FOMES传输协议规范

J

Version 1.0 Published

# 版本声明

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 性质 | 作者 | 日期 |
| Version1.0 | Published | Jerry Zhang | August 28th, 2014 |
|  |  |  |  |
|  |  |  |  |

**目录**

[版本声明 2](#_Toc397014398)

[1. 协议概述 4](#_Toc397014399)

[2. 协议模型 5](#_Toc397014400)

[2.1. 物理层 5](#_Toc397014401)

[2.1.1. 前端设备物理介质 5](#_Toc397014402)

[2.1.2. 后端设备物理介质 5](#_Toc397014403)

[2.1.3. 传输管道介质 5](#_Toc397014404)

[2.2. 数据链路层 6](#_Toc397014405)

[2.2.1. 前端内存组织 6](#_Toc397014406)

[2.2.2. 后端内存组织 6](#_Toc397014407)

[2.2.3. 关键数据结构 7](#_Toc397014408)

[2.2.4. 关键信息字 9](#_Toc397014409)

[2.3. 传输层 10](#_Toc397014410)

[2.3.1. 前端设备传输组织 10](#_Toc397014411)

[2.3.2. 后端设备传输组织 11](#_Toc397014412)

[2.3.3. 普通数据通道（GDP） 12](#_Toc397014413)

[2.3.4. 紧急错误通道（E2P） 13](#_Toc397014414)

[2.3.5. 设备空间和传输地址映射 13](#_Toc397014415)

[2.4. 应用层接口 13](#_Toc397014416)

[2.4.1. 传输管道建立 13](#_Toc397014417)

[2.4.2. 下发指令 14](#_Toc397014418)

[2.4.3. 获取反馈信息 14](#_Toc397014419)

[2.4.4. 传输管道关闭 15](#_Toc397014420)

[2.4.5. 意外处理 15](#_Toc397014421)

[3. 基本通信机制 15](#_Toc397014422)

[3.1. 普通数据传输机制 15](#_Toc397014423)

[3.2. 突发状况反应机制 16](#_Toc397014424)

# 协议概述

FOMES（Fast Organized Memory Exchange Specification）全称快速组织内存交换规范，是应用于表面贴装机系统，用于连接上位机系统与下位机控制单元，基于PCI总线传输和后端内存交换结构的基本通信协议。FOMES基于内存管理模式的指令交互，通过完成上下数据打包传送，实现控制指令下载和反馈信息上传。FOMES具有以下特点：

* 基于OSI协议模型，物理层、数据链路层、传输层协同设计；
* 基于系统内存操作模式，包含缓存数据包的管理、传输和地址映射，实现系统数据传输分级隔离，提高传输系统的可靠性和稳定性；
* 简化的系统应用层接口和基于双口RAM的机电硬件层接口，具有很强的可移植性；
* 包含8位异或校验字；
* 普通下载和上传数据通过普通数据通道GDP（General Data Passage）进行传输，采用主控轮询模式完成数据传送；错误、故障信息通过紧急错误通道E2P（Emergency & Error Passage）直接上传至应用程序界面，该方式基于系统中断。GDP与E2P相互隔离，资源分开，访问不同的内存区，互不影响；
* 支持多种总线操作模式，根据底层存储介质不同可调整传输模式，提高系统性能；
* 内含独立的指令队列和轮询机制，自动处理堆积指令，协调系统与机电设备的速度差别。

基于FOMES开发了应用于表面贴装机的FOMES传输管道（以后简称传输管道）。传输管道基于FOMES模型，实现上层应用系统与底层机电设备的信息交互传输。

# 协议模型

FOMES基于OSI协议模型，完成了物理层、（数据）链路层、传输层的设计。物理层完成传输管道的物理介质实现；链路层完成了物理介质的信息结构组织；传输层完成了传输资源调度和协议机制实现。传输层向系统应用层提供动态链接库形式的接口调用。

## 物理层

### 前端设备物理介质

上位机平台为32位工控机，搭载Windows XP SP3系统，数据字长为32位，存储字长为8位（字节存储），存储模式为BIGEND，即高字低址，低字高址。默认编译32位长字对齐。

上位机接口为32为工控机PCI系统总线，遵守PCI总线协议，数据字长32位、16位可设，电平标准5V-PCI电平。上位机缓存为工控机内存，存储字长8位，存储模式BIGEND。

### 后端设备物理介质

下位机平台为16位DSP，TMS28335（Texas Instruments），数据字长16位，存储字长16位。默认编译32位长字对齐。下位机接口为双口RAM读写端口，遵守双口RAM读写时序，数据字长16位。下位机缓存为双口RAM设备，存储字长16位，容量4K。具体细节请参阅*《双口RAM芯片手册》*。

### 传输管道介质

传输设备平台为PCI传输板卡，数据字长16位，无控制器搭载。传输平台缓存为16位FIFO，内置于PCI9054内部，写FIFO深度32，读FIFO深度16，存储字长16位。

## 数据链路层

### 前端内存组织

#### 数据结构静态组织

前端内存组织包括一个用来存放指令数据包信息的指令向量表和一个用来存放反馈数据包信息的反馈向量表，以及用来存放数据包包体信息的堆栈空间。两个向量表结构完全一致，其区别仅仅是存储的数据包信息不同，指令向量表存放的是指令包信息，反馈向量表存放的是反馈包信息。

向量表的元素是待下发的指令数据包或已上传的反馈数据包的关键信息索引，该索引包括数据包的长度，数据包对应的指令ID和数据包的包体数据指针。

数据包长度：表示数据包体的链路层缓存长度，以字节计数；

数据包指令ID：唯一标识每个数据包链路层身份的ID标识，具体信息详见*特殊信息字*相关章节。

数据包体数据指针：指向被保存在堆栈空间内的数据包体。

#### 向量表的动态运行机制

当系统应用程序下发指令时，传输管道将数据包的属性信息填入指令向量表，将数据包体复制到堆栈缓存空间中，等待下发；指令发下后，传输管道冲洗下发数据，释放堆栈空间，推进向量寻址指针。

后端设备有反馈信息更新时，传输管道读取反馈信息数据，将反馈数据包读入系统堆栈空间，填写反馈信息，等待应用程序读取反馈信息；应用程序读取反馈信息时，传输管道在反馈向量表中搜索数据包信息，将数据包复制到系统提供的缓存指针中，在接口函数返回时冲洗上传数据，释放堆栈缓存。

### 后端内存组织

后端内存的物理介质是双口RAM，内存组织共分四区，分别用来存储下载指令数据、上传反馈数据、紧急错误数据和中断传输控制数据。下载数据和上传数据区还有分段，分别存放数据包体和备注信息。关于数据包体和备注信息的相关信息详见*普通数据传送包*和*数据包备注信息*相关章节。后端内存组织区段地址分配如图 1和表 1所示。



图 1 后端双口RAM地址分配表

### 关键数据结构

#### 普通数据传送包

系统指令集共包含57种指令。每条指令对应一个输出结构体和一个输入结构体。输入结构体包含自前端设备发送至后端设备的指令信息，输出缓存包含后端设备提供给前端设备的反馈信息。系统指令集以及对应的输入输出结构体定义等内容请查阅文档*《上位机控制指令和返回状态说明》*。

输入输出结构体实例化后的数据缓存称为数据传送包，简称数据包，其中输入结构体对应的数据包称为指令（数据）包，或称下载包，输出结构体对应的数据包称为反馈（数据）包，或称上传包。

表 1 后端双口RAM区段分配

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 区地址 | 区名称 | 段地址 | 段名称 | 备注 |
| 0x000 –  0x7FF | 下载区 | 0x000 – 0x7FB | 下载信息区 | 系统下发的指令包缓存区 |
| 0x7FC – 0x7FF | 备注信息区 | 指令包的备注信息 |
| 0x800 –  0xBFF | 上传区 | 0x800 – 0xBFB | 上传信息区 | 待上传至系统的反馈信息 |
| 0xBFC – 0xBFF | 备注信息区 | 反馈信息的备注信息 |
| 0xC00 – 0xFFD | 错误信息缓存 |  |  | 存放错误信息的缓存 |
| 0xFFE | 上传控制字 |  |  | 数据上传中断控制 |
| 0xFFF | 下载控制字 |  |  | 数据下载中断控制 |

#### 数据包备注信息

表 2 数据包备注信息结构表

|  |  |  |
| --- | --- | --- |
| 偏移地址（字节地址） | 名称 | 备注 |
| 0x0 – 0x3 | 指令标号 | 32位指令数据ID |
| 0x4 – 0x5 | 指令长度 | 16位指令长度信息 |
| 0x6 | 校验码 | 16位校验码 |
| 0x7 | 保留 | 全“0” |

每个数据包都对应一个2X32bit的备注信息随包一同下载或上传，数据包体被存入下载或上传信息区，备注信息存入备注信息区。数据包备注信息如表 2所示。

#### 紧急错误信息包

待定

### 关键信息字

#### 指令标号（指令ID）

指令标号不同于指令结构体定义中的指令码，有关指令码的信息请查阅文档*《上位机控制指令和返回状态说明》*。指令标号是为区别不同数据包定义的，相对意义上独一无二的缓存编码，是基于链路层数据传输定义的概念，以便从链路上区分不同的缓存数据。指令标号是由FOMES传输层生成的，随每个普通数据包下发的32位无符号整形编码，位于备注信息区内。每个指令包和其对应的反馈信息包共享相同的指令ID。

之所以说独一无二是相对意义上的，是因为理论上指令ID理论上只能穷举个。FOMES假定系统应用层不再理会相当长一段时间以前的反馈信息。因此，当指令ID用尽之后，会循环使用，被冲突的未处理旧包将会被无条件冲洗。

#### 校验码（8位）

校验码是普通数据块传输正确与否的判断标准。上传数据和下载数据校验码的算法为：将缓存区数据以字节为单位进行算术加和，结果8位截断。

#### 数据上传控制字（16位）

数据上传控制字是后端设备写入双口RAM的0xFFE单元里的16位信息字，向上位机表明有关信息上传的相关表示。信息字定义如表 3所示。

表 3 数据上传控制字定义

|  |  |
| --- | --- |
| 字码 | 字义 |
| 0x1212 | 反馈信息已就绪 |
| 0x2323 | 校验码出错 |
| 0x3434 | 指令码出错，或指令无效 |
| 其他 | 无效 |

#### 数据下载控制字（16位）

数据下载控制字是前段设备写入双口RAM的0xFFF单元里的16位信息字，向下位机传达系统有关指令下载的相关指示。信息字定义如表 4所示。

表 4 数字下载控制字定义

|  |  |
| --- | --- |
| 字码 | 字义 |
| 0x5454 | 指令下载已就位 |
| 其他 | 无效 |

## 传输层

### 前端设备传输组织

传输层通过前端接口为系统应用层提供调用接口动态链接库，包括在数据链路层定义的静态内存组织和动态数据结构，即指令向量表和反馈向量表，以及运行中的轮询线程和中断线程。

轮询线程和中断线程是FOMES传输层的控制中心，控制传输管道的运行和管道资源的动态管理。轮询线程和中断线程在DLL被调用后即开启运行，在主程序发送管道关闭信息后，完成一个最小动作后结束运行。

轮询线程在运行期间定时查询后端设备是否有更新的反馈数据需要上传，以及指令向量表中是否有新指令数据包需要下载，并且对可能出现的传输问题进行独立处理。

中断线程则始终监听PCI系统中断，一旦PCI系统中断触发，该线程会在第一时间读取紧急错误信息，不加处理地迅速发送至系统界面。

### 后端设备传输组织

传输管道在硬件上通过37针并行线与后端设备相连，数据交互基于异步中断传输，便于提高设备效率和实现数据传输隔离。后端并行线定义如表 5和图 2所示。

表 5 后端并行接口引脚定义

| 引脚标号 | 引脚方向 | 引脚名称 | 引脚定义 |
| --- | --- | --- | --- |
| 1 |  | GND | 地线 |
| 2 | BIDIR | SD14 | 16位并行数据线 |
| 3 | BIDIR | SD12 | 16位并行数据线 |
| 4 | BIDIR | SD10 | 16位并行数据线 |
| 5 | BIDIR | SD8 | 16位并行数据线 |
| 6 | BIDIR | SD6 | 16位并行数据线 |
| 7 | BIDIR | SD4 | 16位并行数据线 |
| 8 | BIDIR | SD2 | 16位并行数据线 |
| 9 | BIDIR | SD0 | 16位并行数据线 |
| 10 | DOWN | PCI\_R/W# | RAM读写线 |
| 11 | DOWN | SA11 | 12位并行地址线 |
| 12 | DOWN | SA9 | 12位并行地址线 |
| 13 | DOWN | SA7 | 12位并行地址线 |
| 14 | DOWN | SA5 | 12位并行地址线 |
| 15 | DOWN | SA3 | 12位并行地址线 |
| 16 | DOWN | SA1 | 12位并行地址线 |
| 17 | DOWN | DSP\_INT\_DOWN | 下行直通线（保留） |
| 18 | UP | RAM\_BUSY | RAM双口忙线 |
| 19 |  | GND | 地线 |
| 20 | BIDIR | SD15 | 16位并行数据线 |
| 21 | BIDIR | SD13 | 16位并行数据线 |
| 22 | BIDIR | SD11 | 16位并行数据线 |
| 23 | BIDIR | SD9 | 16位并行数据线 |
| 24 | BIDIR | SD7 | 16位并行数据线 |
| 25 | BIDIR | SD5 | 16位并行数据线 |
| 26 | BIDIR | SD3 | 16位并行数据线 |
| 27 | BIDIR | SD1 | 16位并行数据线 |
| 28 | DOWN | PCI\_OE# | RAM输出使能线 |
| 29 | DOWN | PCI\_CE# | RAM片选线 |
| 30 | DOWN | SA10 | 12位并行地址线 |
| 31 | DOWN | SA8 | 12位并行地址线 |
| 32 | DOWN | SA6 | 12位并行地址线 |
| 33 | DOWN | SA4 | 12位并行地址线 |
| 34 | DOWN | SA2 | 12位并行地址线 |
| 35 | DOWN | SA0 | 12位并行地址线 |
| 36 | UP | RAM\_INT | RAM上传中断线 |
| 37 | UP | PCI\_INT\_UP | DSP紧急中断线 |



图 2 后端并行接口示意图

### 普通数据通道（GDP）

普通数据通道为下载指令和上传反馈提供传输通道，是正常生产数据传输的管道。数据通过双口RAM的上传缓存区和下载缓存区进行数据交互，利用双口RAM的中断功能进行控制交互。传输管道通过定时器轮询机制实现上传下载的时序状态控制。

### 紧急错误通道（E2P）

紧急错误通道为后端机电设备故障后紧急发送错误信息的特殊通道，由下位机直接产生中断信号，绕过双口RAM触发PCI系统中断，错误信息被传输管道的中断线程直接递送到用户界面，确保信息处理的及时性。

### 设备空间和传输地址映射

后端设备提供给用户有2个64K地址空间可用，传输管道仅使用了其中的一个64K空间。设备空间提供了后端RAM的访问和内部基本寄存器的状态查询。

FOMES基于内存交换，后端设备的地址空间将映射到系统内存空间，由于平台差异问题，不同平台操作统一存储单元的地址不同，因此需要实现地址映射，系统应用层和数据链路层访问硬件空间的方式不同。基本地址映射如表 6所示。系统全地址映射请参阅*附录：FOMES传输管道地址全映射总表*。

表 6 FOMES传输管道地址基本映射

|  |  |
| --- | --- |
| 地址范围 | 目标映射 |
| 0x0000 – 0x1FFF | 后端双口RAM空间 |
| 0x2000 – 0x2003 | 中断查询寄存器 |
| 0x2004 – 0xFFFF | 保留 |

## 应用层接口

传输管道以动态链接库的形式为系统应用层提供接口。传输管道与上层系统应用之间仅通过五种方式进行交互。尽可能简化系统接口。

### 传输管道建立

传输管道的建立依靠应用程序对DLL的调用。一旦应用程序开始对DLL的调用，传输管道即建立，完成设备、数据结构、映射表等的初始化，开始管道轮询。

### 下发指令

应用程序下发指令时，通过SetToBuffer指令，向下发指令队列中写入指令。SetToBuffer函数原型为：

*ProcErr SetToBuffer( (unsigned char \*)&m\_ControlComd,*

*int len,*

*int\*ComdID);*

其中，各参数含义如下：

*m\_ControlComd* 下载指令的缓存首地址

*len* 下载指令的缓存长度（字节计数）

*ComdID* 指向数据包指令ID的指针

### 获取反馈信息

应用程序读取反馈信息时，通过GetInfrBuffer指令，向传输管道查询反馈信息。GetInfrBuffer函数原型为：

*ProcErr GetInfoBuffer(int ComdID,*

*(unsigned char \*)&m\_FeedBackInfo,*

*int len,*

*int Waittime);*

其中，各参数含义如下：

*m\_FeedBackInfo* 上传反馈的缓存首地址

*len* 上传反馈的缓存大小（字节计数）

*ComdID* 数据包的指令ID

*Waittime* 系统应用允许的返回时间

### 传输管道关闭

应用程序关闭时，向传输管道发送关闭消息，传输管道在执行完当前的最小动作之后，安全关闭。最小动作指程序、设备安全执行的一组最小动作单元，指定最小动作能够保证管道的正常关闭，不必引起可能的硬件操作故障。

### 意外处理

当传输管道发现不能独自处理的意外情形时，会向主程序发送消息，主程序将根据意外情形执行不同的动作。这些意外情形包括但不仅限于底层硬件故障，传输管道抛出异常，上位机指令发送出错等。

# 基本通信机制

## 普通数据传输机制



图 3 传输管道运行流程

应用程序调用传输管道动态链接库后，传输管道完成系统设备及系统变量、数据结构的初始化，随即启动查询定时器。定时器定时到期后首先查询RAM中断标志寄存器（设备空间0x2000），确定下位后端设备是否有数据传输请求。若下位设备有数据请求，则完成本次数据传输；若下位系统处于空闲状态，则传输管道查询指令向量是否有指令堆积，若有指令堆积，则发起新一轮数据传输。完成一次数据传输后，传输管道将反馈信息填入反馈信息队列。

应用程序通过接口函数SetToBuffer向管道指令队列中填入指令，通过接口函数GetInfoBuffer从管道信息队列中读取反馈信息。

传输管道独立处理一切与机电系统运行状态无关、与上位机故障无关的传输错误和问题。整个信息处理流程如图 3所示。

## 突发状况反应机制

当下层机电设备发生非正常运行状况时，或指令执行失败时，会通过E2P通道向系统应用程序直接发送错误信息。该机制基于PCI系统中断实现。PCI中断触发后，传输管道会在第一时间寻访双口RAM中的错误信息（0x1800 – 0x1FF8），并及时将错误信息发送给系统应用。

1. 相关术语
   1. FOMES

Fast Organized Memory Exchange Specification

* 1. 传输管道

根据FOMES搭建的软件系统和硬件平台，应用于表面贴装机，完成前端设备和后端设备的信息交互，是FOMES的一个实现。

* 1. 物理层

传输管道的物理介质实现；链路层完成了物理介质的信息结构组织；传输层完成了传输资源调度和协议机制实现。传输层向系统应用层提供动态链接库形式的接口调用

* 1. 链路层

FOMES架构的第二层，完成基于物理层的信息结构组织，包括存储系统组织、关键数据结构组织和关键信息字定义

* 1. 传输层

FOMES架构的第三层，完成协议的信息交换组织，实现信息交互的动态控制和状态转换，向系统应用层提供FOMES前端接口

* 1. 应用层

工作在上位机的系统应用程序及其以上的所有软件功能架构

* 1. 上位机

承载系统应用层的硬件介质，完成用户信息交互和动态生产过程控制

* 1. 下位机

总控底层机电设备运行的控制器，完成机电设备各部分的工作协调

* 1. 前端设备

系统应用层及其以上的所有系统架构

* 1. 后端设备

下位机主主板及其以后的所有机电设备架构

* 1. 前端接口

前端设备和传输管道的接口

* 1. 后端接口

后端设备和传输管道的接口

* 1. 地址映射

系统操作地址到底层硬件操作地址的空间转换，亦即自底层自然地址到从系统看过去的地址的转换

* 1. 指令向量表

位于传输层，存放堆积指令数据包的索引信息的数据结构

* 1. 反馈向量表

位于传输层，存放堆积反馈数据包的索引信息的数据结构

* 1. 指令（包）

对应系统指令输入结构体的实例缓存

* 1. 反馈（包）

对应系统指令输出结构体的实力缓存

* 1. 下载

指令数据包从应用层发送到底层硬件设备的过程

* 1. 上传

反馈数据包从底层硬件设备传送到应用层的过程

* 1. 数据包

数据传输中全体数据缓存的总称，包括指令包、反馈包和紧急错误包

1. FOMES传输管道地址空间全映射总表

|  |  |  |
| --- | --- | --- |
| 段地址（字节地址） | 单元名称 | 单元说明 |
| 0x0000 – 0x0FF7 | 下载信息区 | 系统下发的指令包缓存区 |
| 0x0FF8 – 0x0FFF | 备注信息区 | 指令包的备注信息 |
| 0x1000 – 0x17F7 | 上传信息区 | 待上传至系统的反馈信息 |
| 0x17F8 – 0x17FF | 备注信息区 | 反馈信息的备注信息 |
| 0x1800 – 0x1FFB | 错误信息缓存 | 存放错误信息的缓存 |
| 0x1FFC – 0x1FFF | 上传控制字 | 数据上传中断控制 |
| 下载控制字 | 数据下载中断控制 |
| 0x2000 – 0x2003 | 中断查询寄存器 | 标识后端是否请求 |
| 0x2004 – 0xFFFF | 保留 | 未定义 |