

独立提交

注释请求

更新:

类别: 信息

国际标准刊号:

季国松

3 月 2022

网关手册/3.0

摘要

进程间数据通讯使用 C/S

sql 只协助配置,实时值不再写入数据库

关于下段备忘

版权声明

目录

- 1. 架构.....3
 - 1.1. 数据流向.....3
 - 1.2. 启动流程 (systemctl)3
 - 1.3. 进程交互流程.....3
- 2. 协议报头.....4
 - 2.1. 声明.....4
 - 2.2. 示例.....4
 - 2.3. 说明.....4
- 3. 协议主体.....6
 - 3.1. 声明.....6
 - 3.2. 示例.....6
 - 3.3. 说明.....7

1. 架构

1.1. 数据流向

进程间通讯采用 socket，数据提供端全部为 server，收集端全部为 client

突变数据流向：

硬件设备→各种协议程序→服务器对接程序->外部服务器

总召数据流向：

硬件设备→各种协议程序→服务器对接程序->外部服务器

1.2. 启动流程（systemctl）

/etc/systemd/system→/ts/install/ts.sh→sql、monitor 等->各种协议程序

1.3. 进程交互流程

序列号和确认号规则遵循 tcp 协议,防止报文丢失(序列号要加有效数据长度)

Server:

等待连接,收到连接后,将连接信息(fd)添加到总连接链表中.

等待报文,收到报文后,发送所有实时数据(相当于初始化中的总召).

之后发生突变直接发送突变.

Client:

发起连接,发送请求报文

收到数据后解析保存.

2. 协议报头

2.1. 声明

本报头参考 tcp/ip,仅用于串口通讯，网络通讯系统底层已包含

2.2. 示例

```
{
  "ip": {
    "version": 3,
    "source_address": 32,
    "destination_address": 33
  },
  "tcp": {
    "control_bits": 3,
    "source_port": 34,
    "destination_port": 342,
    "sequence_number": 23,
    "acknowledgment_number": 342
  },
  "data": "fdsf"
}
```

2.3. 说明

字段	类型	说明
	Json object	
version	int	版本号
source_address	int	源 id
destination_address	int	目的 id

control_bits	int	控制位
source_port	int	源端口
destination_port	int	目的端口
sequence_number	int	序列号
data	Json object	数据

address:

```
enum struct_address {
    STRUCT_ADDRESS_START        = -1,
    STRUCT_ADDRESS_STDIN        = 0,
    STRUCT_ADDRESS_STDOUT       = 1,
    STRUCT_ADDRESS_STDERR       = 2,

    STRUCT_ADDRESS_SOFTWARE     = 4096,
    STRUCT_ADDRESS_SERVER,
    STRUCT_ADDRESS_CLIENT,

    STRUCT_ADDRESS_OUT          = 4096 + 8,
    STRUCT_ADDRESS_MONITOR      = 4096 + 12,

    STRUCT_ADDRESS_SQL          = 4096 + 16,
    STRUCT_ADDRESS_SQL_UPDATE,
    STRUCT_ADDRESS_SQL_SELECT,
    STRUCT_ADDRESS_SQL2CONFIG,

    STRUCT_ADDRESS_TCP          = 4096 + 20,
    STRUCT_ADDRESS_TCP_BOOST,

    STRUCT_ADDRESS_MODBUS_RTU    = 4096 + 24,

    STRUCT_ADDRESS_KD_ROBOT      = 4096 + 28,
    STRUCT_ADDRESS_TEMPLATE,

    STRUCT_ADDRESS_TCP_MNAI      = 4096 + 32, //mechanical properties
    STRUCT_ADDRESS_HTTP_MECHANICAL, //mechanical properties
    STRUCT_ADDRESS_HTTP_WEICHEN,
    STRUCT_ADDRESS_HTTP_KEHUA,
    STRUCT_ADDRESS_WEBSHELL      = 4096 + 36,
    STRUCT_ADDRESS_WEBSERVER,
```

```
STRUCT_ADDRESS_DEVICE,  
STRUCT_ADDRESS_INTERFACE,  
STRUCT_ADDRESS_POINT,  
STRUCT_ADDRESS_END,  
};
```

3. 协议主体

3.1. 声明

本协议参考 http 协议

3.2. 示例

```
{  
  "http": {  
    "version": 3,  
    "message_type": 1,  
    "message_headers": {  
      "time": "2021-03-04T14:33:22Z"  
    },  
    "message_body": {  
      "request": ["id",  
        "type", "status", "value"  
      ],  
      "condition": [  
        [  
          "type", 3  
        ], 3, 3  
      ]  
    }  
  }  
}  
  
{  
  "http": {
```

```

    "version": 3,
    "message_type": 1,
    "message_headers": {
        "time": "2021-03-04 34:33:22"
    },
    "message_body": {
        "response": [{
            "id": 2,
            "type": 2,
            "status": 0,
            "value": 4
        }, {
            "id": 22,
            "type": 2,
            "status": 0,
            "value": 43
        }]
    }
}

```

3.3. 说明

字段	类型	说明
version	int	版本号
message_types	int	消息类型： 请求：0 回应：1
message_headers	json	消息头
time	utc (2007-03-01T13:00:00Z)	时间
message_body	json	消息主体
request	Json array	要请求的参数名称集合
condition	Json array	条件集合
Response	Json	回应数据集合

condition 字段

带参数的条件放前面：["参数名称", 值]

其他条件放后面 ： 值

condition 直接设值需要从下面选择

```
enum struct_condition_idx {  
    STRUCT_CONDITION_MAX,  
    STRUCT_CONDITION_NUMBER,  
};
```