

[QuickProtocol]协议说明

Protocol Specifications



Quick.Protocol V2

2021-12-10

目录

| | |
|---------------------|----|
| 一、 概览..... | 1 |
| 1. 简介..... | 1 |
| 2. 结构图..... | 1 |
| 二、 数据包..... | 2 |
| 1. 简介..... | 2 |
| 2. 包类型..... | 2 |
| 3. 心跳数据包..... | 2 |
| 4. 通知数据包..... | 2 |
| 5. 命令请求数据包..... | 2 |
| 6. 命令响应数据包..... | 3 |
| 7. 拆分数据包..... | 3 |
| 三、 压缩与加密..... | 3 |
| 1. 压缩..... | 3 |
| 2. 加密..... | 4 |
| 四、 命令与指令集..... | 4 |
| 1. 命令..... | 4 |
| 2. 指令集..... | 4 |
| 五、 基础指令集..... | 4 |
| 1. 连接命令..... | 4 |
| 2. 认证命令..... | 6 |
| 3. 握手命令..... | 7 |
| 4. 私有命令..... | 8 |
| 5. 获取全部指令集信息命令..... | 9 |
| 6. 私有通知..... | 15 |
| 六、 测试工具..... | 16 |
| 1. 下载测试工具..... | 16 |
| 2. 运行..... | 17 |
| 3. 添加..... | 17 |
| 4. 连接..... | 18 |
| 5. 测试接收心跳..... | 19 |
| 6. 测试接收通知..... | 21 |
| 7. 测试命令..... | 23 |

一、概览

1. 简介

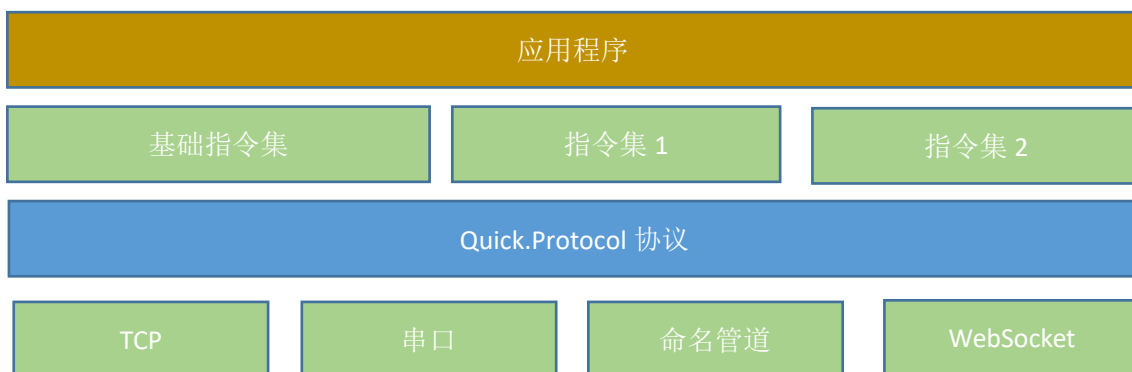
QuickProtocol 协议是基于固定包头已知包长度，使用 JSON 格式传递数据的通讯协议。支持 TCP、串口、命名管道、WebSocket 作为传输通道。

QuickProtocol 协议可以使用任何开发语言实现 QuickProtocol 协议，目前已经有 .NET 和 Java 的实现。

QuickProtocol 协议上通过不同指令集来实现不同的功能集合。

| 项 | 值 |
|---------|---|
| 项目地址 | https://github.com/QuickProtocol |
| .NET 实现 | https://github.com/QuickProtocol/QuickProtocol_CSharp 程序包 ----- https://www.nuget.org/packages/Quick.Protocol/ https://www.nuget.org/packages/Quick.Protocol.Pipeline/ https://www.nuget.org/packages/Quick.Protocol.Tcp/ https://www.nuget.org/packages/Quick.Protocol.SerialPort/ https://www.nuget.org/packages/Quick.Protocol.WebSocket.Client/ https://www.nuget.org/packages/Quick.Protocol.WebSocket.Server.AspNetCore/ |
| Java 实现 | https://github.com/Quick.Protocol/Quick.Protocol_Java |
| C 实现 | 尚未实现 |
| GO 实现 | 尚未实现 |

2. 结构图



二、数据包

1. 简介

数据包是 QuickProtocol 协议中通讯的最小单位。数据包分为包头和包体两部分。包头为 5 个字节，前 4 个字节为包长度，长度包含包头的 5 个字节，是 32 位整数以大端字节序排列，第 5 个字节为包类型。

2. 包类型

| 字节 | 类型 |
|-----|---------|
| 0 | 心跳数据包 |
| 1 | 通知数据包 |
| 2 | 命令请求数据包 |
| 3 | 命令响应数据包 |
| 255 | 拆分数据包 |

3. 心跳数据包

协议连接的双方会定时发送心跳数据包来确保连接的有效。心跳数据包只有 5 个字节的包头，没有包内容。

心跳数据包结构：

| |
|---|
| [4 字节] [1 字节] 包长度 包类型 -----包头----- |
|---|

心跳数据包内容：

| |
|----------------|
| 00-00-00-05-00 |
|----------------|

4. 通知数据包

通知数据包一般用于向对方进行通知，而不关心对方收到通知后的操作。

通知数据包结构：

| | | |
|----------------------------|-----------|-----------|
| [4 字节] [1 字节] [1 字节] | [n 字节] | [n 字节] |
| 包长度 包类型 类名长度 | 类名(A.B.C) | 类内容(JSON) |
| -----包头----- -----包内容----- | | |

5. 命令请求数据包

命令请求数据包用于向对方发送命令后等待对方返回响应结果。

命令请求数据包结构:

| | | | | | |
|----------------------------|--------|---------|--------|-----------|-----------|
| [4 字节] | [1 字节] | [16 字节] | [1 字节] | [n 字节] | [n 字节] |
| 包长度 | 包类型 | 命令编号 | 类名长度 | 类名(A.B.C) | 类内容(JSON) |
| -----包头----- -----包内容----- | | | | | |

6. 命令响应数据包

命令响应数据包用于向对方回复命令的响应结果。

命令响应数据包结构:

| | | | |
|----------------------------|--------|---------|---------------------|
| [4 字节] | [1 字节] | [16 字节] | [1 字节] |
| 包长度 | 包类型 | 命令编号 | 返回码(0 代表成功, 其他代表失败) |
| 成功: [1 字节] [n 字节] [n 字节] | | | |
| 类名长度 类名(A.B.C) 类内容(JSON) | | | |
| 失败: [n 字节] | | | |
| 错误消息(字符串) | | | |
| -----包头----- -----包内容----- | | | |

7. 拆分数据包

当要传输的内容大于单个数据包允许的最大长度时,就需要对数据包进行拆分,分成多个拆分数据包依次发送。

拆分数据包结构:

| | | | |
|----------------------------|--------|--------------------------|-----------------|
| [4 字节] | [1 字节] | [4 字节 0 字节] | [n 字节] |
| 包长度 | 包类型 | 原始包长度, 第一个拆分包才有 4 个字节的长度 | 类内容(Byte Array) |
| -----包头----- -----包内容----- | | | |

三、压缩与加密

1. 压缩

如果启用了压缩,则双方在握手命令处理完毕后,所有的数据包都会被压缩。压缩的内容是不包含包长度前 4 个字节的后面全部数据。前 4 个字节为-压缩后包的长度。压缩方式为 GZip

| | | |
|--------------|--------|--------|
| [4 字节] | [1 字节] | [n 字节] |
| 包长度 | 包类型 | 包内容 |
| ----压缩部分---- | | |
| ↓ GZip 压缩 | | |
| [4 字节] | [n 字节] | |
| 包长度 | 压缩后内容 | |

2. 加密

如果启用了加密，则双方在握手命令处理完毕后，所有的数据包都会被加密。加密的内容是不包含包长度前 4 个字节的后面全部数据。前 4 个字节为加密后包的长度。加密方法为 DES，密码为通讯密码的 MD5 值的前 8 个字节。

| | | |
|--------|--------|--------------|
| [4 字节] | [1 字节] | [n 字节] |
| 包长度 | 包类型 | 包内容 |
| | | ----加密部分---- |
| | | ↓ DES 加密 |
| [4 字节] | [n 字节] | |
| 包长度 | 加密后内容 | |

四、命令与指令集

1. 命令

每一个命令请求数据包需要产生一个 16 字节 GUID 作为命令的唯一编号，命令的类名用来区别命令的不同功能，命令的类内容为命令的详细内容。

收到命令请求数据包后，根据命令的类名作出不同的处理然后返回一个命令响应数据包给对方。

2. 指令集

指令集是指按照业务逻辑区分的不同的命令与通知数据包定义的一个集合。比如基础指令集包含了连接、握手、认证和获取全部指令集命令和私有通知等功能。其他指令集定义了另外的业务逻辑功能。

五、基础指令集

连接双方必须实现基础指令集定义的功能。

1. 连接命令

客户端连接到服务端后，发送连接命令到服务端开始连接流程。

| 项 | 值 |
|----|----|
| 名称 | 连接 |

| | |
|---------|--|
| 命令请求类名称 | Quick.Protocol.Commands.Connect.Request |
| 命令响应类名称 | Quick.Protocol.Commands.Connect.Response |

(1) 请求定义

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Request",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "InstructionIds": {
      "type": [
        "array",
        "null"
      ],
      "items": {
        "type": "string"
      }
    }
  }
}
```

(2) 请求示例

```
{
  "InstructionIds": [
    "SoftCloud.Gateway.Protocol.V2"
  ]
}
```

(3) 请求参数说明

| 参数 | 说明 |
|----------------|--|
| InstructionIds | 客户端要求服务端必须实现的指令集数组，由于基础指令集是必须实现的，这里面就不需要传递基础指令集的编号了，只需要传递业务逻辑的指令集编号。 |

(4) 响应定义

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Response",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "BufferSize": {
      "type": "integer",
      "format": "int32"
    },
    "Question": {
      "type": [
```

```

        "null",
        "string"
    ]
}
}
}

```

(5) 响应示例

```

{
  "BufferSize": 131072,
  "Question": "933789894d0d460d90976122a79416da"
}

```

(6) 响应参数说明

| 参数 | 说明 |
|------------|-------------------------|
| BufferSize | 服务端的缓存大小，大于这个大小的数据包会被抛弃 |
| Question | 服务端发过来的认证问题 |

2. 认证命令

客户端收到连接响应后，发送认证命令到服务端继续连接流程。认证是否成功根据响应包的返回码来确定，如果返回码是 0 代表认证成功。

| 项 | 值 |
|---------|---|
| 名称 | 连接 |
| 命令请求类名称 | Quick.Protocol.Commands.Authenticate.Request |
| 命令响应类名称 | Quick.Protocol.Commands.Authenticate.Response |

(1) 请求定义

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Request",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "Answer": {
      "type": [
        "null",
        "string"
      ]
    }
  }
}

```

(2) 请求示例

```

{
  "Answer": "933789894d0d460d90976122a79416da"
}

```


(3) 请求参数说明

| 参数 | 说明 |
|--------|---|
| Answer | 将服务端的认证问题拼接密码后计算 MD5 值作为认证回答，发送认证命令给服务端。认证回答 = MD5(认证问题 + 密码) |

(4) 响应定义

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Response",
  "type": "object",
  "additionalProperties": false
}
```

(5) 响应示例

```
{}
```

3. 握手命令

服务端返回认证成功后，客户端即可发送握手命令，约定传输的超时时间，是否启用压缩，是否启用加密。至此整个连接流程完成。

| 项 | 值 |
|---------|--|
| 名称 | 握手 |
| 命令请求类名称 | Quick.Protocol.Commands.HandShake.Request |
| 命令响应类名称 | Quick.Protocol.Commands.HandShake.Response |

(6) 请求定义

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Request",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "TransportTimeout": {
      "type": "integer",
      "format": "int32"
    },
    "EnableEncrypt": {
      "type": "boolean"
    },
    "EnableCompress": {
      "type": "boolean"
    }
  }
}
```

(7) 请求示例

```
{
```

```
"TransportTimeout": 15000,
"EnableEncrypt": false,
"EnableCompress": false
}
```

(8) 请求参数说明

| 参数 | 说明 |
|------------------|--------------------------------------|
| TransportTimeout | 传输超时时间，如果在这个时间内没有收到对方的任何数据，则触发超时断开连接 |
| EnableEncrypt | 是否启用加密 |
| EnableCompress | 是否启用压缩 |

(9) 响应定义

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Response",
  "type": "object",
  "additionalProperties": false
}
```

(10) 响应示例

```
{}
```

4. 私有命令

此命令用于一些不想创建指令集实现一些简单命令的情况。

| 项 | 值 |
|---------|---|
| 名称 | 私有命令 |
| 命令请求类名称 | Quick.Protocol.Commands.PrivateCommand.Request |
| 命令响应类名称 | Quick.Protocol.Commands.PrivateCommand.Response |

(7) 请求定义

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Request",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "Action": {
      "type": [
        "null",
        "string"
      ]
    },
    "Content": {
      "type": [
        "null",
```

```
        "string"
      ]
    }
  }
}
```

(8) 请求示例

```
{
  "Action": "Action",
  "Content": "Content"
}
```

(9) 请求参数说明

| 参数 | 说明 |
|---------|------|
| Action | 动作 |
| Content | 请求内容 |

(10) 响应定义

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Response",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "Content": {
      "type": [
        "null",
        "string"
      ]
    }
  }
}
```

(11) 响应示例

```
{
  "Content": "Content"
}
```

(12) 响应参数说明

| 参数 | 说明 |
|---------|------|
| Content | 响应内容 |

5. 获取全部指令集信息命令

此命令用于获取服务端支持的全部指令集以及指令集的详细信息。

| 项 | 值 |
|---------|---|
| 名称 | 获取全部指令集信息 |
| 命令请求类名称 | Quick.Protocol.Commands.GetQpInstructions.Request |

命令响应类名称

Quick.Protocol.Commands.GetQpInstructions.Response

(13) 请求定义

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Request",
  "type": "object",
  "additionalProperties": false
}
```

(14) 请求示例

```
{}
```

(15) 响应定义

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Response",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "Data": {
      "type": [
        "array",
        "null"
      ],
      "items": {
        "$ref": "#/definitions/QpInstruction"
      }
    }
  },
  "definitions": {
    "QpInstruction": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "Id": {
          "type": [
            "null",
            "string"
          ],
          "readonly": true
        },
        "Name": {
          "type": [
            "null",
            "string"
          ],

```

```

        "readonly": true
    },
    "NoticeInfos": {
        "type": [
            "array",
            "null"
        ],
        "items": {
            "$ref": "#/definitions/QpNoticeInfo"
        }
    },
    "CommandInfos": {
        "type": [
            "array",
            "null"
        ],
        "items": {
            "$ref": "#/definitions/QpCommandInfo"
        }
    }
},
"QpNoticeInfo": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
        "Name": {
            "type": [
                "null",
                "string"
            ],
            "readonly": true
        },
        "Description": {
            "type": [
                "null",
                "string"
            ],
            "readonly": true
        },
        "NoticeTypeName": {
            "type": [
                "null",
                "string"
            ]
        }
    }
}

```

```

    ],
    "readonly": true
  },
  "NoticeTypeSchema": {
    "type": [
      "null",
      "string"
    ],
    "readonly": true
  },
  "NoticeTypeSchemaSample": {
    "type": [
      "null",
      "string"
    ],
    "readonly": true
  }
},
"QpCommandInfo": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "Name": {
      "type": [
        "null",
        "string"
      ],
      "readonly": true
    },
    "Description": {
      "type": [
        "null",
        "string"
      ],
      "readonly": true
    },
    "RequestTypeName": {
      "type": [
        "null",
        "string"
      ],
      "readonly": true
    }
  },

```

```

    "RequestTypeSchema": {
      "type": [
        "null",
        "string"
      ],
      "readonly": true
    },
    "RequestTypeSchemaSample": {
      "type": [
        "null",
        "string"
      ],
      "readonly": true
    },
    "ResponseTypeName": {
      "type": [
        "null",
        "string"
      ],
      "readonly": true
    },
    "ResponseTypeSchema": {
      "type": [
        "null",
        "string"
      ],
      "readonly": true
    },
    "ResponseTypeSchemaSample": {
      "type": [
        "null",
        "string"
      ],
      "readonly": true
    }
  }
}

```

(16) 响应示例

```

{
  "Data": [
    {
      "Id": "Id",

```

```

    "Name": "Name",
    "NoticeInfos": [
      {
        "Name": "Name",
        "Description": "Description",
        "NoticeTypeName": "NoticeTypeName",
        "NoticeTypeSchema": "NoticeTypeSchema",
        "NoticeTypeSchemaSample": "NoticeTypeSchemaSample"
      }
    ],
    "CommandInfos": [
      {
        "Name": "Name",
        "Description": "Description",
        "RequestTypeName": "RequestTypeName",
        "RequestTypeSchema": "RequestTypeSchema",
        "RequestTypeSchemaSample": "RequestTypeSchemaSample",
        "ResponseTypeName": "ResponseTypeName",
        "ResponseTypeSchema": "ResponseTypeSchema",
        "ResponseTypeSchemaSample": "ResponseTypeSchemaSample"
      }
    ]
  }
}

```

(17) 响应参数说明

| 参数 | 说明 |
|---|---------|
| Data | 指令集数组 |
| Data/Id | 指令集编号 |
| Data/Name | 指令集名称 |
| Data/NoticeInfos | 指令集通知数组 |
| Data/NoticeInfos/Name | 通知名称 |
| Data/NoticeInfos/Description | 通知描述 |
| Data/NoticeInfos/NoticeTypeName | 通知类名称 |
| Data/NoticeInfos/NoticeTypeSchema | 通知类定义 |
| Data/NoticeInfos/NoticeTypeSchemaSample | 通知类示例 |
| Data/CommandInfos | 指令集命令数组 |
| Data/CommandInfos/Name | 命令名称 |
| Data/CommandInfos/Description | 命令描述 |
| Data/CommandInfos/RequestTypeName | 命令请求类名称 |
| Data/CommandInfos/RequestTypeSchema | 命令请求类定义 |
| Data/CommandInfos/RequestTypeSchemaSample | 命令请求类示例 |
| Data/CommandInfos/ResponseTypeName | 命令响应类名称 |

| | |
|--|---------|
| Data/CommandInfos/ResponseTypeSchema | 命令响应类定义 |
| Data/CommandInfos/ResponseTypeSchemaSample | 命令响应类示例 |

6. 私有通知

此命令用于一些不想创建指令集实现一些简单通知的情况。

| 项 | 值 |
|-------|--------------------------------------|
| 名称 | 私有通知 |
| 通知类名称 | Quick.Protocol.Notices.PrivateNotice |

(18) 请求定义

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "PrivateNotice",
  "type": "object",
  "description": "用于传递私有协议通知。",
  "additionalProperties": false,
  "properties": {
    "Action": {
      "type": [
        "null",
        "string"
      ]
    },
    "Content": {
      "type": [
        "null",
        "string"
      ]
    }
  }
}
```

(19) 请求示例

```
{
  "Action": "Action",
  "Content": "Content"
}
```

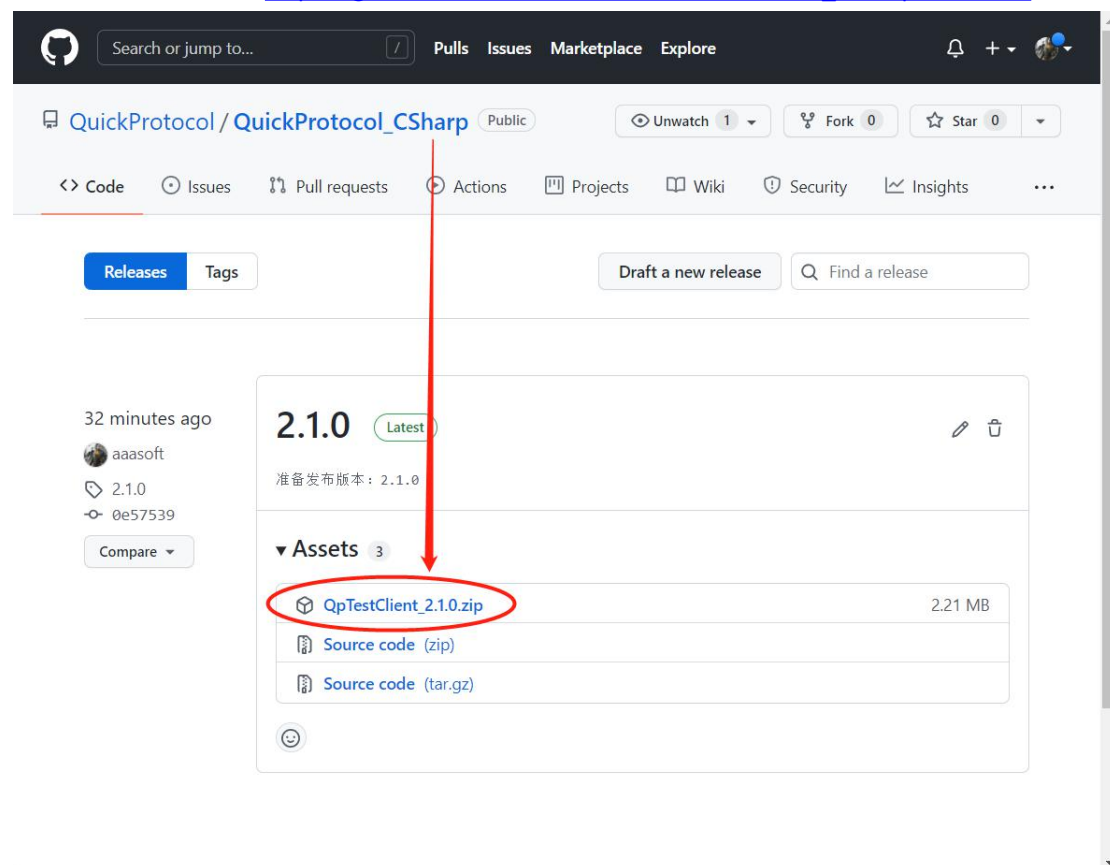
(20) 请求参数说明

| 参数 | 说明 |
|---------|------|
| Action | 动作 |
| Content | 通知内容 |

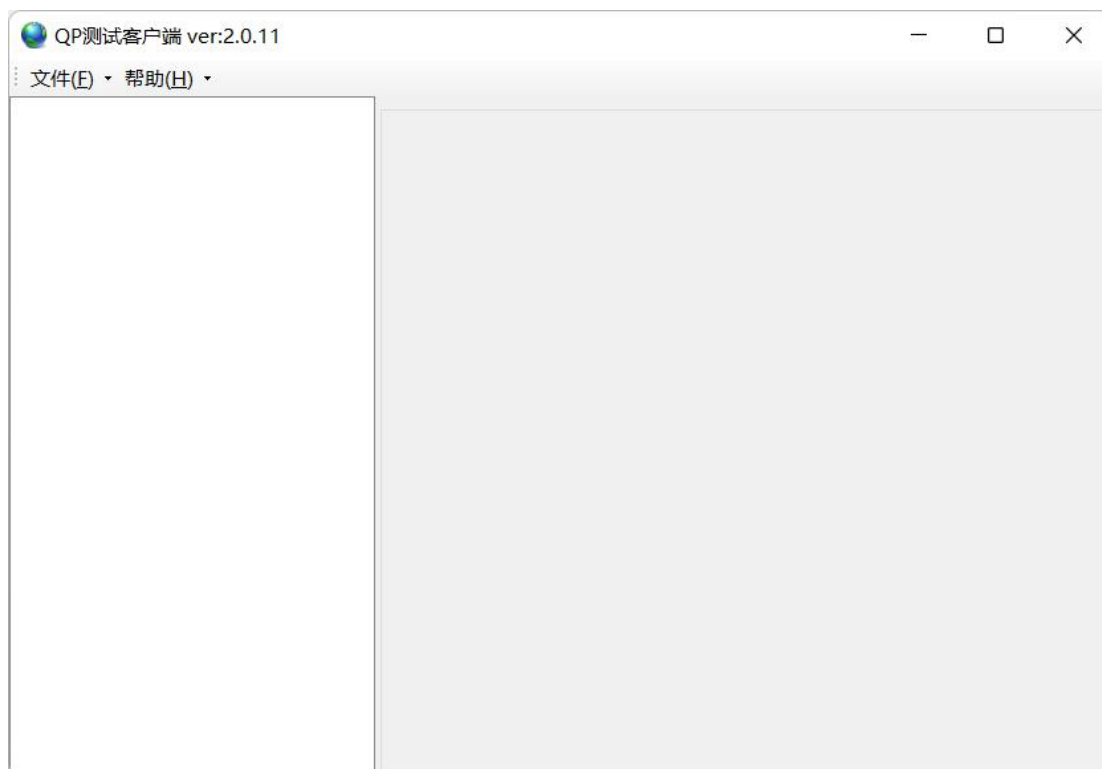
六、测试工具

1. 下载测试工具

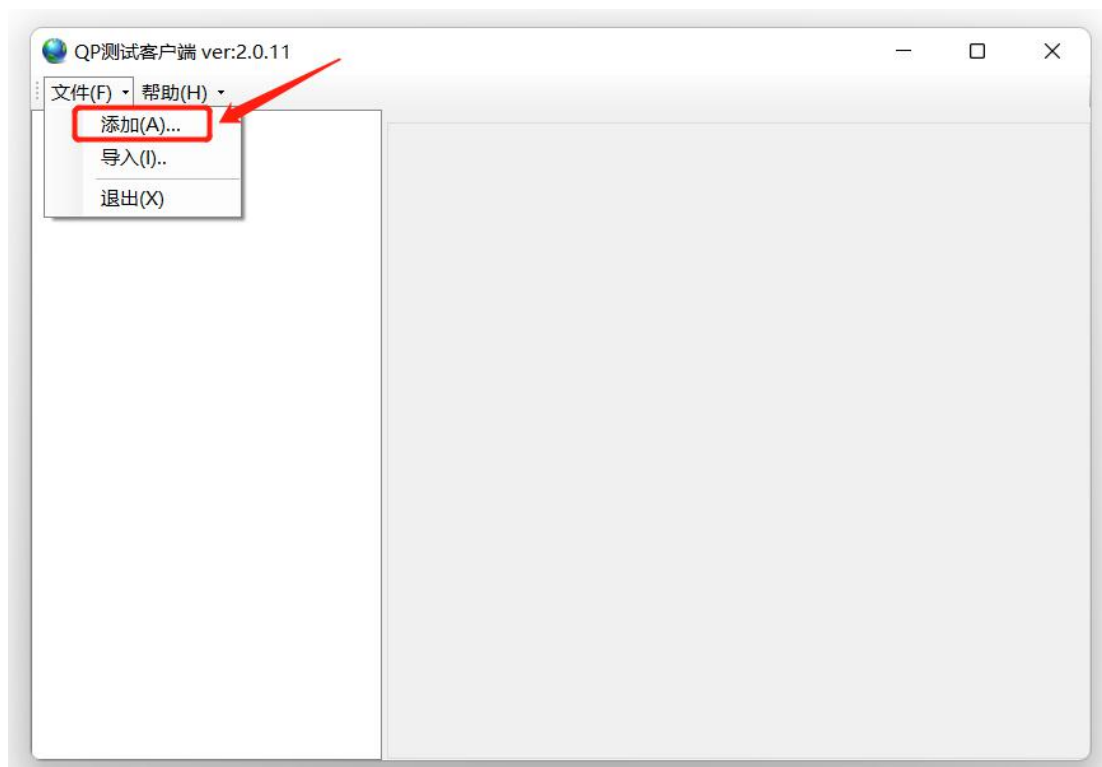
测试工具下载页面: https://github.com/QuickProtocol/QuickProtocol_CSharp/releases/

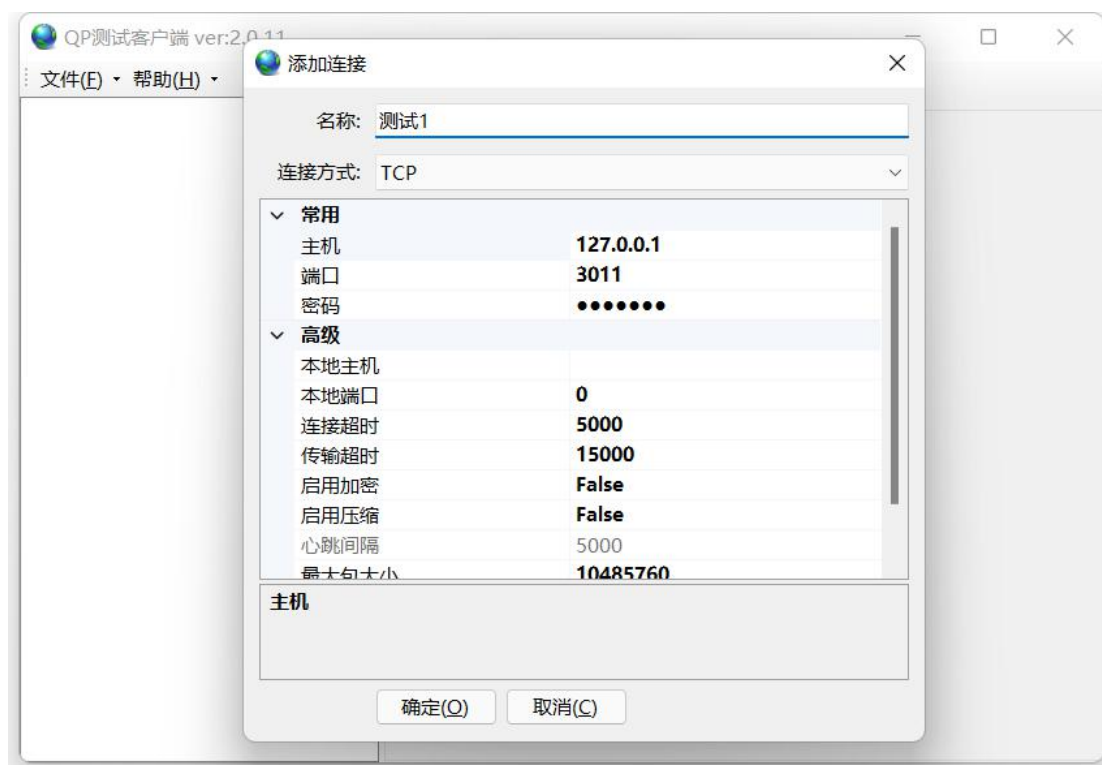


2. 运行

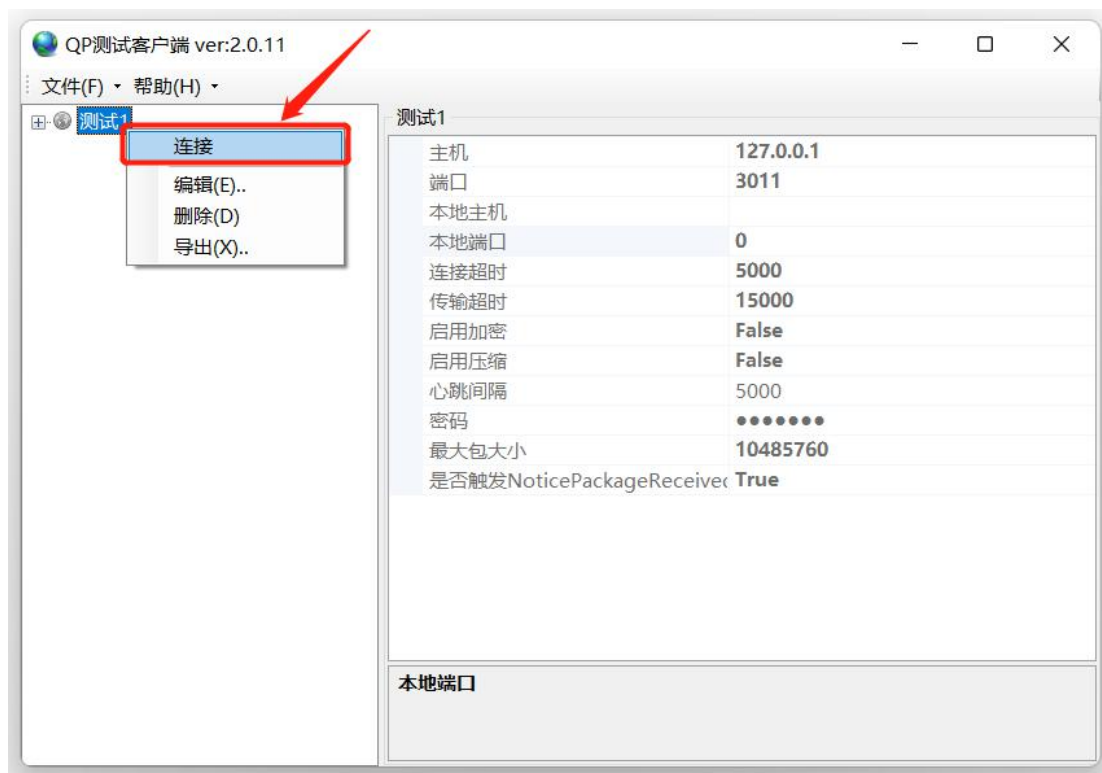


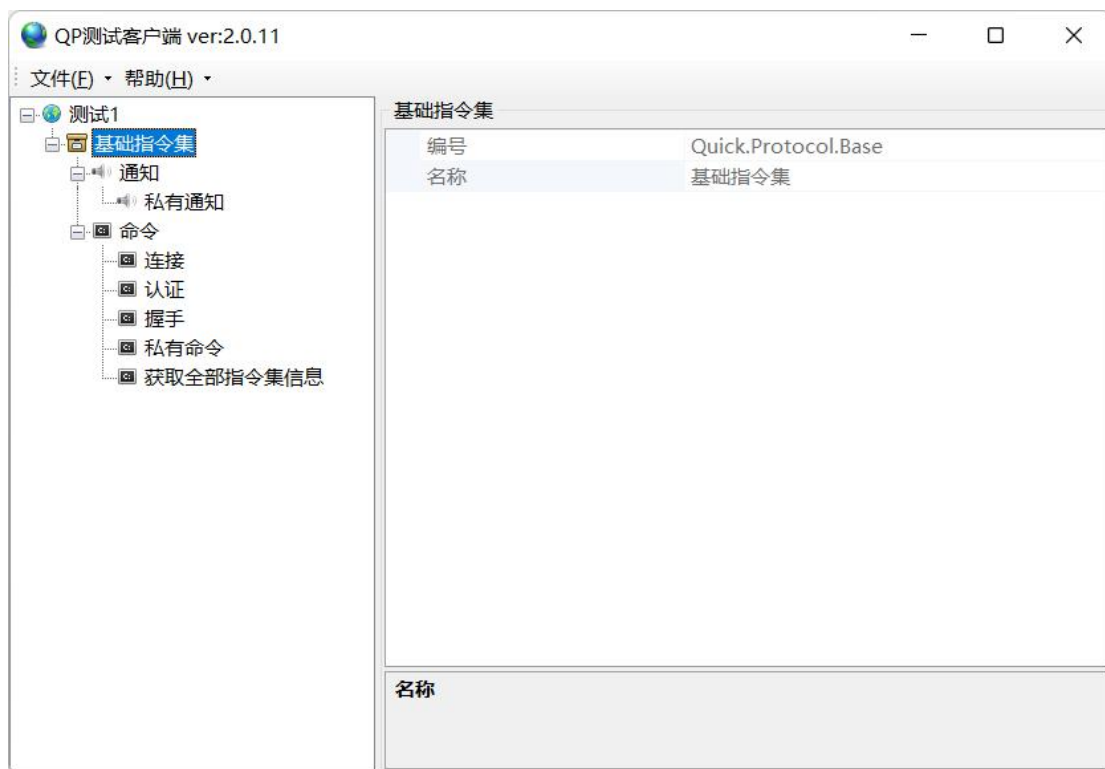
3. 添加



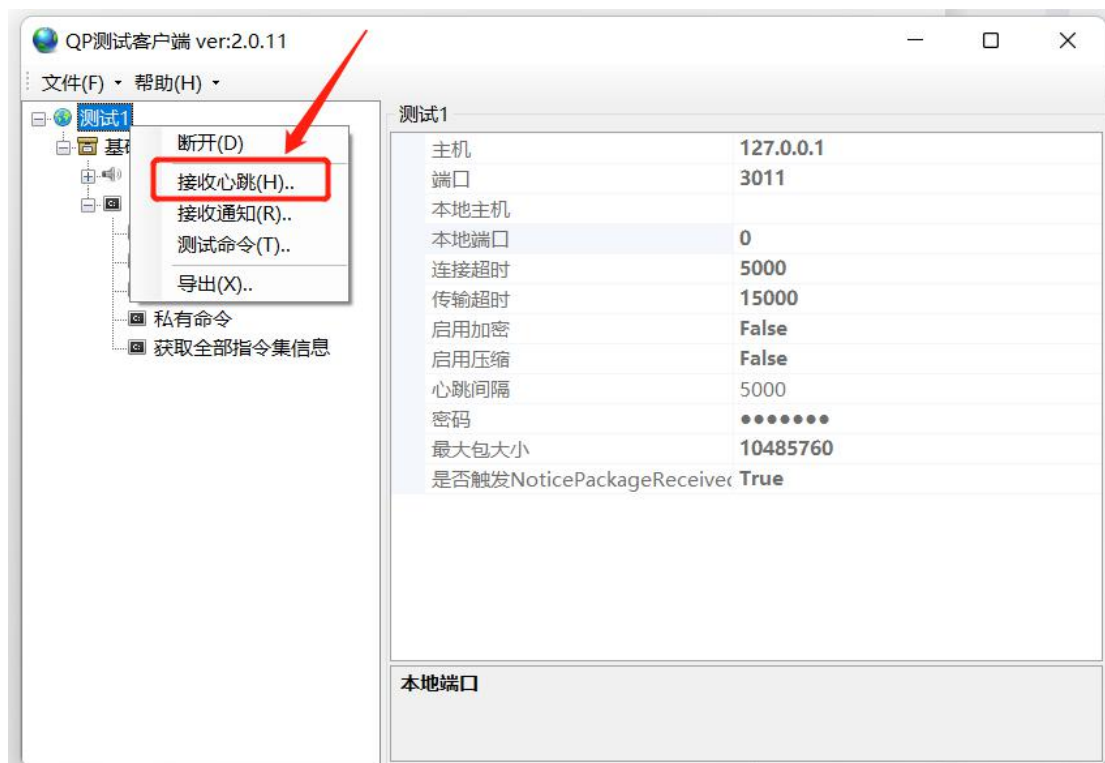


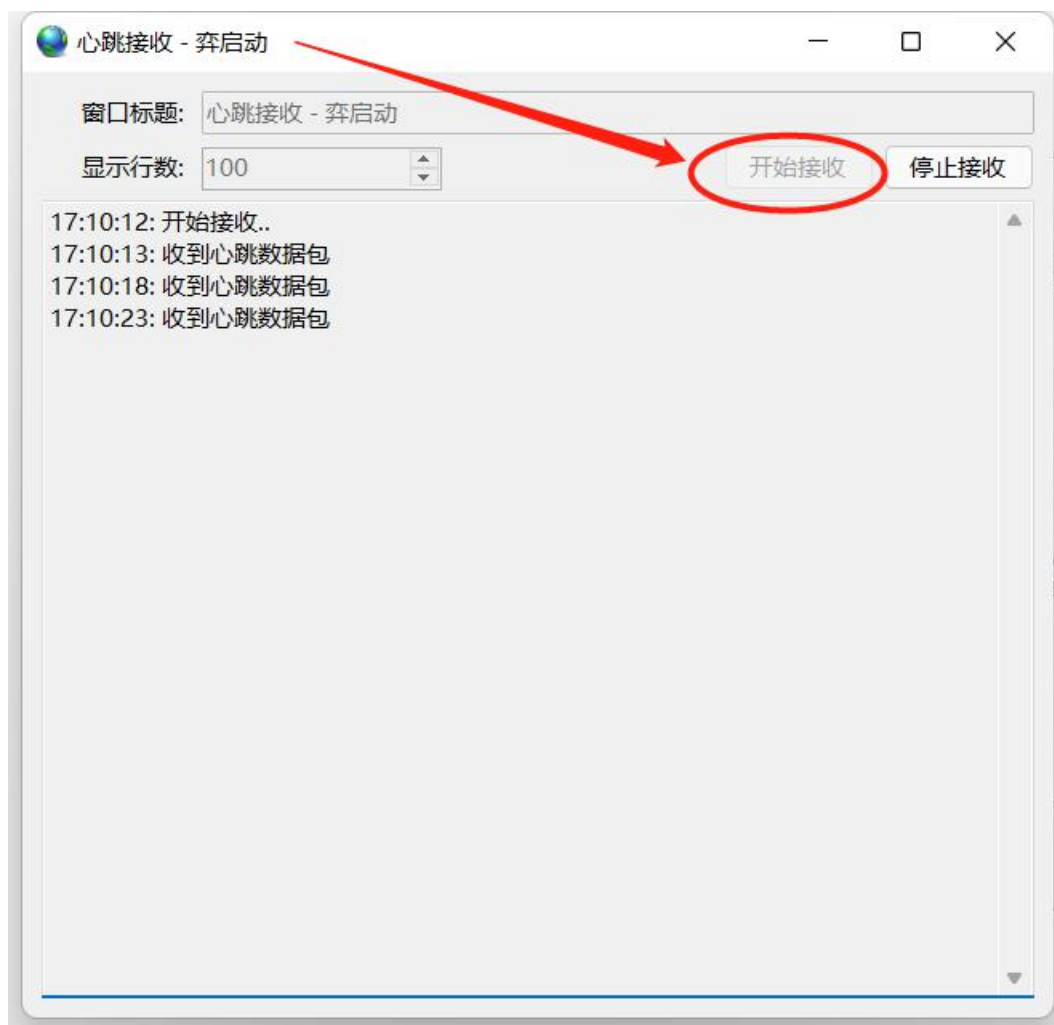
4. 连接



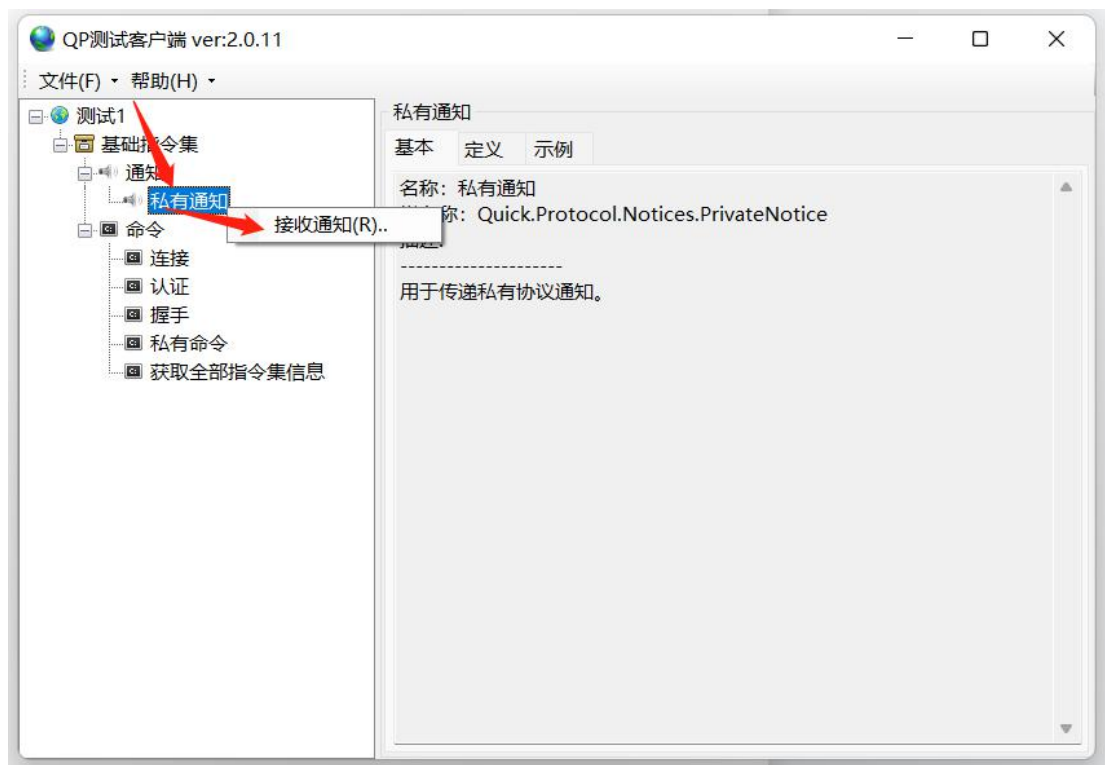


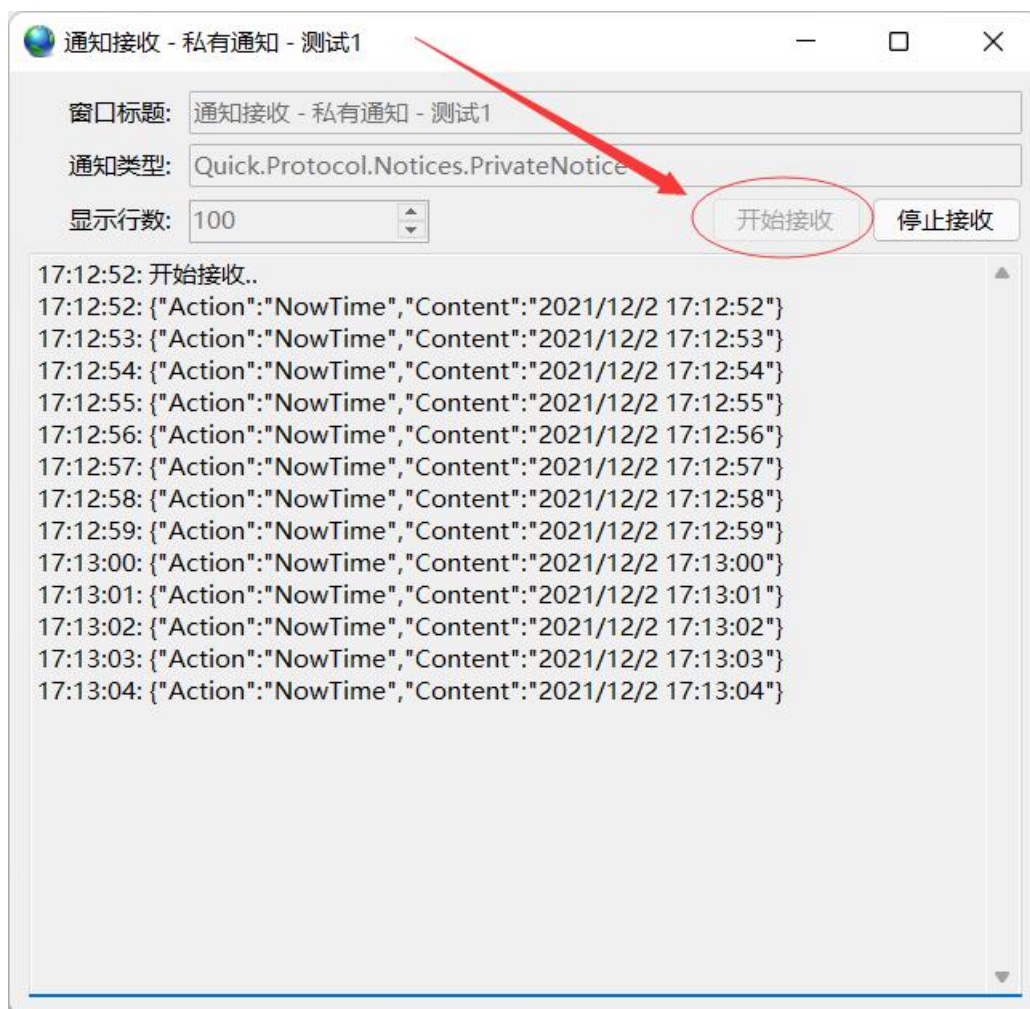
5. 测试接收心跳





6. 测试接收通知





7. 测试命令

