

4G 网关 DLT645 配置说明

一、版本适配说明

说明：本篇文档所适用的网关型号为：**XY-3422**。使用的配置工具的最低版本为 **v1.1.4**。

二、概览。

说明：目前 DLT645 配置只支持配合 mqtt 通道使用。

DLT645 配置与数据流模板的关系：上行数据流先经过 DLT645 解析，然后再经过数据流模板处理；下行数据流先经过数据流模板解析，再经过 DLT645 解析。

配置页面如下图所示：

< 道参数 TCP服务器 预置信息 自动采集任务 GPIO GPS 数据流 预警 任务 ModBus **DLT645** HJ212 >

注：此部分需单独配置，暂只支持配合MQTT通道使用。

↓ 导入JSON文件 ↑ 导出JSON文件 写入配置 读取配置

是否启用: 启用 不启用

绑定串口id: 1 2 3

读表方式: 周期读表 定时读表

定时读表: 选择时间

指令下发间隔: 1000 (单位: ms)

指令超时重发次数: - 0 +

指令超时重发间隔: 300 (单位: s)

唤醒码: 0xFE

唤醒码长度: - 4 +

广播校时: 每天一次 不启用

时间戳: 添加 不添加

上报方式: 整包上传 按设备上报

全部重置

读数据 =====>>>

其他指令 =====>>>

图 1 DLT645 配置

通过配置 DLT645 解析参数，可以实现：网关自动下发 DLT645 读数据指令，并将电表的 DLT645 响应数据转化为 JSON 格式数据包，包内数据的属性名由用户自定义。

三、配置项说明。

1、DLT645 配置



图 2 DLT645 配置项

- **绑定串口 id:** 本串口配置对应的串口 ID。
- **读表方式:**
 - 周期读表：每隔一定时间进行读表操作。
 - 定时读表：指定一个或多个时刻进行读表操作。
- **轮询周期:**（读表方式为“周期读表”）周期读表中两次下发同一条指令之间的间隔。
- **定时读表:**（读表方式为“定时读表”）选择要读表的时刻，仅支持“hh:MM”的格式。
- **指令下发间隔:** 每次轮询中的两条指令之间的时间间隔。
- **指令超时重发次数:**（读表方式为“定时读表”）读表失败后重试读表的次数。
- **指令超时重发间隔:**（读表方式为“定时读表”）读表失败后间隔时长重试读表。
- **唤醒码:** 每条指令前都要添加一串唤醒码，用于唤醒电表。
- **唤醒码长度:** 一般为 4 字节，对于不需要唤醒码的电表，则填 0。
- **广播校时:** 网关联网同步时间后，每隔 24 小时下发一次广播校时指令。
- **时间戳:** 是否在上报的 JSON 包中添加时间戳属性“ts”。若添加，则 JSON 包变为：
`{"addr":"xxxxxxxxxx", "ts":xxxxxx, "data":{ ... } }`。
- **解析方式:** 整包上传——读完所有表的数据后，上报本轮读表中解析的全部数据。
按设备上报——每读完一个表的数据，立即上报该表解析的数据。
- **指令操作**

注：配置读写操作时，“新建一行”即为新增一条指令，读操作中的所有行作为一轮指令。

（一）读数据



图 3 读操作配置表

点击“新增一行”后，添加一行指令配置行。

- (1) 协议：DLT645 协议版本，支持 645-07 协议。
- (2) 表地址：电表的地址（表号）。
- (3) 状态：若为停用，则网关不会下发本组指令。
- (4) 操作：新建解析——在本行（本组指令）下的子表中新增一条指令与解析。
 停用/启用——停用或启用本组指令。
 删除——删除本组指令。

子表是对查询数据的选择并对电表响应数据的解析，每个查询数据都会下发一条读取指令：

上图中的读数据表会生成两条（一组）**DLT645-07 协议**的查询指令，分别为（16 进制）：

- FE FE FE FE 68 01 00 00 00 00 00 68 11 04 33 34 34 35 B6 16 --读取 A 相电压
- FE FE FE FE 68 01 00 00 00 00 00 68 11 04 33 35 34 35 B7 16 --读取 B 相电压

FE FE FE FE —— 唤醒码，4 个字节

68 —— 帧起始符，1 个字节

01 00 00 00 00 00 —— 电表地址，12 位 BCD 码

68 —— 帧起始符，1 个字节

11 —— 控制码，0x11 表示“读数据”，1 个字节

04 —— 数据区长度，1 个字节

33 34 34 35 —— 数据标识（每个字节加 0x33 扰码，低字节在前，高字节在后），n 个字节

B6 —— 帧校验码，1 个字节

16 —— 帧结束符，1 个字节

那么电表响应 A 相电压的数据为（16 进制）：

- 68 01 00 00 00 00 00 68 91 02 57 55 10 16

68 —— 帧起始符，1 个字节

01 00 00 00 00 00 —— 电表地址，12 位 BCD 码

68 —— 帧起始符，1 个字节

91 —— 控制码，将 0x11 控制码的 bit7 变为 1，表示电表的响应帧，1 个字节

02 —— 数据区长度，1 个字节

57 55 —— 数据区（低字节在前），n 个字节，转成高字节在前并减去 0x33 扰码后为 22 24（16 进制），则采集值即为 2224（10 进制），则 A 相电压值为 222.4v

10 —— 帧校验码

16 —— 帧结束符

（1）属性名：所查询数据对应的属性名，用户自定义，即为上报到服务器的 JSON 包中的属性名。

（2）数据标识：645-07 协议规约中定义的数据标识。

（3）比例系数：电表采集值和真实值的比值。如 采集值：真实值 = 10:1，那么比例填 10。

（4）偏移量：采集值按比例转化后，可能由某些因素导致和实际值有微小的误差，可填负数，最终上报的值为：（采集值 / 比例系数）+ 偏移量。

（5）状态：是否解析这个值，若停用，则不会下发查询该数据的指令。

解析完成后上报到服务器的 JSON 包的格式为（不添加时间戳）：

```
{
  "addr": "表号",
  "data": {
    "属性名 1": "值 1",
    "属性名 2": "值 2",
    "属性名 3": "值 3",
    ....
  }
}
```

如果对上报的数据有其他要求，可以在“发送数据流模板”中对这个 JSON 数据包进行进一步的处理。

（二）其他指令

目前支持通过写数据的方式进行设备校时，当设备不识别广播校时指令时，可通过此方式进行校时。写数据校时指令的下发频率为每天一次。

其他指令 =====>>> ∨

	协议	表地址	状态	操作
收起	DLT645/07协议	AAAAAAAAAAAA	启用	新建指令 停用 删除
指令				
指令	密码权限	密码	操作者代码	操作
写数据校时	02	*****	00000000	删除

新增一行

图 4 其他指令配置表格

四、配置示例

以“三相物联采控终端”为例，设备同时支持 modbus-RTU 协议和 645-07 协议。设备的 645 协议地址为 000000000001。

现需采集 A 相电压、A 相电流、瞬时 A 相有功功率等，数据上传到服务器，A 相电压字段名定义为“ua”，A 相电流字段名定义为“ea”，瞬时 A 相有功功率字段名定义为“pa”。下面开始填入配置项。

是否启用: 启用 不启用

绑定串口id: 1 2 3

读表方式: 周期读表 定时读表

轮询周期: (单位: s)

指令下发间隔: (单位: ms)

唤醒码:

唤醒码长度:

广播校时: 每天一次 不启用

时间戳: 添加 不添加

上报方式: 整包上传 按设备上报

读数据 =====>>>

收起	协议	表地址	状态	操作	
	DLT645/07协议	000000000001	启用	新建解析 停用 删除	
属性名	数据标识	比例	偏移	状态	操作
ua	变量 / 电压 / A相 <input type="checkbox"/> 自定义	10	0	启用中	停用 删除
ea	变量 / 电流 / A相 <input type="checkbox"/> 自定义	1000	0	启用中	停用 删除
pa	变量 / 瞬时有功... <input type="checkbox"/> 自定义	10000	0	启用中	停用 删除

新增一行

1、是否启用

启用。

2、绑定串口 ID

网关型号为 X_Y3422，有 3 个 RS485 串口，现在使用第二个 485 串口进行接入，对应为串口 2，因此串口 id 选 2。

3、读表方式

这里使用周期读表的方式。

4、轮询周期

用户自定义。这里希望每 60 秒读取一次电表数据，因此填 60。

5、指令下发间隔

所有指令下发时间总和不能超过轮询周期，这里会下发 3 条指令，填 200ms~2000ms 均可，

不建议间隔时间过久。

6、唤醒码

本电表的指令帧需要携带唤醒码，为 0xFE。

7、唤醒码长度

本电表需要的唤醒码的长度为 4 个字节。

8、广播校时

启用后，网关会每天一次下发广播校时指令。

9、时间戳

用户自定义。是否在组合的 JSON 内添加时间戳属性“ts”。

10、上报方式

用户自定义。

11、读数据

- **协议：**本电表使用的是 645-07 协议，因此选择“DLT645/07 协议”。
- **表地址：**本电表的 645 协议地址为“000000000001”。
- **属性名：**用户自定义（勿填中文）。
- **数据标识：**DLT645-07 协议规约中定义的数据标识。
- **比例：**一般按照 DLT645-07 协议规约中定义的数据标识的数据格式来填写。
- **偏移：**采集值进行计算后，与实际值的偏移量。

到这里配置就完成了。将配置写入网关，可以看到，每隔 60 秒，网关会发送三条 645 协议的查询指令。



```
>[ 2023-08-01 14:06:19:298 -- 接收 ][ 20字节 ]:  
FE FE FE FE 68 01 00 00 00 00 00 68 11 04 33 34 34  
35 B6 16  
  
>[ 2023-08-01 14:06:20:292 -- 接收 ][ 20字节 ]:  
FE FE FE FE 68 01 00 00 00 00 00 68 11 04 33 34 35  
35 B7 16  
  
>[ 2023-08-01 14:06:21:298 -- 接收 ][ 20字节 ]:  
FE FE FE FE 68 01 00 00 00 00 00 68 11 04 33 34 36  
35 B8 16
```

网关会将电表返回的数据最终解析为：

```
{  
  "addr": "000000000001",  
  "ts": 1690785014,  
  "data": {  
    "ua": 224.2,  
    "ea": 0.45,  
    "pa": 101.21  
  }  
}
```

并将此 JSON 数据上传到用户的 mqtt 平台。

用户还可以将这页的配置导出为一个 JSON 文件，以便后续配置相同设备时，直接导入这个 JSON 文件即可。