**前后台通讯规约**

*Ver 1.0, By hxc*

本规约用于录波器DPU板与后台工控之间的通讯，用于下载配置、整定定值、进行录波数据传送等。

1. **功能定义**

前置机主要由DPU板构成，运行实时的嵌入式软件，负责数据的采集，保存，判据计算等。可脱离后台机独立运行，其主要功能定义如下：

* 数据采集
* 启动判据
* 录波数据存储
* 实时量计算
* 记录日志
* 接受配置，包括配置下载，定值整定，零漂刻度校正等

后台机主要有工控板构成，运行Windows或LInux等操作系统，主要实现人机对话，配置编辑，数据转换与存储，远传服务等。主要功能定义如下：

* 与前置机通讯
* 对前置机下载配置
* 召唤前置机产生的录波数据，并进行数据转换
* 实现录波器人机操作
* 进行高级分析
* 实现数据远程传送

1. **通讯规约总体描述**

本规约主要用于以太网通讯场合，也可用于其它通讯网络（如：串行通讯等）。用于以太网通讯时可以选择采用UDP方式，也可以选择TCP方式。UDP方式时，每帧报文最大帧长度不要超过以太网报文帧长度，TCP方式则可以更长。

* 1. **帧定义**

报文的帧格式如下所示，其中为了适应TCP/串口等流式传输，针对这些介质还增加了可选的用于分包的启动码与结束码。在各通讯介质中，报文帧格式定义如下：

* UDP通讯

UDP由于协议层保证的包的完整性，每帧报文大小不要超过1400为佳。

|  |  |
| --- | --- |
| **报文头** | **业务数据** |

* TCP通讯

TCP数据需要应用层分包，故需要启动码与结束码，数据的有效性由协议栈保证，不需要额外的校验码。默认情况下的启动码为2 Bytes（0xA8、0xA8），结束码为2 Bytes（0x5F、0x5F）。

|  |  |  |  |
| --- | --- | --- | --- |
| **启动码** | **报文头** | **业务数据** | **结束码（可选）** |

* 串口通讯

串口通讯不但需要启动码与结束码，一般还需要校验码保证数据的完整性，如果是基于总线的结构，可能还需要加入地址码。串口通讯模式暂时不在本文考虑范畴之中

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **启动码** | **地址码** | **报文头** | **业务数据** | **校验码** | **结束码（可选）** |

* 1. **报文主体格式**

无论什么通讯介质，报文头与业务数据部分是一致的，我们称之为报文主体部分。报文主体部分格式主要如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **域** | **长度** | **描述** |
| 报文头 | 事务标识符 | 2 Bytes | 请求/响应事务处理的识别码 |
| 协议标识符 | 2 Bytes | 0x6101 |
| 备用字段 | 2 Bytes | 默认0 |
| 报文长度 | 2 Bytes | 以后报文字节的数量，到数据结束为止，不包括结束码与校验码等 |
| 功能码 | 2 Byte | Bit15（最高位）错误标识：正常响应=0，发生错误=1  Bit14方向标识：命令发起=0，命令回应=1  Bit13多帧确认标识：正常命令=0，多帧确认与请求=1 |
| 命令码 | 2 Byte | 具体的命令 |
| 帧序号 | 4 Bytes | Bit31（最高位）标识帧结束：还有后续帧=0，最后一帧=1 |
| 业务数据 | …… | …… | …… |
| …… | …… | …… |
| …… | …… | …… |

报文头为 16 个字节长，其中几个特殊字段的说明如下：

* **事务标识符**

用于事务处理配对。在响应中，服务器复制请求的事务处理标识符。 事务标识符用于将请求与未来响应之间建立联系。因此，对于通讯双方来说，在同一次会话（比如一次问答）中，这个标识符必须是唯一的。有几种使用此标识符的方式：

* 作为一个带有计数器的简单“TCP顺序号” ，在每一个请求时增加计数器；
* 用作智能索引或指针，来识别事务处理的内容，以便记忆当前的远端服务器和未处理的请求。
* **协议标识符**

用于系统内的多路复用。私有协议，目前设置为0x6101，标识我们的录波器协议。

* **报文长度**

这里的报文长度是当前字段后续所有有效字节的长度，只包括报文头剩余部分与数据字段长度，不包括此字段本身，也不包括可选的结束码与校验码。

* 1. **数据编码**

协议报文中采用大端（BIG ENDIAN）字节序进行编码，这意味着当发送多字节字段时，先发送最高有效字节。

* 1. **通讯交互流程**

目前为了简单起见，通讯模式设定为一问一答方式。同一时刻只能有一个命令进行通讯。为响应的命令，按超时处理。同样为了保证通讯的高效性，多问多答协议也兼容，但不在本文的考虑范畴。

* **命令问答机制**

通讯采用一问一答，每个命令（无论谁发起）都必须进行应答。一般来说一个命令发起后，可能有三种情况：正确的应答/错误汇报/响应超时。

命令

回应

错误

正确的应答报文的组成一般功能码的回应标识(bit14)，报文头其他字段复制自命令报文。写入命令回应报文的数据字段一般为空，读取命令回应报文则返回读取到的数据，需要多帧传输，则配合使用帧序号，启用多帧传输机制，具体参考后面的多帧传输章节。

错误的应答通过置功能码的错误标识，报文头其他字段同样复制自命令报文。数据区附带错误码，具体参考后面的异常处理章节。

超时则表示命令无反映，一般可能是链路错误导致数据丢失或者损坏，也可能是接收方功能异常。这时候发送方需要等待一小段时间，如果没有回应，直接终止当前命令。由于本文环境基于录波器网络直连，通讯比较可靠，暂时不考虑超时重传机制。

* **多帧读取数据**

报文的多帧传输流程如下，对于每一帧数据报文都必须进行确认。

读取数据

第1帧

确认并请求下帧

第2帧

**……**

最后一帧

确认

读取数据模式

读取数据时，返回的多帧数据报文格式与正常的报文回应类似，需要在功能码字段置回应位（不需要置多帧标识位），帧序号从1开始，依次递增，如果是最后一帧则置帧序号最高位。报文头其他字段复制自初始的命令报文。

目前协议下，每帧数据都需要由命令发起方进行确认，发送确认报文时，需要置位多帧标识（Bit14），标明此报文是确认请求帧，报文头其他字段复制自初始的命令报文。数据区格式如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 确认的帧序号 | UINT32 | 确认接收到的帧序号。0表示无效，不确认任何数据帧，一般在重传时有用，录波器环境可暂时不用考虑重传机制。 |
| 请求的帧序号 | UINT32 | 需要请求下一帧的序号。0表示无效，不请求任何帧，一般用于确认最后一帧，并标识全部结束 |

这里进行简单的举例说明，以一次读取定值为例，其通讯过程如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **发送方** | **功能码** | **命令码** | **帧序号** | **数据** | **说明** |
| 后台 | 0000H | 0022H | 8001H | 1, -1, -1 | 读取1号分区所有定值 |
| 前置机 | 4000H | 0022H | 0001H | xx, xx…(1024 B) | 上送第1包 |
| 后台 | 2000H | 0022H | 8001H | 1, 2 | 确认收到第1包，请求第2包 |
| 前置机 | 4000H | 0022H | 0002H | xx, xx…(1024 B) | 上送第2包 |
| 后台 | 2000H | 0022H | 8001H | 2, 3 | 确认收到第2包，请求第3包 |
| … | | | | | |
| 前置机 | 4000H | 0022H | 8009H | xx, xx…(232 B) | 上送第9包，这是最后一包 |
| 后台 | 2000H | 0022H | 8001H | 9, 0 | 确认收到第9包，结束通讯 |

* **多帧写入数据**

写入数据有时候也可能需要启动，即需要先发一帧通知前置机要写入数据（如图所示）。也可以直接在第一个报文上就直接带上第一帧数据（更常用，具体取决于命令设计）。

直接写入第1帧

确认并请求下帧

第2帧

确认并请求下帧

**……**

最后一帧

确认

直接写入模式

写入数据请求

确认并请求下帧

第1帧

确认并请求下帧

**……**

最后一帧

确认

请求再写入模式

写入数据过程中的数据发送与多帧确认与前面读取数据相似，此处不再详述。

* 1. **异常处理**

当命令发起方发出命令后，命令响应方发现了一些错误，需要进行错误响应报文。错误处理报文主要置功能码最高位标识错误（其他字段复制命令报文），并在数据中附带2个字节的错误码。常见的错误码如下：

|  |  |  |
| --- | --- | --- |
| **错误码** | **名称** | **备注说明** |
| 0001H | 无效命令 | 当前命令不被支持 |
| 0002H | 通讯忙 | 系统繁忙，无法处理当前请求 |
| 0003H | 事务错误 | 事务标识符重复或者事务已经结束 |
| 0004H | 不支持的协议 | 协议码不正确 |
| 0005H | 不支持的功能 | 功能码不正确 |
| 0010H | 无效数据 | 数据格式异常 |
| 0011H | 参数错误 | 命令请求的参数不在允许的范围或者行为中 |
| 0012H | 多帧传输错误 | 多帧传输错误，请求终止 |
| FFFFH | 自定义错误 | 跟具体命令功能相关的错误，错误码只在该命令下有效，并在紧跟着的两个字节中给出（这时数据区至少4字节） |
| <待加> |  |  |

无论任何命令，当回应为错误处理报文时，均表示命令处理失败，应该立刻停止处理。

1. **命令一览**

报文中命令码由两个字节表示，以方便扩展。目前暂且用到其中的低字节，高字节为0.。这里对常见的命令进行列举，如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类型** | **命令码** | **名称** | **说明** | **行为** |
| 系统命令 | 01H | 巡检报文 | 用于维持链路，巡检事件 | 双方都可以发起 |
| 配置命令 | 11H | 配置下送 |  | 后台机发起 |
| 12H | 配置读取 |  | 后台机发起 |
| 定值管理 | 21H | 定值整定 | 整定定值到装置 | 后台机发起 |
| 22H | 定值读取 | 读取所有定值整定值 | 后台机发起 |
| 23H | 定值区切换 | 切换定值区 | 后台机发起 |
| 24H | 定值区号读取 | 读取当前活动定值区号 | 后台机发起 |
| 录波命令 | 31H | 录波召唤 | 召唤最新的录波报告 |  |
| 32H | 录波索引读取 | 读取前置机录波索引 |  |
| 33H | 录波读取 | 读取一组历时录波报告 |  |
| 3AH | 手动录波 | 手动录波启动命令 |  |
| 查询命令 | 41H | 实时波形 | 读取一段实时波形数据 |  |
| 42H | 实时数据 | 读取当前各通道数值 |  |
| 调试命令 | 91H | 自检查询 | 查询装置自检状态 |  |
| 92H | 版本查询 | 读取装置版本信息 |  |
| 93H | 日志查询 | 读取装置日志 |  |
| 94H | 子模块查询 | 读取置其他模块与功能状态 |  |
| A0H | 复位命令 | 此命令将复位录波器 |  |
| A1H | 写IP地址 | 设置装置网口IP地址 |  |
| A2H | 写系统时间 | 设置装置时间 |  |

1. **系统命令**
   1. **巡检命令**

巡检命令最主要的作用是用户通道链路的维持，可以由通讯双方发起，当另一方收到巡检后应该立刻回复。默认情况下，当链路空闲超过10秒后，就应该进行链路巡检。实际巡检时间最好可配置。

巡检命令另外一个作用就是进行事件报告，这里主要是前置机向后台机进行数据报告。比如当有新录波生成时，需要后台机召唤时，在每次巡检的数据字段中都会进行报告。

当前置机一切正常时，巡检报告或者巡检回令可以不带任何数据。当有事件发生时，将带上数据，并在数据字段中包含上送事件的掩码。

|  |  |  |
| --- | --- | --- |
| **掩码位** | **事件** | **说明** |
| Byte0:7 | 自检异常 | 报告自检异常，上位机可进行自检查询，并通知管理人员 |
| Byte0:6 | 录波报告 | 报告由新的录波产生，上位机可以召唤新录波 |
| *<待加>* |  |  |

需要注意的是，目前数据只有1个字节，随着将来事件种类的真多，将会依次扩展到多字节，后台机解析时需要注意支持。

1. **配置命令**

配置的下送与读取都是以文件的方式进行的。配置读写将是一个进行多帧传输的会话。配置下送后不会立刻生效，必须手动重启才能生效。如果配置异常，将会报自检异常。

配置的格式将采用XML文本的方式存储与传输，具体细节请参考配置相关文档。

1. **定值管理命令**

定值的管理仅仅是对定值值的读写，如果定值结构发生变化，请通过配置更新。

* 1. **定值整定**

定值整定将一次写入所有要写的定值值，其数据格式如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 定值区号 | UINT16 | 要整定的定值区 |
| 定值项数目 | UINT16 | 要整定的定值数目 |
| 第1项定值 | SET\_VALUE |  |
| 第2项定值 | SET\_VALUE |  |
| …… |  |  |
| 第n项定值 | SET\_VALUE |  |

每项定值SET\_VALUE的结构如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 定值ID | UINT32 | 定值独一无二的ID，由系统或工具分配 |
| 定值索引 | UINT32 | Bits[0~3]:数据类型，Bits[4~17]: 组内索引，Bits[18~31]: 分组号 |
| 值 | UINT32/FLOAT32 | 定值的值 |

如果一次要整定的定值过多，将启用多帧机制进行传输。

* 1. **定值读取**

定值的读取的数据区如下:

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 定值区号 | UINT16 | 要读取的定值区 |
| 定值组号 | UINT16 | 要读取的定值组号，0xFFFF，表示读取所有分组 |
| 定值组内索引 | UINT16 | 要读取的定值项，0xFFFF，表示读取整组数据 |

读取的数据区格式与定值整定相同，如果读取返回的定值过多，将启用多帧传输机制。

* 1. **定值区切换**

定值区切换的数据区如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 定值区号 | UINT16 | 要切换的定值区 |

切换成功将返回之前正在使用定值区号，以及切换后新的定值区号，数据如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 老定值区号 | UINT16 | 之前正在使用定值区号 |
| 新定值区号 | UINT16 | 切换后新的定值区号 |

* 1. **定值区读取**

定值区读取报文无数据，返回的数据区如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 定值区号 | UINT16 | 当前定值区号 |
| 定值区范围 | UINT16 | 总共的定值区号数目 |

1. **录波管理命令**
   1. **录波召唤**

录波召唤命令用于召唤新产生的录波记录。如果录波器一段时间内新产生了好几条录波记录，录波召唤命令会让前置机将依次将这些录波记录上送后台机。需要注意的是，一条录波召唤命令将只召唤一条录波记录。如果有多条新录波记录，需要多次召唤。

录波召唤命令无附带数据，如果录波器存在新的录波记录，召唤命令将返回新录波记录的索引号，如下所示：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 录波起始索引号 | UINT32 | 新录波记录索引号，从1开始。0表示无新记录 |
| 新录波记录数目 | UINT32 | 当前装置新录波记录的数目 |

当返回的录波记录非0时，后台机需要马上启用录波读取命令，读取具体录波数据。只有被后台机读取过的录波记录，才会被认为不再是新的录波。

* 1. **录波索引读取**

录波索引读取命令将检索前置机内部保留的录波记录信息。

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 起始时间 | UINT32 | 标准秒，1970年到现在的秒数，0表示无限制 |
| 结束时间 | UINT32 | 同上 |
| 最大返回数目 | UINT32 | 本次读取最大数目，0表示无限制 |

返回录波索引信息数据如下表所示，如果数据过长将启动多帧传输。

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 数目 | UINT32 | 当前返回的录波数目 |
| 录波索引1 | REC\_INDEX |  |
| 录波索引2 | REC\_INDEX |  |
| …… |  |  |
| 录波索引N | REC\_INDEX |  |

每条录波索引结构如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 索引号 | UINT32 | 当前返回的录波数目 |
| 启动时间 | UINT32 |  |
| 简要信息 | UINT32 | 主要包括简单的启动原因等信息，具体由录波器定义 |

* 1. **录波读取**

录波读取命令用于实际读取指定索引号的录波记录，数据格式如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 索引号 | UINT32 | 要读取的录波记录索引号 |

将直接返回该条录波记录原始的数据。数据如果过长，将分多帧进行传送。原始记录格式由录波器定义，暂定如下（后期可能根据需求发展进行适量改动）。

|  |  |  |  |
| --- | --- | --- | --- |
| **块** | **名称** | **数据类型** | **说明** |
| 数据头 | 模拟通道数目 | UINT16 |  |
| 开关通道数目 | UINT16 |  |
| 模拟通道一次系数 | FLOAT32 |  |
| 模拟通道二次系数 | FLOAT32 |  |
| 录波时刻（秒） | UINT32 |  |
| 录波时刻（纳秒） | UINT32 |  |
| 故障时间（秒） | UINT32 |  |
| 故障时刻（纳秒） | UINT32 |  |
| 线路频率 | FLOAT32 |  |
| 采样段数目 | UINT32 |  |
| 数据块数目 | UINT32 |  |
| 采样段1 | 采样率 | FLOAT32 |  |
| 数据点 | UINT32 |  |
| … |  |  |  |
| 采样段N |  |  |  |
| 数据块1 | 编号 | UINT32 | 最高位为1时，表示是最后一个数据断面 |
| 模拟通道1 | UINT32 | 暂时用32位整形存储 |
| … |  |  |
| 开关量1~32 | UINT32 | 开关量按位存储，不足32的倍数，后面补0 |
|  |  |  |
| … |  |  |  |
| 数据块M |  |  |  |

当数据录制的过程中发生意外（比如断电），将可能导致数据块不完整。这是可能需要后台机软件进行专门的处理。

* 1. **手动录波**

该命令将产生一条手动录波记录。无附加数据。

1. **查询命令**

查询命令主要用于后台机向前置机读取一些实时的数据，用于展示或者调试。

* 1. **实时波形读取**

实时波形读取命令将返回当前一段实时的数据（一般10个周波以内），命令请求数据如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 采样点数目 | UINT32 | 要读取的实时波形采样点数目，最大不要超过10个周波 |

返回的数据格式比较类似录波原始记录，如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **块** | **名称** | **数据类型** | **说明** |
| 数据头 | 模拟通道数目 | UINT16 |  |
| 开关通道数目 | UINT16 |  |
| 模拟通道一次系数 | FLOAT32 |  |
| 模拟通道二次系数 | FLOAT32 |  |
| 起始时刻（秒） | UINT32 |  |
| 起始时刻（纳秒） | UINT32 |  |
| 数据块数目 | UINT32 |  |
| 数据块1 | 编号 | UINT32 | 最高位为1时，表示是最后一个数据断面 |
| 模拟通道1 | UINT32 | 暂时用32位整形存储 |
| … |  |  |
| 开关量1~32 | UINT32 | 开关量按位存储，不足32的倍数，后面补0 |
|  |  |  |
| … |  |  |  |
| 数据块M |  |  |  |

* 1. **实时数据读取**

实时数据读取命令将返回装置内部计算量值，用于调试或者数据查询。<待加>

1. **调试命令**

调试命令多用于改变装置行为，查看装置自身状态等。这类命令比较简单，部分命令项目后期根据需要再加，此处仅对几个主要的命令进行介绍，其他命令不再一一说明。

* 1. **自检查询**

自检查询将用于检查装置自身状态信息。命令返回如下：

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 装置状态掩码 | UINT32 |  |
| 模块状态掩码 | UINT32 |  |
| 插件状态掩码 | UINT64 | 由最高位开始，分别标识各位置插件是否故障 |

* 1. **版本查询**

用于查询装置的版本信息，返回的数据如下。

|  |  |  |
| --- | --- | --- |
| **名称** | **数据类型** | **说明** |
| 装置平台版本 | UINT32 |  |
| 装置程序版本 | UINT32 |  |
| 装置程序CRC | UINT32 |  |
| <其他待加> |  |  |