the 'mysql' program that user <USERNAME> requires a password. The program will prompt for the password.

5.2 Creating the database users required by the LoRa server

The length of a user name or password is normally limited by the database server (e.g. MySQL) to a maximum of 16 characters.

5.2.1 Delete conflicting users

Some installations of MySQL contain default users.

When a log in request is received, the user making the request is compared sequentially with the last of acceptable users. If the default user is tested before the user identity used by the LoRa server, the server's login request will fail.

Delete the following users, if they exist:

- user name: 'Any' host: 'anyhost' (or '%')
- user name: 'Any' user 'localhost'

5.2.2 Create users required by LoRa servers

Each LoRa server requires SELECT, INSERT, UPDATE, DELETE privileges on the tables of its database. If the database structure needs to be updated to match the current server version, each LoRa server also requires CREATE, DROP, INDEX, ALTER privileges on its database.

The LoRa web server requires read privilege to the same tables. This section grants both users privileges to all databases on the server. An implementation may wish to grant more restricted privilege.

This section assumes that all[†] the LoRa servers use the name 'loraserver' and the password 'lorapass'. The actual values must match those used by the servers' '-dbuser' and '-dbpass' command line parameters, as described in Section 8.1.

[†] It is possible for each LoRa server to use an individual username (and password) to access its own database. For example the network server and the 'lora_network' database can be configured to use a username that is different to that used by the application server and the 'lora_application' database.

Execute the following SQL commands[‡]:

```
CREATE USER "lora_server"@localhost IDENTIFIED BY "loraPassword";
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER ON lora_application.* TO
lora_server@localhost IDENTIFIED BY "loraPassword" WITH MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER ON lora_customer.* TO
lora_server@localhost IDENTIFIED BY "loraPassword" WITH MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER ON lora_network.* TO
lora_server@localhost IDENTIFIED BY "loraPassword" WITH MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER ON lora_networkcontroller.*
TO lora_server@localhost IDENTIFIED BY "loraPassword" WITH MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
```

5.2.3 Create users required by LoRa web server

5.2.3.1 Read

This section assumes that the LoRa web server uses the name 'loraweb_r' and the password 'lorawebpass_r'. The values used by the Semtech demonstration web server are held in the constants DB_USER and DB_PASSWORD in the files misc/constants_database.php and html/map/localconstants.php.

- CREATE USER 'loraweb_r'@'%' IDENTIFIED BY 'lorawebpass_r';
GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'loraweb_r'@'%' IDENTIFIED BY 'lorawebpass_r' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;

5.2.3.2 Read/write

This section assumes that the LoRa web server uses the name 'loraweb_rw' and the password 'lorawebpass_rw'. The values used by the Semtech demonstration web server are held in the constants DBW_USER and DBW_PASSWORD in file misc/constants_database.php.

- CREATE USER 'loraweb_rw'@'%' IDENTIFIED BY 'lorawebpass_rw';
GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'loraweb_rw'@'%' IDENTIFIED BY 'lorawebpass_rw' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;

[‡] An incorrect user may be removed by executing the command `user`@`localhost`;

5.2.3.3 Table create/drop

This section assumes that the LoRa web server uses the name 'loraweb_admin' and the password 'lorawebpass_ad'. The values used by the Semtech demonstration web server are held in the constants DB_USER and DB_PASS in file misc/databasePopulator.php.

- CREATE USER 'loraweb_admin'@'%' IDENTIFIED BY 'lorawebpass_ad';
GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'loraweb_admin'@'%' IDENTIFIED BY 'lorawebpass_ad' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;

6 Configuration

Configuration of the LoRa servers must be via the LoRa 'command console'. The LoRa 'command console' syntax is described in 'LoRa server configuration command description'.

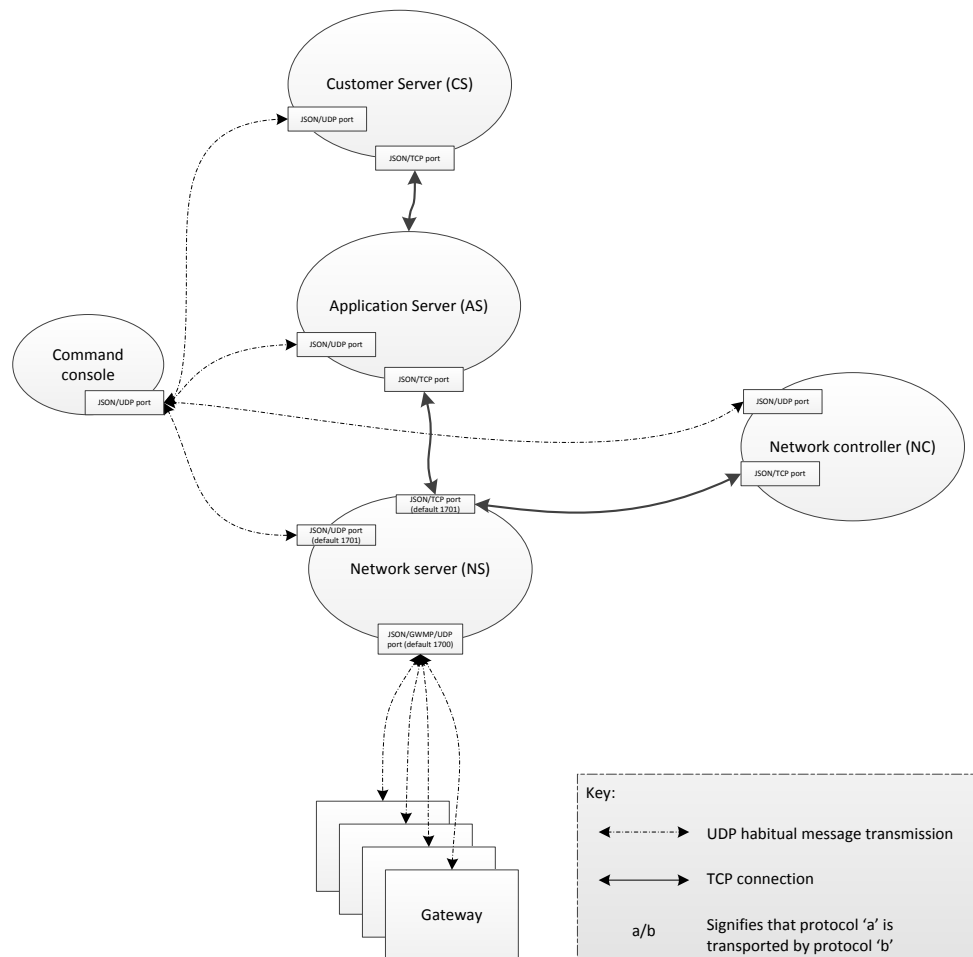


Figure 2: LoRa system communication diagram

7 Assumed LoRa server configuration

The following LoRa server configuration is assumed:

Server	Address
NS	127.0.0.1:1701
AS	127.0.0.1:4000
CS	127.0.0.1:5000
NC	127.0.0.1:6000

Table 1 Assumed LoRa server port addresses

In the 'Windows' version, the LoRa servers' working directory is assumed to be 'C:\run'.

8 Starting the LoRa servers

8.1 Parameters

The server command line parameters are fully described in [1] but the remainder of this sub-section provides a short explanation of the ones recommended in this guide.

The parameter '-dbuser' is required; it sets the name of the database user that the server uses to connect to its relational database.

The parameter '-dbpass' is optional; it sets the password of the database user. The parameter is optional but, if it is omitted, the user will be prompted to enter it.

The command line parameter '-major' sets a server to print only major errors. Replacing it with the parameter '-verbose' causes a server to print more information and may assist during debugging.

The '-port' parameter sets the port on which the AS, CS and NC listen. It is required on the AS, CS and NC. If used on the NS, it sets the UDP port on which the NS receives GWMP messages to that value and the TCP and UDP port on which the NS receives JSON messages to one greater.

The '-netid' parameter sets the range from which the NS selects network addresses.

The '-log' parameter sets the name of the log file.

8.2 Linux

The following commands must be executed. Their order is unimportant:

- `nohup ./loraNS -dbuser <DB user> -dbpass <pass> -noconsole -major -netid 24 -log NSlog.txt &`
- `nohup ./loraAS -dbuser <DB user> -dbpass <pass> -noconsole -major -port 4000 -log ASlog.txt &`
- `nohup ./loraCS -dbuser <DB user> -dbpass <pass> -noconsole -major -port 5000 -log CSlog.txt &`
- `nohup ./loraNC -dbuser <DB user> -dbpass <pass> -noconsole -major -port 6000 -log NClog.txt &`

The parameters are described in Section 8.1 and more fully in [1].

'nohup' is a Linux command. It starts a process but does not terminate when the process's parent processor terminates.

'&' is a Linux command switch that runs a process in the background

For debugging, it is wise to:

- start each server in its own command window
- omit the 'nohub' prefix
- omit the ampersand suffix ('&')
- replace the '-noconsole' parameter with '-console'
- replace the '-major' parameter with '-verbose'

This would, for the network server, result in a command:

```
./loraNS -dbuser <DB user> -dbpass <DB pass> -verbose -netid 24 -log NSlog.txt
```

8.3 Windows

Ensure that the file 'libmysql.dll' has been copied into the directory in with the server executable files are held. This is described in Section 4.2.

The following commands must be executed. Their order is unimportant:

```
start /MIN /B /D C:\run§ NetworkServer -dbuser <DB user> -dbpass <pass>  
-noconsole -major -netid 24 -log NSlog.txt
```

```
start /MIN /B /D C:\run C:\run\ApplicationServer -dbuser <DB user> -dbpass <pass>  
-noconsole -major -port 4000 -log ASLog.txt
```

```
start /MIN /B /D C:\run C:\run\CustomerServer -dbuser <DB user> -dbpass <pass>  
-noconsole -major -port 5000 -log CSLog.txt
```

```
start /MIN /B /D C:\run C:\run\NetworkController -dbuser <DB user> -dbpass <pass>  
-noconsole -major -port 6000 -log NClog.txt
```

The parameters are described in Section 8.1 and more fully in [1].

The 'start /MIN /B' command is a Windows command that starts the program and runs it in the background. The /D switch sets the program's working directory.

For debugging, it is wise to:

- start each server in its own command window
- omit the 'start /B /D C:\run' prefix
- replace the '-noconsole' parameter with '-console'
- replace the '-major' parameter with '-verbose'

This would, for the network server, result in a command:

```
NetworkServer -dbuser myusername -dbpass mypassword -console -verbose -netid 24 -log  
NSlog.txt
```

[§] The LoRa servers' working directory is assumed to be C:\run, but need not be. If the command is executed from the servers' working directory, the '/D' option and its parameter may be omitted.

8.4 Starting the command console

If the command console is to be used on a different computer to one or more of the servers, a command console must first be started on each computer that is running a LoRa server and the command "set allowremotefconfiguration on" executed for each server on which this operation is wanted.

8.4.1 Linux

```
./loracmd -ns<NS address> -as <AS port address> -cs <CS port address> -nc <NC port address>
```

e.g. ./loracmd -ns 127.0.0.1 -as 127.0.0.1:4000 -cs 127.0.0.1:5000 -nc 127.0.0.1:6000

8.4.2 Windows

```
Console -ns<NS address> -as <AS port address> -cs <CS port address> -nc <NC port address>
```

e.g. Console -ns 127.0.0.1 -as 127.0.0.1:4000 -cs 127.0.0.1:5000 -nc 127.0.0.1:6000

8.5 Stopping the command console

The console may be terminated by entering Control-C.

9 Example configuration

The commands in this section, with the exception of the 'server select commands (i.e. 'ns', 'as', 'cs', 'nc') can be written to a file and actioned by entering the command 'include <filename>' in the command console.

The following section assumes that all servers are on the same host, allowing the use of the 127.0.0.1 addresses, and the port addresses are as listed in Section 7.

9.1 General configuration

```
ns                                #select NS
set motereceivewindow 0          # set all motes to receive on first window
set                               #reads the global configuration values

nc
set autcreatemotes on            #set NC to create mote records when it sees them
set                               #reads the global configuration values
```

9.2 Creation of applications

#add the default (zero) application and Application 00-00-00-00-00-aa-bb-cc to each server

```
ns                                #select NS
app add 00-00-00-00-00-00-00-00 defaultApp
app add 00-00-00-00-00-aa-bb-cc testApp testOwner
app list full
```

```
as
app add 00-00-00-00-00-00-00-00 defaultApp
app add 00-00-00-00-00-aa-bb-cc testApp testOwner
app list full
```

```
cs
app add 00-00-00-00-00-00-00-00 defaultApp
app add 00-00-00-00-00-aa-bb-cc testApp testOwner
app list full
```

```
nc
app add 00-00-00-00-00-00-00-00 defaultApp
app add 00-00-00-00-00-aa-bb-cc testApp testOwner
app list full
```

9.3 Connection of application servers

The servers must be connected. A typical server network comprises a single network server (NS) connected to multiple application servers (AS). Each application server may be connected to multiple customer servers (CS).

A connection between a pair of servers will serve the services of one or more applications (identified by the application EUI). A 'service' is a type of data – the available services are described in [1] 'LoRa server configuration command description'.

```
# The following commands specify the remote servers that provide services (e.g. 'user' or 'motetx')
# to the applications served by the currently selected server
# 'active' specifies that the server attempts to connect to the remote server
# 'passive' specifies that the server waits for the remote server to connect
```

ns

```
app server add 00-00-00-00-00-00-00-00 127.0.0.1:4000 active user motetx gwrx joinserver
app server add 00-00-00-00-00-00-00-00 127.0.0.1:6000 active motetx gwrx
app server add 00-00-00-00-00-aa-bb-cc 127.0.0.1:4000 active user motetx gwrx joinserver
app server add 00-00-00-00-00-aa-bb-cc 127.0.0.1:6000 active motetx gwrx
app list full
```

as

```
app server add 00-00-00-00-00-00-00-00 127.0.0.1:5000 active user motetx gwrx joinmonitor
app server add 00-00-00-00-00-00-00-00 127.0.0.1:1701 passive downstream
app server add 00-00-00-00-00-aa-bb-cc 127.0.0.1:5000 active user motetx gwrx joinmonitor
app server add 00-00-00-00-00-aa-bb-cc 127.0.0.1:1701 passive downstream
app list full
```

cs

```
app server add 00-00-00-00-00-00-00-00 127.0.0.1:4000 passive downstream
app server add 00-00-00-00-00-aa-bb-cc 127.0.0.1:4000 passive downstream
app list full
```

nc

```
app server add 00-00-00-00-00-aa-bb-cc 127.0.0.1:1701 passive downstream
app server add 00-00-00-00-00-00-00-00 127.0.0.1:1701 passive downstream
app list full
```

9.4 Configuration of non-application servers

The following commands specify the remote servers that provide services
that are not related to a particular applications
'active' specifies that the NS attempts to connect to the remote server

```
ns
server add 127.0.0.1:6000 active gwst
app list full
```

9.5 Create personalised or provisioned motes

Create provisioned Mote 00-30 using:
Application zero (The application of a provisioned mote is fixed in this version of the server)
Encryption session key of 7e:04:6b:35:8e:36:6e:6a:ef:f5:c1:bc:61:bc:24:d6
Authentication session key of 8f:a8:34:f6:d1:fd:66:51:9c:c8:4e:af:72:6b:b9:b0

```
as
mote add 00-00-00-00-00-00-30-00 7e:04:6b:35:8e:36:6e:6a:ef:f5:c1:bc:61:bc:24:d6
mote list
ns
mote add 00-00-00-00-00-00-30-00 8f:a8:34:f6:d1:fd:66:51:9c:c8:4e:af:72:6b:b9:b0
mote list
nc
mote add 00-00-00-00-00-00-30-00 00-00-00-00-00-00-00-00
mote list
cs
mote add 00-00-00-00-00-00-30-00 00-00-00-00-00-00-00-00
mote list
```

9.6 Create over the air motes

Create provisioned Mote 00-00-fa-78-9a-af-8f-27 using:
Application 00-00-00-00-00-aa-bb-cc
Application Key (AppKey) 0c:2b:4f:12:d4:ea:ea:2e:a4:5f:6a:e6:35:00:3f:49

```
as
mote add 00-00-fa-78-9a-af-8f-27 00-00-00-00-00-aa-bb-cc 0c:2b:4f:12:d4:ea:ea:2e:a4:5f:6a:e6:35:00:3f:49
mote list
```

#These commands are ONLY required if the NC and CS do not have autocreatemotes on (Section 9.1)

```
nc
mote add 00-00-fa-78-9a-af-8f-27 00-00-00-00-00-aa-bb-cc
mote list
```

```
cs
mote add 00-00-fa-78-9a-af-8f-27 00-00-00-00-00-aa-bb-cc
mote list
```

10 Customising the Customer Server

The customer server function

```
void ApplicationDataOutput::UpstreamApplicationDataReceived(EuiType moteEui, uint32  
sequenceNumber, LoRa::FrameApplicationData const& payload, MoteTransmitRecord const&  
transmitRecord, GatewayReceiveList const& gatewayReceiveList)
```

is called when the CS receives data from a mote. It should be rewritten to allow the user to define the future course of the data.

payload.Length() returns the number of byte in the payload

payload.Data() returns a (const) pointer to the first byte of the data

payload.Port() returns the LoRa port number from which the mote when sent the data

The application data may also be accessed using 'payload' as an array of unsigned characters.

11 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
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
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.
Gateway:	A LoRa gateway is transmits LoRa frames to, and receives LoRa frames from, LoRa motes
GNSS:	Global Navigation Satellite System. The most well-known GNSS is GPS.
GPS:	Global Positioning System. A Global Navigation Satellite System.
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 its device-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 its application nonce (a 24 bit random number).
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 [3].
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://loro-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 will communicate with a LoRa gateway. LoRaWAN is defined by the LoRa Alliance [4].
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.
MySQL:	MySQL is an open source database engine available from http://www.mysql.com/
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.
Provisioning:	A synonym for 'personalization'
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'
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.
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

12 References

Each trademark is the property of its owner.

- [1] Semtech Ltd, “LoRaWAN Network Server Demonstration : Configuration Command Definitions,” 2015.
- [2] Semtech Ltd, “LoRaWAN Network Server Demonstration: Gateway to Server Interface Definition,” 2015.
- [3] ECMA International, The JSON Data Interchange Format, 2013.
- [4] LoRa Alliance, “LoRaWAN Specification,” LoRa Alliance, 2015.

© Semtech 2015

All rights reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. 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>