# **采集传输系统软件实现**

在本文研发的基于ARM的海洋油气勘探采集传输系统软件中，由自主研发的通信协议确定外部接口，由PCIe实现CPU与FPGA之间的高速数据传输，由命令接收模块、命令处理模块实现节点与上位机的命令交互，由数据采集发送模块实现声波数据的上传，由板内诊断模块和链路诊断模块协同完成诊断系统。整个节点软件基于自主研发的系统框架研发。本章将首先介绍采集传输系统的外部接口，然后描述系统的主线流程，再详细介绍主线流程下各个模块，论述命令交互、采集传输和故障诊断功能的具体实现。

## 5.1 外部接口

本论文研究设计的采集传输系统针对传输速率指标及小型化指标专门设计了清晰、灵活、可靠性强的外部接口，并基于该外部接口研发了采集传输系统的软件实现及故障维护。外部接口在整个水声探测系统中的作用范围如图5.1红色虚线框所示：

图5.1 节点、网关、上位机通信方式概要图

从图中可知，外部传输接口的作用范围主要集中在采集传输节点与网关的通信上，网关可向采集传输节点发送广播和单播命令，采集传输节点向网关回应命令及上传采集数据，都采用自定义的通信协议。网关在接收到节点或者上位机的通信信息时，对协议进行一个交替转换后再转发给上位机或者节点，除此之外，对命令的内容和数据的转发几乎是透明的，主要业务逻辑集中在采集传输节点上实现。因此，采集传输节点的外部接口是否设计合理至关重要。

整个外部接口可总体概括为传输协议、命令协议、数据协议、处理器与FPGA之间的交互协议以及CRC可靠性验证五个模块，下面将逐个进行介绍。

### 5.1.1 传输协议

传输协议定义了上下行链路通信的帧格式，为实现数据的监听、接收、转发和验证提供系统规范，其具体格式如图5.1所示。

图5.1 外部接口传输协议结构图

传输协议帧格式为网络监听帧头SAV配上数据包DATA的形式，数据包的检测通过网络监听帧头SAV来识别，SAV的固定构成为{8’AA,8’hAA，8’hAB}，大小为3\*8bits，当检测到SAV后，FPGA通过FIFO或者RAM缓存得到一个包的数据，再对包头目的地址进行判断，若目的地址是本地地址则将数据缓存到用以存放本地数据的ram中，若目的地址不是本地地址，则将数据缓存在存放转发数据的ram中。监听处理流程如图5.2所示：

图5.2：数据包监听处理流程图

数据包主体由头部和数据两部分组成。头部主要包含传输协议的交互信息；数据部分则是网络包的主要内容，分为控制命令和实际采集数据，其长度不确定。头部大小为12Bytes，包括2Bytes的目的IP地址，2Bytes的源IP地址，2Bytes的长度信息，1Byte的方向信息，1Byte的端口信息，1Byte的功能选择信息，2Bytes的保留字节以及1Byte的头部CRC校验码。

IP内容主要区分广播包与IP指定包。广播包统一格式为255.255，IP指定包第1个字节固定为192，第二个字节确定节点地址。当接收到目的IP地址为255.255时，接收并转发给处理器，当检验到目的地址为192.xxx时，与本地地址进行校对，若向同，转发给处理器，若不同则转发给下一个节点；若目的地址为其它格式，则直接丢弃。

IP信息定义表参照如表5.1所示：

表5.1 IP信息定义表

|  |  |
| --- | --- |
| 具体意义 | IP(2Bytes) |
| 广播包 | 255.255 |
| IP指定包 | 192.XXX(0 - ?) |

Length中携带了后面Data部分的整体数据长度（不包括Data后面的CRC部分），根据系统指标可得出最大数据包大小为Bytes，故Length的长度取2Bytes。其中2代表2毫秒，数据包每两毫秒传输一次，12代表采样率12kHz，16代表声通道数，24代表分辨率，除以8换算成字节单位。

Direction内容确定硬件上双环路的数据包传输方向，数据包在默认情况下，左右交替发送数据包，当一边链路故障时，不交替发送，所有的数据包都从链路正常的一端发送。内容解析如表5.2所示：

表5.2 Direction字节定义信息表

|  |  |
| --- | --- |
| 具体意义 | Direction(1Byte) |
| 左向 | 0x01 |
| 右向 | 0x02 |
| 双向 | 0x03 |

Port中的内容用于软件内容的判断逻辑。功能选择信息主要用来指示该数据包用于实现的功能，主要分为数据包发送端口、命令包接收及响应端口、调试信息上报端口、诊断描述端口和延时测量等。默认设置为数据包，其具体值与相应的功能对应如表5.3所示：

表5.3 Port字节定义信息表

|  |  |
| --- | --- |
| 具体意义 | Function Select(1Byte) |
| 数据包发送端口 | 0x00 |
| 命令包接收及响应端口 | 0x01 |
| 调试信息上报端口 | 0x02 |
| 诊断模式端口 | 0x03 |
| 延迟测量端口 | 0x10 |
| 时钟切换端口 | 0x11 |

保留字节2Bytes，均默认置1。当功能选择为时钟频率切换时，该保留字节代表将要切换的频率。

### 5.1.2 命令协议

命令协议定义了采集传输节点与上位机之间的命令交互，为实现命令接收、命令处理和命令响应提供系统规范，其具体格式如图5.3所示。

图5.3：外部接口命令协议结构图

命令包主体存在于数据包的DATA，以端口号区分DATA中是数据包还是命令包，为增加系统灵活性，命令包大小不固定，根据其附加通信数据Payload的不同而变化，除附加通信数据外，其通信格式主要包括版本号Version、命令序列号Sequence、命令大类Type、命令内容Context、命令长度Length、掩码Mask和预留位Reserve。

版本号Version：用于保存每次采集传输节点代码更新记录，以便于后期维护管理。

命令序列号Sequence：上位机每向节点发送一次命令，序列号Sequence递增1，节点在进入命令处理前线对序列号进行判断是不是最新序列号，以防止相同命令被处理多次。序列号溢出则从0开始。

命令大类Type：结构清晰，层次感分明的代码往往更便于开发设计及后期维护，随着后期系统需求的增加，节点需要实现的功能更加智能化，后期需要增加的命令肯定也会越来越多，鉴于此，本系统将命令划分为两个层级，提取大类，目前主要分为常规命令类、信息提取类、代码更新类和调试命令类四大类。

命令内容Context：命令内容为上位机具体需要实现的功能，结合命令大类Type共同实现跳转至命令处理函数。

命令长度Length：存放命令包的整体长度；

掩码Mask：Mask长度达14字节，其作用为再次验证命令包是发送至本采集传输节点。14个字节共112个比特位，通过判定Mask对应ID号位的比特位是否为1进行判断，如果是，则判定该命令包是传输至本节点，如果为0，则丢弃。因此，如果Mask全为1，可得知其为广播包。

保留位Reserve用于后期功能拓展；

附加通信数据Payload：附加通信数据一般用于存放节点回应信息的一些数据，例如命令处理成功、命令处理失败、命令异常等。另外，payload还可用于存放调试信息。

### 5.1.3 数据协议

数据协议定义了采集传输节点与数据汇聚系统之间的数据交互，为数据汇聚系统进行所有节点采集的声波数据整提供系统规范，其具体格式如图5.4所示。

图5.4：外部接口数据协议结构图

数据协议定义了采样数据在DATA中的存放格式，便于数据汇聚模块进行打包整理时抽取。其主题由节点号ID、时间标签号timetag和采样数据实体组成。

时间标签号timetag大小为1字节，每个数据包递增，系统根据这个标签号的奇偶性确定发送方向，奇数标签向左发送数据包，偶数标签向右发送数据包；

采样数据sample data中存放了2ms的采样数据，由于采样时钟为12KHz，所以2毫秒一共采样了24次，每次共有16个通道的数据，每个通道的分辨率为24位，因此，一个数据包的采样数据大小为字节。

### 5.1.4 CPU与FPGA交互协议

CPU与FPGA交互的具体实现是在节点板内，但此处讨论主要针对此两者协议上的交互，因协议上的交互最终也是为了配合外部接口，故将其划分至外部接口描述，其数据交互具体实现的驱动设计，将在后续小节作详细讨论。

CPU与FPGA交互协议讨论的要点为为什么数据包在FPGA接收后需要转发给CPU以及什么数据包在FPGA接收后不需要转发给CPU，而是自己进行相关处理后直接转发至下一节点。数据包类型主要分为2种，一种是采样数据包，另一种是功能包。采样数据包包括自身采样数据包和转发自上一节点的采样数据包；功能包主要实现水上系统的命令，包括开始/停止采样、时钟切换、延迟测量等功能。FPGA实现对数据包头部的解析以及CRC检验。根据解析结果，FPGA将数据包发送至相应模块进行后续处理。除自身采样数据包外，FPGA在收到数据包时都首先对数据包包头进行检验，再做后续处理。

若数据包为自身采样数据包，FPGA通过中断的形式通知CPU来取采样数据，CPU通过PCIe接口从FPGA的双口RAM读取采样数据，进行相应处理后再通过PCIe总线发送至FPGA，FPGA根据包头信息产生包头CRC校验码，再转发至下一个节点。

若数据包为转发采样数据包，FPGA在判断目的地址后便可得知，直接将其转发至下一个节点。

若数据包为延迟测量包，延迟测量包由网关发出并统计，对于延迟测量包，只有数据包头部，数据长度为0，FPGA若判定该数据包为延迟测量包，再判断该延迟测量包目的地址是否本地地址，若是则将源地址填充为本地地址，目的地址改为网关地址，并根据包头信息产生包头CRC校验码，从同向端口回包；若不同，则转发至下一个节点。网关统计每个节点的延迟值后给每个节点发延迟配置命令包，数据内容是节点的延迟值。节点同样判断该延迟配置包地址是否与本地地址匹配，若是，则获取该延迟值，并做延迟逻辑（延迟逻辑的实现通过节点的同步时钟计数实现，每个节点按照延迟值的大小先后给将同步时钟发给AD采样的主时钟），否则转发给下一个节点。

若数据包为时钟切换数据包，对于时钟切换数据包，其数据部分包含每个节点地址以及相应的延迟值。FPGA若判定该数据包为时钟切换数据包，应通过PCIe发送模块将该数据包发送至CPU，由CPU去做数据部分的CRC32校验并获取相应的延迟值。CRC32校验成功之后再由CPU通过PCIe总线将数据发送至FPGA，FPGA根据包头信息产生包头CRC校验码，再转发至下一个节点。

若数据包为时钟切换数据包，对于时钟切换数据包，其数据部分包含每个节点地址以及相应的延迟值。FPGA若判定该数据包为时钟切换数据包，应通过PCIe发送模块将该数据包发送至CPU，由CPU去做数据部分的CRC32校验并获取相应的延迟值。CRC32校验成功之后再由CPU通过PCIe总线将数据发送至FPGA，FPGA根据包头信息产生包头CRC校验码，转发至下一个节点。

若数据包为其余功能包，FPGA在检验包头判定为发往本地时，将该数据包通过PCIe发送模块发送至CPU，由CPU去做数据部分的CRC32校验，检验成功并处理后，再由CPU通过PCIe总线将数据发送至FPGA，FPGA根据包头信息产生包头CRC校验码，转发至下一个节点。

### 5.1.5 CRC校验实现

本系统研发的传输协议按照数据包信息的功能分别对数据包包头和数据部分进行CRC校验。对于包头部分，由单独的CRC校验码对包头进行单独的CRC校验。包头CRC校验采用CRC-8校验方式，因此包头CRC长度为1Byte。

数据部分则是网络包的主要内容，分为控制命令和实际采集数据，其长度不确定，但长度信息可以通过包头的Length值得到，同样，数据部分也有单独的CRC校验码用来进行数据部分的CRC单独校验。数据部分的CRC校验采用CRC-32校验方式，因此数据部分的CRC长度为4Bytes。本系统CRC校验流程图如图5.5所示：

图5.x：CRC校验流程图

当系统监听到数据包时，首先对接收到的数据包的包头CRC进行检验，如果解析不通过，则丢弃该数据包继续监听。如果检验通过，则解析数据包包头，判定是否发往本节点，如果不是则转发，如果是，则转发给处理器，处理器在接收到数据包时，首先检验数据包的CRC，如果检验不通过，则丢弃该数据包，继续监听，如果通过，转入命令处理等模块，处理完成后生成回应码以及对应的数据包CRC，发送给FPGA，FPGA根据包头信息及CRC校验码生成规则，生成包头校验码填充至协议数据帧，转发至下一节点。

如果在监听数据帧时发生中断，生成采集数据包，则针对采集数据包产生CRC校验码，发送给FPGA，FPGA根据包头信息及CRC校验码生成规则，生成包头校验码填充至协议数据帧，转发至下一节点。

传统的比特型校验函数每处理1字节的信息码时，需要进行8次以为操作，校验速度很慢，这在RAM资源紧张的嵌入式操作系统中是非常不合理的。因此，本论文采用查表法的方式实现CRC校验。首先通过外部代码计算得出CRC32码码表存入数组crc32\_table[256]中，此后在待校验的字节序列表中的每一个字节参与校验时，将其与前一个CRC校验码异或，再以该异或值为码表索引获得当前CRC校验码。

查表法的方式实现了从比特位到字节位的扩展，从原理上来讲大约能省去87.5%的计算量，相当于7倍的速度提升。经大量的样本测试，CPU利用传统方式对10KB数据进行CRC校验时所消耗的时间大约为22ms，利用查表法方式校验10KB的数据所耗费的时间大约为3ms。所以用查表法的方式可以大大提高校验速率，进而在提高系统可靠性时又尽可能地保证了数据传输不受影响。

图5.6：CRC校验的比特型方法和查表法性能对比图

## 5.2 系统主线

外部接口设计了数据包在外部链路上的传输协议，本节开始将主要介绍节点内部的软件实现，首先从软件主线流程开始讲起，其次分别介绍各功能的具体实现。软件系统主线流程框图如图5.7所示：

图5.7 软件系统主线流程图

采集传输节点在系统上电后，首先运行boot引导程序启动，将系统代码通过DMA从Flash闪存搬运至DDR SDRAM内存，搬运完成后跳转指针，运行系统程序。系统程序启动后，首先进行初始化操作，配置PLL锁相环设置系统工作时钟频率，初始化串口、PCIe等驱动，注册中断处理函数以及配置节点默认参数，完成这些操作后进入命令接收循环。如果收到网络命令，首先判断命令是否有效（是否发往本节点、命令序列号对不对等），如果无效，则发送回应，如果有效，继续判断当前该命令是否为进入诊断模式命令（因当要对系统进行诊断时，不希望发生中断请求，所以对诊断命令单独设置一个系统工作模式——诊断模式），如果是进入诊断模式命令，关闭系统中断，进入诊断模式，否则继续判定其是否为退出诊断模式命令，是则开中断，进入常规模式，否则进入命令处理阶段。诊断命令只能在诊断模式下响应，常规命令只能在常规模式下响应。

## 5.3 命令交互

采集传输节点命令交互的直接对象为网关系统，但是由于网关系统在整个水声探测系统中主要起到中转作用，对命令信息几乎是透明的。因此，节点命令交互的对象实质上就是上位机，命令交互示意图如下图所示：

图5.x 上位机、网关、节点命令交互示意图

如上图所示，网关在接收到上位机命令时不去关注命令的具体内容，直接转发给节点，但其在转发出去后不管节点是否有回应，都会向上位机回应一个转发成功命令，以保证上位机与网关直接通信正常，如果节点异常未回应，也可直接定位异常部位发生在网关与节点之间。节点正常回应后，网关通过协议解析转换为以太网数据包转发给上位机。

节点与上位机的命令交互按照功能可细分为命令接收模块、命令处理模块和命令种类三个方面，下面将逐个进行介绍。

### 5.3.1 命令接收模块

命令接收模块的首要功能为当接收到网络数据包时，对网络数据包进行判断其是否为发往本节点的数据包，其次，判定命令的格式是否符合要求，如是否为协议内定义的命令，是否命令重发，已经处理国该命令等。

命令接收模块的逻辑流程图如图5.x所示：

图5.x 命令接收模块逻辑流程图

如上图所示，命令接收模块在收到数据包时，首先对协议栈进行解析，通过掩码判断该数据包是否为发送至本节点的数据包，如果不是，则直接丢弃，无回应。如果是发往本节点的数据包，则解析命令格式的公共头部，判断命令格式是否正确，是否出现命令多处理，命令不匹配等情况，如果命令格式不正确，则发送格式不正确系列回应，如果命令格式正确，则调用命令处理模块进入命令处理流程。

### 5.3.2 命令处理模块

介绍接受桥接命令，各模块间间交互流程，给出整个交互框图；



### 5.3.3 命令种类



## 5.4 数据交互

### 5.4.1 实现机制

### 5.4.2 乒乓缓存

### 5.4.3 数据采集发送模块

## 5.5 故障诊断

### 5.5.1 链路诊断

### 5.5.2 板内诊断

### 5.5.2.1 FPGA诊断

### 5.5.2.2 串口诊断

### 5.5.2.3 网络诊断

### 5.5.2.4 闪存诊断

### 5.5.2.5 内存诊断

## 5.6 本章小结