

EtherCAT 过程数据传输的实现与时延分析

王培盛, 林 威, 刘 辉, 张钦宇

(哈尔滨工业大学 深圳研究生院, 广东 深圳 518055)

摘要: 针对基于 PC 机的工业以太网控制系统成本较高, 且 Linux、Windows 等非实时操作系统的响应实时性不佳的问题, 设计了基于 $\mu\text{c}/\text{os II}$ 的嵌入式 EtherCAT 工业以太网控制系统; 首先, 阐述了 EtherCAT 技术的基本原理, 然后在设计好主站系统基本框架的基础上, 给出了一个短延时的过程数据收发模型并对其进行时延分析; 最后, 在硬件系统上测量了过程数据的传输延时, 以及各个过程对总体延时的影响; 实验结果表明, 系统在传输 10 个字节过程数据量的情况下传输延时约为 $154 \mu\text{s}$, 该传输速率可以满足大部分的工业实时监控要求。

关键词: EtherCAT; 工业以太网; 过程数据; 实时性;

Realization and Delay Analysis of EtherCAT Process Data Transmission

Wang Peisheng, Lin Wei, Liu Hui, Zhang Qinyu

(Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China)

Abstract: Aiming at the problems of PC-based industrial Ethernet control system cost higher, and the bad real-time performance of some non-real-time operating system such as Linux and Windows, here design a EtherCAT industrial control system based on $\mu\text{c}/\text{os II}$. the paper introduces the basic principle of EtherCAT and then designs the framework of the master system. After that, provide a short delay model of the process data transmission. Last, we test the transmission delay and the effect of each process on the overall delay on the system based on $\mu\text{c}/\text{os II}$. The experimental results indicate that this system can realize transmitting 10 bytes of process data within $154 \mu\text{s}$ and this provide a feasible solution for industrial real-time monitoring.

Keywords: EtherCAT; industry Ethernet; process data; real-time performance

0 引言

近年来, 随着网络技术的迅速发展, 工业现场总线技术逐渐向实时工业以太网技术过渡, 高性能工业控制系统的研究向网络化和高速化方向发展。如何满足并适应高性能运动控制系统对数据传输的实时性要求, 已成为现代工业自动化控制发展的方向以及各种现场总线技术竞争的焦点。针对这种情况, 德国 BECKHOFF 自动化公司开发了一种实时以太网技术 EtherCAT (ethernet for control automation technology)。EtherCAT 基于标准的以太网技术, 可以采用灵活的网络拓扑结构进行组网。它实现简单并且数据传输速率高, 可以在 $30 \mu\text{s}$ 内处理 1 000 数字量 I/O^[5]。实际系统中过程数据的循环时间是指应用层开始发送数据到应用层接收数据完毕的时间, 因而主站的数据收发处理流程对循环时间有很大的影响。然而当前大部分 EtherCAT 主站的设计是基于 PC 机的, 采用的是 windows、Linux 等非实时操作系统, 这种设计的主站设备成本高。本文提出了一种基于嵌入式硬件平台的低成本主站设计方案, 采用实时操作系统 $\mu\text{c}/\text{OS II}$ 来进行 EtherCAT 主站设计, 并对数据收发流程进行优化, 实现了高实时性的 EtherCAT 过程数据传输系统。

1 EtherCAT 技术原理

1.1 网络结构

EtherCAT 系统采用主从式结构, 主站使用标准的以太网接口卡, 从站使用专门的从站控制器 ESC (EtherCAT Slave Controller)。EtherCAT 网段内从站设备在逻辑上布置成一个开口的环形总线, 网络拓扑结构可以是线型、树形或者星形等结构^[5]。数据帧从主站出发, 逐个访问环路上的从站。由于从站控制器 ESC 专用芯片已经集成了数据链路层功能, 当数据帧到达从站时, 从站接口芯片就会从数据帧中抽取数据或者将数据插入到数据帧中, 然后再把整个以太网帧转发到下一个从站, 直至数据帧到达最后一个从站, 数据帧才被返回主站处理。在此过程中从站无需对数据帧进行编解码处理, 使得数据帧的传输时间大大缩短。

主站与从站的连接方式有两种: 单网卡模式和双网卡模式。在单网卡模式下, 数据帧从主站出发, 逐个访问环路上的从站, 到达最后一个从站后数据帧按原路返回主站。在双网卡模式下, 数据帧从主站的第一个网卡出发, 逐个访问环路上的从站, 最后一个从站处理完数据后直接发送给主站的第二个网卡。第二个网卡接收到数据后会返回一个伪数据帧, 该数据帧按原路返回给第一个网卡。由于网卡 2 发送的数据在经过各个从站的时候只转发不处理, 于是在发送周期性过程数据的时候可以省去这个步骤, 从而减小数据收发的延时。一个简单的环状双网卡结构如图 1 所示。

1.2 EtherCAT 数据帧结构

EtherCAT 协议数据直接使用标准的 IEEE802.3 以太网数据帧传输, 其以太网帧类型使用 $0\text{x}88\text{A}4$, 因此很容易实现

收稿日期: 2014-03-21; 修回日期: 2014-04-17。

基金项目: 国家重大科技专项(2012ZX03001027-002)。

作者简介: 王培盛(1988-), 男, 广东省普宁市人, 硕士研究生, 主要从事网络技术方向的研究。

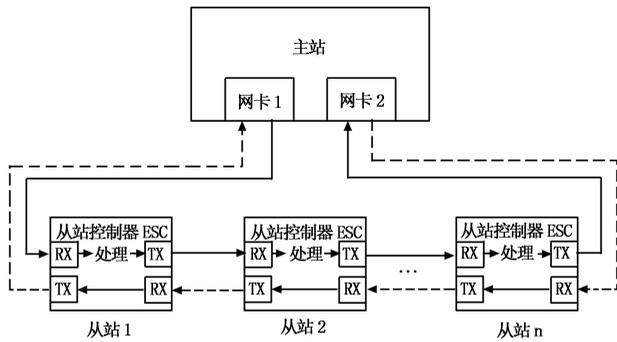


图 1 环形拓扑结构

EtherCAT 协议与标准以太网的兼容。EtherCAT 数据帧包括 14 字节的以太网帧头、2 字节的 EtherCAT 数据头、12~1 498 字节的数据区和 4 字节的帧校验序列 FCS。数据区包含一个或多个 EtherCAT 子报文，每个子报文对应独立的从站设备或者存储区域。子报文包含子报文头、数据域和工作计数器 WKC (Working Counter)。WKC 用于记录子报文被从站处理的情况。当从站处理完某个子报文之后，该子报文的 WKC 会在原来基础上增加一个增量，主站通过比较预期 WKC 和返回子报文的 WKC 来判断子报文是否被正确处理。对于不同的操作方式，WKC 的增量不同。

1.3 逻辑寻址和 FMMU

EtherCAT 设备间的通信通过主站发送数据帧读写从站设备存储区来实现，EtherCAT 报文寻址方式有多种，其中主要包括设备寻址和逻辑寻址。设备寻址通过 16 位的从站设备地址寻址到特定的设备，进而对设备进行读写操作。逻辑寻址主要用于过程数据，它可以进行多播，一个报文能够对多个从站进行操作。逻辑寻址时，将报文内的 32 位地址作为数据的逻辑起始地址，报文可以寻址到的逻辑地址空间为 4GB。逻辑寻址通过现场总线内存管理单元 (FMMU; Fieldbus Memory Management Unit) 来实现，每个 ESC 内部都有 FMMU 控制单元，它将从站的物理内存地址映射到网段内的逻辑地址空间。FMMU 减少了系统对数据进行分类而造成的开销，操作一块逻辑地址空间可以访问相应的多个从站，从而提高了系统的性能。

2 EtherCAT 主站设计

在本设计中，EtherCAT 主站的硬件系统主要由三星公司的 S3C6410 处理器和通用网络接口卡 DM9000 组成。操作系统采用实时操作系统 UC/OS II，并在 UC/OS II 系统下，利用 DM9000 网卡驱动和系统实时定时器向上提供的接口，实现了 EtherCAT 主站协议软件。实时内核 UC/OS II 为 EtherCAT 主站协议软件提供了任务调度、时间管理、信号量等接口。网卡驱动向 EtherCAT 主站软件提供网卡操作接口，使主站可以通过网卡实现数据帧的发送和接收。EtherCAT 主站设计架构如图 2 (左) 所示。

主站软件架构包括 UC/OS II 系统初始化、网卡初始化和 EtherCAT 协议架构。其软件实现主要包括以下流程：

(1) $\mu\text{c/os II}$ 系统初始化：该过程主要实现系统的初始化，并创建 EtherCAT 协议运行所需的任务。

(2) 网卡初始化：配置网卡的相关寄存器，并创建一个网卡接收任务，使网卡能够及时处理接收数据。

(3) 初始化各从站：对各个从站的相关寄存器进行初始化，通过读取从站的配置文件，进而配置从站的设备信息和同步管理等，并控制从站的状态切换。

(4) I/O 地址映射：进行输入输出过程数据地址映射，配置各个从站的 FMMU 单元。

(5) 启动过程数据传输：启动过程数据通信，实现主站与从站之间的实时数据传输。

主站软件设计流程如图 2 (右) 所示。

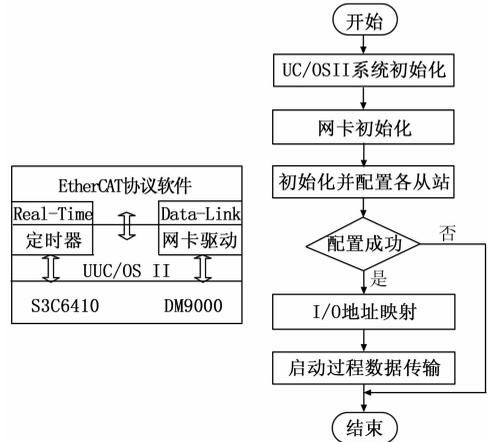


图 2 EtherCAT 主站系统架构 (左) 及软件流程 (右)

3 周期性过程数据传输

3.1 过程数据驱动模型

网卡驱动接收数据包的方式主要有两种：轮询式接收和中断接收。轮询接收方式即驱动程序每隔一段时间检查网卡是否接收到数据，如果是则从网卡接收 SRAM 中读取数据；中断接收方式是利用网卡收到网络数据后会产生硬件中断信号，CPU 可以相应中断信号，调用中断服务程序将数据从网卡接收 SRAM 中取走。由于中断接收方式实时性比较好，因此本设计采用中断接收的方式。

DM9000 驱动模型如图 3 所示。该模型分为 3 个部分：网络层、驱动层和物理层。网络层主要提供数据的打包与检测功能，并调用网卡的接口进行数据收发；驱动模块实现数据的收发操作，它可以将上层传递进来的数据写到网卡进行发送，也可以响应网卡的接收中断，读取网卡接收到的数据并通知上层进行后续操作；物理层则实现实际的数据传输，是主站与从站的通信媒介。下面将分析过程数据发送与接收的具体流程。

3.2 过程数据收发流程

3.2.1 发送流程

周期开始的时候，应用层调用过程数据发送函数请求发送过程数据。发送的具体操作如下：

(1) 获取输入输出过程数据，计算其长度。

(2) 获取数据帧的索引号，并将数据封装成 EtherCAT 数据帧；

(3) 等待网卡互斥信号量可用，如果没有其它任务占用网

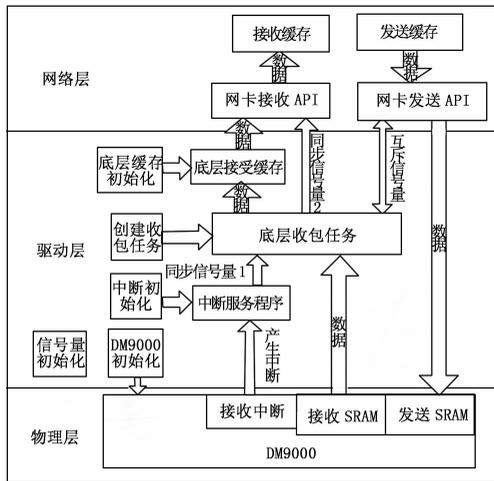


图 3 DM9000 驱动模型

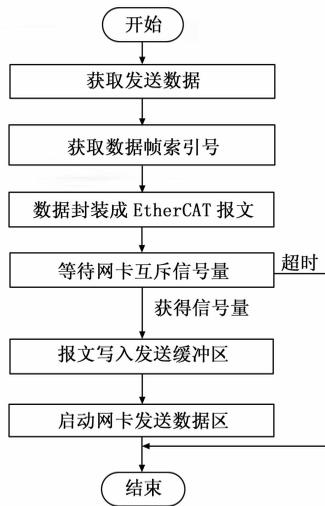


图 4 过程数据发送流程

卡设备, 则执行下一步;

(4) 将数据帧写到发送数据缓存中, 然后启动网卡发送数据。

由于网卡是独占设备, 不能同时被两个进程调用, 因此需要设置一个信号量来进行网卡的互斥访问, 进程只有在获得网卡互斥信号量之后才能对网卡操作。过程数据的具体发送流程如图 4 所示。

3.2.2 接收流程

接收过程可以分为两个部分, 底层接收任务和上层接收任务。

底层接收任务 MACPacket () 是在系统初始化过程中创建的, 该进程优先级比较高。MACPacket () 函数主体是一个循环体。它主要实现以下功能:

- (1) 等待同步信号量 hEMACInput, 当同步信号量可用时执行 (2);
- (2) 调用底层收包函数, 该函数等待网卡互斥信号量可用, 如果没有其它任务占用网卡设备, 则执行 (3);

- (3) 将网卡接收到的数据拷贝到底层接收缓存中;
- (4) 释放网卡互斥信号量和数据同步信号量 EthernetInput, 通知上层进行后续的工作。

当网卡接收到数据的时候, 网卡会产生中断信号以通知处理器, 处理器接收到中断信号后调用中断服务程序从而发出同步信号量 hEMACInput, 底层接收任务获取信号量后就可以进行后续工作了。其具体操作流程如图 5 (左) 所示。

上层接收任务是在网卡发送数据函数退出之后调用的, 其主要的操作流程如下:

- (1) 等待数据同步信号量 EthernetInput。底层收包任务接收到数据后会释放信号量 EthernetInput, 该任务接收到信号量之后可以进入第二步继续操作;
- (2) 将底层接收缓存中的数据提取到临时缓存中;
- (3) 对数据帧进行分析, 如果数据帧正确, 则执行第四步, 否则进行出错处理;
- (4) 将数据拷贝到过程数据缓冲中, 至此整个接收流程结束。

上层接收任务的具体操作流程如图 5 (右) 所示。

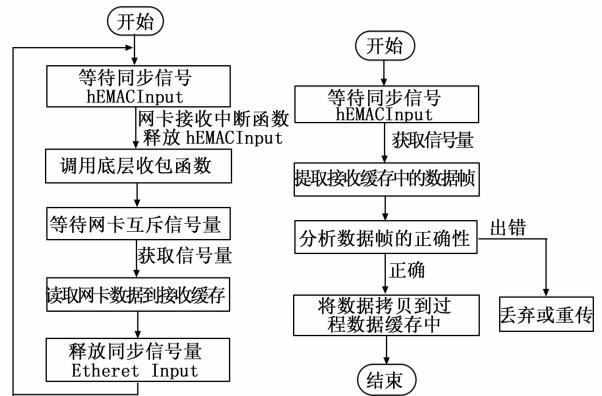


图 5 接收任务流程

3.2.3 延时分析

根据以上的传输模型可以得出如图 6 所示的延时模型。其中各个时间表示的含义如下。

- t_s : 表示从用户层调用发送函数到启动网卡发送数据的处理时间;
- t_b : 表示启动网卡发送到退出发送函数的处理时间;
- t_{r1} : 表示用户层调用上层接收函数到开始等待数据同步信号量的时间;
- t_{r2} : 表示获得数据同步信号量到数据处理完毕的时间;
- t_w : 表示在网卡中断信号发生之前等待数据同步信号量的时间;
- t_d : 表示从网卡接收中断发生到底层接收任务处理数据完毕的时间;
- t_p : 表示网卡发送到网卡接收中断发生的时间, 也就是物理层的处理时间。

过程数据传输总的延时为:

$$T = t_s + t_b + t_{r1} + t_{r2} + t_w + t_d$$

其中: t_s 、 t_{r2} 、 t_d 涉及到数据的操作与搬运, 与操作流程

及数据长度有关，由于操作流程是固定的，因此其大小只随着数据长度的增加而增大。另外， t_w 与 t_p 及 t_b 、 t_{r1} 有关， t_p 与 t_b 、 t_{r1} 在时间上是并行的，当 $t_p \leq t_b + t_{r1}$ 时， $t_w = 0$ ；当 $t_p > t_b + t_{r1}$ 时， $t_w = t_p - (t_b + t_{r1})$ 。因此，当 $t_p \leq t_b + t_{r1}$ ，总的延时与物理层的传输延时无关，而只跟数据的处理过程及数据长度有关；当 $t_p > t_b + t_{r1}$ 时，总的延时与物理层的传输延时、数据的处理过程及数据帧长度有关。另外，物理层的传输延时与数据帧的长度和从站的个数等有关。

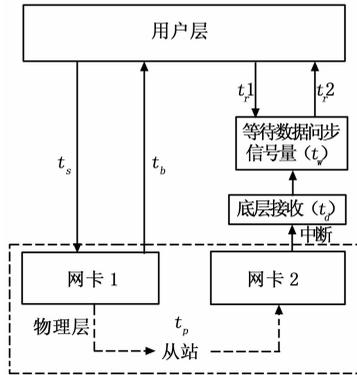


图 6 传输延时模型

4 系统延时测试

测试平台由主站和从站两部分组成。主站系统由 S3C6410 处理器和 DM9000 以太网卡组成；从站系统由 stm32f103 处理器及 BECKHOFF 公司开发的从站控制器 ET1100 组成。

对过程数据传输延时有影响的因素主要有 3 个：主站数据收发的处理流程、物理层的传输以及数据帧的长度。为了测试系统的实时性能，本文测量了过程数据传输延时，以及各个过程的延时。并测试不同从站个数以及在一个从站情况下不同过程数据量对延时的影响。延时的测试采用软件测试的方法。主站底层提供了接口 gettimeofday()，该接口可以获取精度为微妙级的系统时间。对于需要测试时延的过程，可以利用该接口在进入和退出点进行打点，两次的时差减去调用时间接口的时间即为该过程的延时。

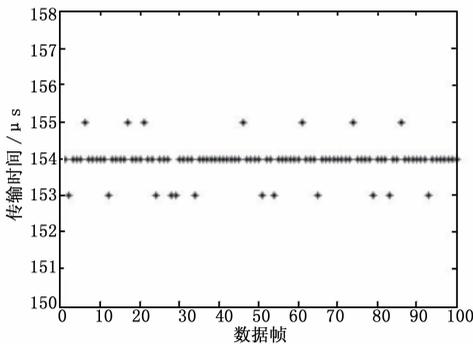


图 7 数据帧传输时间波动情况

表 1 记录了在不同从站个数情况下各个点之间的延时，其中每个从站只定义了 4 个字节的输入过程数据和 6 个字节的输出过程数据，即每个从站有 10 个字节的过数据量。从表中可以看出，在一个从站的情况下，循环周期为 154 μs，其中大部分时间集中在主站的处理过程上，而物理层上的处理时间大概为 6 μs。同时可以看出，每增加一个从站，循环周期延时大概增加 3 μs。表 2 记录了在一个从站情况下不同数据过程量的延时。其中保持了 6 字节的输出过程数据，改变的只是输入过程数据的长度。从表中可以看出，循环周期随着过程数据量的增加基本上呈现线性增大。图 7 表示的是在一个从站 10 个字节过程数据量情况下连续 100 个数据帧的传输时间，可以看出平均传输时间大约为 154 μs，在传输的过程中处理时间会偶尔上下波动 1 μs。

表 1 不同从站个数的延时情况

从站数	应用层发送至网卡发送	网卡发送至网卡接收中断	网卡接收中断至应用层接收完毕	总的延时/μs
1	42	6	106	154
2	44	7	107	158
3	46	7	108	161

表 2 一个从站不同数据量的延时情况

输入过程数据量/byte	4	8	12	16	20	24	28	32
延时/μs	154	155	156	158	160	161	163	166

5 结论

本文在 S3C6410 处理器平台上，采用实时操作系统 μc/os II 来进行 EtherCAT 主站的计，并对数据收发流程进行优化，实现了实时性良好的 EtherCAT 过程数据传输系统。经过测试，过程数据从应用层开始发送到应用层接收到数据平均延时可以达到 154 μs，该延时符合实时工业以太网的性能要求，对现代工业现场控制的实时性数据传输有很好的借鉴作用。

参考文献：

- [1] 王磊, 李木国, 王静, 等. 基于 EtherCAT 协议现场级实时以太网控制系统研究 [J]. 计算机工程与设计. 2011, 32 (7): 2294 - 2297.
- [2] 阮倩茹, 王辉, 施大发, 等. 基于 EtherCAT 的高性能交流伺服控制系统设计 [J]. 科技导报, 2010, 28 (20): 58 - 61.
- [3] 单春荣, 刘艳强, 郇极. 工业以太网现场总线 EtherCAT 及驱动程序程序设计 [J]. 制造业自动化, 2007, 29 (11): 79 - 82.
- [4] 党安喜, 裴少婧, 尚耀东, 等. 以太网时延仿真与性能分析 [J]. 计算机工程与应用, 2009, 45 (2): 119 - 121.
- [5] 德国倍福公司. 实时以太网: I/O 层超高速以太网 [J]. 自动化博览, 2004, 21 (4): 48 - 50.
- [6] 赵伯阳, 杨建武. 一种简单有效的实时以太网研发 [J]. 计算机测量与控制, 2012, 20 (11): 3021 - 3024.