

InPlant SCADA

MQTT 应用指导

声 明

- 严禁转载本手册的部分或全部内容。
- 在不经预告和联系的情况下，本手册的内容有可能发生变更，请谅解。
- 本手册所记载的内容，不排除有误记或遗漏的可能性。如对本手册内容有疑问，请与我公司联系，联系邮箱：SMS@supcon.com。

商 标

中控、SUPCON、PLANTMATE、AI-POET、InPlant、dOps、ESP-iSYS、Webfield、ics、MultiF、SupField、APC 等均是中控技术股份有限公司注册商标，拥有商标的所有权。未经中控技术股份有限公司的书面授权，任何个人及企业不得擅自使用上述商标。对于非法使用我司商标的行为，我司将保留依法追究行为人及企业的法律责任的权利。

文档标志符定义

	警告：标示有可能导致人身伤亡或设备损坏的信息。 WARNING: Indicates information that a potentially hazardous situation which, if not avoided, could result in serious injury or death.
	电击危险：标示有可能产生电击危险的信息。 RISK OF ELECTRICAL SHOCK: Indicates information that Potential shock hazard where HAZARDOUS LIVE voltages greater than 30V RMS, 42.4V peak, or 60V DC may be accessible.
	防止静电：标示防止静电损坏设备的信息。 ESD HAZARD: Indicates information that Danger of an electro-static discharge to which equipment may be sensitive. Observe precautions for handling electrostatic sensitive devices
	注意：提醒需要特别注意的信息。 ATTENTION: Identifies information that requires special consideration.
	提示：标记对用户的建议或提示。 TIP: Identifies advice or hints for the user.

目 录

1 概述	1
2 配置 MQTTClient	1
2.1 打开 MQTTClient 界面	1
2.2 配置 MQTTClient 参数	3
2.2.1 配置服务参数	3
2.2.2 配置全局属性	4
2.2.3 配置实时数据主题参数	5
2.2.4 配置历史数据主题参数	6
2.2.5 配置报警主题参数	6
2.3 开启连接	6
3 查询位号数据	6
3.1 发布和订阅实时数据	7
3.1.1 支持的位号类型	7
3.1.2 数据格式	7
3.2 查询历史数据	11
3.2.1 支持的位号类型	11
3.2.2 数据格式	11
4 查询报警信息	13
4.1 订阅实时报警消息	13
4.1.1 订阅实时报警消息的格式	13
4.1.2 返回数据的格式	14
4.2 查询历史报警消息	15
4.2.1 查询响应时间	15
4.2.2 查询历史报警消息的格式	15
4.2.3 返回数据的格式	16
5 错误提示和操作建议	17
6 资料版本说明	18

MQTT 应用指导

1 概述

InPlant SCADA 支持通过 MQTT（Message Queuing Telemetry Transport，消息队列遥测传输）协议向外发布实时数据、历史数据和报警信息，并支持修改实时数据。

启动 InPlant SCADA 监控、配置 MQTTClient 的参数并成功建立连接后，软件可以通过 MQTTClient 将数据推送到 MQTT Broker 代理服务器，供第三方客户端订阅。

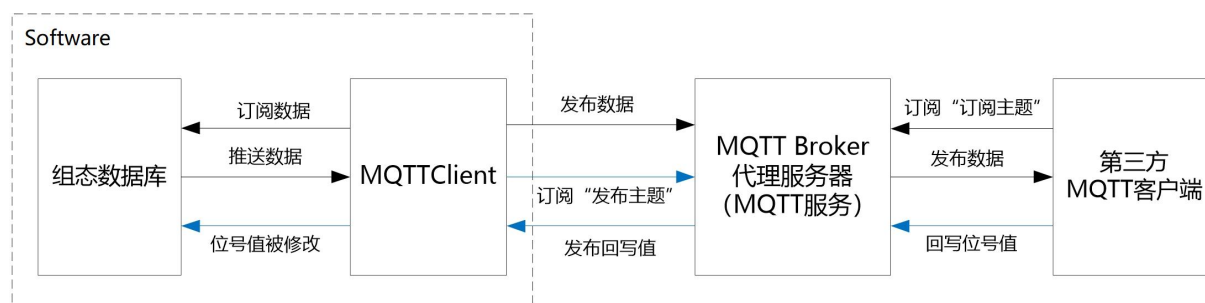


图 1-1 MQTT 应用流程



提示：

除了通过 MQTT 服务对外发布数据，您也可以通过 MQTT 驱动向 MQTT Broker 代理服务器订阅数据。关于 MQTT 驱动的详细说明，请参见《IO 驱动使用手册》的“MQTT 驱动”章节。

2 配置 MQTTClient

本章主要说明 InPlant SCADA 中配置 MQTT 客户端属性的步骤。

2.1 打开 MQTTClient 界面

1. 在工程管理软件中打开全局设置界面，选择“启动选项”页签，勾选“启动 MQTT”，如下图所示，MQTTClient 会随监控软件启动。

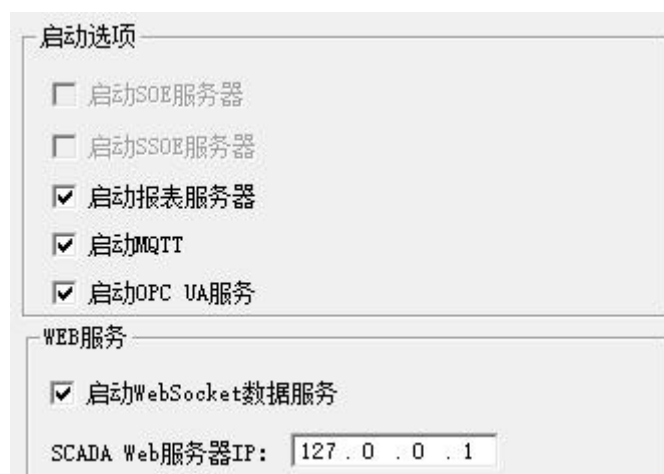


图 2-1 启动 MQTT

2. 启动监控并登陆后，在系统托盘处右键单击 ，选择“属性”，打开 MQTTClient。

2.2 配置 MQTTClient 参数

图 2-2 MQTTClient 服务页

MQTTClient 的界面如上图所示。以下分别说明各页签的参数配置要求。每次修改配置后，请单击下方“保存”按钮。

2.2.1 配置服务参数

“服务”页如图 2-2 所示，参数说明如下：

- 按 MQTT Broker 代理服务器的参数，填写地址、端口、用户名和密码。

- 填写地址和端口后，单击“生成”按钮，生成客户端 ID，以便于 MQTT Broker 代理服务器识别当前 MQTTClient。

2.2.2 配置全局属性

MQTT VxMQTTClient

服务 全局属性 实时数据主题 历史数据主题 报警主题

MQTT 版本: Default MQTT

☒ 变化推送

☐ 周期全部推送 30 秒

* 开启周期全部推送时，变化推送的同时，按照设定的时间周期推送全部数据。

☐ 周期推送

推送周期: 5 秒

* 按照设定时间，周期性的推送全部数据。

状态

连接状态: 未连接

启动 保存

图 2-3 MQTTClient 全局属性页

“全局属性”页参数如上图所示，参数说明如下：

- MQTT 版本：根据 MQTT Broker 代理服务器支持的协议选择。本软件支持的 MQTT 协议版本为 Default MQTT、3.1、3.1.1 或 5.0。
- 推送方式：可设置变化推送或周期推送，默认为变化推送。

- 变化推送：当位号值出现变化时，会推送变化位号；变化推送时开启周期全部推送，变化推送的同时，按照设定的时间周期向代理服务器推送全部位号的实时值。
- 周期推送：可配置推送周期，按配置向代理服务器推送全部位号的实时值，时间范围为 5~3600 秒。

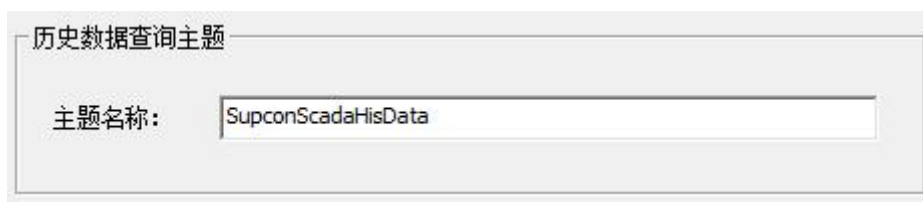
2.2.3 配置实时数据主题参数

图 2-4 MQTTClient 实时数据主题页

“实时数据主题”页参数如上图所示，参数说明如下：

- 发布主题：第三方客户端想要获取本地库的位号实时值时，需要订阅“发布主题”。此时，MQTTClient 中需要设置订阅主题及主题中包含的位号，以便于获取位号实时值。
 - 勾选“开放所有位号”并设置主题名。当第三方客户端订阅主题时，会获取所有位号的实时数据。
 - 如果只希望开放部分指定位号的数据，请勾选“自定义开放位号”，然后配置主题和开放的位号清单：“添加”按钮，新增一个主题，默认主题名为“Topic+数字”；选中主题后单击“重命名”按钮，可以修改主题名，支持英文字符、数字、下划线、斜杠、加号、减号和井号，不超过 32 个字符。选中一个主题，在右侧单击“添加”按钮，在弹出的位号选择器中选择需要订阅的位号。不同订阅主题可以包含相同位号。
- 订阅主题：当位号读写属性为可读写时，第三方客户端可以通过 MQTTClient 向位号写值。此时，需要订阅该主题。此处用于设置订阅主题的名称，支持英文字符、数字、下划线、斜杠、加号、减号和井号，不超过 32 个字符。

2.2.4 配置历史数据主题参数



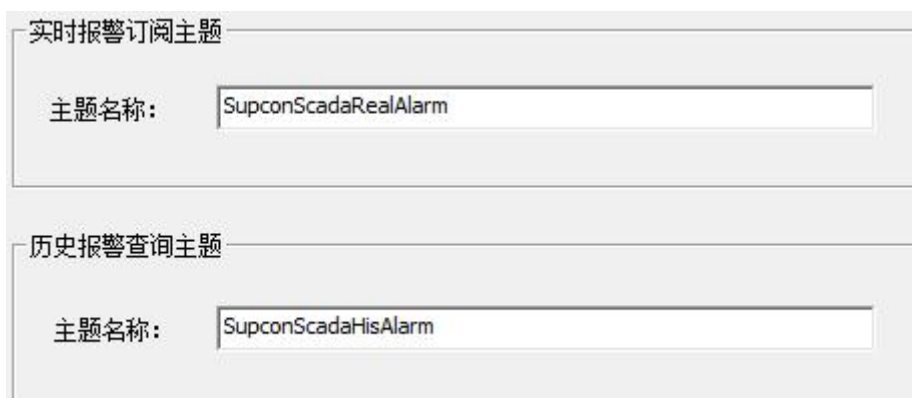
历史数据查询主题

主题名称:

图 2-5 MQTTClient 历史数据主题页

“历史数据主题”页参数如上图所示，需设置历史主题。主题名称支持英文字符、数字、下划线，不超过 32 个字符。

2.2.5 配置报警主题参数



实时报警订阅主题

主题名称:

历史报警查询主题

主题名称:

图 2-6 MQTTClient 报警主题

“报警主题”页参数如上图所示，需设置实时报警或历史报警的主题。主题名称支持英文字符、数字、下划线，不超过 32 个字符。

2.3 开启连接

完成配置后，单击“启动”按钮，进行服务连接。连接成功后，下方“连接状态”显示已连接，本地库开始向 MQTT 代理服务器推送数据。同时，“启动”按钮变为“重启”按钮。

每次修改属性参数后，都需要重启 MQTT。

如果 MQTTClient 和 MQTT broker 代理服务器因网络原因断开连接，等网络恢复正常后，MQTT 连接会自动重连，重连成功后连接状态会变为“已连接”。

3 查询位号数据

第三方客户端订阅对应的主题后，可以获取位号的实时数据；发送查询消息后，可以获取历史数据。本章主要说明可查询到的位号类型和需要遵循的 JSON 数据格式。

3.1 发布和订阅实时数据

第三方客户端订阅实时数据主题，即可获取位号的实时数据。实时数据的位号数量上限为 5 万点。

3.1.1 支持的位号类型

MQTT 通信支持的实时数据类型如表 3-1 所示。

表 3-1 实时数据类型表

位号类型	类型值	数据类型	备注
开关量	1	BOOL	取值范围 0 或 1
字符串	2	CHAR	用以表示 ASCII 字符
整型	3	BYTE	8 位字符串
	4	WORD	16 位字符串
	5	DWORD	32 位字符串
	6	LWORD	64 位字符串
	7	INT8	8 位有符号整型，取值范围[-127, 127]
	8	UINT8	8 位无符号整型，取值范围[0, 255]
	9	INT16	16 位有符号整型，取值范围[-32767, 32767]
	10	UINT16	16 位无符号整型，取值范围[0, 65535]
	11	INT32	32 位有符号整型，取值范围[-2e16, 2e16]
	12	UINT32	32 位无符号整型，取值范围[0, 2e32]
	13	INT64	64 位有符号整型，取值范围[-2e32, 2e32]
	14	UINT64	64 位无符号整型，取值范围[0, 2e64]
实型	15	FLOAT	单精度浮点
	16	DOUBLE	双精度浮点
字符串	17	STRING	字符串，NULL 终结

3.1.2 数据格式

当第三方客户端通过 MQTT 通信收发数据时，得到的返回数据和发送的写值消息都需要遵循指定的格式，如下表所示。在返回的实时数据消息中，“__DATA__”会被替换为 [SingleDataSchema] 里的内容。“__TAGNAME__”、“__TAGVALUE__”、“__DATATYPE__”、“__QUALITY__”、“__TIMESTAMP__”分别代表实际的位号名、位号值、数据类型、质量码和时间戳。同时也支持位号描述、单位、报警限值、报警使能和报警状态点域，关键字为“__点域名__”，详见图 3-3。接收

实时数据时，推荐您使用默认消息格式。如果想要修改消息格式，可以在工程运行目录下找到配置文件 MQTTRTDataJsonSchema.conf 进行修改（默认路径为 D:\ECSRun\工程名\HMI\子工程\Run，此文件默认不存在，首次启动后默认生成，或者手动新建），修改后需要重启软件的 MQTT 服务。详细说明请咨询中控工程师。

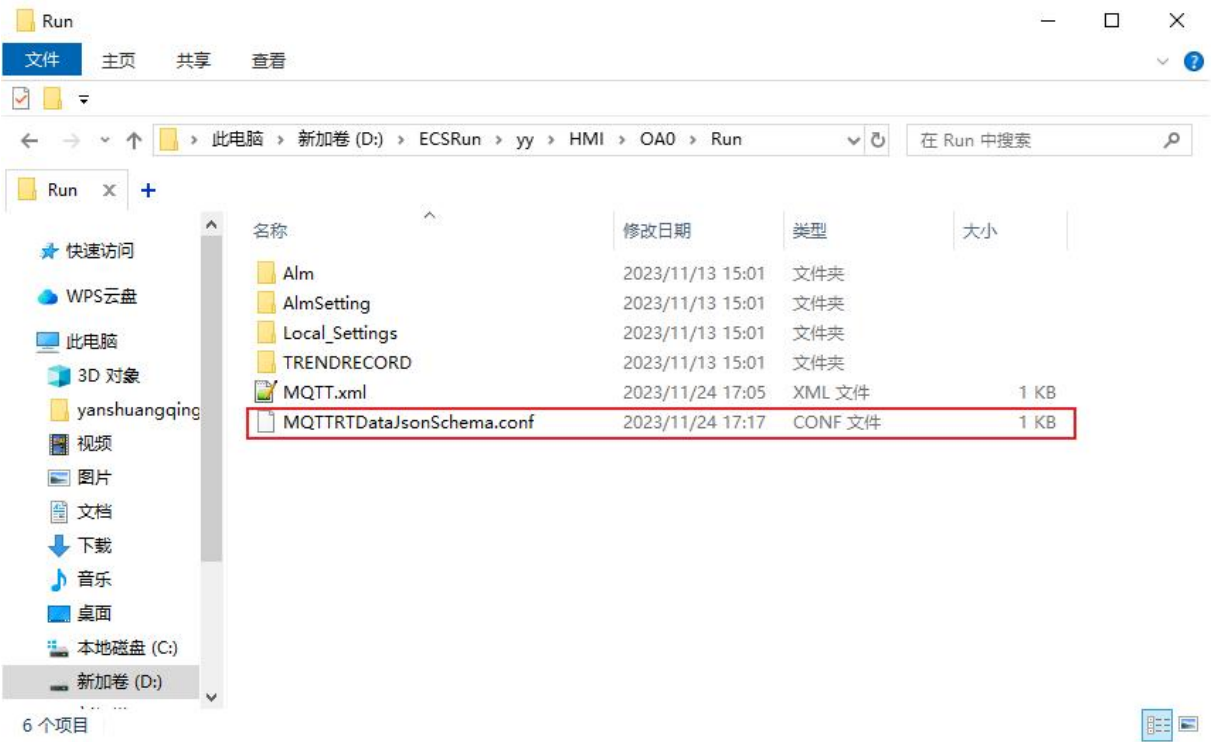


图 3-1 MQTT Json 配置文件位置

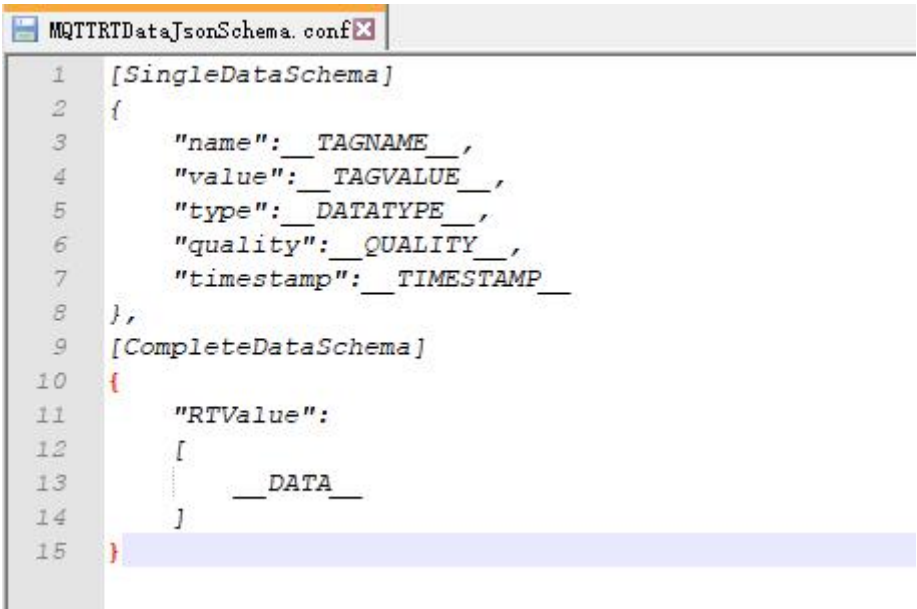
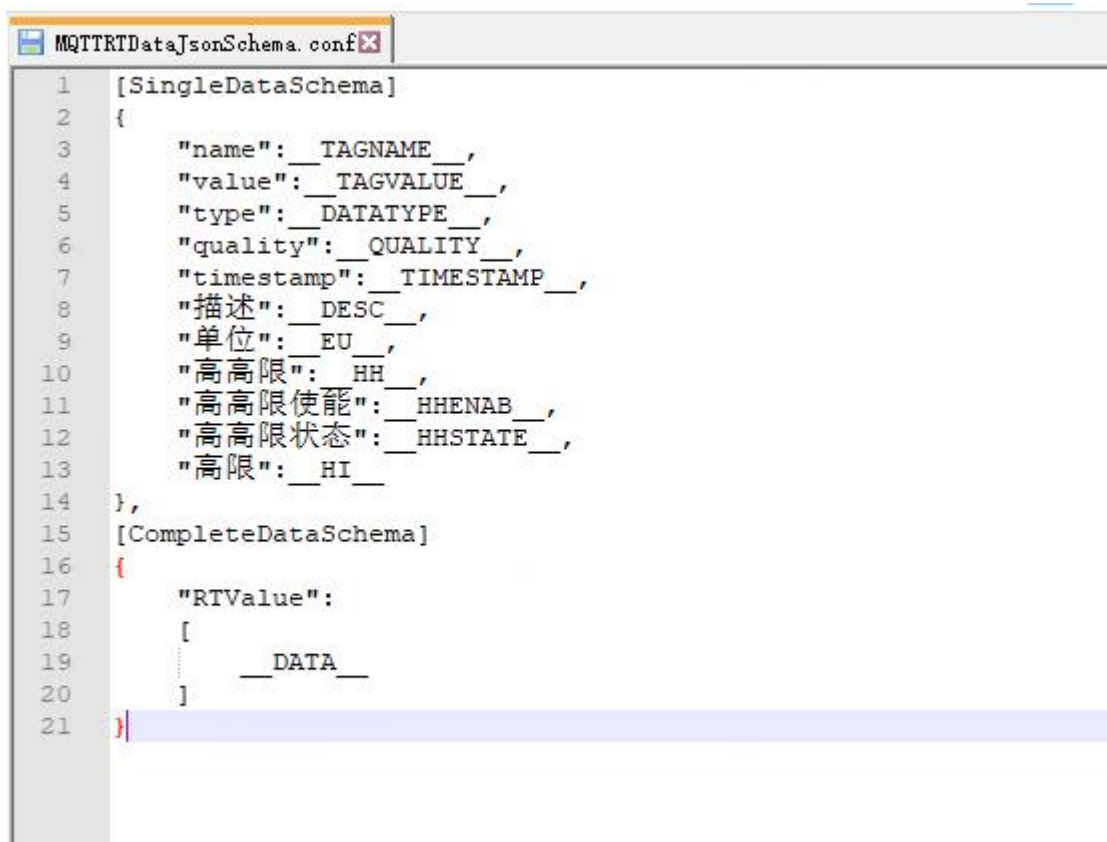


图 3-2 MQTTRTDataJsonSchema.conf 文件默认内容



```
1 [SingleDataSchema]
2 {
3     "name": __TAGNAME__,
4     "value": __TAGVALUE__,
5     "type": __DATATYPE__,
6     "quality": __QUALITY__,
7     "timestamp": __TIMESTAMP__,
8     "描述": __DESC__,
9     "单位": __EU__,
10    "高高限": __HH__,
11    "高高限使能": __HHENAB__,
12    "高高限状态": __HHSTATE__,
13    "高限": __HI__
14 },
15 [CompleteDataSchema]
16 {
17     "RTValue":
18     [
19         __DATA__
20     ]
21 }
```

图 3-3 MQTTTRTDataJsonSchema.conf 支持点域

表 3-2 数据格式一览表

订阅实时数据的消息格式	返回的实时数据	写值消息格式
<pre>[SingleDataSchema] { "name": __TAGNAME__, "value": __TAGVALUE__, "type": __DATATYPE__, "quality": __QUALITY__, "timestamp": __TIMESTAMP__ }, [CompleteDataSchema] { "RTValue": [__DATA__] }</pre>	<pre>{ "RTValue": [{ "name": "tag1", "quality": 1, "timestamp": 1331778905197, "type": 16, "value": "1.000000" }, { "name": "tag2", "quality": 1, "timestamp": 1331778905197, "type": 16, "value": "1.000000" }] }</pre>	<pre>{ "RTValue": [{ "name": "tag1", "type": 16, "value": "2" }, { "name": "tag2", "type": 16, "value": "2" }] }</pre>
<pre>[SingleDataSchema] __TAGNAME__: { "value": __TAGVALUE__, "type": __DATATYPE__, "quality": __QUALITY__, "timestamp": __TIMESTAMP__ }, [CompleteDataSchema] { "RTValue": { __DATA__ } }</pre>	<pre>{ "RTValue": { "tag1": { "quality": 1, "timestamp": 1331778905197, "type": 16, "value": "1.000000" }, "tag2": { "quality": 1, "timestamp": 1331778905197, "type": 16, "value": "1.000000" } } }</pre>	



注意：

位号必须为已订阅的位号，格式严格保持一致，“：”两侧无空格。

3.2 查询历史数据

第三方客户端查询历史数据时，需要向 MQTT 服务发送查询消息。每次查询时都需要重新发送消息。查询成功后，即可获取目前最新的历史数据。每次最多可以查询到 8 个位号，最多可以查询到每个位号的 5000 点数据。

3.2.1 支持的位号类型

MQTT 通信支持的历史数据类型如表 3-3 所示。

表 3-3 历史数据类型表

位号类型	类型值	数据类型
开关量	1	BOOL
8 位/16 位有符号整数	2	INT（16 位）、SINT（8 位）
8 位/16 位无符号整数	4	UINT（16 位）、USINT（8 位） WORD（16 位）、BYTE（8 位）
32 位有符号整数	5	DINT
32 位无符号整数	6	UDINT、DWORD
32 位浮点型单精度	7	REAL
64 位浮点数双精度	8	LREAL
64 位有符号整数	9	LINT
64 位无符号整数	11	ULINT、LWORD

3.2.2 数据格式

当第三方客户端通过 MQTT 通信收发数据时，需遵循约定格式。

查询历史数据消息的格式

```
{
  "method": "HistoryData",
  "topic": "5",
  "names": ["MEM1", "MEM2"],
  "seq": 2,
  "mode": 0,
  "begintime": 1676966248000,
  "endtime": 1676966308000,
  "count": 20,
  "interval": 1000,
  "timeout": 20000
}
```

表 3-4 查询主题参数说明表（历史数据）

参数项	参数说明	备注
method	查询历史数据标识	固定值 “HistoryData”
topic	第三方接收历史数据的主题	由第三方维护
names	位号名	-
mode	默认为 0，表示查历史数据	预留参数，请勿更改
seq	序号，代表一次查询标识，INT64 值	由第三方维护
count	查询数量	● 至少设置一个 ● 若二者同时设置，以 interval 为准
interval	时间间隔，最小为 1000 毫秒	
timeout	超时时间，默认值 25000 毫秒	-
begintime	开始时间戳，单位为毫秒	-
endtime	结束时间，单位为毫秒	-

返回数据的格式

```
{
  "method": "HistoryData",
  "code": 0,
  "msg": "success",
  "seq": 1236,
  "result": {
    "data": [
      {
        "name": "MEM1",
        "datalist": [
          {
            "q": 1,
            "time": 1673599170000,
            "type": 8,
            "val": 0
          }
        ]
      }
    ]
  }
}
```

表 3-5 返回数据参数说明表（历史数据）

参数项	参数说明
code	0 表示成功，其他值表示存在错误，详见“错误提示和操作建议”
msg	"success"表示成功
seq	一次请求的唯一标识，值为订阅消息的 seq 字段
result	返回结果对象，其中 data 数组为查询到的数据
name	查询的位号名
datalist	单个位号的历史数据，包括： ● q: 质量码 1 表示正常，0 表示位号 BAD ● time: 毫秒时间戳， ● type: 数据类型，详见表 3-3 ● val: 值

4 查询报警信息

第三方客户端向对应主题发送实时报警的订阅消息后，即可获取实时报警消息；发送历史报警的查询消息后，可以获取历史报警数据。获取的报警信息不包括事件报警。

本章主要说明订阅或查询报警信息时需要遵循的 JSON 数据格式。

4.1 订阅实时报警消息

第三方客户端发送一次订阅消息后，MQTT 服务会向第三方客户端实时推送实时报警数据；第三方客户端发送一次取消订阅消息后，MQTT 服务就会停止发送实时报警数据。本节主要说明消息的数据格式。

4.1.1 订阅实时报警消息的格式

```
{
  "method": "RealAlarm",
  "state": 1,
  "topic": "5"
}
```

表 4-1 订阅主题参数说明表（实时报警）

参数项	参数说明
method	实时报警的标识，固定值“RealAlarm”
state	0 表示订阅实时报警，1 表示取消订阅实时报警
topic	第三方接收实时报警的主题，由第三方维护，需要第三方订阅

4.1.2 返回数据的格式

```
{
  "method": "RealAlarm",
  "alarms": [
    {
      "name": "MEM1",
      "state": 0,
      "trigger": 51,
      "limit": 50,
      "operate": "admin",
      "results": "确认报警",
      "newtime": 1655368157,
      "acktime": 0,
      "cancelttime": 0,
      "level": 1,
      "type": "L",
      "almdesc": "低限报警",
      "desc": "位号描述"
    }
  ]
}
```

表 4-2 返回数据参数说明表（实时报警）

参数项	参数类型	参数说明
alarms	JSON 对象数组	报警数据
name	string	位号名
newtime	INT64	产生时间
acktime	INT64	确认时间
cancelttime	INT64	消除时间
state	int	报警状态。0 表示报警产生，1 表示报警已确认，2 表示报警已消除
trigger	string	报警触发值
limit	string	报警限值
operate	string	操作人
results	string	处理结果
level	string	报警等级（优先级）
type	string	报警类型（L3/LL/L 等）
desc	string	位号描述
almdesc	string	报警描述

4.2 查询历史报警消息

查询历史报警时，每次查询时需要重新发送查询消息。查询成功后，即可获取目前最新的历史数据。单次可查询最多 10000 条报警；每次最多推送 1000 条报警，超过 1000 条则会分批推送。

4.2.1 查询响应时间

查询历史报警的响应时间取决于历史报警数量和查询数量，详情请见下表。

表 4-3 响应时间的影响因素和参考结果

查询数量 历史报警总数	200 条	500 条	1000 条	5000 条	10000 条
50 万条	3 秒	4 秒	4 秒	6 秒	8 秒
100 万条	3 秒	5 秒	9 秒	10 秒	11 秒

4.2.2 查询历史报警消息的格式

```
{
  "method": "HistoryAlarm",
  "topic": "5",
  "seq": 0,
  "type": [
    "L",
    "LL"
  ],
  "names": [
    "MEM1",
    "MEM2"
  ],
  "eventtype": 0,
  "begintime": 1673761156000,
  "endtime": 1681881422000
}
```

表 4-4 查询主题参数说明表（历史报警）

参数项	参数说明
method	历史报警的标识，固定值“HistoryAlarm”
topic	第三方接收历史报警的主题，由第三方维护，需要第三方订阅
seq	查询序号，INT64 参数，由第三方维护，用于标识一次查询
type	报警类型，String 类型参数；为空则查询全部历史报警

参数项	参数说明
name	要查询的位号名，String 类型参数；为空则查询全部历史报警
eventtype	默认为 0
beginntime	开始时间戳，单位为毫秒
endtime	结束时间戳，单位为毫秒

4.2.3 返回数据的格式

```
{
  "method": "HistoryAlarm",
  "code": 0,
  "msg": "success",
  "seq": 0,
  "data": [
    {
      "name": "MEM2",
      "eventtype": 0,
      "trigger": 51,
      "limit": 50,
      "operate": "admin",
      "results": "确认报警",
      "newtime": 16553681570000,
      "acktime": 16553681570000,
      "canceltime": 16553681570000,
      "level": 1,
      "type": "L",
      "almdesc": "低限报警"
    }
  ]
}
```

表 4-5 返回数据参数说明表（历史报警）

参数项	参数说明
code	0 表示成功，其他值表示存在错误，详见“错误提示和操作建议”
msg	"success"表示成功
seq	推送报警的序号，标识一次推送，同一批次报警的序号相同

参数项	参数说明
data	查询到的报警数据，包括： ● name: string 参数，表示位号名 ● eventtype: int 参数，默认为 0，表示位号报警 ● newtime: INT64 参数，表示报警产生时间 ● acktime: INT64 参数，表示报警确认时间 ● canceltime: INT64 参数，表示报警消除时间 ● trigger: string 参数，表示报警触发值 ● limit: string 参数，表示报警限值 ● operate: string 参数，表示操作人 ● results: string 参数，表示处理结果 ● level: string 参数，表示报警等级（优先级） ● type: string 参数，表示报警类型（L3/LL/L 等） ● almdesc: string 参数，表示报警描述

5 错误提示和操作建议

查询历史数据时，如果出现错误提示，请按下表说明操作。

表 5-1 错误提示和操作建议一览表

错误码	错误信息	操作提示
0	成功	-
1000	每次最多可查询到 8 个历史数据	检查查询消息里包含的位号数据是否超过了 8 个
1001	位号名不合法	检查查询消息里包含的位号名是否符合规则
1002	查询的方法不支持	检查“method”字段是否符合格式，如不符合请修改
1004	单位号最大查询点数 5000 点（历史数据）	查询的时间跨度过大或时间间隔过小
1005	参数错误	JSON 结构的参数不全或错误，请参考数据格式修改
1006	查询历史报警的返回结果为失败，原因是 InPlant SCADA 查询历史数据失败	检查 InPlant SCADA 软件监控服务是否已正常运行
1007	查询历史报警的返回结果为失败，原因是 InPlant SCADA 未正确配置历史数据服务器	检查 InPlant SCADA 软件工程组态中的历史趋势节点是否配置了历史数据服务器
1008	查询历史报警的返回结果为失败，原因是消息无法发送到 InPlant SCADA 数据服务	检查 InPlant SCADA 软件监控服务是否已正常运行
1010	客户端等待服务器反馈超时	检查 InPlant SCADA 软件监控服务是否已正常运行
1011	method 与目标主题不匹配	检查查询消息的“method”字段和目标主题是否一致

6 资料版本说明

表 6-1 版本升级更改一览表

资料版本号	适用软件版本	更改说明
V1.0（20230411）	InPlant SCADA V5.50.01.00	第一版
V1.1（20230629）	InPlant SCADA V5.50.02.00	新增历史数据和报警信息的配置方法和查询说明
V1.2（20231115）	InPlant SCADA V5.50.03.00	查询报警消息新增查询响应时间；勘误