IOT 通用物联网系统

协议支持文档

版本记录

序号	时间	内容
1	2018/10/16	增加 TCP 连接的方式的协议说明
2	2018/11/11	增加 MQTT 连接方式，支持 MQTT
3	2019/06/20	完善 UDP/HTTP 连接通讯协议
4	2019/02/08	升级完善各个协议
5	2020/12/20	完善部分说明

1 系统概念

1.1 协议概念

日常使用中，我们所谈论的协议主要分为两种，一为通讯协议，这个是网络传输层的概念，目前市面上用的到普遍有 TCP、MQTT、DUP 和 HTTP 等，另一为设备协议，设备协议所指设备发送数据格式的规约（如 Modbus RTU），目前通讯协议是统一一致的，设备协议存在多样性，目前市面存在各种各样的设备协议，modbus 算当中用的比较多一些的。

1.2 设备号

设备号[device_code] 它是指的负责连接网络的设备唯一 SN 号，区分不同设备连接上来。如下图：它可以自动生成，也可以手动填入，格式可以为数字也可以为字符串；

图 a 系统中设备号位置

1.3 地址号和寄存器号

地址号[sensor_device_id]和寄存器号[port_id]，它的作用主要是用来区分设备传输的数据。它支持一台网关设备下传输多个不同地址传感器，也可以为一台网关设备下传输多种传感器数据。地址号主要用来区分不同传感器设备的，寄存器号是用来标注同一台传感器设备不同的数据类型的。

图 b 添加传感点中地址号和寄存器号

2 TCP 连接协议

2.1 烽源智能协议 [私有]

小名智能精简协议是使用字符流格式的数据协议，它表达的数据协议更加直接，易于理解。目前数据协议包含登录包、数据上传、控制下发 数据指令。

连接地址： www.iot2yun.com（或者对应 IP） 端口：50001

- 登录包（设备 -> 服务器）[TCP 首次连接时发送]

  UHGDZ1SSGFSH

  数据内容为设备管理信息中设备号[device_code],直接发送设备号即可。

- 上传传感点数据（设备 -> 服务器）

  S23.8,46.7,1,E

  数据指令以 S 开头 E 结尾

  数据信息会按照排列顺序值默认对应传感器的地址号和寄存器号上，如下

  23.8 数值发送到 地址号（sensor_device_id）=0 和寄存器(port_id) =0 的传感器上；

  46.7 数值发送到 地址号（sensor_device_id）=1 和寄存器(port_id) =1 的传感器上；

  1 数值发送到 地址号（sensor_device_id）=2 和寄存器(port_id) =2 的传感器上；

  其中，sensor_device_id 和 port_id 的 0,1,2 代表是数据排序顺序，其他依次类推。

  备注：如果是开关的话，则 0 代表 关闭状态，1 代表是打开状态

- 控制命令下发（服务器->设备）

  2:1

  2 为地址号 sensor_device_id=2 和寄存器 port_id=2 的传感器

  ： 分隔符

  1 为下发数值，此处代表打开

  备注：如果是开关的话，则 0 代表 关闭状态，1 代表是打开状态

2.2 小名智能协议 [私有]

小名智能协议是使用字符流格式的数据协议，目前数据协议包含登录包、心跳包、数据上传、控制下发、控制返回，配置下发、配置返回等数据指令。

- 登录包（设备 -> 服务器）【首次建立 TCP 连接发送】
  [device_code]
  例子：1u76yshydtdhh ，其中 1u76yshydtdhh 为设备 SN 号；
  设备返回：
  loginok 登录成功

- 心跳包（设备 -> 服务器）维持 TCP 连接，设备偏好发送
  Q
  固定格式

- 上传传感点数据（设备 -> 服务器）
  S[sensor_device_id]:[port_id]*[value],[sensor_device_id]:[port_id]*[value]E
  例子：S1:0*12.5,1:1*-0.05,1:2*1,1:3*119.5107+31.64459E
  sensor_device_id 地址号
  port_id 寄存器号
  value 数值

  备注：其中 GPS 经纬度数值，用+连接在一起；

  服务器收到数据成功后，返回 rok 。

- 服务器下发 控制/配置 命令（服务器 -> 设备）
  S[sensor_device_id]:[port_id]*[value]E
  例子：S1:0*0E
  备注：按钮开关命令，0 为关，1 为开；

- 设备接收命令返回（设备 -> 服务器）
  S[sensor_device_id]:[port_id]*[value]E
  例子：S1:0*0E

2.3 Modbus RTU 协议（标准）

系统兼容标准的 modbus RTU 协议。
DTU 配置时，需要配置成透传模式，并且设置自定义注册包，注册包为设备的 SN 即可；

连接地址： www.iot2yun.com（或者对应 IP） 端口：50001

2.4 Modbus TCP 协议（标准）

系统兼容标准的 modbus TCP 协议。
DTU 配置时，需要配置成透传模式，并且设置自定义注册包，注册包为设备的 SN 即可；
DTU 需设置 modbus 模式

连接地址： www.iot2yun.com（或者对应 IP） 端口：50001

2.5 环境 212 标准

系统支持 GB-HJ212 协议。

连接地址： www.iot2yun.com（或者对应 IP） 端口：50001

数据配置格式如下，地址号填写类型，寄存器号默认为 0

序号	传感点名称	类型	地址号	寄存器号	操作
1	PH值	数据	001-Rtd	0	
2	化学需氧量	数据	011-Rtd	0	
3	总磷	数据	101-Rtd	0	
4	氨氮	数据	060-Rtd	0	

3 MQTT 连接协议

3.1 MQTT 连接方式和数据格式

Broker Address：www.iot2yun.com
Broker Port：1883
Client ID：[device_code]　　[设备号]
User Name：iot2yun
Password：iot2yun

- 客户端 -> 服务器
  客户端发布到下方主题数据

  Topic：/dev/coo/[device_code]

  数据格式：

  ```
  [{"sensor_device_id":0,"port_id":0,"sdata":1.0},
   {"sensor_device_id":1,"port_id":1,"sdata":18.3}]
  ```

  sensor_device_id 地址号
  port_id 寄存器号
  Sdata 数据数值

- 服务器 -> 客户端
  客户端关注下方主题，服务器推送数据

  Topic：/server/coo/[device_code]

  数据格式：

  ```
  {"sensor_device_id":0,"port_id":0,"sdata":0.0}
  ```
  sensor_device_id 地址号
  port_id 寄存器号
  Sdata 传感器数值

备注：如果是继电器开关的话，则 ***0*** *关闭，* ***1*** *打开；*