IoT over LoRaWAN Protocol

Supported device types and versions

The protocol allows communication with devices communicating using the LoRaWAN protocol. The LoRaWAN protocol is a protocol designed for occasional, energy-efficient wireless transmission of a small amount of data called **payload** (typically several bytes) over long distances between linked objects - in LoRaWAN terminology called *mote* (typically battery-powered sensors) and LoRaWAN gateway.
The gateway can then communicate directly with the D2000 COM process or send the payload to the network servers or to the cloud, where data processing is performed (deduplication, filtration), followed by the data sent to the D2000 COM process. In any case, the payload is packed in an **envelope** (e. g. such as a field in a JSON message or in a CSV file) and transferred up to the D2000 COM process. The payload is decoded there (using Base64 encoding or Base64 + Base16 encoding) and processed.
Payload processing is dependent on device type (defined in parameter of Device Type protocol).

The communication was tested between sensors and LoRaWAN gateway Kerlink IoT Station 868. It was, depending on the installed firmware:

- sending data directly to the D2000 COM process (JSON messages in UDP packets)
- sending data to the cloud TheThings.Network, from where it was read by D2000 KOM process (JSON messages in MQTT protocol through TCP connection)
- sending data to the cloud Loriot.io, which was forwarding them via MQTT to iot.eclipse.org, from where it was read by D2000 KOM process (JSON messages in MQTT protocol through TCP connection)

Communication line configuration

- Communication line category: SerialOverUDP Device Redundant, TCP/IP-TCP.
- Configuration of SerialOverUDP Device Redundant line:
  - Local port: port, where the D2000 KOM process receives UDP packets
  - Primary / Backup Device: IP address LoRaWAN gateway
    (pre Connection Type=*Kerlink IoT Station SPN*)
  - Port: port, where the LoRaWAN gateway receives UDP packets (currently unused, as there is no implemented record)
- Configuration of TCP/IP-TCP line:
  - Host: IP address of server, to which the D200 KOM process connects or redundant addresses separated by a comma or semicolon
    (for Connection Type=*MQTT client*)
  - Port: server port, to which the D200 KOM process connects

Communication line configuration

Communication line - configuration dialog - **Protocol parameters** tab.
The parameters influence some optional protocol parameters. The following protocol line parameters can be used:

Table 1

Parameter	Description	Unit / size	Default Value
Connection Type	Type of connection between D2000 KOM process and other party (LoRaWAN gateway, network server, cloud). Currently supported are: • **Kerlink IoT Station SPN (JSON via UDP packets)**: communication with Kerlink IoT Station with firmware SPN (Small Private Network). Line must be of the SerialOverUDP Device Redundant type. • **MQTT Client (JSON via MQTT)**: communication with network server or cloud using MQTT protocol. Line must be of the TCP/IP-TCP type.	-	Kerlink IoT Station SPN
Mote Field Name	Name of field with identifier of LoRaWAN device (mote). **Note:** For JSON messages that can be structured, the syntax *level1.level2.level3 ...* is supported e.g. *rx.moteeui* and if they contain fields (indexed from 1) then also the syntax *level1[index1].level2[index2].level3 ...* is supported e.g. *rx.gwrx[1].time*. For examples see description of I/O tags of the Envelope type.	-	rx.moteeui
Payload Field Name	Name of field with payload. See the note next to the Mote Field Name parameter.	-	rx.userdata. payload
Payload Encoding	A method of payload encoding in the message. Supported encoding: • **Base16 + Base64 encoding (Kerlink SPN)** - for Connection Type=*Kerlink IoT Station SPN* • **Base64 encoding (TheThings.network)** - for Connection Type=*MQTT Client* communicating with TheThings.network cloud • **Base64 encoding (Loriot, Slovanet)** - for Connection Type=*MQTT Client* communicating with Loriot and Slovanet clouds • **None** - message contains a payload without encoding - not yet used	-	Base16 + Base64 encoding

Time Field Name	Name of field with time stamp. If the field is not found, the current time is assigned to the values. See the note next to the Mote Field Name parameter.	-	rx.gwrx[1].time
Time Mask	Mask for parsing a value in field with time stamp. **Note:** from settings of time station parameters depends whether the time is interpreted as local or UTC with configured offset. Special masks are: • **UNIX** - the numeric value represents the number of seconds from epoch 00:00:00 01.01.1970 UTC. • **UNIXMS** - the numeric value represents the number of miliseconds from epoch 00:00:00.000 01.01.1970 UTC.	-	yyyy-mm-dd hh:mi:ss
Frame Type Field Name	The name of field that indicates the message type. If the value is empty, the message type is not distinguished. (For example, cloud Loriot sends messages of a various types.)	-	
Frame Type Field Required Value	If the message type differentiation is active (non-empty value of Frame Type Field Name parameter), the message type must match the specified value, otherwise, the message is ignored.	-	
Full Debug	Enabling the detailed statements about sending and receiving values.	YES/NO	NO
Parameters specific for Connection Type=*MQTT Client*.			
MQTT User Name	See the description of the User Name parameter in the MQTT protocol documentation.		
MQTT Password	See the description of the Password parameter in the MQTT protocol documentation.		
MQTT Topic Filter	See the description of the Topic Filter parameter in the MQTT protocol documentation.		+/+/+/up
MQTT Subscribe QoS	See the description of the Subscribe QoS parameter in the MQTT protocol documentation.		
MQTT Client ID	See the description of the Client ID parameter in the MQTT protocol documentation.		
MQTT Clean Session Flag	See the description of the Clean Session Flag parameter in the MQTT protocol documentation.		
MQTT Publish Format	Format of JSON message used while writing a value. The content of I/O tag of Write type will be encoded (depending on the Payload Encoding parameter) and inserted into the message, where it will replace the *#PAY#* string. The default value *"{"port":1, "confirmed":false, "payload_raw":#PAY#}* was tested when sending data to cloud TheThings.Network.	-	"{"port":1, "confirmed":false, "payload_raw": #PAY#}
MQTT Publish QoS	See the description of the Publish QoS parameter in the MQTT protocol documentation.		
MQTT Ping Interval	See the description of the Ping Interval parameter in the MQTT protocol documentation.		
MQTT Reply Timeout	See the description of the Reply Timeout parameter in the MQTT protocol documentation.		
MQTT Wait Timeout	See the description of the Wait Timeout parameter in the MQTT protocol documentation.		
MQTT Max. Wait Retry	See the description of the Max. Wait Retry parameter in the MQTT protocol documentation.		

Line parameters tested for Connection Type=*Kerlink IoT Station SPN* towards Kerlink IoT Station 868 with firmware SPN

Parameter	Value
Connection Type	Kerlink IoT Station SPN
Mote Field Name	rx.moteeui
Payload Field Name	rx.userdata.payload
Payload Encoding	Base16 + Base64 encoding
Time Field Name	rx.gwrx[1].time
Time Mask	yyyy-mm-dd hh:mi:ss
Frame Type Field Name	
Frame Type Field Required Value	

Line parameters tested for Connection Type=*MQTT client* towards TheThings.network

Parameter	Value
Connection Type	MQTT client
Mote Field Name	dev_id or hardware_serial
Payload Field Name	payload_raw
Payload Encoding	Base64 encoding
Time Field Name	metadata.time

Time Mask	yyyy-mm-dd hh:mi:ss.mss
Frame Type Field Name	
Frame Type Field Required Value	
MQTT User Name	ipesoft-test
MQTT Password	***
MQTT Topic Filter	+/+/+/up
MQTT Client ID	D2000kom
MQTT Clean Session Flag	NO
MQTT Publish Format	{"port":1, "confirmed":false, "payload_raw":#PAY#}
MQTT Publish QoS	QoS_0, QoS_1, QoS_2
MQTT Ping Interval	60
MQTT Reply Timeout	20
MQTT Wait Timeout	00.100
MQTT Max. Wait Retry	3

Line parameters tested for Connection Type=*MQTT client* towards Loriot.io with following setup:

- Output via protocol MQTT
- MQTT broker: iot.eclipse.org
- MQTT topic: com/ipesoft/iot

Parameter	Value
Connection Type	MQTT client
Mote Field Name	EUI
Payload Field Name	data
Payload Encoding	Base16 encoding
Time Field Name	ts
Time Mask	UNIXMS
Frame Type Field Name	cmd
Frame Type Field Required Value	rx
MQTT User Name	
MQTT Password	
MQTT Topic Filter	com/ipesoft/iot
MQTT Client ID	D2000kom
MQTT Clean Session Flag	NO
MQTT Publish Format	
MQTT Publish QoS	QoS_1
MQTT Ping Interval	60
MQTT Reply Timeout	20
MQTT Wait Timeout	00.100
MQTT Max. Wait Retry	3

Line parameters tested for Connection Type=*MQTT client* towards LoraLINK Slovanet:

Parameter	Hodnota
Connection Type	MQTT client
Mote Field Name	devEUI
Payload Field Name	dataHex
Payload Encoding	Base16 encoding
Time Field Name	timeStamp
Time Mask *	yyyy-mm-ddThh:mi:ss.mss

Frame Type Field Name	
Frame Type Field Required Value	
MQTT User Name	(poda AppEUI)
MQTT Password	***
MQTT Topic Filter	app/(appEUI)/node/+/rxdata
MQTT Client ID	D2000kom
MQTT Clean Session Flag	YES
MQTT Publish Format	{"reference":"","confirmed":true,"fPort":3,"dataHex":#PAY#}
MQTT Publish QoS	QoS_0
MQTT Ping Interval	60
MQTT Reply Timeout	20
MQTT Wait Timeout	00.100
MQTT Max. Wait Retry	3

* Note.: Timestamp is sent in local time. Station time settings are to be configured accordingly.

Communication station configuration

- Communication protocol "**IoT over LoRaWAN**".
- Station address: the address of the station is the identifier of the specific device (mote) that is in the Mote Field Name field.
    - for Connection Type=*Kerlink IoT Station SPN* is address a text representation of 8 byte LoRaWAN address (e.g. 00-00-00-00-21-1a-e3-c8)
    - for Connection Type=*MQTT Client* the address may be a text representation of 8 byte LoRaWAN address (e.g. 0018B2000000147D) or a symbolic address defined within MQTT server (e.g. fieldtestdevice)

Station parameters

Dialog station configuration - **Protocol Parameter** field.
They affect some optional protocol parameters. The following station parameter parameters can be entered:

Table 2

Parameter	Description	Unit	Default Value
Device Type	LoRaWAN type of device. Each device type may have its own structure of transmitted data (payload). The list of supported devices will gradually increase. Currently supported devices are: • **None** - no device • **OEM device** - payload parsing is performed by an external dll library • **Adeunis RF Field Test Device** - test device sending GPS position data and temperature data • **SolidusTech IndoorUNI Sensor** - indoor temperature and humidity meter • **SolidusTech miniUNI DS18B20 Sensor** - temperature meter for outdoor use • **Adeunis RF LoRaWAN TEMP (ARF8180BA)** - temperature meter for outdoor use with two independent temperature sensors.	-	None
External Dll Name	Name of external DLL library with code for payload parsing for Device Type=*OEM device*.	-	
No Data Timeout	Timeout after which the station goes into communicatoin error state if no data has been received.	hh:mi:ss	01:00:00
MQTT Topic (for writing)	Topic used when writing the value (for Connection Type=*MQTT client*). **Note:** for *ipesoft-test* user and *fieldtestdevice* device was tested towards TheThings.network writing with MQTT_TOPIC=*ipesoft-test/devices/fieldtestdevice/down.*	-	

I/O tag configuration

Possible value types of I/O tags: **Ai**, **Di**, **Ci**, **TxtI**, **Qi**, **TxtO**.

Value type	Address (address type)	Description

Ai, Di, Ci, Qi, TxtI	Payload	I/O tags parsed from payload. Address (Address) depends on device type (Device Type parameter). Address is not case sensitive. A special case is a blank address - the I/O tag will contain the entire payload (after relevant decoding depending on Payload Encoding parameter). The following tables indicate the addresses for each device type: Payload addresses for device type of **OEM Device** address depends on the specific implementation (dll library).

Payload addresses for device type of **Adeunis RF Field Test Device**

Address	Description
Status	Status byte of device.
TriggerAccelerometer	The True value means that the message sending was initiated by an accelerometer.
TriggerButton	The True value means that the message sending was initiated by a button.
Temperature	Measured temperature (-128 °C .. 127°C).
GpsLatitude	Latitude (0-90 degrees) from GPS sensor. **Note: GPS data may be missing if the device has no GPS signal.**
HemisphereSouth	The True value means that the latitude is south (the device is in the southern hemisphere).
GpsLongitude	Longitude (0-180 degrees) from the GPS sensor.
HemisphereWest	The True value means that the latitude is west (the device is in the western hemisphere).
GpsQualityReception	GPS signal reception quality: 1 Good, 2 Average, 3 Poor
QpsQualitySatellites	Number of visible GPS satellites.
UplinkCounter	Packet uplink counter (packets sent from the device to the LoRaWAN gateway).
DownlinkCounter	Packet uplink counter (packets sent to the device from the LoRaWAN gateway).
BatteryLevel	Battery voltage in mV.
RSSI	Indicator of strength of the received signal (Received Signal Strength Indicator) - value between 0-255. Payload contains the field only if it has previously been written to the device (sending data from the LoRaWAN gateway to the device).
SNR	Signal Noise Ratio v dB (-128 .. 127). Payload contains the field only if it has previously been written to the device (sending data from the LoRaWAN gateway to the device).

Payload addresses for device type of **SolidusTech IndoorUNI Sensor**

Address	Description
ADR	Adaptive Data Rate (optimizing data transfer speed and energy consumption). Value True indicates that ADR is on.
DataRate	Data Rate (data transmission rate) 0-5.
SNR	Signal Noise Ratio v dB (-128 .. 128).
BatteryLevel	Battery voltage in mV.
Temperature	Temperature (-125.99°C .. 125.99°C) with 0.1°C resolution.
Humidity	Relative humidity (0.0%-100%) with 0.1% resolution.
PowerAdapter	The True value means that the device is connected to a power adapter, the False value means that it is powered by battery (always False for firmware version FW 0.2.2 and lower).
Contact	The True value means that an auxiliary contact is switched on (always False for firmware version FW 0.2.2 and lower).

Payload addresses for device type of **SolidusTech miniUNI DS18B20 Sensor**

Address	Description
BatteryLevel	Battery voltage in mV.
SNR	Signal Noise Ratio of previous payload in dB. It applies after ACK is received. Value 127 indicates an undefined value (no ACK or downlink packet was received from LoRaWAN gateway).
Temperature	Temperature (-25°C .. 85°C) with resolution 0.1°C.

Payload addresses for device type of **Adeunis RF LoRaWAN TEMP (ARF8180BA)**

Address	Description
FrameCounter	Internal message counter going from 0 to 7.
BatteryLow	Low battery indicator. Has values True or False.
HWError	Indicator of hardware error in a device (temperature sensor error etc.).
InternalTemp	The value of the temperature sensor located in the device housing with a resolution of 0.1 °C.
ExternalTemp	The value of the temperature sensor located on external wire with a resolution of 0.1 °C.

Ai, Di, Ci, Qi, TxtI	Envelope	I/O tag parsed from envelope of message. The address is the name of the field in the envelope of message. **Note:** For JSON messages that can be structured, the syntax *level1.level2.level3 ...* is supported, e.g. *rx.moteeui* and if they contain fields (indexed from 1) then also *level1[index1].level2[index2].level3 ...* syntax, e.g. *rx.gwrx[1].time*. Example of JSON message for Connection Type=*Kerlink IoT Station SPN* (added spacing and alignment for better legibility):

```
{
  "rx": {
    "moteeui": "00-00-00-00-00-1e-fc-1d",
    "userdata": {
        "seqno": 77,
        "port": 1,
        "payload": "NzM3RjAwZTgwMA==",
        "motetx": {
            "freq": 868500000,
            "modu": "LoRa",
            "datr": "SF7BW125",
            "codr": "4/5"
        }
     },
     "gwrx": [
        {
         "time": "2017-07-05 16:06:52",
         "chan": 2,
         "rfch": 0,
         "rssi": -33,
         "lsnr": 7.5
        }
     ]
  }
}
```

I/O tags of envelope may have addresses e.g. *rx.moteeui*, *rx.userdata.seqno*, *rx.userdata.motetx.freq*, *rx.gwrx[1].time*.

Example of JSON message for Connection Type=*MQTT Client (JSON via MQTT)* (added spacing and alignment for better legibility):

```
{
  "app_id":"ipesoft-test",
  "dev_id":"fieldtestdevice",
  "hardware_serial":"0018B2000000147D",
  "port":2,
  "counter":549,
  "payload_raw":"niNJElVwAYQ5UBYfBBBN",
  "metadata":{
    "time":"2017-08-10T08:12:26.06860368Z",
    "frequency":867.5,
    "modulation":"LORA",
    "data_rate":"SF7BW125",
    "coding_rate":"4/5",
    "gateways":[
      {
        "gtw_id":"eui-000000000003080b",
        "timestamp":705621508,
        "time":"2017-08-10T08:12:26.434682Z",
        "channel":5,
        "rssi":-34,
        "snr":7.8,
        "latitude":49.20927,
        "longitude":18.73184,
        "altitude":359
      }
    ]
  }
}
```

I/O tags of envelope may have addresses e.g. *dev_id*, *metadata.time*, *metadata.gateways[1].latitude*.

TxtI	All data	I/O tag, that will contain the complete received message - the whole envelope (e.g. JSON message). The I/O tag is intended for debugging purposes and for eventual processing of the entire message in the script.
TxtO	Write (MQTT)	I/O tag for writing. Currently implemented just for Connection Type=*MQTT client* and tested towards cloud TheThings.Network. The value of I/O tag is considered to be a payload that will be encoded (depending on the Payload Encoding parameter) and inserted into the message template defined by the MQTT Publish Format parameter, where it will replace the *#PAY# string*. The resulting message will be sent to the MQTT server.

Literature

Links
Official website of LoRaWAN alliance https://www.lora-alliance.org/technology
Official website of MQTT protocol http://mqtt.org

Specifications and Standards
MQTT 3.1.1 specification http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html
ISO/IEC 20922:2016 http://www.iso.org/standard/69499.html

Descriptions of Data Formats and API
www.loriot.io - Application API Data Format https://www.loriot.io/home/documentation.html#docu/app-data-format
www.thethingsnetwork.org - API Reference https://www.thethingsnetwork.org/docs/applications/mqtt/api.html

Document revisions

- Ver. 1.0 - August 10th, 2017 - Document creation.
- Ver. 1.1 - August 25th, 2017 - Extended line configuration (Frame Type, Time Mask - UNIX, UNIXMS, PayloadEncoding - Base16), support of AdeunisRF LoRaWAN TEMP device and communication with Loriot.io.

Related pages:

Communication protocols