MODBUS-TCP

欢迎访问设备云门户网站<http://open.iot.10086.cn/>注册用户，获取最新文档。

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订日期** | **修订内容** | **说明** |
| V1.0 | 2014.10.30 | 新增 |  |
| V1.2 | 2015/8/5 | Dtu设备登录传输项目ID |  |
| V1.3 | 2016/02/18 | 增加接入流程例子 |  |
| V1.4 | 2016/07/06 | 实现01，02，03，04功能码周期查询通过URL获取服务IP和PORT设备定时发送心跳包通过restfull api主动下发在线命令 |  |

# 说明

基于《modbus协议.pdf》实现01，02，03，04功能码周期查询。

适用于设备云定期主动发送modbus-tcp命令到设备查询寄存器值，设备回复寄存器值的情况。

# 名词解释

**项目ID**：在平台注册账号后，添加新项目平台生成数字ID；

**Api-key**：管理每个项目下资源，用于restful api鉴权。

# 设备接入

1. 访问设备云门户注册用户，可以查看（项目ID）和master\_key(api-key)；
2. 业务平台调用RESTFUL API 新增设备，并指定如下设备属性：

|  |  |
| --- | --- |
| 设备属性名 | 说明 |
| **interval** | 设备云主动查询设备的时间间隔，单位秒 |
| **auth\_info** | 设备鉴权信息，格式如下：{“登录报文的phone字段”:”登录报文的SVRPWD字段”} |

1. 设备上电，发送登录报文（52字节）到设备云接入服务器进行鉴权，设备通过链接[**http://api.heclouds.com/s?t=5&v=1**](http://api.heclouds.com/s?t=5&v=1)拉取服务器IP和PORT，其中格式如下：



登录报文固定格式如下：（其中绿色三项为必填）

|  |  |  |
| --- | --- | --- |
| 报文字段 | 固定长度 | 说明 |
| **type** | 11字节 | 设备类型字符串，以’\0’结尾，目前作为保留用途。 |
| **name** | 9字节 | 设备名称字符串，以’\0’结尾，目前作为保留用途。 |
| **phone** | 12字节 | 设备通信卡号码，以’\0’结尾。注册设备时作为auth\_info属性子域json对象的key，用于鉴权。 |
| **svrpwd** | 9字节 | 服务器登录密码，以’\0’结尾。注册设备时作为auth\_info属性子域json对象的value，用于鉴权。 |
| **id** | 11字节 | 项目ID，以’\0’结尾，平台申请项目时分配的项目ID字符串。 |

1. 设备登录成功后，服务器不会返回数据，如果登录失败，服务器会主动断开连接。

# 数据流

* 1. 业务平台调用restfull api为设备新增数据流，数据流需要指定如下属性：

|  |  |
| --- | --- |
| 数据流属性名 | 说明 |
| **cmd** | 设备云为获取数据定期下发访问寄存器的命令，用十六进制标示。如“83A245”表示三个字节的命令0x83A245。 |
| **formula** | 可选配置。对命令返回各个寄存器值的处理公式，多个公式用分号间隔。例如: "(A0+A1)\*A2",表示对命令返回的前两个寄存器值的和再乘上第三个寄存器的值得到综合结果。注意：公式不能写成这样 (A0\*A1)，请去掉最外层的无效括号，否则公式会被认为不合法，不会生效 |

* 1. 01、02命令返回结果（不支持formula属性）

{“起始地址”:”线圈值1,线圈值2,…,线圈值n”}

其中，每个线圈值对应设备返回数据的一个byte

 起始地址、线圈值都以10进制表示

* 1. 03命令返回结果（支持formula属性）
		1. 未设置formula属性

{“地址1”:”寄存器值1”, ”地址2”:”寄存器值2”, … ,”地址n”:”寄存器值n”}

其中，地址、寄存器值都以10进制表示

* + 1. 已设置formula属性

{“1”:”formula计算结果1”, “2”: “formula计算结果2”, … , “n”:”formula计算结果n”}

其中，计算结果为double类型值

formula属性可以有多个计算公式组合在一起，通过”;”隔开。

1、2、…、n，分别表示第1、2、…、n个计算公式

A0、A1、…、An, 分别表示第1、2、…、n个寄存器值

例如：(A0+A1)\*A2 – (A3-A4)/A5 + (A6+A7)%A8 + A9fA10

 目前formula支持运算符如下：

 **+ - \* / % ()**

**f** 表示前后操作数作为uint16，然后合并组成float

例如 **A0fA1**

**union**

**{**

**uint32\_t u32;**

**float f32;**

**} u\_f;**

**u\_f.u32 = (((uint32\_t)(uint16\_t)A0)<<16) | (uint16\_t)A1;**

**u\_f.f32 即为f运算符运算结果**

* 1. 04命令返回结果（不支持formula属性）

{“地址1”:”寄存器值1”, ”地址2”:”寄存器值2”, … ,”地址n”:”寄存器值n”}

其中，地址、寄存器值都以10进制表示

* 1. 注意事项

数据流中定时下发的命令尽量保持不同

# 保持设备在线

当设备处于空闲的时候，为了保持与ONENET平台的连接不断开，需要定期发送心跳包（推荐0x0000），以确保网络连接不断开。其中时间间隔须小于5分钟，推荐1～3分钟。

# 通过restfull api下发在线命令到设备

区别于数据流的定时下发命令到设备，resfull api通过使用HTTP请求的方式，主动下发命令到设备。

其中,通过restfull api主动下发的命令，尽量与数据流中定时下发的命令保持不同。

|  |  |
| --- | --- |
| HTTP方法 | POST |
| URL | http://api.heclouds.com/cmds |
| HTTP头部 | api-key:xxxx-ffff-zzzzz，必须master key |
| URL参数 | device\_id = //接收该数据的MODBUS设备ID，必填。 |
| HTTP内容 | { "cmd":”xxxxxxxx” //需要下发MODBUS设备的命令（16进制字符串）} |
| 成功返回 | {"errno": 0,"error":“succ”，"data":{*//不超过64个字符字符串*"cmd\_uuid":“2302-312-FWs”}} |

查看命令发送状态

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://api.heclouds.com/cmds/<cmd\_uuid> |
| HTTP头部 | api-key:xxxx-ffff-zzzzz，必须master key |
| 成功返回 | {"errno": 0,"error":“succ”，"data":{*//0：设备不在线|Device not online**1：命令已创建| Command Created* *2：命令已发往设备| Command Sent* *3：命令发往设备失败| Send Command Failed* *4：设备正常响应| Command Response Received* *5：命令执行超时| Command Response Timeout**6：设备响应消息过长 | Command Response Too Large*"status":1"desc":“sending”}} |

获取终端执行命令后的响应内容。

|  |  |
| --- | --- |
| HTTP方法 | GET |
| URL | http://api.heclouds.com/cmds/< cmd\_uuid >/resp |
| HTTP头部 | api-key:xxxx-ffff-zzzzz， 必须master key |
| HTTP body | MODBUS设备响应数据(二进制) |

响应的HTTP状态码对应的错误信息如下

|  |  |
| --- | --- |
| 400 | No api-key |
| 403 | Auth Failed |
| 404 | Not Found |
| 400 | Invalid Parameter |
| 220 | Command Created |
| 221 | Command Sent |
| 520 | Send Command Failed |
| 200 | Command Response Received |
| 432 | Command Response Timeout |
| 433 | Command Response Too Large |

# 接入例子--命令03读取保持寄存器（以下截图来自《MODBUS协议.pdf》）

1. 利用restfull api创建设备

POST http://api.heclouds.com/devices HTTP/1.1

User-Agent: Fiddler

api-key: nCxNdCoX6eHGfergpyytu66TyuY= /\*必须为MasterKey\*/

Host: api.heclouds.com

注意长度和空行，建议使用fiddler调试，会自动填充长度

Content-Length: 155

{

 "title":"12345670",/\*填写MODBUS设备卡号\*/

 "desc":"test modbus",

 "private":true,

 "protocol":"HTTP",/\*填写HTTP协议，没有专门的Modbus协议\*/

 "auth\_info":{"12345670":"12345670"},/\*这里就是鉴权信息的卡号和密码，长度有限制，参考登录报文格式\*/

 "interval":20 /\*命令的查询间隔\*/

}

此api没有针对modbus相关配置出错处理返回，必须仔细填写各个字段。

1. 为该设备创建数据流

使用restfull api创建，请求数据如下：

POST http://api.heclouds.com/devices/[your\_devid]/datastreams HTTP/1.1

User-Agent: Fiddler

api-key: nCxNdCoX6eHGfergpyytu66TyuY= /\*设备对应的key或masterkey\*/

Host: api.heclouds.com

注意长度和空行，建议使用fiddler调试，会自动填充长度

Content-Length: 66

{

"id":"test\_cmd",

"cmd":"1103006B00038776",

"interval":20

}

CMD字段解释如下："1103006B00038776"对应的MODBUS RTU命令如下图所示：

{0x11,0x03,0x0,0x6B,0x0,0x3,0x87,0x76}，其中0x87和0x76存储的是CRC16结果0x8776，从Slave Address 0x11开始计算到最后一个寄存器地址(此处为0x3)。



1. 上传登录报文

如前述登录报文格式，例子如下：

unsigned char req[52];

unsigned char type[11] = "type";

 unsigned char name[9] = "name";

 unsigned char phone[12] = "12345670";//填写创建设备时的卡号

 unsigned char pwd[9] = "12345670";//填写创建设备时的密码

unsigned char p\_id[11] = "47434";//modbus设备所在项目的ID

 memset(req, 0, 52);

 memcpy(req, type, 11);

 memcpy(req + 11, name, 9);

 memcpy(req + 11 + 9, phone, 12);

 memcpy(req + 11 + 9 + 12, pwd, 9);

 memcpy(req + 11 + 9 + 12 + 9, p\_id, 11);

向登录服务器上传整个req即可，登录成功无返回，否则服务器断开连接。

Raw数据如下：

0x74 0x79 0x70 0x65 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x6e 0x61 0x6d 0x65 0x0 0x0 0x0 0x0 0x0 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x30 0x0 0x0 0x0 0x0 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x30 0x0 0x33 0x39 0x34 0x38 0x34 0x0 0x0 0x0 0x0 0x0 0x0

四个颜色标注的数据依次对应登录报文的各字段。

1. 与服务器端建立TCP连接后，并成功登录，设备云会定期发送查询命令，设备一分钟内无回复，服务器会断开连接。

设备端收到的命令RAW数据如下：

0x11 0x3 0x0 0x6b 0x0 0x3 0x87 0x76

和增加数据流时配置的数据一模一样，如"cmd":"1103006B00038776"。

1. 设备端回复查询命令的结果

按照如下截图中RTU格式回复16进制数据：

0x11 0x3 0x6 0x2 0x2b 0x0 0x0 0x0 0x64 0xc8 0xba ，采用CRC16校验，不需要转字符串。



1. 查看上传结果
2. 在平台的web端显示，进入对应项目和设备的页面查看，情况如下：



1. 利用restfull api查询结果，这里使用Fiddler工具发送查询设备数据点的命令。



核对下数据，是不是我们上传的。

上传数据：0x11 0x3 0x6 **0x2 0x2b 0x0 0x0 0x0 0x64** 0xc8 0xba ，

0x22b=555,0x00=0,0x64=100，上传正确。

附测试用Restfull api：

POST http://api.heclouds.com/devices HTTP/1.1

User-Agent: Fiddler

api-key: **yours**

Host: api.heclouds.com

Content-Length: 155

{

 "title":"12345670",

 "desc":"test modbus",

 "private":true,

 "protocol":"HTTP",

 "auth\_info":{"12345670":"12345670"},

 "interval":20

}

POST http://api.heclouds.com/devices/**yours**/datastreams HTTP/1.1

User-Agent: Fiddler

api-key: **yours**

Host: api.heclouds.com

Content-Length: 66

{

"id":"test\_cmd",

"cmd":"1103006B00038776",

"interval":20

}

GET http://api.heclouds.com/devices/**yours**/datastreams/test\_cmd HTTP/1.1

User-Agent: Fiddler

api-key: **yours**

Host: api.heclouds.com

Content-Length: 0