ThingPlus Embedded Software Implementation Guide

ver: 0.5

This document is for helping H/W manufacturers to implement an embedded software to connect and work with ThingPlus Cloud Platform. This is including ThingPlus MQTT Protocol Specification(MQTT Topic and Message) with the sample source code. Connection and data communication management, generation and analysis of MQTT Message and time synchronization should be implemented. This allows to implement the optimized embedded software for any devices.

1.Mandatory Features of an Embedded Software integrated with ThingPlus Platform

An Embedded Software integrated with ThingPlus Cloud Platform should meet the below requirements

- A device should support the MQTT Communication over SSL/TLS for the secured communication
- The local time of a device should be synchronized with ThingPlus Platform Server Time
- A device should have the capability to store the sensor, temporarily, for preventing the loss of sensor data when the device suffers the connection and/or network issue. Data Store for latest 3 hours is recommended
- When the network connection between a device and ThingPlus Cloud Platform is recovered after the connection issue, the stored data should be transferred to ThingPlus cloud Platform
- A device should report the status of device and sensor(s) to ThingPlus Cloud Platform, periodically. This is a separate process from the sensor data transmission
- When a device has a sensor generating values periodically, the sensor data should be sent with the predefined frequency by the business term. If you need to use the shorter frequency than predefined frequency, please contact Daliworks Engineering Team.
- A device only can subscribe the allowed MQTT Topic by ThingPlus Cloud Platform

2.Definition of Terminologies defined by ThingPlus Platform

2.1.Category of Sensors based on data transfer timing

- **Event Sensor:** It transmits the sensor data right after the data is generated and/or changed
- **Series Sensor:** It transmits the sensor data periodically. In this case, **Report Interval** means the frequency of the data transmission

2.2.List of Sensors and Actuators

ThingPlus defines the supported sensors and actuators. Sensor/Actuator type is the unique set of strings. H/W Manufacturers should use the defined sensor/actuator types. This is for displaying an icon and graph of the sensor/actuator. If you uses unmatched sensor/actuator type, ThingPlus Cloud Portal will display the unmatched icon and graph for your sensor/actuator

- List of Sensors: This is a part of the full list of sensors

Sensor Type	Category	Sensor Type	Category
number	series	onoff	event
string	series	motion	event
percent	series	countEvent	event
temperature	series	door	event
humidity	series	accelerometer	series

In case of Actuator, ThingPlus defines types of actuators and command for the actuator

- List of Actuators: This is a part of the full list of actuators

Actuator Type	Command	Actuator Type	Command
led	on off blink	lcd	print clear
powerSwitch	on off	buzzer	on off
camera	snapPicture		

You can get the full list of sensors and actuators over the ThingPlus api and/or ThingPlus portal after you logged in the portal

If you should use a sensor and/or actuator which is not defined by ThingPlus, yet. Please contact ThingPlus Engineering Support Team.

2.3 Gateway ID, APIKEY

- **Gateway ID**: It is an unique ID of a thing to identify it from others. It is set of 12 characters. All things integrated with ThingPlus platform should have an Gateway ID. In general, the Mac Address of the ethernet on the device.
- **APIKEY:** APIKEY is the token for MQTT Authentication of a thing. ThingPlus Portal provide a APIKEY when a Gateway ID is registered. A thing should store the APIKEY permanently and use the APIKEY for the authentication process of the MQTT connection with ThingPlus Cloud Platform.

3.ThingPlus MQTT Specification

This chapter describes the MQTT Topic and data format used by ThingPlus Platform. A thing integrated with ThingPlus Cloud Platform should follow the given ThingPlus MQTT Specification. When ThingPlus Cloud Platform send a command to an actuator, the command uses the same ThingPlus MQTT Specification.

- ThingPlus MQTT Message uses QoS1. ‘RETAIN’ of all messages except ‘Will’ Message is not used.
- Timezone of the message is UTC.
- A thing transmit data regarding the status of MQTT connection, gateway and sensor.
- The value for each status is defined like below,

Status	Value(strings)
turned-on	on
turned-off	off
Error occurred	err

3.1. ThingPlus MQTT Specification - MQTT connection

The version of MQTT used by ThingPlus platform is 3 and 8883 port is used for the MQTT communication. 8883 port should be opened for communicating with ThingPlus Platform.

- **Details of the MQTT Connection Spec**

MQTT Connection SPEC	Defined Value by ThingPlus	Comments
MQTT Version	3	**Fixed Value.** Version of MQTT used by ThingPlus
PORT	8883	**Fixed Value.** Port used by ThingPlus for MQTT Communication
MQTT Client ID	*_gatewayID_*	Unique Value for a Thing
Clean Session	TRUE	**Fixed Value**
MQTT ID	*_gatewayID_*	Unique Value for a Thing
MQTT Password	*_APIKEY_*	APIKEY for a Thing
Will Topic	v/a/g/*_gatewayID_*/mqtt/status	**Fixed value except *_gatewayID_* part**
Will Message	err	**Fixed Value**
Will Message Retain	TRUE	**Fixed Value**
Keep Alive	*_KeepAliveValue_*	Keep Alive value for the MQTT Connection in sec. **2 times longer value than the report interval for a thing is recommended**

3.2. ThingPlus MQTT Specification - transmission of the MQTT Connection status data

A Thing should transmit the status of the MQTT Connection to ThingPlus Platform after the MQTT Connection is created successfully.

- **Transmit the MQTT Connection Success Status**

{
TOPIC: /v/a/g/*_gatewayID_*/mqtt/status
MESSAGE : on
}

3.3. ThingPlus MQTT Specification - transmission of the H/W status data

A Thing should transmit its status and the valid time of the status data periodically. When ThingPlus fails to get the status data of a specific H/W within the valid time period, ThingPlus defines its status as the error status

- **Transmit the H/W Status Data**

{
TOPIC: /v/a/g/*_gatewayID_*/status
MESSAGE : *_hwStatus_*, *_validTime_*
}

Sensor Status Data can be transmitted with the H/W status data. It is an efficient way to send the sensor status data with the H/W status data in one topic for saving the network bandwidth cost. When a thing has multiple sensors and/or actuators, each sensor/actuator should have an unique value for identifying it from other sensors/actuators on the thing. You can define it by yourself. If you use this method, you can skip chapter 3.4.

- **Transmit the H/W Status Data with sensor status data**

{
TOPIC: /v/a/g/*_gatewayID_*/status
MESSAGE : *_hwStatus_*, *_validTime_*, *_sensorID_*, *_sensorStatus_*,*_validTime_*,..., *_sensorID_*, *_sensorStatus_*,*_validTime_*
}

3.4. ThingPlus MQTT Specification - transmission of the sensor status data

A Thing should transmit the status of a sensor on it and the valid time of the status data periodically. When ThingPlus fails to get the status data within the valid time period, ThingPlus defines its status as the error status

- **Transmit the sensor status data**

{
TOPIC: /v/a/g/*_gatewayID_*/s/*_sensorID_*
MESSAGE : *_sensorStatus_*,*_validTime_*
}

3.5. ThingPlus MQTT Specification - transmission of the sensor value data

A single sensor value and multiple sensor values for a specific sensor can be transmitted at a time. A sensor value should be paired with the time value which the thing read it. When multiple sensor values are transmitted at a time, they should be ordered by time. Array of multiple sensor values is allowed

- **Transmit the sensor value data**

{
TOPIC: /v/a/g/*_gatewayID_*/s/*_sensorID_*
MESSAGE : *_time_*,*_value_*,*_time_*,*_value_*...*_time_*,*_value_*
}

- **Transmit the sensor value data as array**

{
TOPIC: /v/a/g/*_gatewayID_*/s/*_sensorID_*
MESSAGE : [*_time_*,*_value_*,*_time_*,*_value_*...*_time_*,*_value_*]
}

3.6. ThingPlus MQTT Specification - transmission of the sensor value data for multiple sensors

Multiple sensor values for multiple sensors can be transmitted at a time. At this time, the sensor value should for a single sensor should be grouped and ordered by time

- **Transmit the multiple sensor values for multiple sensors**

{
TOPIC: /v/a/g/*_gatewayID_*
MESSAGE : { "*_sensorID_*":[*_time_*,*_value_*,*_time_*,*_value_*...*_time_*,*_value_*],
"*_sensorID_*":[*_time_*,*_value_*,*_time_*,*_value_*...*_time_*,*_value_*], … ,
"*_sensorID_*":[*_time_*,*_value_*,*_time_*,*_value_*...*_time_*,*_value_*] }
}

3.7. ThingPlus MQTT Specification - transmission of the result of a request from ThingPlus Platform

A thing should be report the result of a request from ThingPlus Platform. the request can be the command for an actuator, the configuration request.

- **Transmit the result of a request from ThingPlus Platform - Success Case**

{
TOPIC: /v/a/g/*_gatewayID_*/res
MESSAGE : {"id":"*_messageID_*","result":"***_result_***"}
}

Value of ***_result_*** should follow the schema of the value defined by ThingPlus. You can get the data schema from the senor/actuator definition after logged in to ThingPlus Portal.

- **Transmit the result of a request from ThingPlus Platform - Failed Case**

{
TOPIC: /v/a/g/*_gatewayID_*/res
MESSAGE : {"id":"*_messageID_*","error":{"code":*_errCode_*, "message":*_errMessage_*}}

}

In case of _errCode_, it is the error code of the failed case. ThingPlus follows the JSONRPC Error Code Rule. ***_errMessge_*** should have the details about the root cause of the error.

3.8. ThingPlus MQTT Specification - MQTT Messages from ThingPlus Platform

A thing should subscribe below topic for getting the MQTT Message published by ThingPlus Platform. Each thing can subscribe a MQTT message only for it.

- **MQTT Topic and Message subscribed by a Thing**

{
TOPIC: /v/a/g/*_gatewayID_/req*
MESSAGE : "id":"*_messageID_*","method":"*_method_*","params":"*_params_*"
}

A MQTT Message from ThingPlus Platform consists of message ID(*_messageID_*), method(*_method_)* and parameter(*_params_*).

- **Message ID:** it is an unique id for identifying each message and reporting the result of each request.
- **Method:** List of requests from ThingPlus Platform.
- **Parameters**: Parameters for a method. each method has its own parameters

List of Methods defined by ThingPlus platform is below,

- **List of Methods and Parameters for a method**

Method	**Description**	**Parameters**	**Description**
timeSync	**Synchronize the local time of a thing with ThingPlus Platform server time**	**time**	**current time in UTC**
setProperty	**Environment Configuration**	**reportInterval**	**frequency for reporting sensor value in msec**
controlActuator	**execute a command on an actuator**	**id**	**ID of an actuator**
		cmd	**Command for an actuator Actual value of this paramter depends on the type of an actuator**
		options	**options for a command**
poweroff	**turn off the device**	**none**	**none**
reboot	**restart the H/W**	**none**	**none**
restart	**restart the embedded software**	**none**	**none**
swUpdate	**Upgrade the embedded software**	**none**	**none**

swInfo	**Provide the version of embedded software**	**none**	**none**

The mandatory methods which should be implemented an embedded software integrated with ThingPlus Platform should are **timeSync and setProperty**. Others are optional.

- **Example of timeSync Method**

When a thing get an MQTT message inclduing this method, it should reset the local time on it as the received time value.

Message spec

{

TOPIC: /v/a/g/*_gatewayID_/req*

MESSAGE : "id":"*_messageID_*","method":"timeSync","params":{“time”:*_time value in UTC_*}

}

Example - Request

{

TOPIC: v/a/g/1928dbc93871/req

MESSAGE: {"id":"e1kcs13b9","method":"timeSync","params":{"time":1372874401865}}

}

- **Example of setProperty Method**

This method is for changing the report interval. the unit of report interval value is msec. ThingPlus Portal provides the UI for changing the report interval.

Message spec

{

TOPIC: /v/a/g/*_gatewayID_/req*

MESSAGE : "id":"*_messageID_*","method":"setProperty","params":{“reportInterval”:*_interval_*}

}

Example - Request

{

TOPIC: v/a/g/1928dbc93781/req

MESSAGE: {"id":"e1kcs13bb","method":"setProperty","params":{"reportInterval":"60000"}}

}

- **controlActuator Method**

This method is for sending command to an actuator. Command can be sent from ThingPlus Portal UI and an actuator should execute the delivered command from ThingPlus Platform. Commands and options for each actuator are defined by ThingPlus. The examples of commands and option for some actuators are below,

Actuator	Command	option
led	on	duration
	off	
	blink	duration
		interval
powerSwitch	on	duration
	off	

You can get the full list of commands/options for actuators defined by ThingPlus from ThingPlus Portal.

Message spec

{

TOPIC: /v/a/g/*_gatewayID_/req*

MESSAGE : "id":"*_messageID_*","method":"controlActuator","params":{“id”:*_actuatorID_,* “cmd”:*_cmd_*, “options”: *_cmdOptions_*}

}

Example - Request

{

TOPIC: v/a/g/1928dbc93781/req

MESSAGE:

{"id":"46h6f8xp3","method":"controlActuator","params":{"id":"led-1928dbc93781-r","cmd":"on","options":{"duration":3000}}}

}
