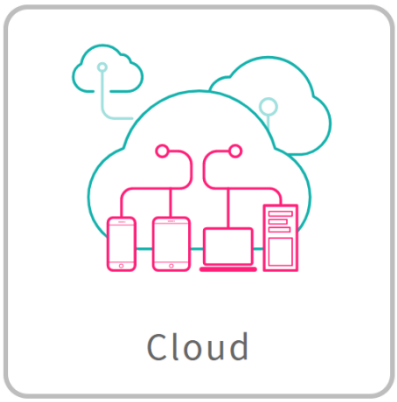
REYAX
TECHNOLOGY CORPORATION, LTD

1-OCT-2020 56312E35

RYC1001

MQTT IoT Cloud Platform

Datasheet



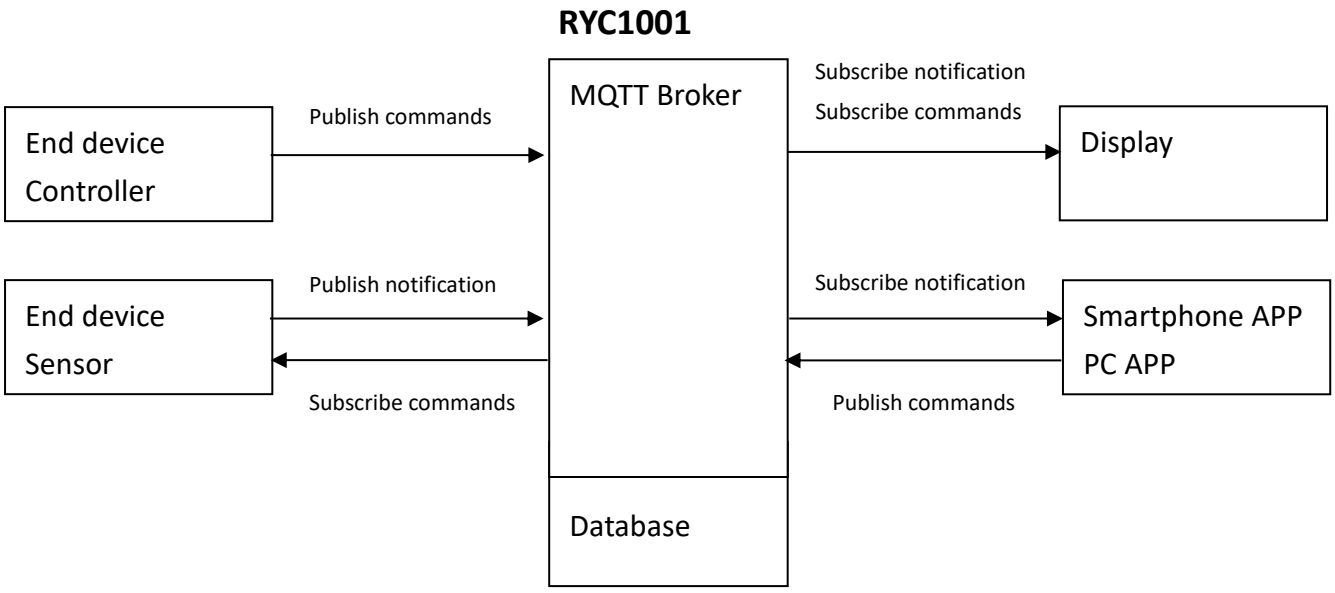
PRODUCT DESCRIPTION

RYC1001 is a cloud platform suitable for low-data-volume and power-saving devices. Using the MQTT protocol, you can use simple commands for your applications, monitor end devices, and easily establish your IoT connection.

FEATURES

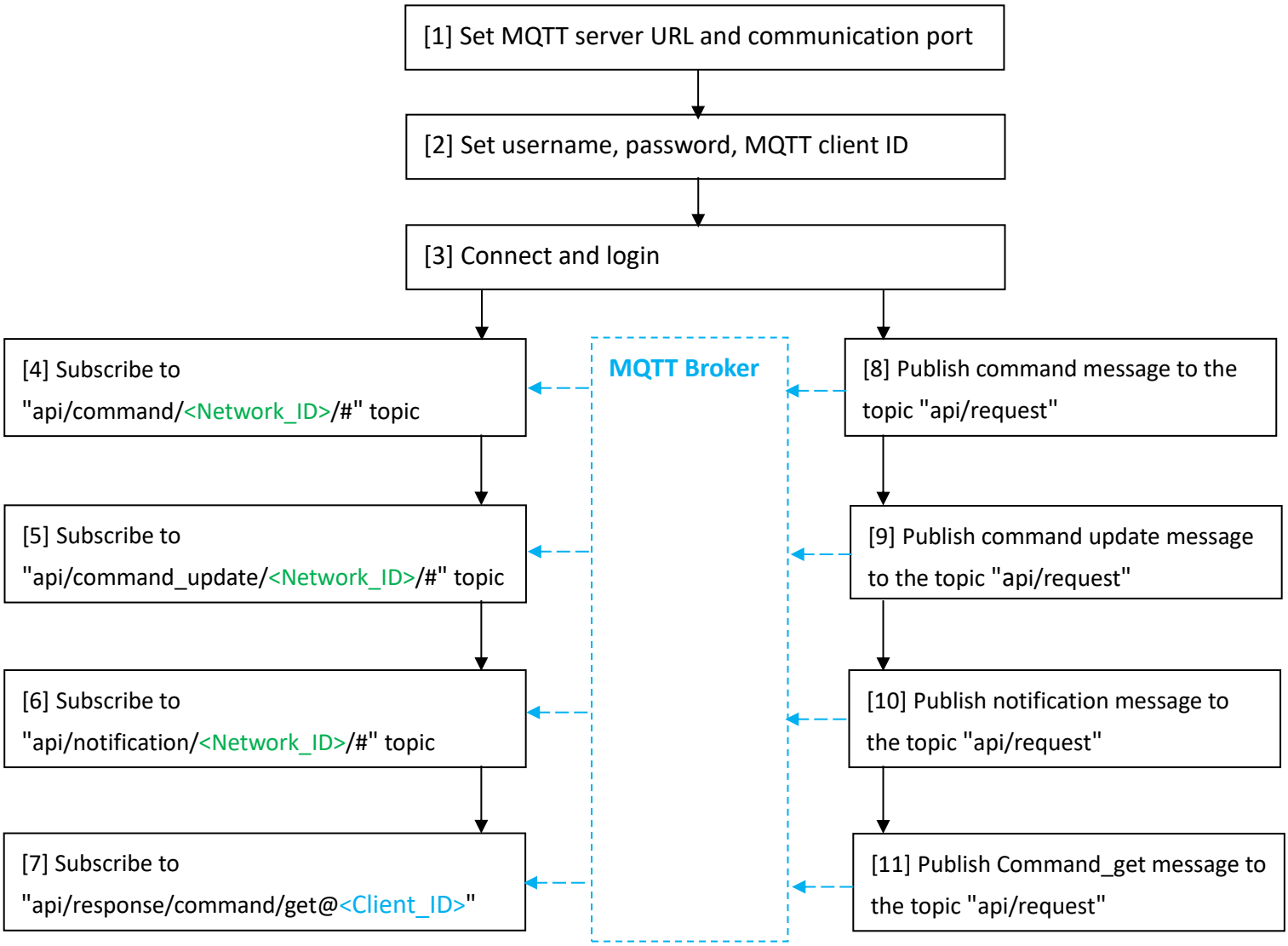
- Built on a stable AWS service
- Use MQTT protocol suitable for low-data-volume and power-saving mode
- Devices that support the MQTT protocol can be use.
- Compatibility testing of all mainstream IoT communication modules is completed
- Support Android, iOS, Windows, Linux
- Low-cost access to the cloud platform
- Use simple instructions to quickly integrate the IoT cloud platform

MQTT BLOCK DIAGRAM



The MQTT protocol uses Subscribe and Publish system. When Subscribed Notification or Commands Published, and conforms to the subscription Topic, MQTT Broker will take the initiative to notify subscribers, so subscribers do not need to continue to actively ask, to save power consumption and transmission volume. And the records of these events and commands will be recorded in the Database, you can use the <commandId> provided by the system to query and modify the stored data.

OPERATION FLOW



*The operation process is not fixed from [1]~[11], it can be executed according to the use situation.

*Scenario 1: To display the value of "end device controller" in the "display", the process of “display” is [1] [2] [3] [4], the process of "end device controller" is [1] [2] [3] [8].

*Scenario 2: Display want to query the previously stored value of “end device sensor” in the Database, the process of "display" is [1] [2] [3] [7] [11].

*Scenario 3: To continue the storage time of "end device controller" in the Database, the process of "display" is [1] [2] [3] [5], the process of "end device controller" for [1] [2] [3] [9].

OPERATION PROCESS & INSTRUCTION FORMAT DESCRIPTION

***The following parameters can be found in the chapter: PARAMETER DESCRIPTION**

[1] Set MQTT broker URL and port: <URL> is defaulted to iot.reyax.com and the port is 1883. <URL> is based on the information at the time of purchase.

[2] Set username and password: <Username>, <Password>, and MQTT client ID: <Client_ID>.

[3] Connect and login: execute the connect and login action.

[4] Subscribe to the "api/command/<Network_ID>/#" topic: This topic is defined by the platform service to receive command messages.

When Publish is executed, the received format is as follows:

{"action":"command/insert","command":{"id":<commandId>,"command":"<Command_Name>","timestamp":"<Time_Stamp>","lastUpdated":"<Time_Stamp>","userId":6,"deviceId":"<Device_ID>","networkId":6,"deviceTypeId":4,"parameters":{"<Parameter1>":"<Parameter2>"},"lifetime":null,"status":"<Status>","result":{"<Result1>":"<Result2>"}},"subscriptionId":<subscriptionId>}

[5] Subscribe to the topic "api/command_update/<Network_ID>/#": This topic is defined by this platform service to receive command_update messages.

When Publish is executed, the received format is as follows:

{"action":"command/update","command":{"id":<commandId>,"command":"<Command_Name>","timestamp":"<Time_Stamp>","lastUpdated":"<Time_Stamp>","userId":6,"deviceId":"<Device_ID>","networkId":6,"deviceTypeId":4,"parameters":{"<Parameter1>":"<Parameter2>"},"lifetime":null,"status":"Done","result":{"<Result1>":"<Result2>"}},"subscriptionId":<subscriptionId>}

[6] Subscribe to the topic "api/notification/<Network_ID>/#": This topic is defined by the platform service to receive notification messages.

When Publish is executed, the received format is as follows:

{"action":"notification/insert","notification":{"id":<notificationId>,"notification":"<Notification_Name>","deviceId":"<Device_ID>","networkId":6,"<Time_Stamp>":"<TimeStamp>","parameters":{"<Parameter1>":"<Parameter2>"}},"subscriptionId":<subscriptionId>}

[7] Subscribe to the topic "api/response/command/get@<Clinet_ID>": This topic is defined by the platform service and is used to query the results of previous Publish command according to the <commandId>.

When Publish is executed, the received format is as follows:

{"action":"command/get","requestId:null,
"status":"success","command":{"id":<commandId>,"command":"<Command_Name>","timestamp":"<Time_Stamp>","lastUpdated":"<Time_Stamp>","userId":6,"deviceId":"<Device_ID>","networkId":6,"deviceTypeId":4,"parameters":{"<Parameter1>":"<Parameter2>"},"lifetime":null,"status":"<Status>","result":{"<Result1>":"<Result2>"}}}

[8] Publish the command message to the “api/request” topic: The format of command message is defined by this platform. The message will be stored in the database of this platform.

The command format is as follows:

{"action":"command/insert","deviceId":"<Device_ID>","command":{"command":"<Command_Name>","parameters":{"<Parameter1>":"<Parameter2>"},"status":"<Status>",
"result":{"<Result1>":"<Result2>"}}}

[9] Publish the command_update message to the "api/request" topic: The format of the command_update message is defined by this platform. The message will be stored in the database of this platform.

The command format is as follows:

{"action":"command/update","deviceId":"<Device_ID>","commandId":<commandId>,"command":{"status":"<Status>","result":"<Result1>":"<Result2>"}}

[10]Publish notification messages to the "api/request" topic: The format and content of notification messages are defined by this platform. The message will be stored in the database of this platform.

The command format is as follows:

{"action":"notification/insert", "deviceId": "<Device_ID>","notification": {"notification": "<Notification_Name>", "parameters": {"<Parameter1>":"<Parameter2>"}}}"

[11] Publish Command_get message to "api/request" topic: The format and content of Command_get message are defined by this platform.

The command format is as follows:

{"action":"command/get","deviceId":"<Device_ID>","commandId":<commandId>}

PARAMETER DESCRIPTION

*<XXX>Green and <XXX>Blue are the parameters and data provided by the user to the system.

*<XXX>Red is the information provided by the system.

1. The <URL>, <Username>, <Password>, <Device_ID> and <Network_ID> used in the following procedures need to be obtained after purchase.

*<URL> is the name of the connected URL, please follow the URL provided when purchasing the information.

*<Username> Account name for logging in to the system.

*<Password> Password for logging in to the system.

When the password is incorrect for ten consecutive times, the account will be locked

*<Device_ID> The unique device code on the system.

*<Network_ID> The unique network code on the system.

2. <commandId> Description: When executing "[8] Post command message to command subject", the system will save the data and generate <commandId>, this <commandId> is the number of the data, you can use.

"[9] Post command_update message to command_update topic" to modify the content.

3. <Command_Name>, <Parameter1>, <Parameter2>, <Status>, <Result1>, <Result2>, <Notification_Name>, <Client_ID> description:

*<Command_Name> Users define content by themselves, avoid {",:} <> these symbols.

*<Parameter1> User-defined content, avoid {",:} <> these symbols.

*<Parameter2> Users define content by themselves, avoid {",:} <> these symbols.

*<Status> User-defined content, avoid {",:} <> these symbols.

*<Result1> Users define content by themselves, avoid {",:} <> these symbols.

*<Result2> Users define content by themselves, avoid {",:} <> these symbols.

*<Notification_Name> Users define content by themselves, avoid {",:} <> these symbols.

*<Client_ID> is recommended as username and number(0001~9999).

4. <Time_Stamp> Description: Time stamp in ISO 8601 format, yyyy-mm-ddThh:mm:ss.xxx.
The example is: 2020-10-13T08:09:13.033

5. <subscriptionId> Description: The subscriber ID passed to the subscriber by the system to identify multiple different subscribers.
   The example is: 1602576993294310

6. Each message contains the command format and the limit of the amount of data transfer is 1500Bytes.

7. To receive published message, you must Subscribe first. Otherwise, you cannot receive previously Publish message.

8. The published message can only be kept in the Database for 25 days. If you need to keep it, you need to use "[8] Post command_update message to command_update topic" to update and extend the retention time.

LIMITED LIABILITY & SERVICE DECLARATION

1. You need to take good care of your account password and related confidential information to avoid damage due to data leakage.
2. We have tried our best to use the best services and equipment. However, due to natural disasters, telecommunication equipment problems, power outages, man-made sabotage, war damage, hacking, equipment maintenance and other problems in Internet communications, we will deal with these problems. We do not assume any responsibility.
3. We will try our best to provide detailed documentation for this product to help you use it, but for further technical assistance beyond the documentation, a fee must be charged.
4. If it is clearly discovered that there is an act of using account information to invade or damage the system, we will have the right to terminate your right to use it without making any refund.
5. If you have important data, it is recommended that you back it up regularly. We cannot guarantee that your data will be saved.

ORDER INFORMATION

Part No.	Monthly Message Limit	Service Duration
RYC1001	200K times	5 years

*The monthly message limit will be reset to zero at GMT+0 00:00 on the 1st of each month.

*Any Publish or Subscribe action will count the number of messages once

*There will be one year buffer period after the purchase, the 5 years usage period will start after buffer period.

* You can add money to increase the use time of your account..

REYAX
TECHNOLOGY CORPORATION, LTD

Taiwan: sales@reyax.com
China: sales@reyax.com.cn
http://reyax.com