广州致远电子有限公司-设备与服务器之间的通讯协议

版本	说明
v0.0	初稿
v0.1	支持TLS加密，去掉应用层的加密。支持透传上报和下发。
v0.2	INFO改为STATUS；增加主题/device/broadcast/${devtype}，用于向指定的设备类型发送广播。
v0.3	增加上报命令的执行结果和请求设备上报最新数据。
v0.4	去掉HeartBeat消息，使用MQTT的ping和遗言（LWT）判断离线。设备类型和ID从消息体移到主题中。
v0.5	包的头部增加1字节的保留字段。
v0.6	增加透传数据的定义。为了避免透传数据与透传命令的概念混淆，去掉内置透传命令，小量数据仍然可以通过命令放送，但是需要自己定义。
v0.7	增加登录流程。
v0.8	增加设备上报日志的主题。
v0.9	修改文档结构。
v1.0	移除设备型号以及设备注册主题；修改MQTT主题结构，新增owner层级；新增网关发布¥订阅相关的主题。

设备与服务器之间使用MQTT协议通讯，MQTT协议包括主题(topic)和消息(message)两部分。本文档定义所涉及的主题和消息的格式。

> 下面所引用的宏请参考：common/client.h

一、消息格式

1. 基本消息格式

一般情况下，服务器和设备之间的消息都使用基本消息，其格式如下：

- 第一个字节保留(设置为0x00)，方便支持其它打包方式。
- 消息由Key/Value组成，Key/Value是字符串格式，并自带结束的空字符。

例：

```
"\0key1\0value1\0key2\0value2\0"
```

```
其主要好处有：
```

1. 易于扩展。增加新的字段不会破坏兼容性。
2. 解析速度快，由于自带结束的空字符，不需要额外的解析过程和内存空间。

数字需要转换为**字符串。**

2. 内置的key

目前预先定义的key有：

宏	key	类型	意义
STR_KEY_NAME	"name"	字符串	(服务器发送给设备的)命令名称或其它的名称，表示消息的功能/用途。
STR_KEY_MSG_TYPE	"mtype"	整型	(服务器发送给设备的)消息的类型，目前有两种：PACKET_COMMAND (值为2)是命令消息，PACKET_RESPONSE (值为1)是响应消息。比如设备发布上线消息后，服务器用PACKET_RESPONSE回复设备上线结果。
STR_KEY_CMDID	"cmdid"	整型	命令的ID，一般是自增ID，用来标志消息的唯一性。
STR_KEY_DEVICE_ID	"devid"	字符串	设备的ID。
STR_KEY_DEVICE_TYPE	"devtype"	字符串	设备的类型。
STR_KEY_DEVICE_ID	"devid"	字符串	设备的ID。
STR_KEY_SUBDEV_TYPE	"sub_devtype"	字符串	子设备类型，用于网关动态绑定¥解绑子设备。
STR_KEY_SUBDEV_ID	"sub_devid"	字符串	子设备的ID，用于网关动态绑定¥解绑子设备。
STR_KEY_TIME	"time"	整型	时间戳(seconds since the Epoch)。
STR_KEY_URL	"url"	字符串	url，如固件的网址。

宏	key	类型	意义
STR_KEY_VERSION	"version"	字符串	版本号。
STR_KEY_RESULT	"result"	布尔值	命令的执行结果。
STR_KEY_MSG	"msg"	字符串	信息，用来表示命令执行结果相关的信息。

除了预定义的key，可根据需求自定义其他key

3. 透传数据格式

不同于基本消息的Key/Value字符串格式，透传数据使用自定义的二进制数据格式，用来满足不同设备的不同需求。如数据量比较大时，可以发送压缩后的数据。注意，服务器是无法直接解析透传数据的。透传数据的头部需要需要包含一下字段：

命令	意义
version	版本号。
save	标志服务器是否保存该数据。
reserved	保留位。
magic	起始位。

> 透传数据的头部定义请参考：common/client.h

二、设备发送消息给服务器(数据上行)

设备通过发布预定义的MQTT主题(topic),把消息(message)发送给服务器,下面是预定义的MQTT主题。

1. STR_TOPIC_ONLINE (/d2s/${owner}/${devtype}/${devid}/online)

Key	Value
功能	设备上线时，通知服务器更新状态。
发布方	设备。
订阅方	设备服务器或其它客户端。
消息内容	可为空，也可附加设备的状态信息，如固件版本、型号、IP和MAC地址。
回应消息	返回服务器当前时间(可用于设备同步时间)。

4. STR_TOPIC_OFFLINE (/d2s/${owner}/${devtype}/${devid}/offline)

Key	Value
功能	设备下线时，通知服务器更新状态。设备异常断电掉线时，服务器会主动监测设备已经下线。
发布方	设备。
订阅方	设备服务器或其它客户端
消息内容	无。
回应消息	无。

MQTT遗言（LWT）使用本消息。

5. STR_TOPIC_REPORT_STATUS (/d2s/${owner}/${devtype}/${devid}/status)

Key	Value
功能	设备向服务器上报设备的状态信息，如SIM的有效期和剩余流量等等。设备在升级固件之后，也需要通过本消息上报给服务器。
发布方	设备。
订阅方	设备服务器或其它客户端。
消息内容	设备状态信息。
回应消息	无。

STATUS类的消息在服务器上不记录历史，只保存当前状态。

6. STR_TOPIC_REPORT_DATA (/d2s/${owner}/${devtype}/${devid}/data)

Key	Value
功能	设备向服务器上报设备的数据，如传感器采用到的数据。上报的数据内容必须符合device_schema中定义。
发布方	设备。
订阅方	设备服务器或其它客户端。
消息内容	时间戳及数据。
回应消息	无。

DATA类的消息在服务器上会记录历史，可以查看整个数据的变化过程。

7. STR_TOPIC_REPORT_LOG (/d2s/${owner}/${devtype}/${devid}/log)

Key	Value
功能	设备向服务器上报日志信息。上报的日志内容必须符合device_schema中定义。
发布方	设备。
订阅方	设备服务器或其它客户端。
消息内容	时间戳、日志的名称及相关参数。
回应消息	无。

8. STR_TOPIC_REPORT_WARNING (/d2s/${owner}/${devtype}/${devid}/warning)

Key	Value
功能	设备向服务器上报警告信息，如温度偏高。上报的告警内容必须符合device_schema中定义。
发布方	设备。
订阅方	设备服务器或其它客户端。
消息内容	时间戳、警告的名称及相关参数。
回应消息	无。

9. STR_TOPIC_REPORT_ERROR (/d2s/${owner}/${devtype}/${devid}/error)

Key	Value
功能	设备向服务器上报错误信息，如温度过高。上报的错误内容必须符合device_schema中定义。
发布方	设备。
订阅方	设备服务器或其它客户端。
消息内容	时间戳、错误的名称及相关参数。
回应消息	无。

10. STR_TOPIC_REPORT_RESULT (/d2s/${owner}/${devtype}/${devid}/result)

Key	Value
功能	设备向服务器上报命令执行的结果。 设备收到的命令的参数中带有cmdid时，才需要上报执行的结果。
发布方	设备。
订阅方	设备服务器或其它客户端。

Key	Value
消息内容	cmdid、结果和详情。
回应消息	无。

11. STR_TOPIC_ADD_BIND (/g2s/${owner}/${devtype}/${devid}/bind)

Key	Value
功能	网关设备绑定节点设备（自动创建相应的节点设备）。
发布方	网关设备。
订阅方	设备服务器。
消息内容	节点信息。
消息示例	"¥0sub_devtype¥0yourtype¥0sub_devid¥0yourid¥0"（也可附加设备的状态信息等）。
回应消息	绑定子设备成功时返回"OK"。

11. STR_TOPIC_REMOVE_UNBIND (/g2s/${owner}/${devtype}/${devid}/unbind)

Key	Value
功能	网关设备解绑节点设备（只是解除绑定关系，不会移除节点设备）。
发布方	网关设备。
订阅方	设备服务器。
消息内容	节点信息。
消息示例	"¥0sub_devtype¥0yourtype¥0sub_devid¥0yourid¥0"。
回应消息	解绑子设备成功时返回"OK"。

12. STR_TOPIC_REPORT_RAW (/d2s/${owner}/${devtype}/${devid}/raw)

Key	Value
功能	设备向服务器上报透传数据。
发布方	设备。
订阅方	设备服务器或其它客户端。
消息内容	透传数据除了固定的消息头外，内容格式无限制。
回应消息	无。

透传数据的头部标识决定服务器是否保存该数据；服务器无法直接解析透传数据，如果要解析透传数据，需要针对每种设备定义相应的解析脚本，并上传到服务器端。

四、服务器发送消息给设备(数据下行)

1. STR_TOPIC_DEVICE (/s2d/d/${owner}/${devtype}/${devid})

Key	Value
功能	服务器向指定的设备发送命令。
发布方	设备服务器或其它客户端。
订阅方	设备。
消息内容	命令+命令ID（可选）+命令参数。
回应消息	消息携带cmdid时，需要通过STR_TOPIC_REPORT_RESULT主题返回命令执行结果；没有cmdid时，则无需回应。

> 消息格式，如："¥0name¥0cmdname¥0cmdid¥0id¥0key1¥0val1¥0key2¥0val2¥0"。

系统内置命令：

- STR_CMD_SET_CONFIG ("set_config")
  功能：配置设备参数。
  参数内容：参数名称+参数值。
- STR_CMD_SYNC_TIME ("sync_time")
  功能：时间同步，用来配置设备的时间，服务器每天与设备同步一次时间。
  参数内容：STR_KEY_TIME为服务器的时间。
- STR_CMD_NOTIFY_UPGRADE ("notify_upgrade")
  功能：通知设备有新固件，设备收到通知后，可提示用户有更新但不执行更新，也可后台自动更新固件。
  参数内容：STR_KEY_URL为申请固件下载地址的url，STR_KEY_VERSION为版本，
  STR_KEY_DEVICE_TYPE为设备类型。
- STR_CMD_EXEC_UPGRADE ("exec_upgrade")
  功能：通知设备更新固件，设备收到通知后，需立即更新固件。
  参数内容：同notify_upgrade的参数内容。
- STR_CMD_REQ_REPORT ("req_report")
  功能：请求设备上报最新的数据。 参数内容：无。
- STR_CMD_PASS_THROUGH ("pass_through")
  功能：发送透传消息给设备。 参数内容：base64加密的字符串。

除了上述预定义的命令，可以根据需求自定义命令，命令的定义参考device_schema。

2. STR_TOPIC_GATEWAY (/s2g/d/${owner}/${devtype}/${devid}/${sub_devtopic})

Key	Value
功能	服务器向指定网关的指定节点发送命令（**sub_devtopic**代表一个完整的设备主题）。
发布方	设备服务器或其它客户端。
订阅方	设备。
消息内容	命令+命令ID（可选）+命令参数。
回应消息	无。

3. STR_TOPIC_DEVICE_RAW (/s2d/d/${owner}/${devtype}/${devid}/raw)

Key	Value
功能	服务器向指定的设备发送原始数据。
发布方	设备服务器或其它客户端。
订阅方	设备。
消息内容	原始数据。
回应消息	无。

4. STR_TOPIC_BROADCAST (/s2d/b/all)

Key	Value
功能	服务器向所有设备发送广播命令。
发布方	设备服务器或其它客户端。
订阅方	设备。
消息内容	命令+命令参数。
回应消息	无。

5. STR_TOPIC_BROADCAST_DEVTYPE (/s2d/b/${devtype})

Key	Value
功能	服务器向指定设备类型的设备发送广播命令。
发布方	设备服务器或其它客户端。
订阅方	设备。
消息内容	命令+命令参数。
回应消息	无。

批量升级固件可以使用本主题。

6. STR_TOPIC_BROADCAST_OWNER_DEVTYPE (/s2d/b/${owner}/${devtype})

Key	Value
功能	服务器向指定用户的指定设备类型的设备发送广播命令。
发布方	设备服务器或其它客户端。
订阅方	设备。
消息内容	命令+命令参数。
回应消息	无。

批量升级固件也可以使用本主题。

五.MQTT登录

设备登录

设备类型

- 全局设备：使用服务器内置密码登录，因为是全局密码，所以必须保证密码的安全性。
- 自定义设备：使用创建设备时返回的devsecret作为密码登录。

登录流程

- 登录认证服务器https://auth.zlgcloud.com:8143/login：
  1. 设置用户名、密码和设备信息。其中，**用户名**为设备类型，**密码**为设备密钥。
  2. 登录获取MQTT的相关信息，如设备的持有者和地址、MQTT服务器地址和端口以及唯一的Token。
- 连接到MQTT服务器：
  1. 设置MQTT连接参数。其中，**登录密码**为认证服务器返回的Token，**ClientId**为格式化字符串"${devtype}：${devid}"的值。
  2. 连接到服务器，成功即可按协议通讯。

当连接断开重连时，需要根据情况（如设备欠费被服务器强制下线）重新获取Token。

六、其它

设备状态的更新

状态中可以包含JSON数据。type指定为『string』，format指定为『json』即可。可以在schema中加入json具体的格式描述(服务器不做检查，仅为APP开发时提供参考)。如：

```
"CfgInfo":{
   "comment": "CAN配置",
   "type": "string",
   "format":"json",
   "schema" : {
    ...
```

```
    }
},
```

服务器支持更新JSON的部分内容。更新部分内容时，key使用『.』分隔的路径，该路径下的值被替换成新的值(不是合并)。

假设状态中有settings的格式为JSON，我们看看如何部分更新：

- 更全部内容

```
key: settings
value:
{
  "can": {
    "chn1": {
      "Mode": 1,
      "Baud": 115200
    },
    "chn2": {
      "Mode": 2,
      "Baud": 115200
    }
  },
  "di": {
    "chn1": {
      "Mode": 1,
      "Baud": 115200
    },
    "chn2": {
      "Mode": 2,
      "Baud": 115200
    }
  }
}
```

- 更全部内容(原有的di字段会被删除)

```
key: settings
value:
{
  "can": {
    "chn1": {
      "Mode": 1,
      "Baud": 115200
    },
    "chn2": {
      "Mode": 2,
      "Baud": 115200
    }
```

```
  }
}
```

- 只更新settings.can的内容(原有的di字段的内容不变)

```
key: settings.can
value:
{
  "chn1": {
    "Mode": 1,
    "Baud": 115200
  },
  "chn2": {
    "Mode": 2,
    "Baud": 115200
 }
}
```

- 只更新settings.can.chn1的内容

```
key: settings.can.chn1
value:
{
  "Mode": 1,
  "Baud": 115200
}
```

- 只更新settings.can.chn1.Mode的内容

```
key: settings.can.chn1.Mode
value: 1
```