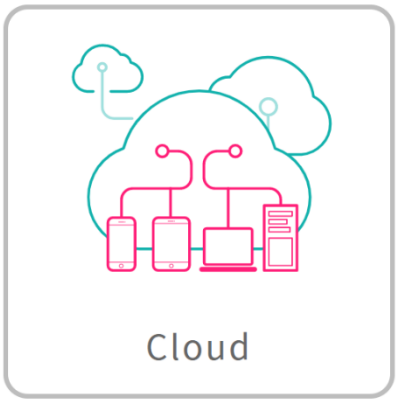
REYAX
TECHNOLOGY CORPORATION, LTD

30-OCT-2020 56312E31

RYC1001

MQTT 物联网云端平台

Datasheet



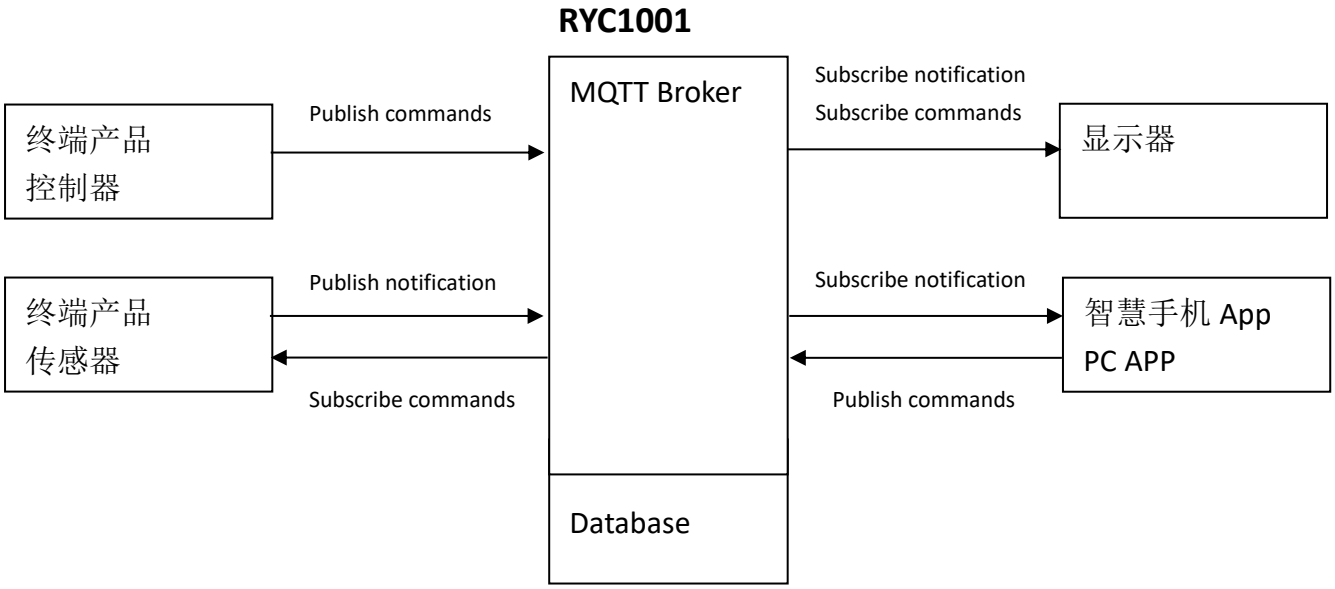
PRODUCT DESCRIPTION

RYC1001 是一个适合低数据量与省电装置运用的云端平台，使用 MQTT 传输协议，可以利用简单的指令，置入应用程序与监控终端产品，轻易建立你的物联网联机。

FEATURES

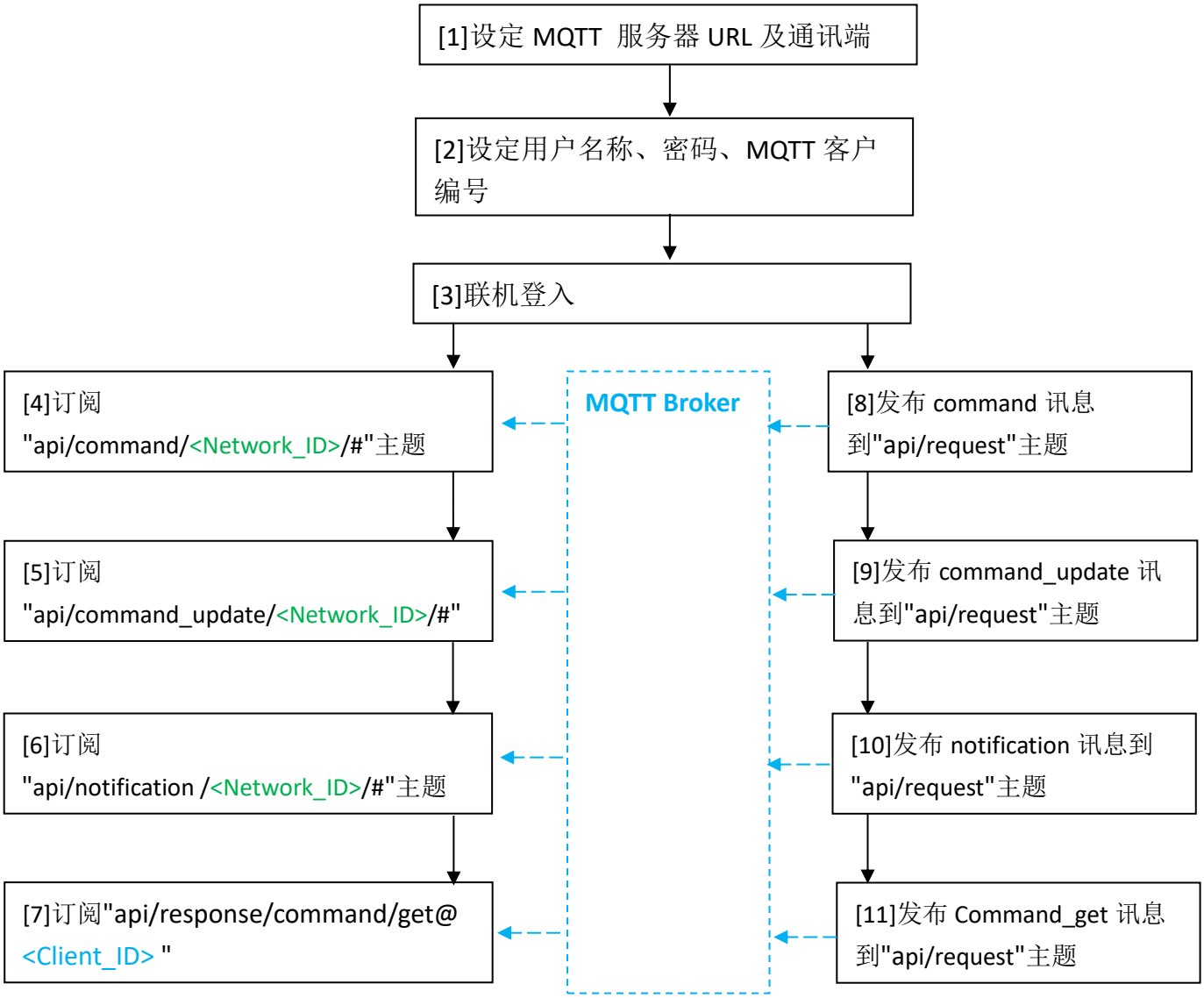
- 建构在稳定的 AWS 系统
- 使用适合低数据量与省电机制的 MQTT 传输协议
- 支持 MQTT 通讯协议的装置都可以使用
- 各主流物联网通讯模块兼容性测试完成
- 支援 Android、iOS 、Windows、Linux
- 低成本进入云端平台
- 使用简单指令快速整合物联网云端平台

MQTT BLOCK DIAGRAM



MQTT 传输协议是使用订阅(Subscribe)与发布(Publish)的机制，当被订阅的通知(Notification)或命令(Commands)被发布(Publish)时，并且符合订阅的主题(Topic)，MQTT Broker 会主动通知订阅者，因此订阅者不用持续主动询问，达到节省电能消耗与传输量的目的，并且这些事件与命令的纪录会记录在数据库(Database)里，可利用系统提供的<commandId>查询与修改储存的数据。

操作流程



*操作流程并非固定从[1]~[11]，可依照使用情境执行。

*情境 1:要在"显示器"中显示"终端产品控制器"的数值，"显示器"的流程为[1] [2] [3] [4]，"终端产品控制器"的流程为[1] [2] [3] [8]。

*情境 2:要查询"终端产品传感器"在数据库(Database)里之前储存的数值，"显示器"的流程为[1] [2] [3] [7] [11]。

*情境 3:要延续"终端产品控制器"在数据库(Database)里的储存时间，"显示器"的流程为[1] [2] [3] [5]，"终端产品控制器"的流程为[1][2][3][9]。

操作流程与指令格式说明

*以下参数可以查阅参数说明章节

[1]设定 MQTT 服务器 URL 及通讯端口: <URL>预设为 iot.reyax.com，通讯端口为 1883。
<URL>需依照购买时的信息决定。

[2]设定 MQTT 用户名称与密码: <Username>、<Password>、MQTT 客户编号<Client_ID>。

[3]联机登入: 执行联机登入动作。

[4]订阅"api/command/<Network_ID>/#"主题: 该主题是由本平台服务所定义，用来接收 command 讯息。
当发布(Publish)执行后，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]订阅"api/command_update /<Network_ID>/#"主题: 该主题是由本平台服务所定义，用来接收 command_update 讯息。
当发布(Publish)执行后，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]订阅"api/notification/<Network_ID>/#"主题: 该主题是由本平台服务所定义，用来接收 notification 讯息。
当发布(Publish)执行后，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]订阅"api/response/command/get@<Client_ID>"主题: 该主题是由本平台服务所定义，依照 <commandId>用来查询之前的发布(Publish)command 内容。

当发布(Publish)执行后，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]发布(Publish)command 讯息到"api/request"主题: Command 讯息的格式内容由本平台所定义。该讯息会存入本平台的数据库。

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]发布(Publish)command_update 讯息到"api/request"主题: command_update 讯息的格式内容由本平台所定义。该讯息会存入本平台的数据库。

指令格式如下:

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

[10]发布(Publish)notification 讯息到"api/request"主题: notification 讯息的格式内容由本平台所定义。该讯息会存入本平台的数据库。

指令格式如下:

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

[11]发布(Publish)Command_get 讯息到"api/request"主题: Command_get 讯息的格式内容由本平台所定义。

指令格式如下:

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

参数说明

*<XXX>绿色与<XXX>蓝色为用户提供给系统的参数与数据。

*<XXX>红色为系统主动提供的信息。

1. 下面程序中使用的<URL>、<Username>、<Password>、<Device_ID>、<Network_ID>需要在购买后获取。
   *<URL>为连接的网址名称，请依照购买信息时提供的网址连接。
   *<Username>登入系统用的账号名称。
   *<Password>登入系统用的账号密码，当密码连续错误十次时账号会被锁定。
   *<Device_ID>系统上唯一的设备编码。
   *<Network_ID>系统上唯一的网络编码。

2. <commandId>说明：当执行" [8]发布 command 讯息到 command 主题"后系统会储存数据并产生<commandId>，此<commandId>即为该笔资料的编号，可以使用。
   "[9]发布 command_update 讯息到 command_update 主题"修改内容。

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

   *<Command_Name>使用者自行定义内容，避免{ " , : } < >这些符号。
   *<Parameter1>使用者自行定义内容，避免{ " , : } < >这些符号。
   *<Parameter2>使用者自行定义内容，避免{ " , : } < >这些符号。
   *<Status>使用者自行定义内容，避免{ " , : } < >这些符号。
   *<Result1>使用者自行定义内容，避免{ " , : } < >这些符号。
   *<Result2>使用者自行定义内容，避免{ " , : } < >这些符号。
   *<Notification_Name>使用者自行定义内容，避免{ " , : } < >这些符号。
   *<Client_ID>建议为用户名称加上数字(0001~9999)。

4. <Time_Stamp>说明: ISO 8601 格式的时间戳，yyyy-mm-ddThh:mm:ss.xxx。
   范例为: 2020-10-13T08:09:13.033

5. <subscriptionId>说明:系统传给订阅者的订阅者 ID，用以识别多个不同的订阅者。
   范例为: 1602576993294310

6. 每个讯息含指令格式与数据传输量的限制总量为 1500Bytes。

7. 要接收发布(Publish)的数据必须先订阅(Subscribe)，发布(Publish)后再订阅(Subscribe)是无法收到之前发布(Publish)的数据。
8. 发布(Publish)的数据只能在 Database 保留 25 天，如果需要继续保留需使用"[8]发布 command_update 讯息到 command_update 主题"更新，延长保留时间。

有限责任与服务宣告

1. 您需要善尽保管账号密码与相关机密信息，避免因数据泄密造成损害。
2. 我们已经尽力使用最好的服务与设备，但在因特网通讯上由于存在着天然灾害、电信设备问题、电力中断、人为破坏、战争毁坏、黑客入侵、设备维护等问题，我们对这些问题将不承担任何责任。
3. 我们会尽力对此产品提供详细的说明文件以帮助您使用，但对于文件说明以外的进一步的技术协助那是必须收费的。
4. 如果明确发现有利用帐户信息进行入侵或破坏系统的行为，我们将有权终止你的使用权利并不做任何退款。
5. 您如果有重要的数据建议您定时备份，我们将无法保证一定能救回您的数据。

型号	每月讯息传输次数限制	账号使用年限
RYC1001	200K次	5年

ORDER INFORMATION

*每月讯息传输次数限制会在每月的 1 日的 GMT+0 00:00 归零。
*任何发布(Publish)或是订阅(Subscribe)的动作都会计算一次讯息数。
*购买后会有一年的缓冲期，经过一年后才会开始计算五年使用时间。
*账号是可以储值延长使用时间的。

REYAX
TECHNOLOGY CORPORATION, LTD

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