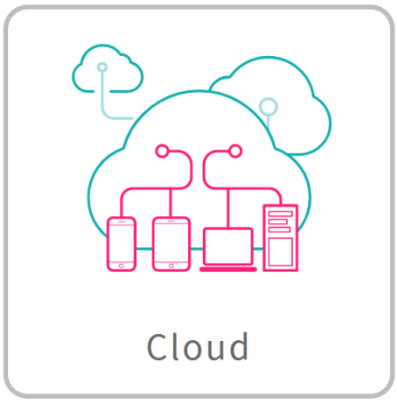
REYAX
TECHNOLOGY CORPORATION, LTD

1-OCT-2020 56312E35

RYC1001

MQTT 物聯網雲端平台

Datasheet



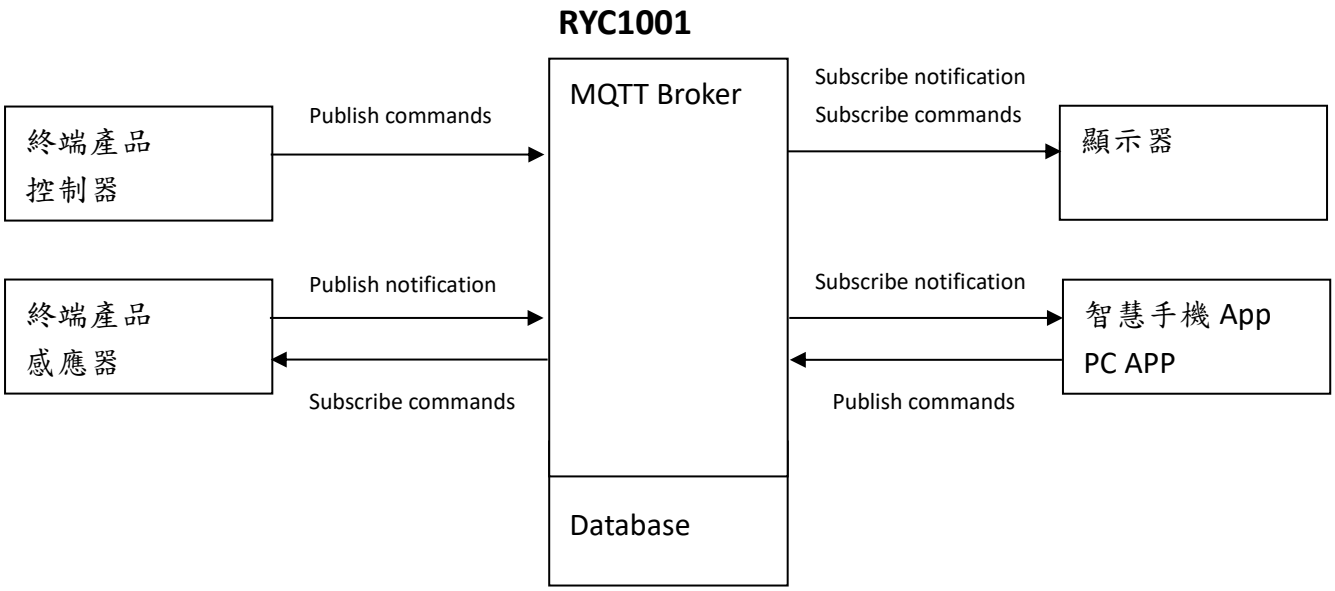
PRODUCT DESCRIPTION

RYC1001 是一個適合低資料量與省電裝置運用的雲端平台，使用 MQTT 傳輸協定，可以利用簡單的指令，置入應用程式與監控終端產品，輕易建立你的物聯網連線。

FEATURES

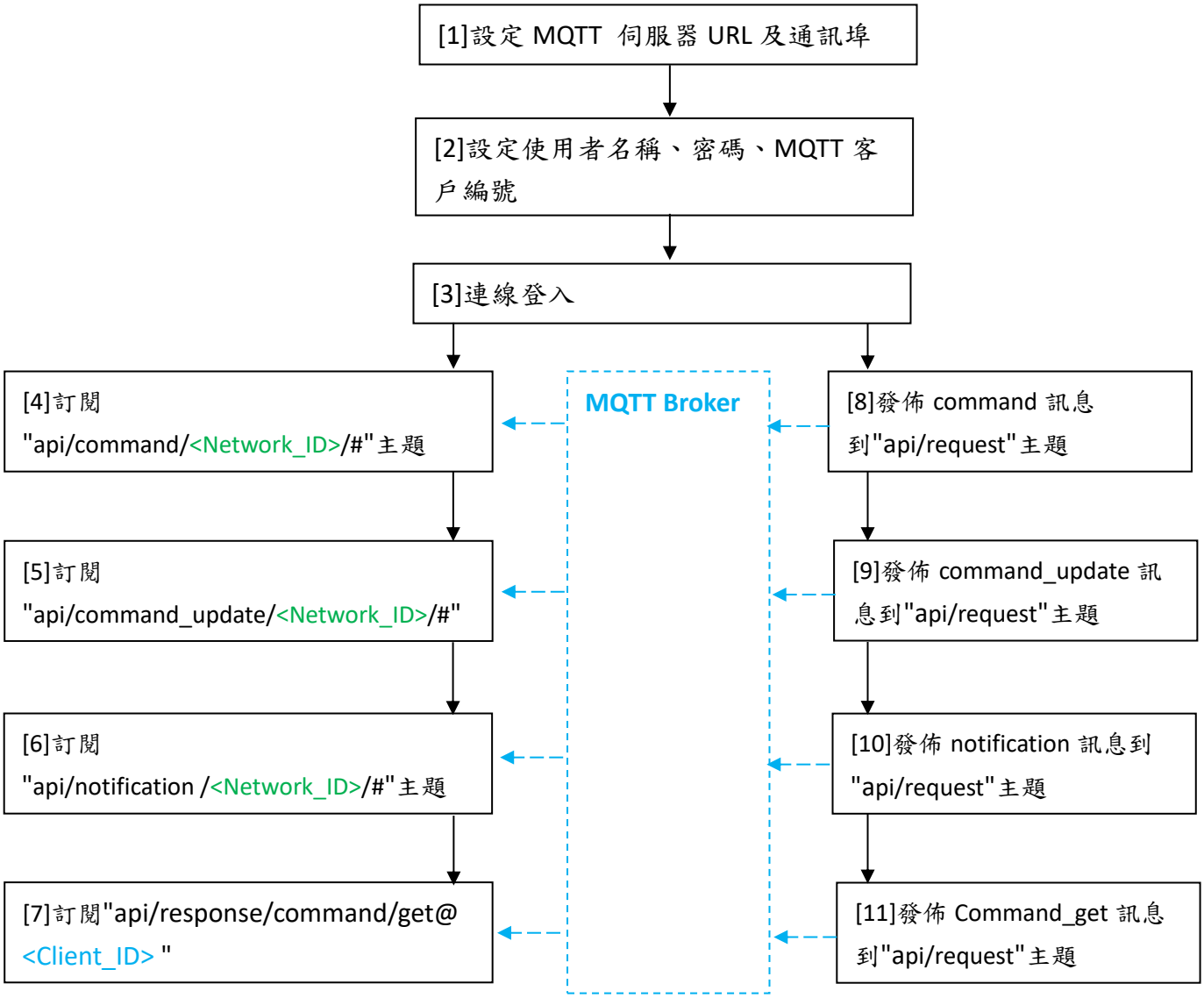
- 建構在穩定的 AWS 系統
- 使用適合低資料量與省電機制的 MQTT 傳輸協定
- 支援 MQTT 通訊協定的裝置都可以使用
- 各主流物聯網通訊模組相容性測試完成
- 支援 Android、iOS 、Windows、Linux
- 低成本進入雲端平台
- 使用簡單指令快速整合物聯網雲端平台

MQTT BLOCK DIAGRAM



MQTT 傳輸協定是使用訂閱(Subscribe)與發佈(Publish)的機制，當被訂閱的通知(Notification)或命令(Commands)被發佈(Publish)時，並且符合訂閱的主題(Topic)，MQTT Broker 會主動通知訂閱者，因此訂閱者不用持續主動詢問，達到節省電能消耗與傳輸量的目的，並且這些事件與命令的紀錄會記錄在資料庫(Database)裡，可利用系統提供的<commandId>查詢與修改儲存的資料。

操作流程



*操作流程並非固定從[1]~[11]，可依照使用情境執行。

*情境 1:要在"顯示器"中顯示"終端產品控制器"的數值，"顯示器"的流程為[1] [2] [3] [4]，"終端產品控制器"的流程為[1] [2] [3] [8]。

*情境 2:要查詢"終端產品感應器"在資料庫(Database)裡之前儲存的數值，"顯示器"的流程為[1] [2] [3] [7] [11]。

*情境 3:要延續"終端產品控制器"在資料庫(Database)裡的儲存時間，"顯示器"的流程為[1] [2] [3] [5]，"終端產品控制器"的流程為[1][2][3][9]。

操作流程與指令格式說明

*以下參數可以查閱**參數說明**章節

[1]設定 MQTT 伺服器 URL 及通訊埠: <URL>預設為 iot.reyax.com，通訊埠為 1883。
<URL>需依照購買時的資訊決定。

[2]設定 MQTT 使用者名稱與密碼: <Username>、<Password>、MQTT 客戶編號<Client_ID>。

[3]連線登入: 執行連線登入動作。

[4]訂閱"api/command/<Network_ID>/#"主題: 該主題是由本平台服務所定義，用來接收 command 訊息。
當發佈(Publish)執行後，接收到的格式如下:
{"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]訂閱"api/command_update /<Network_ID>/#"主題: 該主題是由本平台服務所定義，用來接收 command_update 訊息。
當發佈(Publish)執行後，接收到的格式如下:
{"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]訂閱"api/notification/<Network_ID>/#"主題: 該主題是由本平台服務所定義，用來接收 notification 訊息。
當發佈(Publish)執行後，接收到的格式如下:
{"action":"notification/insert","notification":{"id":<notificationId>,"notification":"<Notification_Name>","deviceId":"<Device_ID>","networkId":6,"<Time_Stamp>":"<TimeStamp>","parameters":{"<Parameter1>":"<Parameter2>"}},"subscriptionId":<subscriptionId>}

[7]訂閱"api/response/command/get@<Client_ID>"主題: 該主題是由本平台服務所定義，依照<commandId>用來**查詢**之前的發佈(Publish)command 內容。

當發佈(Publish)執行後，接收到的格式如下:

{"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]發佈(Publish)command 訊息到"api/request"主題: Command 訊息的格式內容由本平台所定義。該訊息會存入本平台的資料庫。

指令格式如下:

{"action":"command/insert","deviceId":"<Device_ID>","command":{"command":"<Command_Name>","parameters":{"<parameter1>":"<parameter2>"},"status":"<Status>",
"result":{"<result1>":"<result2>"}}}

[9]發佈(Publish)command_update 訊息到"api/request"主題: command_update 訊息的格式內容由本平台所定義。該訊息會存入本平台的資料庫。

指令格式如下:

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

[10]發佈(Publish)notification 訊息到"api/request"主題: notification 訊息的格式內容由本平台所定義。該訊息會存入本平台的資料庫。

指令格式如下:

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

[11]發佈(Publish)Command_get 訊息到"api/request"主題: Command_get 訊息的格式內容由本平台所定義。

指令格式如下:

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

參數說明

*<XXX>綠色與<XXX>藍色為使用者提供給系統的參數與資料。

*<XXX>紅色為系統主動提供的資訊。

1. 下面程序中使用的<URL>、<Username>、<Password>、<Device_ID>、<Network_ID>需要在購買後獲取。
   *<URL>為連接的網址名稱，請依照購買資訊時提供的網址連接。
   *<Username>登入系統用的帳號名稱。
   *<Password>登入系統用的帳號密碼，當密碼連續錯誤十次時帳號會被鎖定。
   *<Device_ID>系統上唯一的設備編碼。
   *<Network_ID>系統上唯一的網路編碼。

2. <commandId>說明: 當執行" [8]發佈 command 訊息到 command 主題"後系統會儲存資料並產生<commandId>，此<commandId>即為該筆資料的編號，可以使用。
   "[9]發佈 command_update 訊息到 command_update 主題"修改內容。

3. <Command_Name>、<Parameter1>、<Parameter2>、<Status>、<Result1>、<Result2>、<Notification_Name>、<Client_ID>說明:

   *<Command_Name>使用者自行定義內容，避免{",:}<>這些符號。
   *<Parameter1>使用者自行定義內容，避免{",:}<>這些符號。
   *<Parameter2>使用者自行定義內容，避免{",:}<>這些符號。
   *<Status>使用者自行定義內容，避免{",:}<>這些符號。
   *<Result1>使用者自行定義內容，避免{",:}<>這些符號。
   *<Result2>使用者自行定義內容，避免{",:}<>這些符號。
   *<Notification_Name>使用者自行定義內容，避免{",:}<>這些符號。
   *<Client_ID>建議為使用者名稱加上數字(0001~9999)。

4. <Time_Stamp>說明: ISO 8601 格式的時間戳記，yyyy-mm-ddThh:mm:ss.xxx。
   範例為: 2020-10-13T08:09:13.033

5. <subscriptionId>說明:系統傳給訂閱者的訂閱者 ID，用以識別多個不同的訂閱者。
   範例為: 1602576993294310

6. 每個訊息含指令格式與資料傳輸量的限制總量為 1500Bytes。

7. 要接收發佈(Publish)的資料必須先訂閱(Subscribe)，發佈(Publish)後再訂閱(Subscribe)是無法收到之前發佈(Publish)的資料。
8. 發佈(Publish)的資料只能在 Database 保留 25 天，如果需要繼續保留需使用"[8]發佈 command_update 訊息到 command_update 主題"更新，延長保留時間。

有限責任與服務宣告

1. 您需要善盡保管帳號密碼與相關機密資訊，避免因資料洩密造成損害。
2. 我們已經盡力使用最好的服務與設備，但在網際網路通訊上由於存在著天然災害、電信設備問題、電力中斷、人為破壞、戰爭毀壞、駭客入侵、設備維護等問題，我們對這些問題將不承擔任何責任。
3. 我們會盡力對此產品提供詳細的說明文件以幫助您使用，但對於文件說明以外的進一步的技術協助那是必須收費的。
4. 如果明確發現有利用帳號資訊進行入侵或破壞系統的行為，我們將有權終止你的使用權利並不做任何退款。
5. 您如果有重要的資料建議您定時備份，我們將無法保證一定能救回您的資料。

型號	每月訊息傳輸次數限制	帳號使用年限
RYC1001	200K次	5年

訂購資訊

*每月訊息傳輸次數限制會在每月的 1 日的 GMT+0 時間 00:00 歸零。
*任何發佈(Publish)或是訂閱(Subscribe)的動作都會計算一次訊息數。
*購買後會有一年的緩衝期，經過一年後才會開始計算五年使用時間。
*帳號是可以儲值延長使用時間的。

REYAX
TECHNOLOGY CORPORATION, LTD

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