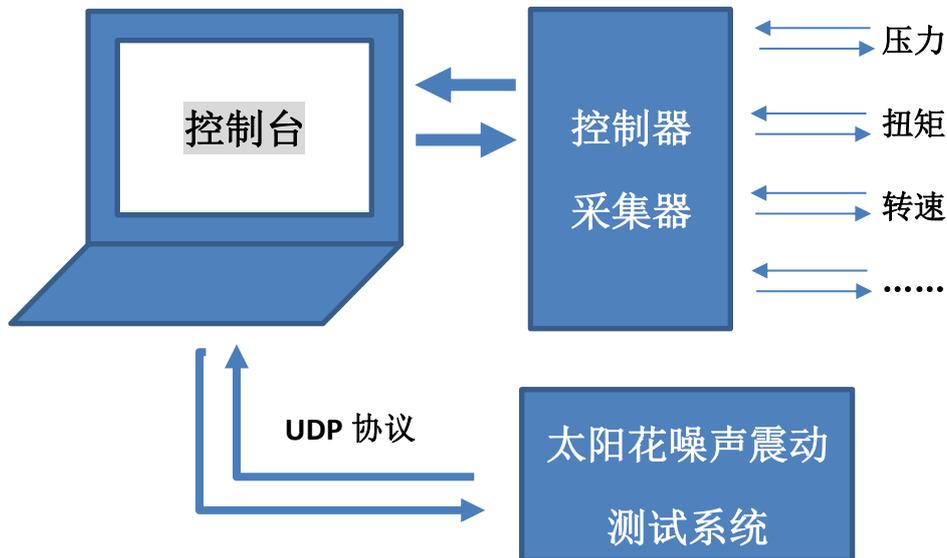


噪声振动测试与已有测试台架的连接

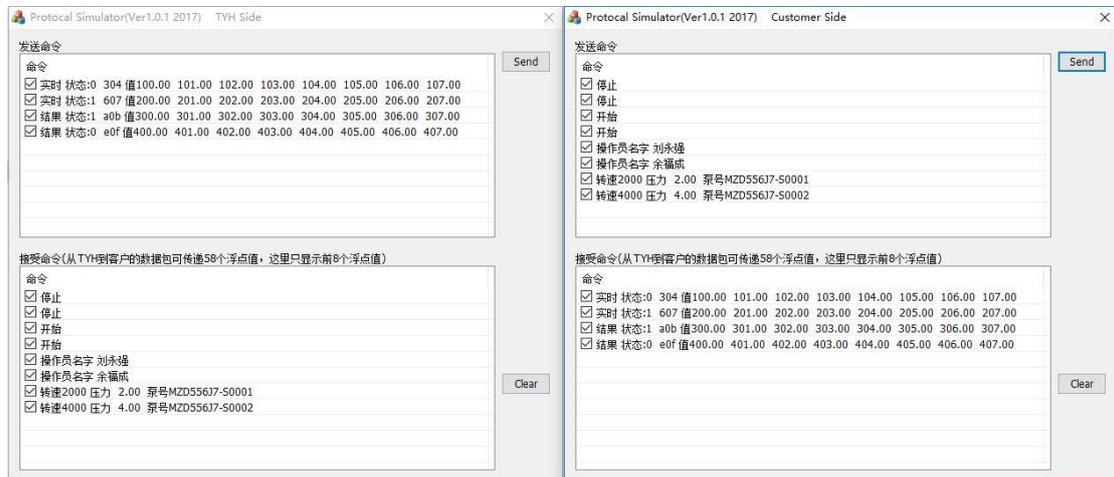
油泵与水泵测试涉及很多环节，太阳花噪声振动测试可以像扭矩测试等其他测试一样融入已有的测试台架或生产线中。为确定测试台架控制台与太阳花噪声振动测试的通信格式，我们提供了相应的通信协议（详见附录 1：简化的单工位通信协议、附录 2：简化的双工位通信协议）和调试器帮助客户进行调试。如有需要，我们还可以为客户提供 VB 和 Labview 的代码例子。

测试台架控制台与太阳花噪声振动测试系统之间可以按照通信协议中指定的格式传递命令和数据。控制台可向太阳花噪声振动测试系统发送零件号、转速和压力等信息，并控制测试系统的启动和停止。太阳花噪声振动测试系统将根据接收到的指令进行工作，并在检测结束后把检测结果发送至测试台架控制台。



图表 1：噪声振动测试链接示意图

1. 调试器间的信息及命令传递：



图表 2：单工位-调试器间的信息传递

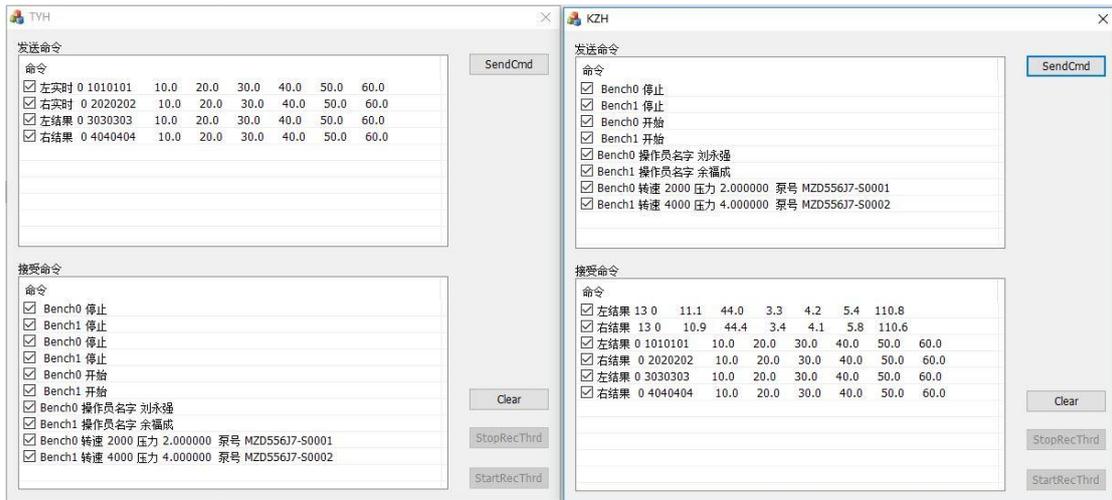


图 3：双工位-调试器间的信息传递

上图中的 Customer Side 和 KZH 调试器是模拟的测试台架控制台程序；TYH 调试器是模拟的太阳花噪声振动测试系统。若客户完成的控制台程序可以成功地与 TYH 调试器收发信息，则一定可以与太阳花噪声振动测试系统正常通信。

2. 简化的单工位实例：

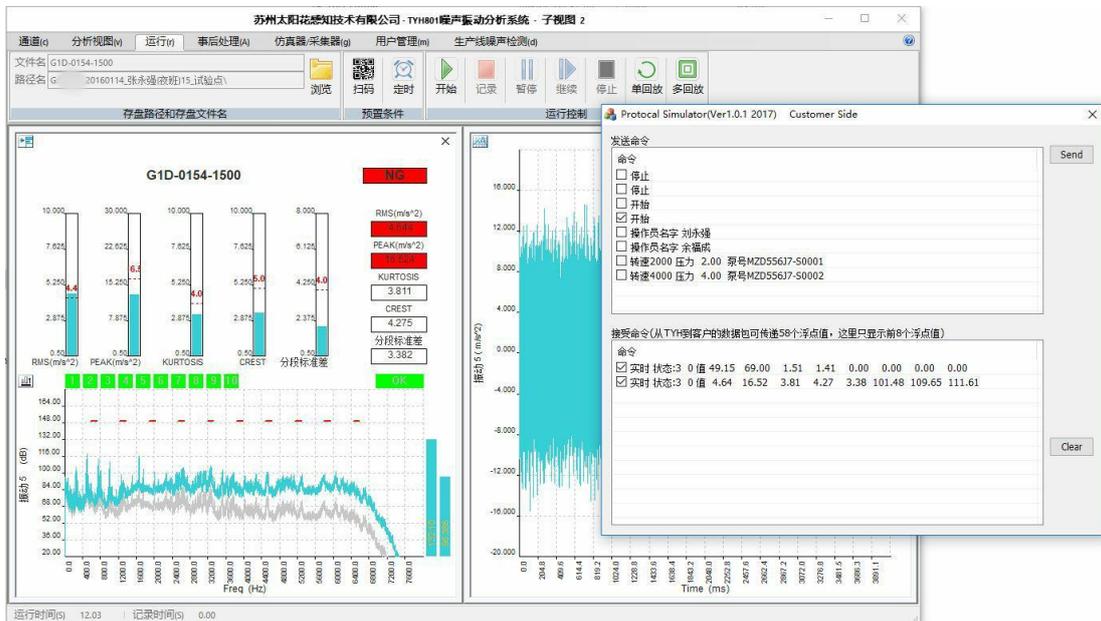
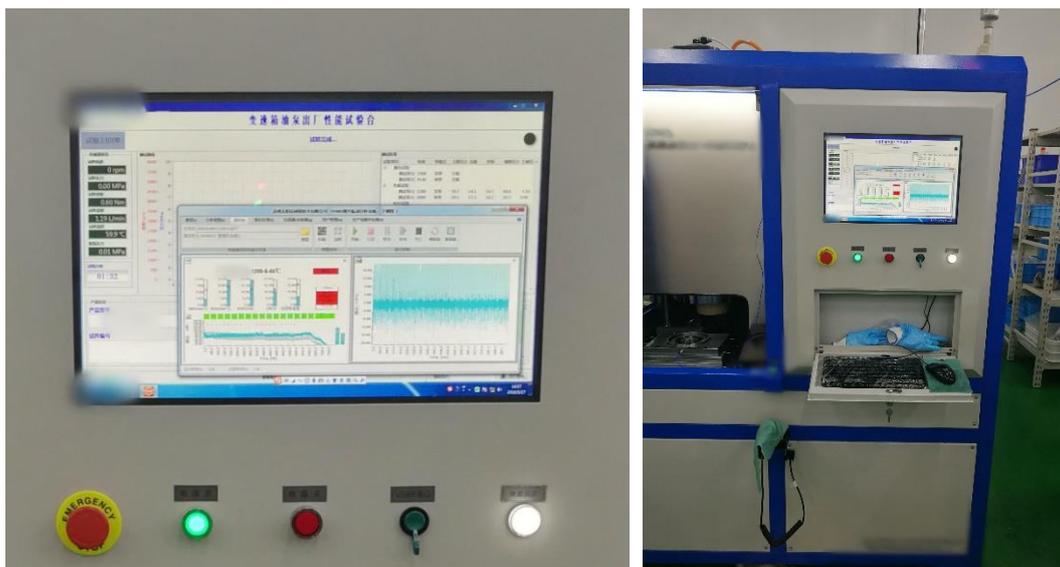


图 4：简化的单工位实例

上图中，程序 Customer Side（模拟的简化控制台程序）发送了开始命令，太阳花噪声振动测试程序收到信息后将文件名进行更改并开始测试。当程序 Customer Side 发送停止命令后，太阳花噪声振动测试程序停止测试并将测试结果发送至 Customer Side，显示在右下角接受框内。

3. 已完成的控制台实例（客户信息被进行了模糊处理）：



图表 5：油泵测试台架控制台单工位实例 1



图表 6：油泵测试台架控制台单工位实例 2



图表 7：油泵测试台架控制台单工位实例 3



图表 8:油泵测试台架控制台单工位实例 4



附录 1：简化的单工位通信协议

客户(9888)与太阳花(5921)通信协议（UDP）

一、客户发往太阳花数据包格式

1 停止太阳花工作

序号	字节位置	名称	类型	含义
1	0-3	标识码	Long	固定值 0x00230512 客户发往太阳花
2	4-7	命令码	Long	固定值 0xC0D00000 停止工作命令
3	8-11	数值	Long	以后根据需要再定义哪些结果要求传回
4	12-15	数值	Long	以后根据需要再定义
5	

```
struct PKG_CUSTUMER2TYH_STOP {  
    int direction;  
    int cmd;  
    int resv1;  
    int resv2;  
    BYTE resv[512 - sizeof(int) * 4];  
};
```

2 启动太阳花工作

序号	字节位置	名称	类型	含义
1	0-3	标识码	Long	固定值 0x00230512 客户发往太阳花
2	4-7	命令码	Long	固定值 0xC0D00001 开始工作命令
3	8-11	数值	Long	以后根据需要再定义采样率
4	12-15	数值	Long	以后根据需要再定义通道号
5

```
struct PKG_CUSTUMER2TYH_START {  
    int direction;  
    int cmd;  
    int resv1;  
    int resv2;  
    BYTE resv[512 - sizeof(int) * 4];  
};
```

3 传递操作员名字

序号	字节位置	名称	类型	含义
1	0-3	标识码	Long	固定值 0x00230512 客户发往太阳花
2	4-7	命令码	Long	固定值 0xC0D00002 操作员名字传递
3	8-263	名字	unicode	操作员名字 unicode 字符串
4	264-....	缺省		以后根据需要再定义

```

struct PKG_CUSTUMER2TYH_OPERATORNM{
    int direction;
    int cmd;
    char operatornm[256];
    BYTE resv[512-sizeof(int)*2-256];
};

```

4 传递转速、压力、零件号参数

序号	字节位置	名称	类型	含义
1	0-3	标识码	Long	固定值 0x00230512 客户发往太阳花
2	4-7	命令码	Long	固定值 0xC0D00003 传递转速、压力、零件号
3	8-11	转速	Long	测试转速 RPM
4	12-15	压力	Long	测试压力接收值/10000 为实际压力单位 MPa
5	16-271	零件号	unicode	零件号 unicode 字符串
6	272-....	缺省	Long	以后根据需要再定义

```

struct PKG_CUSTUMER2TYH_RPMPRESSPARTNUMBER {
    int direction;
    int cmd;
    int rpm;
    int pressure;
    char pumpnmb[256];
    BYTE resv[512-sizeof(int)*4-256];
};

```

二、太阳花发往客户数据包格式

1 零件测试的最终结果

序号	字节位置	名称	类型	含义
1	0-3	标识码	Long	固定值 0x05120023 太阳花发往客户
2	4-7	命令码	Long	固定值 0xC0D01001 零件最终测试结果值
3	8-11	转速	Long	测试转速 RPM
4	12-15	压力	Long	测试压力接收值/10000 为实际压力单位 MPa
5	16-271	零件号	unicode	零件号 unicode 字符串
6	272-275	状态码 1	Long	是否通过，0 通过，1 拒绝 第 0 位（LSB）表示 RMS 是否通过 第 1 位表示 PEAK 是否通过 第 2 位表示 Kurtosis 是否通过 第 3 位表示 CrestFactor 是否通过 第 4 位表示“分段标准差”是否通过
7	276-279	状态码 2	Long	每位（bit）和一个测试指标分项相对应，0 通过 1 拒绝，共 22 个频率段
8	280-511	v	Long (58 个)	最后统计值 前 5 个值为 RMS 、 PEAK、 Kurtosis、 CrestFactor 和“分段标准差”具体应用定义值内容 实测值 =v/10000 第 6-26 各值 为 22 个频率子段的振动值，实测值 =v/10000

每次检测零件结束后，发 1 次检测结果。

```
struct PKG_CUSTOMER2TYH_RESULT {
    int direction;
    int cmd;
    int rpm;
    int pressure;
    char pumpnmb[256];
    int status1;
    int status2;
    int v[58];
};
```

每个零件测试大致通信流程

- 1 客户发太阳花操作员名字
- 2 客户发太阳花转速、压力、零件号
- 3 客户发开始检测命令
- 5 客户发太阳花停止检测命令（可以省去，由定时决定何时停止）
- 6 太阳花停止工作，发回零件测试结果数据包

如果在同一台 PC 上双方收发地址均为 127.0.0.1,

太阳花插口号为 5921, 客户的插口号为 9888

通信方式: UDP 格式

包长: 固定包长, 每个数据包 512 字节

对齐方式: 传输的结构体单字节对齐 `#pragma pack(1) ... #pragma pack()`

发送前整个数据包字节全部置零, 然后再赋值发送。

字符串先调试 `unicode`, 因为这样可以传汉字, 如果字符串 `Unicode` 不好调通, 后面就改为 `multi byte` 形式。(传 `Unicode` 没有问题)

附录 2：简化的双工位通信协议

客户(9888)与太阳花(9885)信协议

三、客户发往太阳花数据包格式

1 停止太阳花振动检测

序号	字节位置	名称	类型	含义
1	0-3	标识码	Long	固定值 0x00230512 客户发往太阳花
2	4-7	命令码	Long	固定值 0xC0D00000 1号台架停止工作 固定值 0xC0D10000 2号台架停止工作
3	8-11	数值	Long	以后根据需要再定义哪些结果要求传回
4	12-15	数值	Long	以后根据需要再定义
5	

```
struct PKG_KZH2TYH_STOP{  
    int direction;  
    int cmd;  
    int resv1;  
    int resv2;  
    BYTE resv[512-sizeof(int)*4];  
};
```

2 启动太阳花振动检测

序号	字节位置	名称	类型	含义
1	0-3	标识码	Long	固定值 0x00230512 客户发往太阳花
2	4-7	命令码	Long	固定值 0xC0D00001 1号台架开始工作 固定值 0xC0D10001 2号台架开始工作
3	8-11	数值	Long	以后根据需要再定义采样率
4	12-15	数值	Long	以后根据需要再定义通道号
5

```
struct PKG_KZH2TYH_START{  
    int direction;  
    int cmd;  
    int resv1;  
    int resv2;  
    BYTE resv[512-sizeof(int)*4];  
};
```

3 传递操作员名字

序号	字节位置	名称	类型	含义
1	0-3	标识码	Long	固定值 0x00230512 客户发往太阳花
2	4-7	命令码	Long	固定值 0xC0D00002 传递 1号台架操作员名字

				固定值 0xCOD10002 传递 2 号台架操作员名字
3	8-263	名字	unicode	操作员名字 unicode 字符串
4	264-....	缺省		以后根据需要再定义

```

struct PKG_KZH2TYH_OPERATORNM{
    int direction;
    int cmd;
    char operatornm[256];
    BYTE resv[512-sizeof(int)*2-256];
}

```

4 传递转速值和泵号字符串

序号	字节位置	名称	类型	含义
1	0-3	标识码	Long	固定值 0x00230512 客户发往太阳花
2	4-7	命令码	Long	固定值 0xCOD00003 传递 1 号台架泵号 固定值 0xCOD10003 传递 2 号台架泵号
3	8-11	转速	Long	泵的测试转速 RPM
4	12-15	压力	Long	泵的测试压力 接收值/10000 为实际压力 单位 MPa
5	16-271	泵号	unicode	泵号 unicode 字符串
6	272-....	缺省	Long	以后根据需要再定义

```

struct PKG_KZH2TYH_RPMPUMPNUMBER{
    int direction;
    int cmd;
    int rpm;
    int pressure;
    char pumpnmb[256];
    BYTE resv[512-sizeof(int)*4-256];
};

```

四、太阳花发往客户数据包格式

1 振动检测最终结果

序号	字节位置	名称	类型	含义
1	0-3	标识码	Long	固定值 0x05120023 太阳花发往客户
2	4-7	命令码	Long	固定值 0xCOD01001 1 号台架实时统计值 固定值 0xCOD11001 2 号台架实时统计值
3	8-11	转速	Long	测试转速 RPM
4	12-15	压力	Long	测试压力接收值/10000 为实际压力单位 MPa
5	16-271	零件号	unicode	零件号 unicode 字符串
6	272-275	状态码	Long	是否通过, 0 通过, 1 拒绝

		1		第 0 位 (LSB) 表示 RMS 是否通过 第 1 位表示 PEAK 是否通过 第 2 位表示 Kurtosis 是否通过 第 3 位表示 CrestFactor 是否通过 第 4 位表示“分段标准差”是否通过
7	276-279	状态码 2	Long	每位 (bit) 和一个测试指标分项相对应, 0 通过 1 拒绝, 共 22 个频率段
8	280-511	v	Long (58 个)	最后统计值 前 5 个值为 RMS、PEAK、Kurtosis、CrestFactor 和“分段标准差”具体应用定义值内容 实测值=v/10000 第 6-26 各值 为 22 个频率子段的振动值, 实测值=v/10000 (对于 11 齿泵, 第 11 阶为第 6 个浮点数, 第 22 阶为第 7 个浮点数, 以此类推) (对于 13 齿泵, 第 13 阶为第 6 个浮点数, 第 26 阶为第 7 个浮点数, 以此类推)

每次检测零件结束后, 发 1 次检测结果。

```

struct PKG_CUSTOMER2TYH_RESULT {
    int direction;
    int cmd;
    int rpm;
    int pressure;
    char pumpnmb[256];
    int status1;
    int status2;
    int v[58];
};

```

地址: 127.0.0.1 客户端口号: 9888 太阳花端口号: 9885

每个泵测试大致通信流程

- 1 客户发太阳花操作员名字
- 2 客户发太阳花转速、**压力**、泵号
- 3 客户发开始检测命令
- 4 客户发太阳花停止检测命令
- 6 太阳花停止工作, 向客户发回检测结果包

通信方式: UDP 格式

包长: 固定包长, 每个数据包 512 字节

对齐方式: 传输的结构体单字节对齐 `#pragma pack(1) ... #pragma pack()`

发送前整个数据包字节全部置零, 然后再赋值发送。

字符串先调试 `unicode`, 如果字符串 `Unicode` 不好调通, 后面就改为 `multi byte` 形式。