

## 摘要

工业以太网是当前工业控制现场总线技术的一个重要发展方向，与使用传统技术的现场总线相比，以太网现场总线具有传输速度快，数据容量大，传输距离长，性能较好的优点。EtherCAT 作为实时工业以太网技术中的一种典型技术，2003 年由德国 BECKHOFF 自动化公司提出的。它充分利用了以太网的全双工特性，使用主从模式介质访问控制，主站发送以太网帧给各从站，从站从数据帧中抽取数据或者将数据插入数据帧。其从站节点使用专用的控制芯片，主站使用标准的以太网控制器。EtherCAT 具有适用广泛，无需从属子网，支持多种设备连接拓扑结构，高效率，刷新周期短，同步性能好等特点，并且完全符合以太网标准。整个 EtherCAT 网络化控制系统在现代工业生产中具有相当的通用性，只是针对不同的应用层设备而存在不同。EtherCAT 网络化控制技术在很大程度上提高了现场总线的性能。

本文主要研究 EtherCAT 网络化控制系统中 EtherCAT 从站微处理器对主站与从站数据交换的实现过程。而系统中所用到的数据通信协议为 CANopen 协议。课题首先对整个 EtherCAT 网络化控制系统的系统结构做了介绍，然后针对 EtherCAT 从站设计出了从站微处理芯片 LPC1754 的硬件电路，并且完成了所设计芯片的 PCB 设计以及芯片元器件焊接工作。接着对 EtherCAT 网络化控制系统中的通信协议 CANopen 协议做了认真地学习与研究，根据协议中的规定设计了 EtherCAT 从站微处理器芯片 LPC1754 的驱动程序。最后结合主站控制器对所设计的驱动程序进行了 PDO 和 SDO 数据通信的测试，并且在自动化软件 TwinCAT 平台上利用所设计的芯片 LPC1754 上的数模转换模块对所设计的微处理器驱动程序进行了测试，对所得到的测试结果进行了分析。

从最终测试结果可以发现，所设计的从站驱动程序可以在设计的从站微处理器芯片 LPC1754 上正确运行，所设计的从站驱动程序可以正确地完成对 CoE 格式的数据帧的封装与解析工作，能够准确地实现主站与从站之间的数据交换功能。所设计的 LPC1754 芯片能够实现整个 EtherCAT 网络化控制系统中从站微处理器的功能。整个模块工作稳定可靠，能够满足基本的工业化网络控制系统需求，能够投入到现代化工业生产与相关制造业中。

关键字： EtherCAT， CANopen， 对象字典， PDO， SDO

## Abstract

Industrial Ethernet is an important development direction in current industrial control of fieldbus technology. Compared to the traditional techniques of fieldbus, Ethernet fieldbus has characteristics of transmission speedily, packets capacity, long distance transmission and good performance. As a typical technology of real-time industrial Ethernet technology, put forward by Germany BECKHOFF automation company in 2003. EtherCAT made full use of the Ethernet full-duplex characteristics, use master-slave media access control model, master send the Ethernet frame to slave, slave find data from the Ethernet frame or insert data to the Ethernet frame. Slave use specialized control chip, master use the standard Ethernet controller. EtherCAT has characteristics of widely applicable, with out affiliate subnet, support a variety of equipment connection topological structure, high efficiency, short refresh cycle, good synchronization performance. And full compliance with Ethernet standard. The EtherCAT networked control system in modern industry is universal. Just different in different application layer equipment. EtherCAT networked control system greatly improve the performance of the field bus.

This paper mainly research the realization process of data exchange between master and slave by microprocessor in EtherCAT networked control system. The data communication protocol of this system is CANopen. First we introduced the system structure, then we designed the hardware of slave microprocessor, PCB design and chip welding. And then we learn and research the CANopen protocol. According to the CANopen protocol, we designed the driver of microprocessor LPC1754. At last, we test our driver by PDO and SDO communication with the master. and test our driver with the AD converter of LPC1754 by automation software TwinCAT and analyzed the test results.

We can find that from the final results, the driver we designed could run well in the LPC1754 we designed, and could complete the data package and data analytical of CoE frame, could realize accurately the data exchange of master and slave. The LPC1754 we designed could do the work of EtherCAT slave microprocessor. The

module is stable and reliable, and can meet the basic needs of networked control systems, to contribute to the modern industrial production and related industry.

**Key words:** EtherCAT CANopen Object Dictionary PDO SDO

# 第 1 章 绪 论

## 1.1 研究背景

计算机信息技术、现代控制技术和网络技术的迅猛发展正深刻地改变着人们的生活、工作方式，尤其对于工业控制领域，对工厂自动化和企业信息化领域的发展产生了巨大的影响。与此同时，计算机信息技术的发展带动了嵌入式系统应用研究的快速发展。面对当前工业不断走向自动化、智能化、网络化的阶段，将所有高可靠性的局部嵌入式系统集成成为整个网络化控制系统的要求日益显著<sup>[1]</sup>。

当今世界工业生产中用到的最广泛的控制技术是现场总线技术，它作为工程自动化、制造自动化、楼宇和交通等领域现场只能之间的互联通讯网络，沟通了生产过程现场控制设备之间及其与更高控制管理网络之间的联系。现场总线控制系统 FCS (Fieldbus Control System) 为网络集成式全分布控制系统，其特点是控制功能由过去的控制室设备变为智能现场仪表来承担，可以组成大型的开放式控制系统，从而实现从最高决策到最低设备层的管理和控制<sup>[2]</sup>。

网络控制系统 NCS (Networked Control System) 是最近几年新提出的概念，该类控制系统中，被控制对象与控制器以及控制器与驱动器之间是通过一个公共的网络平台连接的。这种网络化的控制模式具有信息资源能够共享、连接线数大大减少、易于扩展、易于维护、高效率、高可靠性及灵活性等优点，是未来控制系统的发展模式<sup>[3]</sup>。

目前，基于以太网的工业网络控制系统是使用最广泛的网络控制系统。采用工业以太网技术，不同的网络控制系统都有相同的通讯协议，能实现办公自动化网络和工业控制网络的无缝连接。另外，嵌入式实时操作系统和嵌入式平台的快速发展，嵌入式控制器、智能现场测控仪表都可以很方便地为整个网络控制系统服务，直至与 Internet 相连。近几年来，以太网技术已开始广泛应用于工业控制领域，已在工业控制的各个领域得到应用，是实现数字化、网络化制造的关键技术。作为工业以太网中一种最具有代表性的技术 EtherCAT (Ethernet for Control Automation Technology) 具有很广的应用领域以及极强的

竞争力。

## 1.2 研究现状

### 1.2.1 EtherCAT 研究现状

EtherCAT 是由德国 BECKHOFF 自动化公司于 2003 年提出的实时工业以太网技术。它具有高速和高数据有效率的特点,支持多种设备连接拓扑结构。其从站节点使用专门的控制芯片,主站使用标准的以太网控制器。

EtherCAT 的主要特点如下<sup>[4]</sup>:

- (1) 广泛适用性,任何带商用以太网控制器的控制单元都可作为 EtherCAT 主站;
- (2) 符合以太网标准, EtherCAT 可以与其他以太网设备及协议并存于同一总线;
- (3) 无须从属子网,复杂节点或只有 2 位的 I/O 节点都可以作为 EtherCAT 从站;
- (4) 高效率,最大化利用以太网带宽进行用户数据传输;
- (5) 刷新周期短,可以达到小于 100  $\mu$  S 的数据刷新周期,可以用于伺服技术中底层的闭环控制。

EtherCAT 系统配置简单,具有高速、高有效数据率等特点,其有效数据利用率达到 90% 以上,全双工特性得以完全利用<sup>[5]</sup>。2005 年 2 月, EtherCAT 通过 96% IEC 会员投票表决,正式成为 IEC 规范——IEC/PAS62407,国际标准组织 (ISO) 已将 EtherCAT 纳入 ISO15745 标准。EtherCAT 技术引起了自动化技术领域的广泛关注,并于 2003 年成立了 EtherCAT 技术组织,简称 ETG<sup>[6]</sup>。

随着以太网技术在实时性、通信效率和总线供电等方面的障碍的突破,本质安全也完成开发工作,以太网技术在工业上的应用将继续向下延伸。而以太网技术最近的一个新发展方向是向机箱级的背板总线延伸。而 EtherCAT 正是利用这一技术开发了用于现场控制柜的 E-bus, I/O 机箱的第一个模块使用总线耦合器,该耦合器将标准的双绞线或光缆电气信号转换为 E-bus 信号, I/O 模块之间信息通过 E-bus 传输。这样一来,以太网帧可以不受影响地传送到 I/O 输入的端口,从某种意义上讲,以太网已经延伸到现场设备级<sup>[7]</sup>。

## 1.2.2 CANopen 研究现状

目前,现场总线在欧美一些工业发达的国家和地区发展迅速,并形成广阔的市场应用。如今国外现场总线技术已经较大范围地形成了公认的标准,对其的研究及发展正逐渐走向成熟,具有广阔的市场前景。而在国内,从很早就开始了现场总线技术的研究及应用,到现在,已经取得了比较显著的成果,并在实际中得到了应用<sup>[2]</sup>。

CAN (Controller Area Network) 是一种多主方式的串行通信总线,能有效地支持具有很高安全等级的分布实时控制,特别适合于几台智能设备的联网,CAN 总线已被广泛应用到各个自动化控制系统中<sup>[8]</sup>。CANopen 协议是目前在嵌入式网络应用中较为流行的一种高层协议,具有良好的互操作性和兼容性,通过扩展可以应用到广泛的应用领域中。当然,CANopen 并不是唯一被 CiA (CAN-in-Automation) 认可的 CAN 应用层协议。同样运用于 CAN 网络的应用层协议还有 SAEJ1939、DeviceNet、CANkingdom 和 SDS。在欧洲和美国,CANopen 协议已被广泛应用于各种控制场合,但在我国,CAN 的应用层协议主要应用 DeviceNet,因此 CANopen 协议需要在我国进行进一步的推广。当然,由于 CANopen 技术在工业控制方面有极强的适用性和优点,所以国内对 CANopen 的研究也在不断的向前发展中。目前主要应用有:

对 CANopen 协议分析与研究,如对其原理,协议实现等的分析<sup>[9]</sup>;对 CANopen 协议在工业控制中的应用研究,设计 CANopen 从节点和主节点。如对 CANopen 协议在工业控制中适用性的分析<sup>[10]</sup>;作为嵌入式控制设备嵌入到智能装载机和智能摊铺机控制系统中<sup>[11]</sup>;对基于 CANopen 协议构成的网络控制系统进行优化分析的研究。

此外,国内对 CANopen 的研究大部分是应用于 CAN 总线的,然而将 CANopen 应用于实时工业以太网的不多,本课题将研究应用于 EtherCAT 协议下的 CANopen,即 CANopen over EtherCAT (CoE),其实 CoE 早在几年前就有 EtherCAT 的开发商德国 BECKHOFF 公司提出并被广泛支持,在欧洲得到广泛的应用,但在国内的应用很少,故本课题的研究内容比较新颖。

## 1.3 研究内容

本课题针对工业网络化的发展趋势,选择 EtherCAT 这种实时工业以太网作

为研究对象，并分析 EtherCAT 协议下的多种应用层协议，提出基于 EtherCAT 网络的 CANopen 协议网络化控制系统的设计方案，在着重分析 EtherCAT 协议规范和 CANopen 协议原理的基础上，设计开发具有高实时性的 CANopen 从站协议栈，组成具有主从式结构形式的网络控制系统，实现高效的数据通讯和完善的网络管理。

课题的具体研究内容安排如下：

第一章：简要介绍本课题的研究背景与意义，以及对于 EtherCAT 协议和 CANopen 协议的国内外研究现状做了简单的阐述，最后对全文的研究内容做一简单介绍。

第二章：简要分析 EtherCAT 协议的规范，介绍了整个 EtherCAT 网络化控制系统的系统结构，然后针对系统中所用从站微处理器芯片 LPC1754 进行了硬件设计。

第三章：简单介绍 CANopen 通信协议。对 EtherCAT 从站微处理器芯片 LPC1754 的驱动程序进行了设计。

第四章：先对所设计的从站程序进行了通信测试，然后在装有 TwinCAT 软件的 Windows XP 平台上，分析从站调试运行结果，检验设计的控制网络是否达到了设计要求，运行是否稳定、快速、精确。

第五章：结论与展望，对本文所做的工作做了总结，提出工作中的不足之处，并对接下来的工作做了展望。

## 第 2 章 EtherCAT 从站微处理器硬件设计

### 2.1 EtherCAT 协议简介

EtherCAT 是 2003 年由德国 BECKHOFF 自动化公司于提出的实时工业以太网技术。其具有以下优点：高速和高数据效率，支持多种拓扑结构设备之间连接。其从站节点使用专门的从栈控制芯片，主站使用标准的以太网控制器。

#### 2.1.1 EtherCAT 系统结构

EtherCAT 是一种实时工业以太网技术，它充分利用了以太网的全双工特性。其工作原理采用主从模式介质访问控制（MAC），主站将以数据帧发送给各从站，从站从数据帧中抽取输入数据或者将输出数据存入数据帧。主站接口卡使用标准的以太网接口卡。而从站使用专门的 EtherCAT 从站控制器 ESC（EtherCAT Slave Controller）来负责数据帧的处理。物理层使用标准的以太网物理层器件。

##### 2.1.1.1 EtherCAT 主站组成

EtherCAT 主站使用标准的以太网控制芯片，传输介质为 100BASE-TX 规范的 5 类 UTP 线缆，如图 2-1 所示。通信控制器的功能是完成以太网数据链路的介质访问控制（MAC, Media Access Control）工作，实现主站控制芯片与物理层芯片 PHY 之间的数据交互功能。物理层芯片 PHY 用来进行数据帧的编码、译码和收发工作，它们之间通过 MII（Media Independent Interface）接口进行数据交换。MII 定义了与传输介质无关的标准电气和机械接口，使用这个接口将以太网数据链路层和物理层完全分开，使以太网可以方便的选用任何传输介质。隔离变压器可以实现信号的隔离，防止信号之间的相互干扰，从而能更有效的进行通信。

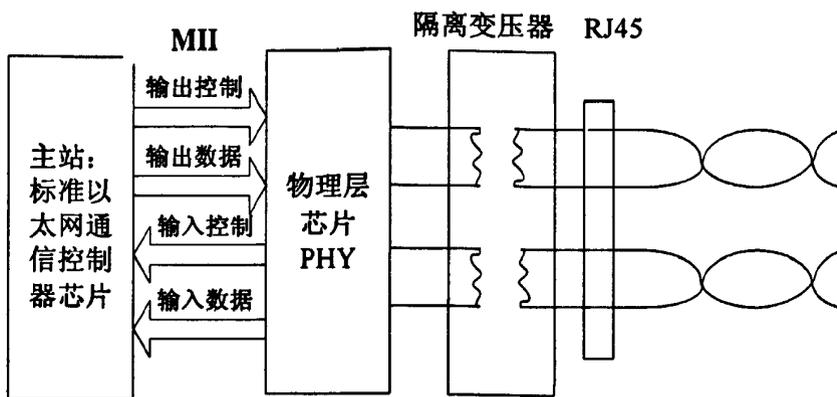


图 2-1 EtherCAT 物理层连接原理图

### 2.1.1.2 EtherCAT 从站组成

EtherCAT 从站设备具有以下 2 个功能：实现通信和控制应用层事件，其结构如图 2-2 所示，由以下几个部分组成<sup>[4]</sup>。

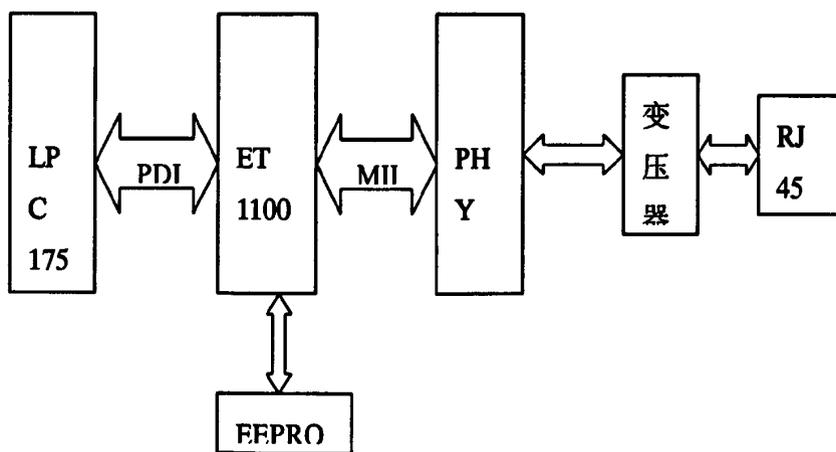


图 2-2 EtherCAT 从站组成结构图

#### (1) EtherCAT 从站控制器 ET1100

EtherCAT 从站控制器芯片 ET1100 (EtherCAT Slave Controller) 的功能是处理 EtherCAT 数据帧，并使用双端口存储区 (SM) 实现 EtherCAT 主站与从站设备应用数据的交换。每个从站 ESC 按照其在环路 (EtherCAT 各个从节点按节点 0, 1, 2, 3... 组成一个环形回路) 上的物理位置顺序移位读写数据帧。报文经过从站时，ESC 从报文中提取发送给自己的输出命令数据并将其存储到内部存储区，输入数据从内部存储区又被写到相应的子报文中。数据的提取与插入

都是由数据链路层硬件完成的。

(2) 从站控制微处理器 LPC1754

微处理器 LPC1754 负责处理 EtherCAT 通信和完成控制任务。微处理器从 ESC 中读取控制数据，通过解析数据实现设备控制功能，并且采样从设备的反馈数据，然后写入 ESC 存储区域中，等待主站主站读取。整个通信过程完全由 ESC 控制。

(3) 物理层器件

从站使用 MII 接口时，需要使用物理层芯片 PHY 和隔离变压器等标准的以太网物理层器件。使用 EBUS 时不需要任何其他芯片。

2.1.2 EtherCAT 数据帧结构

EtherCAT 数据直接使用以太网数据帧传输，数据帧使用帧类型 0x88A4。EtherCAT 数据包括两个字节的帧头和 44~1498 字节的数据。数据区有一个或者多个 EtherCAT 子报文组成，每个子报文对应独立的设备和从站存储区域，如图 2-3 所示。表 2-1 给出了 EtherCAT 数据帧定义。

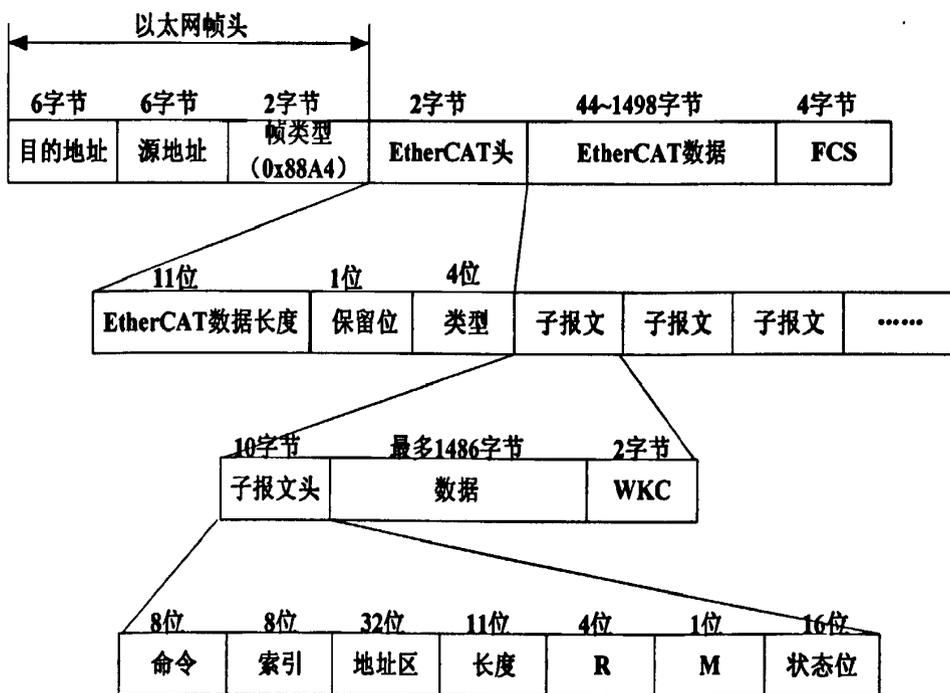


图 2-3 EtherCAT 报文嵌入以太网数据帧

表 2-1 EtherCAT 数据帧定义

名称	含义
目的地址	接收方 MAC 地址
源地址	发送方 MAC 地址
帧类型	0x88A4
EtherCAT 头: 数据长度	EtherCAT 数据长度, 即所有子报文长度总和
EtherCAT 头: 类型	1: 表示与从站通信; 其余保留
FCS (Frame Check Sequence)	帧校验序列

每个 EtherCAT 子报文包括子报文头, 数据域和相应的工作计数器 (WKC, Working Counter)。WKC 记录了子报文被从站操作的次数, 主站为每个通信服务子报文设置预期的 WKC。发送子报文中的工作计数器初值为 0, 子报文被从站正确处理, 工作计数器的值将增加一个增量, 主站比较返回的数据帧的同时进行处理, 不同的通信服务对 WKC 的增加方式不同。表 2-2 给出了 EtherCAT 子报文的结构定义<sup>[4]</sup>。

表 2-2 EtherCAT 子报文结构定义

名称	含义
命令	寻址方式及读写方式
索引	帧编码
地址区	从站地址
长度	报文数据区长度
R	保留位
M	后续报文标志
状态位	中断到来标志
数据区	子报文数据结构, 用户定义
WKC	工作计数器

### 2.1.3 通信模式

在实际自动化控制系统中，应用程序之间通常有两种数据交换形式：时间关键和非时间关键。时间关键表示特定的动作必须在确定的时间窗口内完成。如果不能在要求的时间窗口内完成通信，则很有可能引起控制失效。时间关键的数据通常周期性发送，称为周期性过程数据通信。非时间关键数据可以非周期性发送，在 EtherCAT 中采用非周期性邮箱 (mailbox) 数据通信。本项目所用的通信方式为非周期性邮箱数据通信，下面就主要介绍一下这种通信方式。

EtherCAT 协议中非周期性数据通信称为邮箱数据通信，它可以双向进行——主站到从站和从站到主站。它支持全双工、两个方向独立通信和多用户协议。从站到从站的通信是由主站作为路由器来管理。邮箱通信数据头中包括一个地址域，使主站可以重寄邮箱数据。邮箱数据通信时实现参数交换的标准方式，如果需要配置周期性过程数据通信或需要其他非周期性服务时需要使用邮箱数据通信<sup>[13]</sup>。

邮箱数据报文结构如图 2-4 所示，其数据头中数据元素的解释如表 2-3 所列。

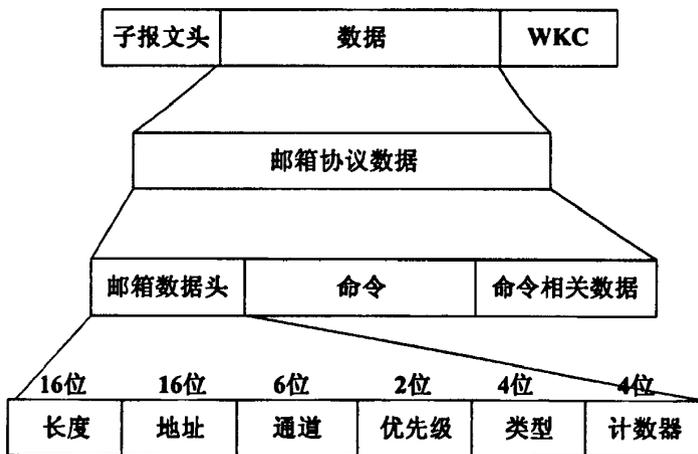


图 2-4 邮箱数据单元结构

表 2-3 邮箱数据头

数据元素	位数	描述
长度	16 位	跟随的邮箱服务数据长度
地址	16 位	从站到主站通信时，为数据源从站地址 主站到从站通信时，为数据目的从站地址

通信	6 位	保留
优先级	2 位	保留
类型	4 位	邮箱类型，即后续数据的协议类型， 0: 邮箱通信出错 2: EoE (Ethernet over EtherCAT) 3: CoE (CANopen over EtherCAT) 4: FoE (File Access over EtherCAT) 5: SoE (Servo Drive over EtherCAT) 15: VoE (Vendor specific profile over EtherCAT)
计数器 Ctr	4 位	用于重复检测的顺序编号，每个新的邮箱服务将加 1 (为了兼容老版本而只使用 1~7)

本项目中子报文的命令相关数据的协议类型为 CoE (CANopen over EtherCAT)，故类型就为 3。

#### (1) 主站到从站通信——写邮箱命令

主站发送写数据区命令将邮箱数据发给从站。主站需要检查从站邮箱命令应答报文中工作计数器 WKC。如果工作计数器为 1，表示写命令成功。反之，如果工作计数器没有增加，通常因为从站没有完成读完上一命令，或在限定的时间内没有响应，主站必须重新发邮箱数据写命令。

#### (2) 从站到主站通信——读邮箱命令

从站有数据需要发送给主站，必须先将数据写入输入邮箱缓存区，然后由主站来读取。主站发现从站 ESC 输入邮箱数据区有数据等待发送时，会尽快发送适当的读命令来读取。主站有两种方法来测定从站是否有数据填入数据区。一种是使用 FMMU 周期性读取某一个标志位。另一个方法是简单地轮询 ESC 输入邮箱的数据区。读命令的工作计数器增加 1 表示从站已经将新的数据填入数据区。

### 2.1.4 状态机和通信初始化

EtherCAT 状态机 (ESM, EtherCAT State Machine) 负责协调主站和从站应用程序在初始化和运行时的状态关系<sup>[4]</sup>。

EtherCAT 设备必须支持四种状态, 另外还有一个可选状态: Init(初始化)、Pre-Operational(预运行)、Safe-Operational(安全运行)、Operational(运行)、和可选状态 Boot-Strap(引导状态)。

以上各状态之间的转化关系如图 2-5 所示。从初始化状态到运行状态转化时, 必须按照“初始化—预运行—安全运行—运行”的顺序转化, 不可以越级转化。从运行状态返回时可以越级转化。

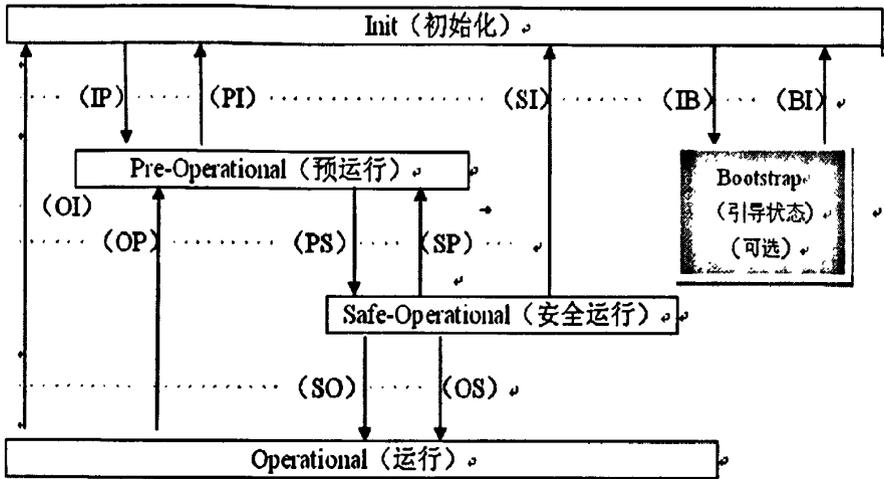


图 2-5 EtherCAT 状态转化关系

### 2.1.5 应用层协议

应用层 AL (Application Layer) 是 EtherCAT 协议最高的一个功能层, 是直接面向控制任务的一层, 它为控制程序访问网络环境提供手段, 提示为控制程序提供服务。应用层不包括控制程序, 它只定义了控制程序与网络交互的接口, 是符合此应用层协议地方各种应用程序可以协同工作, 如图 2-6 所示。

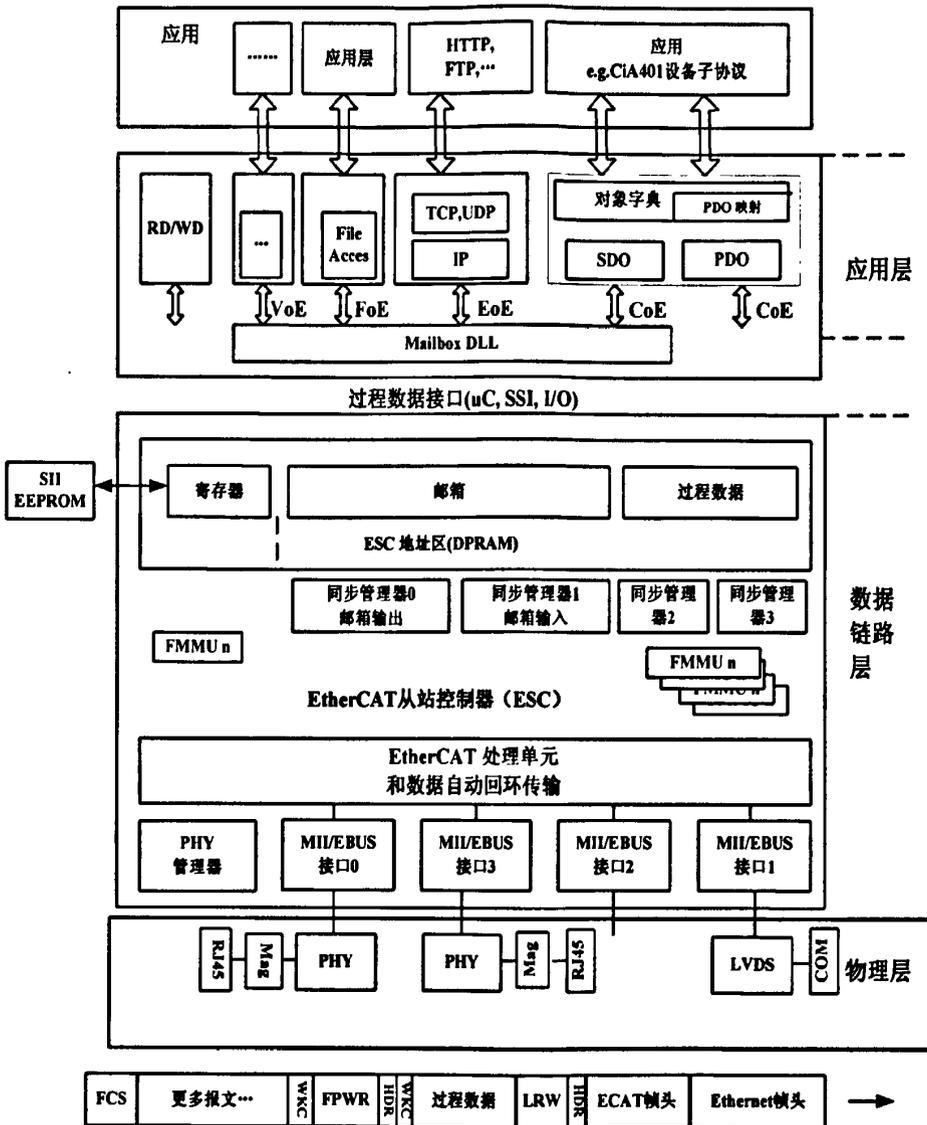


图 2-6 EtherCAT 协议结构

EtherCAT 包括很多种应用层协议，其中包括 CoE (CANopen over EtherCAT), SoE (Servo Drive over EtherCAT), EoE (Ethernet over EtherCAT), FoE (File Access over EtherCAT)<sup>[4]</sup>。本项目中使用 CoE (CANopen over EtherCAT) 作为数据通信的协议。

## 2.2 从站微处理器 LPC1754 的主要电路设计

### 2.2.1 模块概述

开发基于 EtherCAT 从站结构,要求开发出嵌入 AD/DA 模块的从站微处理器芯片 LPC1754。芯片包括 AD/DA 模块,运算放大器,数字隔离器,模拟电源,数字电源以及电路外围接口等。

### 2.2.2 AD 模块

- 1、模拟量输入通道数: 8 路
- 2、采样频率: 每通道不小于 20ksps
- 3、AD 位数: 16bit
- 4、AD 模块量程: 0-10V
- 5、模块工作电压: 24V

根据要求,选择的 AD 芯片为电荷再分配逐次逼近寄存器 (SAR) 型模数转换器 AD7689BCPZRL7,其技术参数如表 2-4。

表 2-4 AD7689 概述

参数	说明
位数	16bit
通道数	8 通道 (可选择输入、单极性单端输入、差分输入、伪双极性输入)
吞吐速率	250kSPS
-3dB 输入带宽	全带宽 1.7MHz, 1/4 带宽 0.425MHz
INL	典型值 $\pm 0.4$ LSB, 最大值 $\pm 1.5$ LSB ( $\pm 23$ ppm 或 FSR)
动态范围	93.8dB
SINAD	92.5dB (20kHz)
THD	-100dB (20kHz)
模拟输入范围	0V-VREF, VREF 可达 VDD
基准源	内部可选 2.5V 或 4.096V 外部缓冲基准源 (可达 4.096V) 外部基准源 (可达 VDD)

电源	单电源工作：2.3V-5.5V，逻辑接口电压：1.8V-5.5V
接口	SPI、MICROWIRE、QSP 和 DSP
功耗	3.5mW (2.5V/200kSPS)、12.5mW (2.5V/250kSPS)、 15.5mW (5V/250kSPS)
待机电流	50nA
封装	20 引脚 4mm*4mmLFCSP
工作温度	-40℃——+85℃
模拟输入漏电流	1nA
增益误差温漂	±1ppm/℃
失调误差温漂	±1ppm/℃
内部基准电压源温漂	±10ppm/℃
输入电压调整率	±15ppm/V
外部基准源漏电流	50uA (250ksps, REF=5V)

该 AD 芯片吞吐率最大为 250ksps，大于所需 20ksps。当 VDD=5V，使用 250ksps 吞吐速率，内部基准源时，最大功耗为 21mW。当 VDD=5V 时，吞吐率才能达到 250ksps，因此选用 5V 供电。

其中，如果采用单极性模式的模拟输入电压范围是 0 至 VREF，而 VREF 的电压需要小于 VDD，可选用内部参考源或外部参考源。由于采样电压为 0-10V，而 VREF 不能大于 VDD，为稳妥且设计需要，因此取 VREF=2.5V。由于 2.5V 的内部基准源的温漂为 10ppm/℃，那么按一般高温 45℃计算，误差则达 500ppm=500uV，而 2.5V 的 16bit 的 1LSB=38.1uV，不算增益误差和失调误差，误差已经达到 13.1LSB。因此选用一个精度较高的 2.5V 外部参考源 ADR381ARTZ-REEL7（详细参见该部分）。

另外，采样电压为 0-10V，而选定的参考电压为 2.5V，所以前段的模拟输入部分需要作降压处理。具体是通过电阻对输入信号进行分压，再接上一个电压跟随器，最后接入模拟输入端口。其电路图如图 2-7。

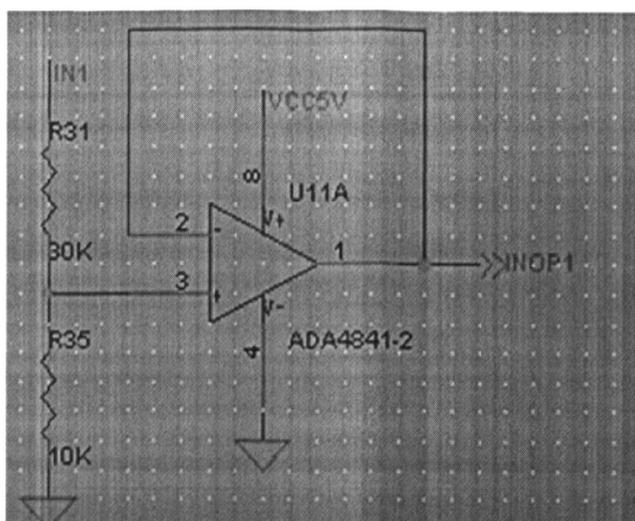


图 2-7 AD 前端模拟信号处理电路

数字信号电源部分，VIO 同样采用 5V 供电。

SPI 通讯的时序有 3 种模式：转换器件读取/写入，适用快速主机；转换后读取/写入，适用任何速度的主机；转换全程读取/写入，适用任何速度的主机。由于选取的 250ksps 速率大于 20ksps，因此选用较为简单而精确的转换后读取/写入的模式。

另外，对应以上三种模式中，都有带繁忙指示器和无繁忙指示器这两种模式。二者区别在于，采用无繁忙指示器时，主机必须根据转换时间进行数据传输，而连接使用 SPI 且带中断输入的主机时。需要采用繁忙指示器的驱动模式。因为采用 LPC1754 的 ARM 主机是带中断的，因此采用繁忙指示器的驱动模式。

内部可选低通滤波器：全带宽 1.7MHz，1/4 带宽：425kHz。

### 2.2.3 DA 模块

- 1、 模拟量输出通道数：2 路
- 2、 更新频率：每通道不小于 20KHz
- 3、 DA 位数：16bit
- 4、 DA 模块输出量程：0-10V
- 5、 模块工作电压：24V

根据需求，所选的 DA 芯片为轨到轨输出的 nanoDAC 数模转换器

AD5663ARMZ-REEL7, 其技术参数如表 2-5。

表 2-5 AD5663 技术参数

参数	说明
位数	16bit
通道数	2 通道
相对精度	最大±12LSB
封装	10 个引脚的 3mm*3mm 的 MSOP
电源	2.7-5.5V
功耗	0.6mW (3V), 1.25mW (5V)
特性	硬件的 LDAC、CLR 功能
工作温度	-40℃——+105℃
SPI 传输速率	50MHz
-3dB 输出带宽	340kHz
输出模拟电源	0-VDD
输出阻抗	0.5 Ω
输出噪声	45uVp-p
外部参考源输入电压	0.75V-VDD
外部参考源输入电流	典型 170uA, 最大 200uA
外部参考源输入阻抗	26k Ω

该芯片的更新速率为 50MHz, 大于要求的 20kHz。选用 5V 供电。最大功耗为 1.25mW。由于此芯片没有外部参考源, 因此使用外部参考源, 且与 AD 芯片的参考源相同的 2.5V。需要注意外部参考源输入电流最大值为 200uA。由于参考电压为 2.5V, 但要求输出模拟量为 0-10V, 所以在模拟量输出端口要接入一个正向放大器, 把电压还原为 0-10V。电路图如图 2-8。

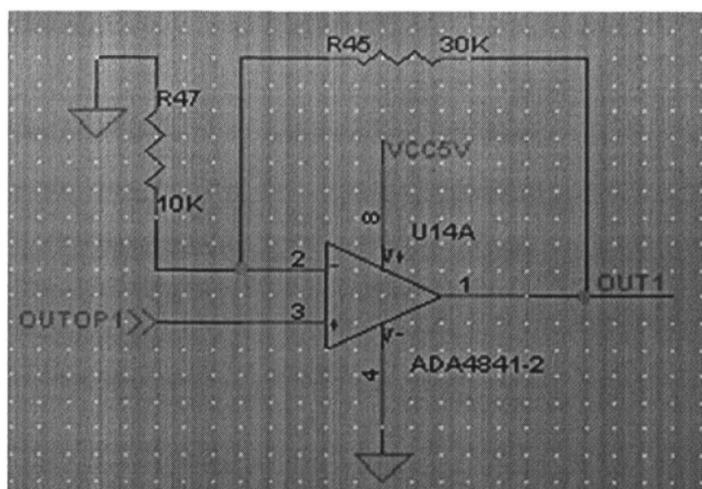


图 2-8 DA 模拟信号端放大电路

另外，芯片带硬件的 LDAC 和 CLR 功能。其中 LDAC 可由一个配置位确定其是否使能。而 CLR 相当于一个异步复位键，但应用中不需用到 CLR 的清零功能。因此，对这两个引脚都置为高电平。

### 2.2.4 运算放大器

选用 ADA4841-2YRMZ-R7。其参数如表 2-6 所示。

表 2-6 ADA4841-2 参数

参数	说明
功耗	1.1mA/amp
带宽噪声	2.1 nV/√Hz , 1.4 pA/√Hz
失真	-105 dBc @ 100 kHz, V <sub>O</sub> = 2 V p-p
-3dB 带宽	80MHz (G=+1)
压摆率	12 V/μs
输入失调电压	典型 40uV, 最大值 300uV
输入失调电压温漂	1uV/°C
电源电压	2.7V-12V
工作温度	-40°C——+125°C

直流输入偏移电压	典型值 40uV, 最大值 300uA
输出电压摆幅	最小值 0.045V-2.955V, 典型值 0.023-2.988V

在增益为 1 的时候, 带宽达 80MHz, 满足 AD 芯片的 1.7MHz 输入带宽; 在增益为 5 时, 带宽也有 11MHz, 也满足 DA 芯片的 340kHz。

另外, 对于使用跟随器的 AD 输入端, 运放的输入失调电压典型值位 40uV, 再加上温漂的 20uV, 也达到 60uV=1.57LSB。但是它的最大失调电压非常大, 达到 300uV, 再加上温漂的 20uV, 输入的误差达到 320uV=8.4LSB 了。对于使用了增益 G=+4 的 DA 输出, 输出误差最严重的时候达 1.28mV。所以, 在环境恶劣的情况下, 该运放存在较大的误差。

按输出功率算, 每个 AD 输入的功耗大概为  $2.5V * (2.5V / 2.2k\Omega) = 2.84mW$ , 那么 8 个 AD 端输入的功耗约为 24mW。至于 DA 部分的运放功耗未知, 估算为 10mW。而为减少输出电流, 可考虑在输出端接上一个限流电阻。那么 5 个运放的功耗大概为 44mW。

### 2.2.5 参考电源

选用参考电压为 2.5V 的 ADR381ARTZ-REEL7, 其技术参数如表 2-7。

表 2-7 ADR381 参数

参数	说明
输入电压范围	2.8V—18V
输出电压	最小 2.494V, 典型 2.5V, 最大 2.506V
静态电流	最大 140uA
温漂	典型 3 ppm/°C, 最大 25 ppm/°C
电压噪声	5uVp-p (0.1Hz-10Hz)
纹波抑制比	85dB
初始精度	±5mV, 最大±6mV
工作温度	-40°C—+80°C

其典型温漂为 3 ppm/°C, 按 2.5V, 45 °C 计算, 温漂带来的误差为  $2.5 * 3 * 20 = 150uV$  即为 3.94LSB。

## 2.2.6 数字隔离器

选用 ISO7241ADWR，其技术参数表 2-8。

表 2-8 ISO7241 参数

参数	说明
电源电压范围	-0.5V—6V
IN, OUT, EN 电压	-0.5V—6V
输出电流	±15mA
高电平输入电压	0.7VCC—VCC
低电平输入电压	0—0.3VCC
工作温度	-40°C—+125°C

为了使 AD/DA 芯片对处理器进行隔离，使用两个数字隔离器对其进行隔离，且使用对隔离器的两端分别数字电源、数字地和模拟电源、模拟地以达到完全隔离。电路图如图 2-9。

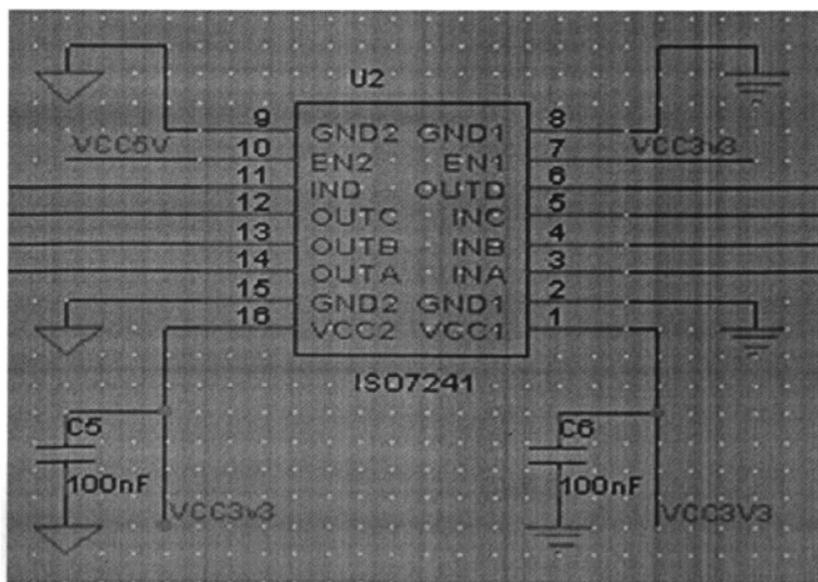


图 2-9 数字隔离器应用示例

每个隔离器的最大功耗为 220mW。两个加起来的功耗为 440mW。

### 2.2.7 模拟电源

根据需求，选用 LM317AEMP，其技术参数如表 2-9。

表 2-9 LM317 参数

参数	说明
电压线性调整率	$3V \leq (V_{IN} - V_{OUT}) \leq 40V$
参考电压	最小 1.225V，典型 1.250V，最大 1.270V
输出电流	最小 1.5A，典型 2.2A，最大 3.3A
调整引脚 ADJ 电流	典型 50uA，最大 100uA
纹波抑制比	最小 66dB，典型 80dB
工作温度	-40℃——+125℃

LM317 是线性电源，给 AD，DA，模拟输入部分的运放，参考源等芯片供电。可把直流输入电压 24V 转化为 5V 输出电压。另外，其典型输出电流为 2.2A，即它的功率可以达到  $2.2 \times 5W = 11W$ ，足够给模拟部分的芯片进行供电。

利用该芯片，设计出 24V-5V 的直流稳压电源，其电路图如图 2-10。

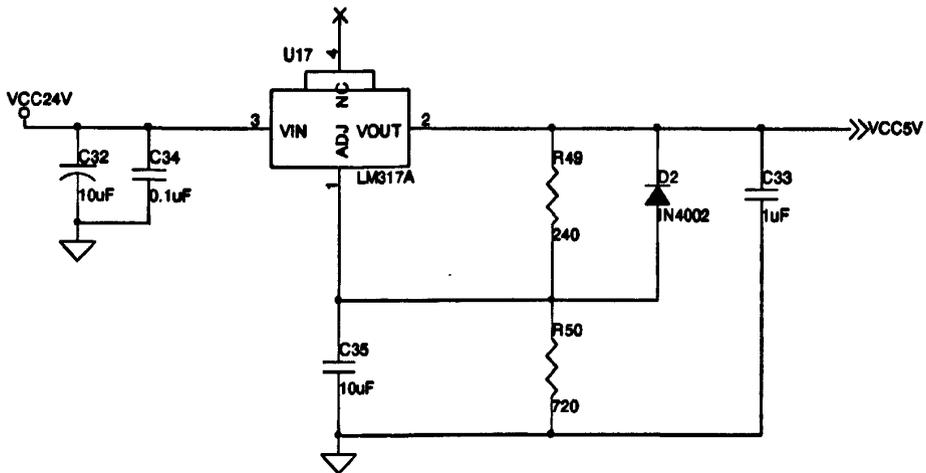


图 2-10 模拟电源 24V-5V 电路图

由于 DA 输出范围是 0-10V，因此需要一个 10V 以上的电源供电。12V 模拟电源的电路图如图 2-11。

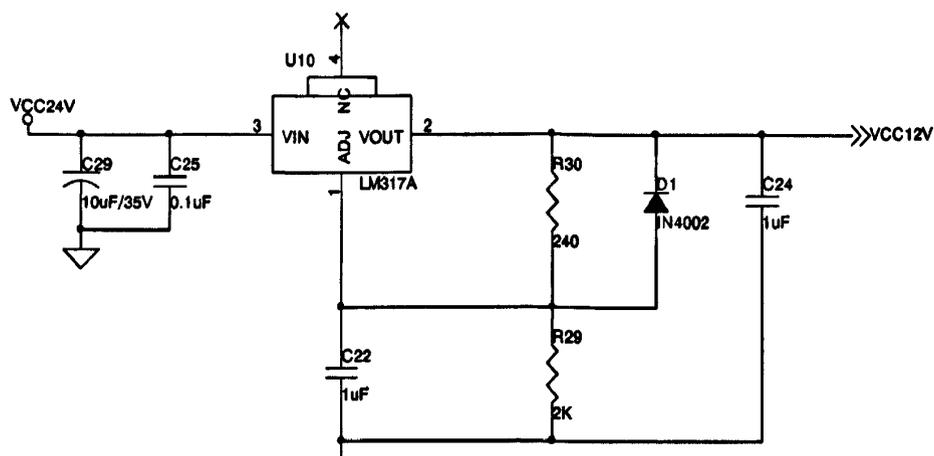


图 2-11 模拟电源 24V-12V 电路图

## 2.2.8 数字电源

根据需求，选用一个 DC-DC 隔离开关电源 WRB2405YMD-6W 以提供隔离的 5V 电源，以及一个低压差的线性电源 AMS1117-3.3 以提供 3.3V 电源。这两块芯片的技术参数如表 2-10。

表 2-10 WRB2405YMD-6W 参数

参数	说明
电压输入范围	18V-36V, 最大 40V
输出电压	5V
输出电流	最小 150mA, 最大 1500mA
调整引脚 ADJ 电流	典型 50uA, 最大 100uA
效率	79%
输出功率	最大 6W
纹波和噪声	典型 20 mVp-p, 最大 50 mVp-p (20MHz 带宽)
转换频率	300KHz (100%负载, 标称输入电压)
工作温度	-40℃——+85℃

由于数字供电部分中的 ARM 处理器有 1.5W 的功耗, RS232 接口和 EtherCAT 接口的供电功率大概需要 1—2W, 因此再加上其它的复位芯片、与门、RS-232

的供电，达到 4W，再加上 10%-20%的裕度。

利用该芯片，设计出 24V-5V 的直流隔离电源，其电路图如图 2-12。

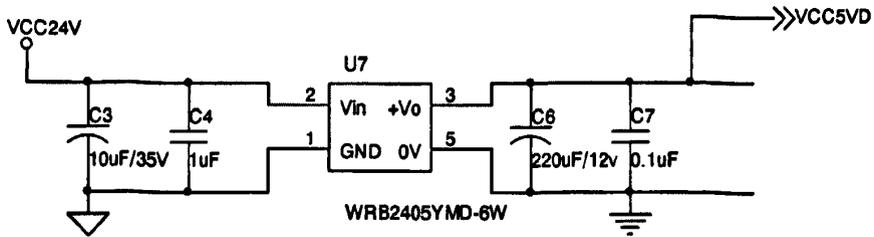


图 2-12 DC-DC 隔离电源 24V-5V 电路图

表 2-11 AMS1117-3.3 参数

参数	说明
输入电压	4.75V—12.0V
输出电压	最小 3.235V，典型 3.300V，最大 3.365V
输出电流	最小 0.9A，典型 1.1A，最大 1.5A
调整引脚 ADJ 电流	典型 55uA，最大 120uA
纹波抑制比	最小 60dB，典型 72dB
工作温度	-40℃——+125℃

ARM 处理器需要 3.3V 的供电。而该输出电流电源典型值为 1.1A，即功率为 3.63W，足够给 ARM 及其它芯片供电。

利用该芯片，设计出 5V-3.3V 的直流稳压电源，其电路图如图 2-13。

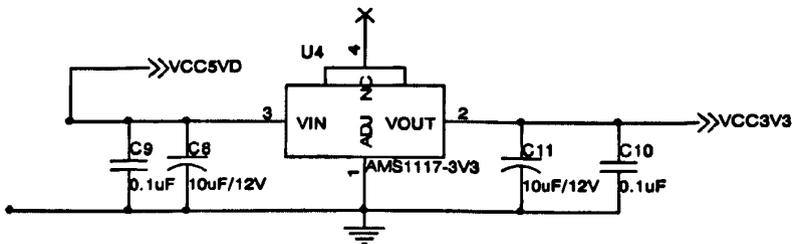


图 2-13 线性稳压电源 5V-3.3V 电路图

### 2.2.9 电路外围接口

加入通用异步收发器(UART)方便 ARM 调试、程序下载。其中使用了 LPC1754 上的 P0.3 引脚 RXD0 作为串行输入引脚, P0.2 引脚 TXD0 作为串行输出引脚。另一方面, ARM 需要使用 UART 接口与 RS-232C 接口的设备(PC 机)相连以进行基本的通讯, 所以需要一个 RS-232C 转换器将 TTL (Transistor-Transistor Logic) 电平转换成 RS-232C 电平, 如图 2-14 所示。

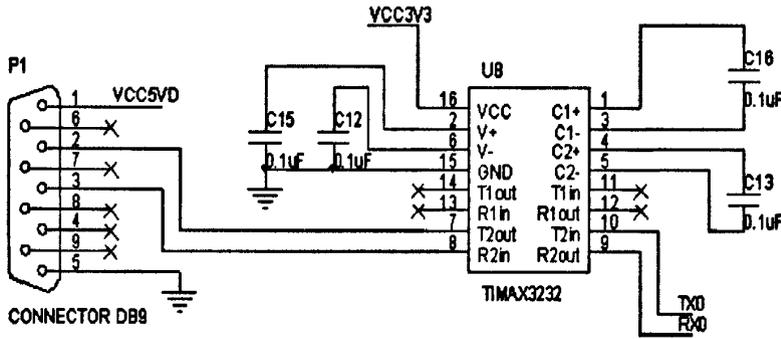


图 2-14 RS-232C 接口电路图

此外, 添加 JTAG (Joint Test Action Group) 接口以实现 ARM 的在线调试 (ICE) 或在线编程等功能。其中, LPC1754 的引脚 1、2、3、4、5 分别是 JTAG 信号的测试数据串行输出口 (TDO)、测试数据串行输入口 (TDI)、测试模式选择口 (TMS)、测试系统复位信号口 (J\_nRST)、测试时钟口 (TCK)。另外, J\_nRST 与系统复位信号相连, 可以检测系统的复位情况。JTAG 使用 20 针的接口, 其引脚图如图 2-15。

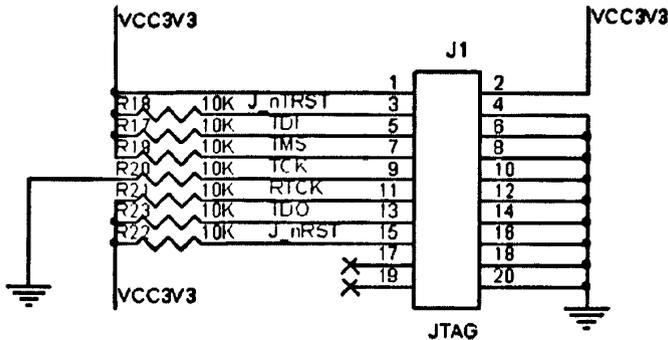


图 2-15 JTAG 接口电路图

在系统复位电路中，ARM 的外部复位芯片使用了 CAT1025。它的电源监控电路和复位电路可在系统上/下电时保护存储器和系统控制器。而存储器则采用的高速 IIC (Inter Integrate Circuit) 总线接口，支持手动按键复位输入和写保护输入。其中，LPC1754 上的 P0.10、P0.11 分别作为 IIC 总线的 SDA 接口、SCL 接口。其复位电路图如图 2-16。

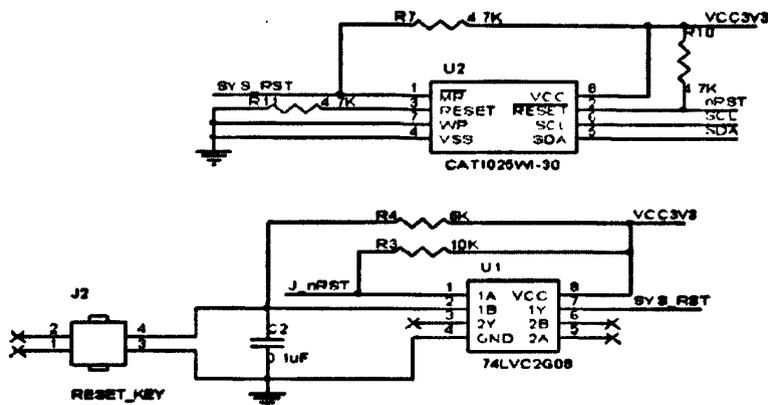


图 2-16 复位电路电路图

### 2.2.10 微处理器

处理器芯片需要至少为16位，而模块需要3个 SPI或SSP接口，RS-232接口，JTAG接口，复位键等。因此，选用的LPC1754FBD80为32bit，工作频率为100MHz，且带有128KB的flash。其参数如表2-12。

表 2-12 LPC1754 参数

参数	说明
电源电压范围	2.4V-3.6V
Flash	128KB
SRAM	32KB
封装	LQFP80
功耗	最大 1.5W
保存温度	-65℃——+150℃

在通讯上, AD、DA 使用处理器上的两个 SSP 接口, 而处理器的 SPI 接口用作 EtherCAT 接口。另外有 JTAG 接口、RS-232 接口、复位键接入处理器。

## 2.3 从站微处理器 LPC1754PCB 制作

印刷电路板 PCB(Printed Circuit Board), 既能大大地缩小电路板体积, 也能使电路具有更好的性能。由于电路中的芯片封装都为贴片, 且电路板制作的大小要求是名片大小, 除此之外, 更重要的是因为使用的 16 位的高精度 DA 转换器, 为了电路的信号完整性及电磁兼容性考虑, 所以需要制作一个 4 层的 PCB 电路板以使模块获得更好的性能。

根据 PCB 设计中的电磁兼容原理<sup>[26][27]</sup>和电源、信号完整性原理, 在该模块中, 主要作了以下处理:

(1) 在布局上, 把模拟信号的元件放在一边, 把数字信号的元件放在另一边。

(2) 把 DC-DC 直流隔离开关电源和微处理器的晶振部分远离模拟信号, 防止高频信号对模拟信号造成干扰。

(3) 电源部分的元器件都放在一个区域里, 可以防止电流环路和产生的感生电感对整体电路造成严重的干扰。

(4) 每个芯片的电源引脚处均并联上一个 10uF 和一个 100uF 的电容器进行旁路的滤波去耦, 且小电容更靠近芯片引脚处。

(5) 为了避免多条走线产生串扰, 特别地, 把接地走线和电源走线均通过一个过孔, 在走一段很小距离的走线连接到相应的引脚处, 使其能够相应地接到地层和电源层。

PCB 设计中的电磁干扰 EMI (Electro Magnetic Interference)、电磁兼容 EMC (Electro Magnetic Compatibility)、电源完整性 PI (Power Integrity)、信号完整性 SI (Signal Integrity) 是一门博大精深的学科, 要做出一块完全没有电磁干扰的 PCB 是不可能的, 我们只能通过对以上的学科进行研究、分析, 才能更好地避免电磁干扰, 做出性能更好地 PCB 电路板。LPC1754 电路图如图 2-17, 制作好的 PCB 设计图如图 2-18, 焊接好的电路板如图 2-19 和 2-20 所示。

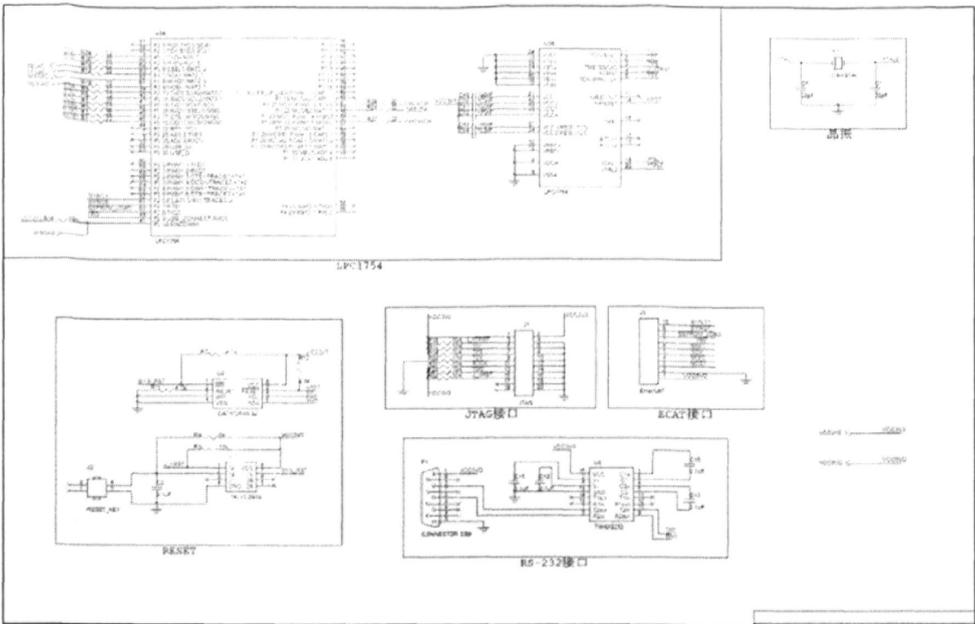


图 2-17 LPC1754 主要部分电路图

下图为制作好的 PCB 设计图。

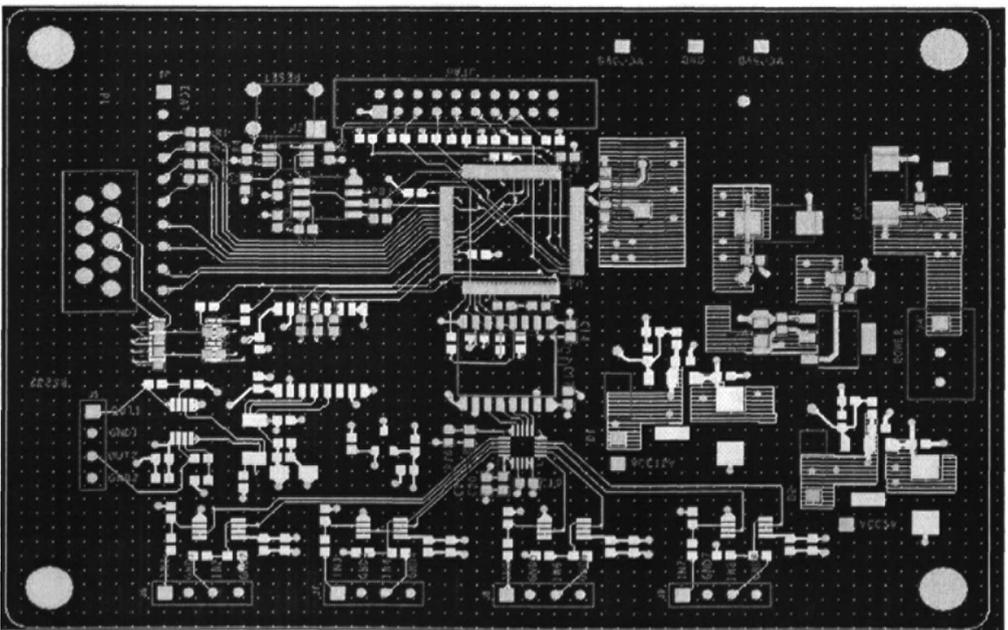


图 2-18 LPC1754PCB 版图

图 2-19, 2-20 为焊接好的电路板实物图。

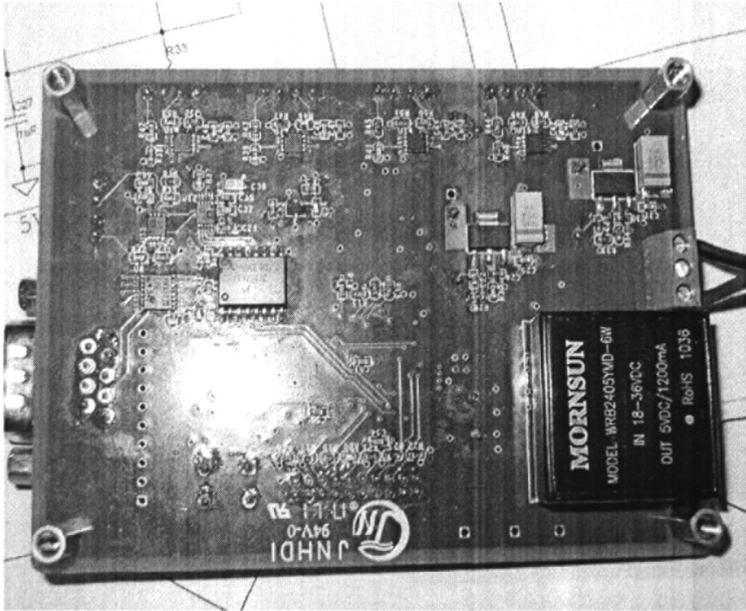


图 2-19 电路板图

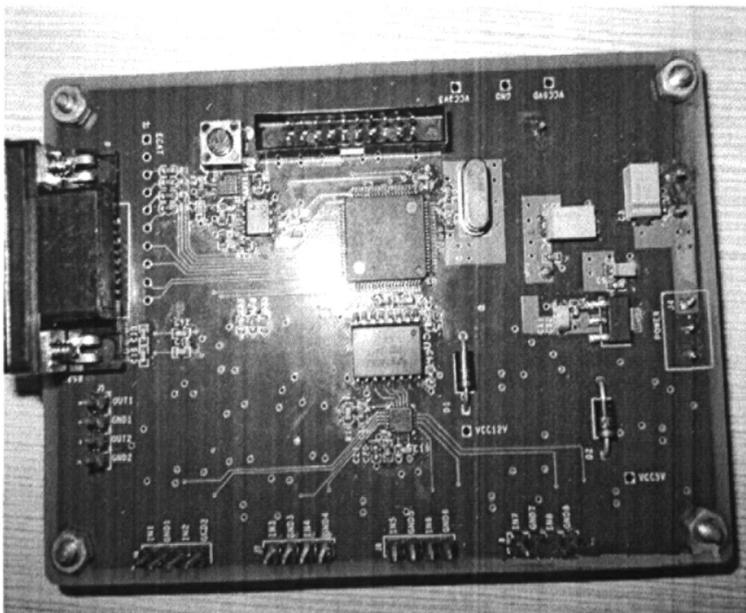


图 2-20 电路板图 (反面)

## 2.4 本章小结

本课题致力于开发高精度和可靠性、实时性好的从站控制芯片，能够实现正常通信，运行稳定并且反应快速，以满足各项要求。故本章首先分析了整个从站系统的硬件结构和硬件选择，然后介绍了本课题中从站微处理器芯片 LPC1754 的主要电路设计。接着给出了 PCB 设计图，最后给出了焊接好的实物图。

## 第3章 EtherCAT 从站微处理器驱动程序设计

### 3.1 CANopen 协议简介

CANopen 最初是为 CAN (Control Area Network) 总线控制系统所开发的应用层协议。EtherCAT 协议在应用层上支持 CANopen 协议,并做了相应的扩充。主要功能有:

- a. 使用邮箱通信访问 CANopen 对象字典以及对象,实现网络初始化。
- b. 使用 CANopen 应急对象和可选的时间驱动 PDO 消息,实现网络管理。
- c. 使用对象字典映射过程数据,周期性传输指令数据和状态数据。
- d. CoE 在整个 EtherCAT 网络中的结构如图 3-1 所示<sup>[13]</sup>。

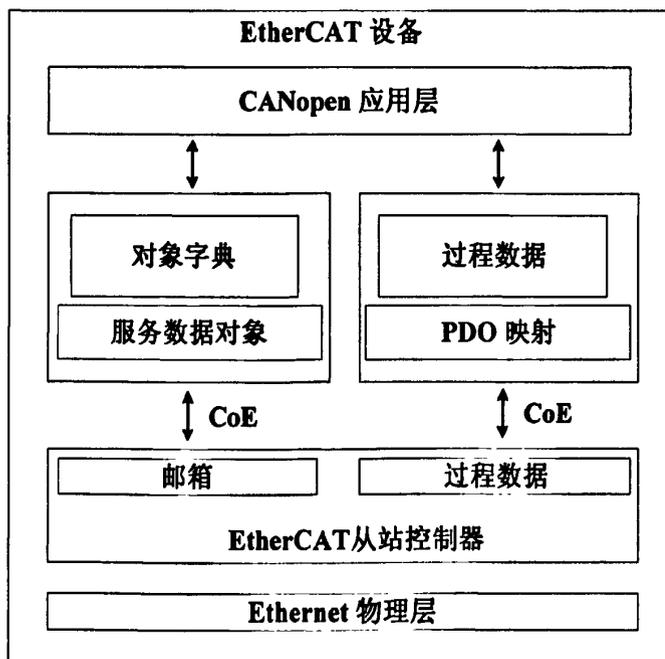


图 3-1 CoE 设备结构图

EtherCAT 作为网络通信技术,支持 CANopen 协议中的行规 CiA401 和 CiA402

等，称为 CoE。CoE 协议完全遵从 CANopen 协议，其对象字典的定义以及报文的类型等也是相同的。故下面重点介绍一下 CANopen 协议。

### 3.1.1 CANopen 设备模型

要理解 CANopen 规范，核心是要理解 CANopen 的设备模型和各类型的通讯对象<sup>[14]</sup>。一个 CANopen 设备由 3 个部分组成，分别是通讯部分、对象字典和应用部分，此设备一端接到控制网络，另一端接到应用相关的 I/O 数据端口，如图 3-2 所示。

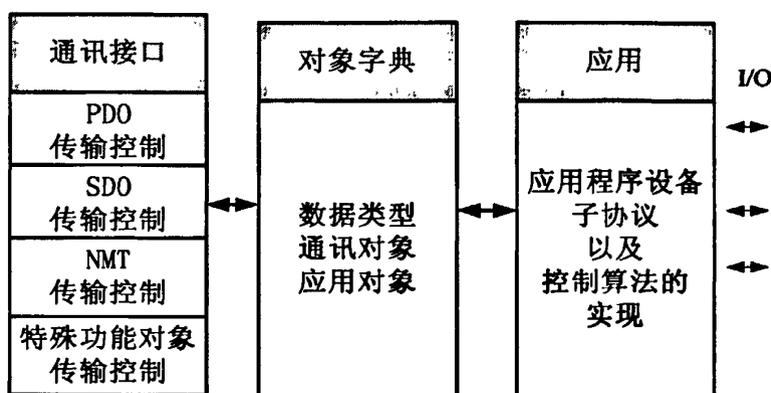


图 3-2 CANopen 节点的设备模型

通讯部分根据 CANopen 协议的要求在上收发 CoE 报文，不同的是 CANopen 设备间的通信都是通过交换通讯对象完成的。应用部分由用户根据实际的应用要求编写具体的应用软件。应用部分与控制网络接口部分之间是对象字典，这是一个 CANopen 设备的核心部分，每个设备都有一个自己的对象字典，其中描述了设备使用的各种数据对象和参数。

### 3.1.2 CANopen 通信模型

前面说明了 CANopen 中设备模型的概念，现在来介绍在 CANopen 网络中的通讯消息，它们的内容和功能，换句话说：CANopen 通讯模型。

CANopen 通讯模型定义了 4 种报文（通讯对象）<sup>[14][16]</sup>：

- (1) 管理报文 NMT (Network Management Object)

主要功能是层管理,网络管理和 ID 分配服务:如初始化,配置和网络管理(包括:节点保护)。这些服务都是基于主从通讯模式:在一个 CAN 网络中,只能有一个 NMT 主节点,可以有一个或多个 NMT 从节点。

#### (2) 服务数据对象 SDO(Service Data Object)

通过使用 16 位索引和 8 位子索引(在 CoE 报文的前几个字节),SDO 使客户机能够访问设备(服务器)对象字典中的项(对象)。协议是确认服务类型。

#### (3) 过程数据对象 PDO(Process Data Object)

PDO 用来传输实时数据,提供数据的为生产者,消费数据的为消费者。数据从一个生产者传到一个或多个消费者。数据传送长度限制在 1 到 8 个字节(例如,一个 PDO 可以传输最多 64 个数字 I/O 值,或者 4 个 16 位的 AD 值)。

本课题就是利用 PDO 将 16 位的数字量通过 PDO 报文传送给 DA 芯片,完成数模转换<sup>[16][21]</sup>。

#### (4) 预定义报文或者特殊功能对象

包括同步对象(SYNC Object)、紧急事件(Emergency Object)和时间标记对象(Time Stamp Object)。这些对象用于处理网络中的错误以及时间同步。

### 3.1.3 对象字典

#### (1) 对象字典概述

对象字典(Object Dictionary, OD)<sup>[16]</sup>是 CANopen 协议中最重要的部分,而且 CoE 协议完全遵从 CANopen 协议,目的是对各类对象进行规范化管理。对象字典就像一个数据表,它包含了所有一个网络可到达的数据,可以通过网络读取的一组预先定义的对象。每一个 CANopen 节点都必须要实现自身的对象字典。对象字典包含了描述这个 CANopen 节点和它的网络行为的所有参数,另外,一个节点或设备的对象字典可以被其他 CANopen 节点访问和修改。

在一个节点中,当读取节点的对象字典条目时,其它的节点就能够得知关于这个节点在做什么和它是如何操作的信息;当把一个数据写到对象字典的条目时,这个节点将会被指示执行某一类型的操作。该把哪些描述信息存入对象字典,是可以根据网络设计的需求而改变的。然而,有些信息在对象字典中是必需的,一定要包含在内。这些必需的信息取决于在节点中所实现的 CANopen 特性。因此通过网络改变特定对象的值就可以很方便的改变设备的配置和状态。

CANopen 使用面向对象的方法来定义标准设备:每个设备都表现为一组对

象的集合。对象字典存放着设备的各种参数，包括数据类型定义、通讯参数、设备参数和接口参数等。对象字典是通讯对象与应用对象的接口，起着连接内部与外部的桥梁作用。

利用索引与子索引对对象字典进行访问。通常索引为 16 位，子索引为 8 位。索引用于访问设备中的每个对象，子索引用于访问数据结构中的单个元素。

(2) 对象字典结构

一个标准的 CANopen 对象字典的结构<sup>[6]</sup>，如表 3-1 所列。

表 3-1 CANopen 对象字典

索引	对象
0000	Not used
0000-001F	静态数据类型 (标准数据类型, 如 Boolean Integer 16)
0020-003F	复杂数据类型 (预定义的由简单类型合成的结构 PDO CommPar, SDO Parameter)
0040-005F	制造商规定的复杂数据类型
0060-007F	设备子协议规定的静态数据类型
0080-009F	设备子协议规定的复杂数据类型
00A0-0FFF	Reserved
1000-1FFF	通讯子协议区域 (如设备类型, 错误寄存器, 支持的 PDO 数目)
2000-5FFF	制造商规定的子协议区域
6000-9FFF	标准的设备子协议区域 (例如 “DSP401 I/O 模块设备子协议”: Read State 8 Input Lines)
A000-FFFF	Reserved

表 3-2 列出了 CoE 通讯数据对象。其中针对 EtherCAT 通信扩展了相关的通讯对象 0x1C00~0x1C4F，用于设置存储同步管理器的类型、通讯参数和 PDO 数据分配<sup>[4]</sup>。

表 3-2 CoE 通讯数据对象

索引值	含义
0x1000	设备类型：32 位整数 位 0~15：所使用的设备行规 位 16~31：基于使用行规的附加信息
0x1001	错误寄存器，8 位 位 0：常规错误 位 1：电流错误 位 2：电压错误 位 3：温度错误 位 4：通信错误 位 5：设备行规定义错误 位 6：保留 位 7：制造商定义错误
0x1008	设备商设备名称，字符串
0x1009	制造商硬件版本
0x100A	制造商软件版本
0x1018	设备标识符，结构体类型 子索引 0：参数体数目 子索引 1：制造商 ID (Vendor ID) 子索引 2：产品码 (Product Code) 子索引 3：版本号 (Revision Number) 子索引 4：序列号 (Serial Number)
0x1600-0x17F F	RxPDO 映射，结构体类型 子索引 0：参数体数目 子索引 1：第一个映射的输出数据对象 ： 子索引 n：最后一个映射的输出数据对象
0x1A00-0x1BF F	TxPDO 映射，结构体类型 子索引 0：参数体数目 子索引 1：第一个映射的输入数据对象 ： 子索引 n：最后一个映射的输入数据对象
0x1C00	同步管理器通信类型，子索引 0 定义了所使用的 SM 的个数，子索引 1~32 定义了相应的 SM0~SM31 通道的通信类

	<p>型，相关通信类型有：</p> <p>0：邮箱输出，非周期数据通信，1个缓存区写操作；</p> <p>1：邮箱输入，非周期数据通信，1个缓存区读操作；</p> <p>2：过程数据输出，周期性数据通信，3个缓存区写操作；</p> <p>3：过程数据输入，周期性数据通信，3个缓存区读操作</p>
0x1C10-0x1C2 F	<p>过程数据通信同步管理器 PDO 分配：</p> <p>子索引 0：分配的 PDO 数目；</p> <p>子索引 1~n：PDO 映射对象索引号。</p>
0x1C30-0x1C4 F	<p>同步管理器参数：</p> <p>子索引 1：同步类型（有自由运行、同步于数据输入或输出事件和同步于分布时钟同步事件）；</p> <p>子索引 2：周期时间，单位为 ns；</p> <p>子索引 3：AL 事件和相关操作之间的偏移时间，单位为 ns。</p>

### (3) 对象字典描述

对象字典时一组有序的对象，就像一个表格，每个对象通过 16 位的索引来寻址。对于数组和结构等复杂类型的对象，使用 8 位子索引来访问其内部成员。对于每个对象来说，它都有一组属性。对象字典是一系列有序条目的集合。一个对象字典通常被描述成一个表格的形式，其属性如下表 3-3 所示。例如索引值为 0x1000，名称为设备类型的对象条目的描述如表 3-4 所示。

表 3-3 对象字典属性

标题	描述
索引 (Index)	对象字典的索引值
子索引 (Sub-index)	对象字典的子索引值
名称 (Name)	对象字典内容的名称
大小 (Size)	对象字典内容所占字节数
值 (Value)	对象字典对应数据值
内容 (Comment)	对象字典简介

表 3-4 设备类型对象字典

Index	Sub-index	Name	Size	Value	Comment
0x100 0	0	设备类型	4 bytes	0x191	CiA 定义

## 3.2 EtherCAT 从栈程序结构概述

在 EtherCAT 应用中，一般是基于主从模式构建网络。EtherCAT 从站实现通信和控制应用两部分功能，其主要部件是从站控制器 ESC 和从站控制微处理器。ESC 的主要功能是用于数据转发，从站控制微处理器实现数据处理以及控制任务功能<sup>[4]</sup>。图 3-3 为从站协议栈结构图。本章主要介绍 EtherCAT 从站驱动、CANopen 协议栈对象字典的设计与实现、CoE 报文的封装和解析以及协议栈的通信实现。

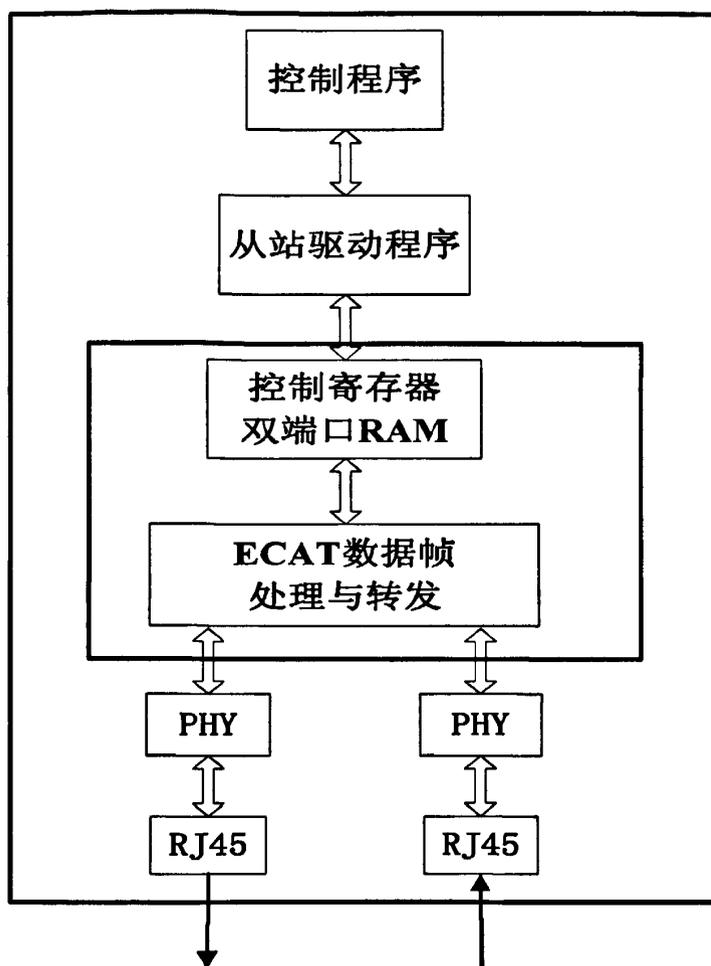


图 3-3 EtherCAT 从站协议栈

### 3.2.1 EtherCAT 从站程序结构

EtherCAT 从站以 EtherCAT 从站控制器 ESC 芯片为核心,ESC 实现 EtherCAT 数据链路层,完成数据的接受和发送以及错误处理。从站使用微处理器操作 ESC 芯片,实现应用层协议,包括以下任务:

- (1) 微处理器初始化、通信变量和 ESC 寄存器初始化。
- (2) 通信状态机处理,完成通信初始化:查询主站的状态控制器,读取相关配置寄存器,启动和终止从站相关通信服务。
- (3) 非周期性数据处理,实现邮箱通信:查询应用层非周期事件。

图 3-4 为支持邮箱通信的从站驱动程序流程<sup>[13] [16]</sup>。

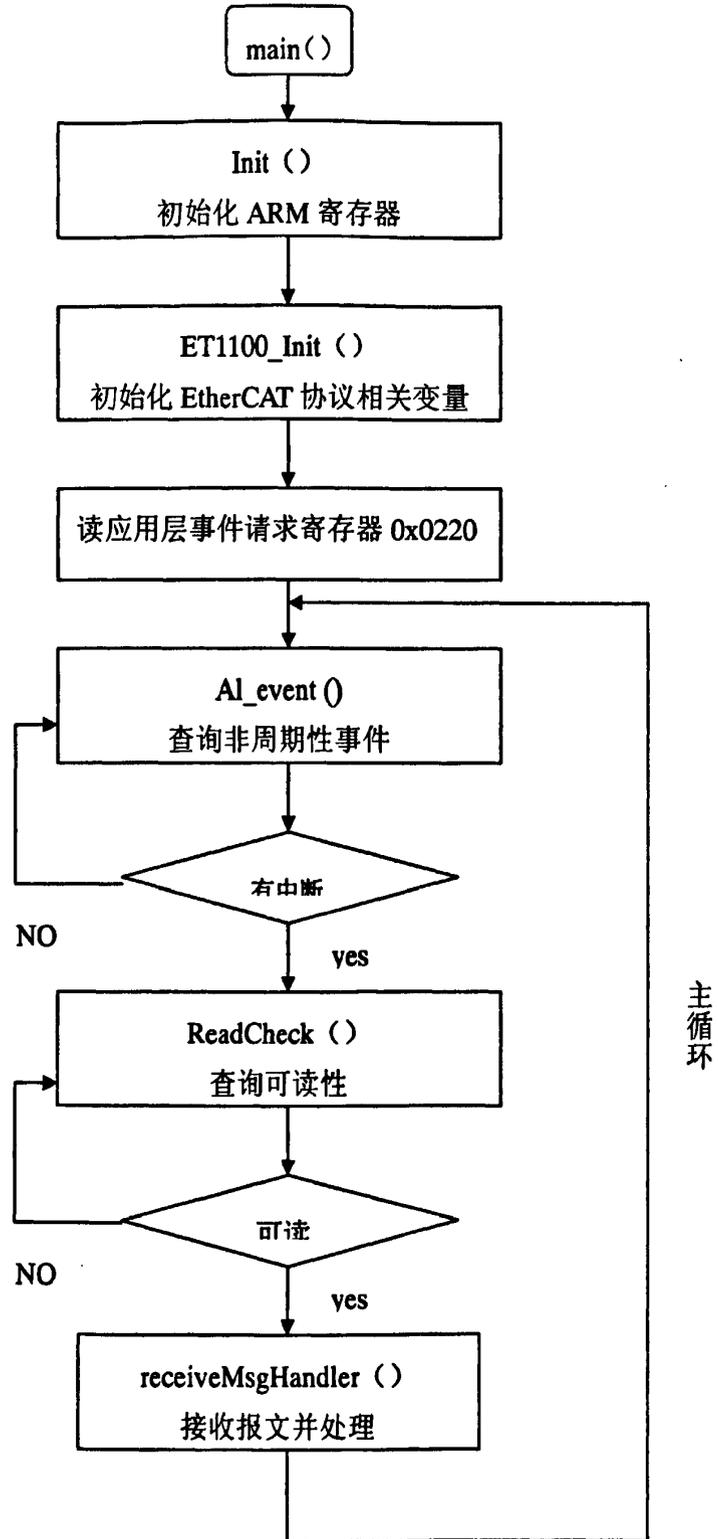


图 3-4 从站程序整体流程图

ESC 通信寄存器由主站进行配置，从站程序只需要从中读取有效数据即可，相关基本寄存器如表 3-5。

表 3-5 从站操作相关的基本寄存器

编号	地址(长度)	名称	读/写	操作
1	0x0120 (2)	应用层状态控制	读	读取主站发出的状态改变指令
2	0x0130 (6)	应用层状态即状态码	写	返回从站实际状态及状态码
3	0x0204 (4)	应用层事件中断屏蔽	写	设置事件触发中断信号
4	0x0220 (4)	应用事件请求	读	运行中查询发生的事件
5	0x0800(32)	SM 配置数据	读	读取 SM 通道内存中的起始地址和长度
6	0x098E	SYNCO 信号状态	读	读取此寄存器响应 SYNCO 中断

### 3.2.2 EtherCAT 从站程序基本操作函数

#### 1、LPC1754 初始化函数 SPI\_LPC1754 ()

##### (1) 函数功能

- ①初始化 LPC1754 引脚以及相关寄存器，配置 SPI 通信相关参数；
- ②初始化 DA 芯片 (AD5663)，向其发送相关配置字；

##### (2) 函数功能说明

①配置 SPI 与 ET1100 相连，LPC1754 为 SPI 的主机，设置单次发送数据的位数为 16 位，数据在 SCK 的第一个时钟沿采样，SCK 为高有效，SPI 数据传输 MSB 且 SPI 中断使能。

②配置 SPI0 与 DA 芯片 (AD5663) 相连，LPC1754 为 SPI0 的主机，设置单次发送数据的位数为 12 位，数据分两次发送，数据在 SCK 的第一个时钟沿采样，SCK 为高有效，SPI 数据传输 MSB 且 SPI 中断使能。编写 LPC1754 向 AD5663 发送数据的函数 SPIDA\_SendData (unsigned long data)，入口参数 data 为向 AD5663 发送的配置字或者 16 位数字量，初始化时 data=0x00300003，配置

AD5663 工作模式为在新数据输入的第 24 个时钟沿自动将数据更新到 DAC 寄存器完成数模转换。

## 2、ESC 初始化函数 ET1100\_Init ( )

### (1) 函数功能

① LPC1754 向 ET1100 通过寻址读取以及写入数据的函数，从 ET1100 读取相关寄存器状态；

② 初始化同步管理器 SM 数目以及相关参量和邮箱相关参量。

### (2) 函数功能说明

① AddressingEsc( unsigned int address, unsigned short char Command ) 是寻址函数，address 是要寻址的地址，Command 表示读/写命令。通过函数 EscReadAccess( unsigned int \*pData, unsigned int Address, unsigned int Len ) 读取 ET1100 相关寄存器的状态并保存。

② 读取 ET1100 寄存器 0x0800 得到有关 SM 的配置数据并保存，根据数据初始化邮箱通信的相关参量。

## 3、设置应用层状态函数 setAlStatus( )

### (1) 函数功能

函数用于设置从站 AL 状态寄存器。

### (2) 函数功能说明

一旦从站接受到主站发送的 NMT 从站状态转换报文，便调用此函数来设置 ET1100 中的 AL 状态寄存器。从站处理器芯片 LPC1754 读取 AL 状态寄存器中的数据后将从站的状态切换到所对应的状态。

## 3.3 EtherCAT 从站非周期性事件的处理

从站非周期性事件主要是邮箱通信事件。在程序主函数的主循环中读取应用层事件请求寄存器 0x220 以查询事件的发生。AL\_event() 的代码结构图如图 3-6。

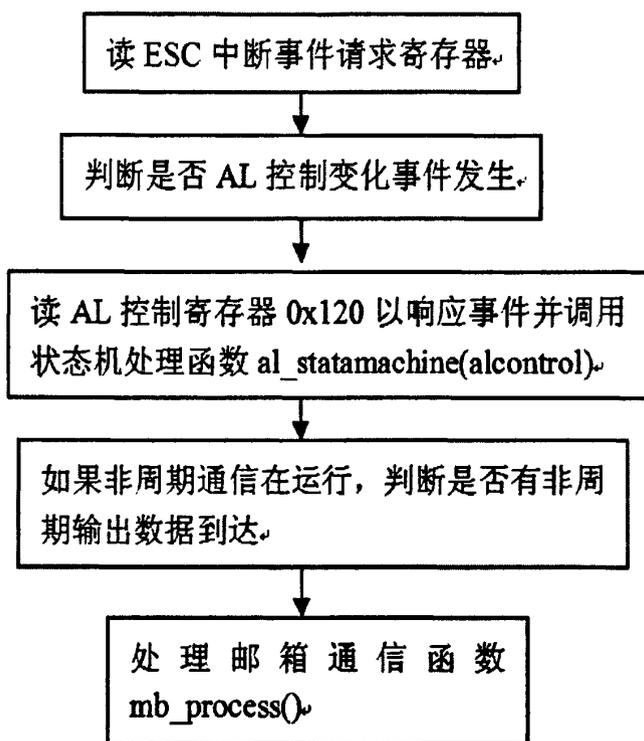


图 3-6 AL\_event() 的代码结构

### 3.3.1 初始化 CANOpen 节点(CAN\_Init())

#### (1) 函数功能

初始化对象字典，包括本节点 ID，接收 PDO 数目，输出模拟量补偿，缩放比例，以及输出模拟量的范围。

#### (2) 函数功能描述

在程序设计中，定义了对象字典的结构体<sup>[23]</sup>如下：

```

typedef struct td_subindex
{
    enum e_accessAttribute  bAccessType; // 读写属性(RO, RW, ...)
    UNS8                    bDataType;  // 对象的数据类型
    UNS8                    size;       // 变量的大小（字节位为单位）
    void*                   pObj;       // 指向变量的指针
} subindex;
    
```

包括该对象字典的读写属性，数据类型，数据大小以及数据的入口地址，

这些是在访问对象字典时最常用到的属性<sup>[22]</sup>。

对于各种通讯对象 ID 的设置采用的是功能码加上节点 ID 的方法，例如从节点使用的 ID 为 5，则初始化接收 PDO(RPDO) 对象字典代码为：

```
“RxPDO.cob_id=0x300+Node_ID;”
```

程序代码中 RPDO 的功能码为 0x300，Node\_ID 为节点 ID 值，其值为 5，由上面接收 PDO 的 COB\_ID 计算式子可以知道 RPDO 报文的 ID 为 0x305H。程序没有采取直接赋值 0x305H，这样做的好处是如果节点 ID 改变，各种通讯对象的 ID 可以随之变化，而不用对每个通讯对象进行一次重新的赋值操作，尤其是在从节点比较多的系统中，只需修改每个节点所对应的 Node\_ID，就可以配置好本节点所有通讯对象的 COB\_ID。

### 3.3.2 初始化 RPDO(CAN\_InitRPDO())

#### (1) 函数功能

①检查 RPDO 所允许的最大范围并且检查节点 ID 的范围是否为 1-127，以及检查 PROD 映射到对象字典过程中映射数组的偏移量(offset)有没有超过映射数组的最大范围。

②初始化通讯子索引(通讯参数的个数)的数目、COB-ID 和传输类型。

#### (2) 函数说明

对 RPDO 的参数(通信参数，映射参数)进行检查是为了保证数据通信的正确性，一旦用户参数设置不正确，比如出现对象的数目超出了程序中设定的范围，就会向用户发出错误警告。而通讯子索引数目的设置则是为了符合协议中的要求，以满足用户查询 PDO 中使用的参数个数。对于 PDO 的 ID 设置，程序中用了前面所介绍的报文功能码加上节点 ID 的方法，此种方法要求 TPDO 和 RPDO 的个数不多于 4 个，因为本课题中用到的 RPDO 的数目为 1，故可以采用上述方法。

RPDO 接收的是设备需要执行的命令或数据，虽然从字面意思上看传输类型的定义更符合 TPDO，但 RPDO 也要根据传输类型为同步或异步进行相应的动作。传输类型为 254、255 的异步 RPDO 在接收到新数据后立即执行新数据的内容。本课题中使用的通信方式是非周期邮箱通信，故采用异步触发中的事件触发 255，该事件是来自于应用层的中断。

### 3.3.3 CoE 报文的接收处理

在 CANopen 设备初始化状态结束后，设备进入了可以由用户控制执行的运行状态 (Operational)，即 CANopen 协议栈定义的各个过程函数。在这一部分中，CANopen 协议栈各个执行函数要在主函数中循环运行，以处理各种命令和数据。

#### 3.3.3.1 对接收到的报文进行处理(receiveMsgHandler())

##### (1) 函数功能

节点进入预运行状态后，用户和设备可以给其它节点发送数据或者命令，一个节点可以配置为主节点，其它节点配置为从节点。报文的格式为 CoE 数据帧。当从节点接收到主节点发送过来的数据帧后，首先判断报文的类型，然后对接收到的报文进行处理。如图 3-7 所示。

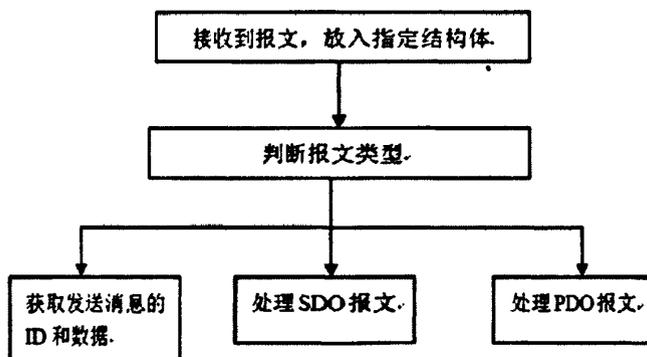


图 3-7 处理接收报文流程

(1) 首先将接收到的 CoE 报文放到指定结构体中；

(2) 判断如果本节点为主节点，则获取发送消息的从节点的功能 ID、节点 ID 以及数据长度，将数据保存在指定的数组中；

(3) 如果收到的是 SDO 报文，判断节点是否处于预运行或运行状态，是则处理 SDO 报文请求；

(4) 如果收到的是 PDO 报文，判断节点是否处于运行状态，然后根据传输类型(255，此时为异步)和 ID 找到对应的 RPDO，再根据对象字典中 RPDO 的映射参数将接收的报文中的数据复制到对应 RPDO 的过程数据映射数组中。

##### (2) 函数功能说明

### (1) 函数介绍

对接收到的报文的处理函数是 CANopen 从栈协议栈中一个主要的函数，函数对每一个节点收到的数据或者命令都进行处理。当设备进入预运行状态后，除了 PDO 报文不能够被执行，其他类型的报文（NMT, SDO 等）都可以在设备中执行，但在此状态下可以分配 PDO 的映射。函数定义了存放 CoE 报文的结构体<sup>[22] [23]</sup>，格式为：

```
typedef struct _MESSAGE
{
    unsigned int Length;           //后续数据长度
    unsigned int Address;         //数据源地址
    unsigned int Form;           //报文类型
    unsigned int coecmd;         //coe 命令
    unsigned char DATA[8];      //帧数据
}Message;
```

这是标准的 CoE 报文格式。

### (2) PDO 报文的处理

对于传输过程数据的 PDO 报文，在预运行状态下不能通讯，在对 PDO 的配置和设备参数配置好后(如果需要此时进行)，在运行状态，所有类型的对象都可以进行数据交互。PDO 的传输有同步传输和异步传输，同步传输实在同步时间对象的控制下进行的，异步传输往往由事件触发。在此函数中运用异步 PDO 传输，因为同步传输实时性要求比较高，在接收到同步对象后立即执行，所以在同步 PDO 的操作一般用中断子程序中实现，PDO 报文的处理是由函数 proceedPDO() 完成的。proceedPDO() 的函数结构如图 3-8。

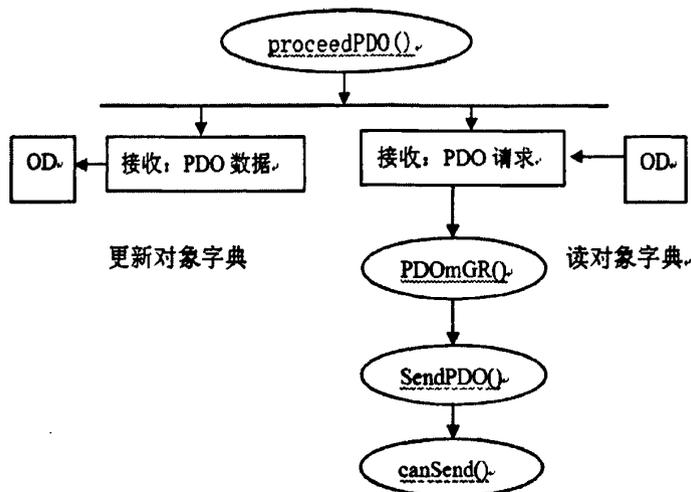


图 3-8 PDO 处理函数结构图

(3) SDO 报文的处理

SDO (Service Data Object, 服务数据对象) 主要用于更改或获取对象字典中的数据或参数, 也可在初始化时传输配置文件。当从站接收到一个 SDO, 首先根据 SDO 的 Command Specifier 判断是上传还是下载“Upload/Download”, 然后获取要读写的索引和子索引值、数据大小、传输方式等。如果是 Upload, 读 OD 相应项, 将数据或参数写入 SDO 数据项中并发送; 如果是 Download, 写 OD 相应项, 并发送一个 SDO 告知主站是否成功。SDO 处理函数结构图如图 3-9。

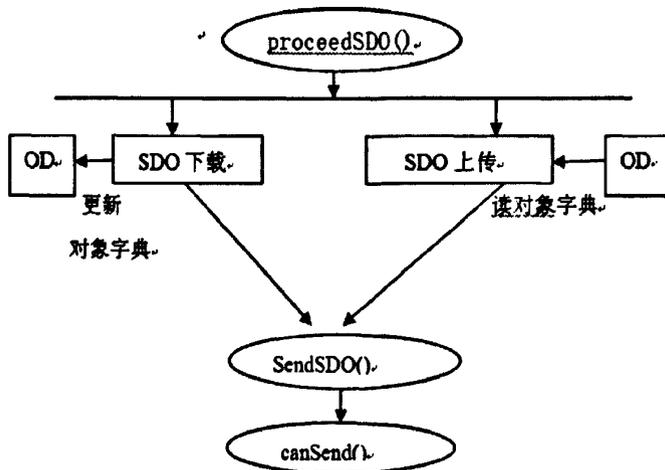


图 3-9 SDO 处理函数结构图

### 3.3.3.2 中断函数的设计

LPC1754 在每次接收 CoE 报文时,是由中断触发接收的,另外数字量输出至 AD5663 芯片也是有中断(同步信号中断)触发的。中断函数的流程图如图 3-10 所示<sup>[12]</sup>。

#### (1) 中断子程序具体功能

①检查邮箱中是否有新的 RPDO 报文,如果有则将接收报文并将其放到对应结构体中,并调用报文处理函数将数据提取出来放到过程映射数组中,等待同步信号中断激发;

②如果没有新的 RPDO 报文,则判断上一次收到的 RPDO 报文数据是否已执行,如果已执行,中断返回;

③若未执行,则判断是否有同步信号中断,若有则执行收到的 RPDO 的内容,若没有则中断返回。

#### (2) 同步机制的说明

同步激发通过同步 RPDO 输出实现:当收到一个同步对象后,立即处理收到的 RPDO 报文,提取数据内容,然后等待同步信号以激发数据。设计中采用的是从站控制器 ESC 向 LPC1754 发送一组 RPDO 数据,定义了一个专门接收同步 RPDO 的结构体,在进行同步激发前,将数据提取出来放到映射数组中,LPC1754 在接收带一个同步输出信号中断时激发数据,输出至 AD5663,完成数模转换。

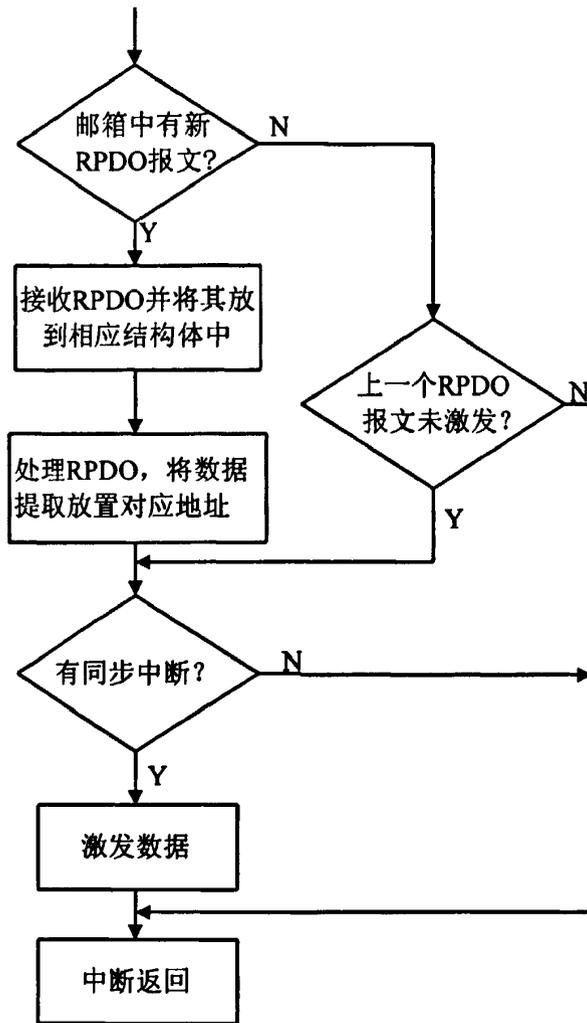


图 3-10 接收同步报文中断子程序

(1) 查询邮箱是否可以读取报文 (ReadCheck())

函数功能: 所有报文的接收都是通过查询邮箱是否有新数据的方法实现的, 如果有新的数据并且可以读取新消息时, 则进行处理。每次处理完一个邮箱的消息后, 需要清除这个邮箱的接收消息标志位, 以便下次查询。程序中定义了一个标志邮箱中是否有新数据并且是否可读的全局变量和标志报文处理的全局变量, 若判断邮箱中有新消息并且可读, 则标志位置 1, 激发 LPC1754 接收新消息, CANopen 协议的处理程序处理完一个报文后, 标志置 1, LPC1754 根据这个标志来判断消息是否被处理以便在同步信号到来之际是否激发数据。

(2) 查询是否有同步信号 (Sync0Check())

函数功能：所有报文的数据都是由同步信号触发输出的，程序中定义了一个标志是否有同步信号到来的全局变量，若判断消息已经被处理，并且有同步信号，则激发数据输出至 AD5663 完成数模转换。

### 3.4 本章小结

从站控制器是实现从站微处理器与主站相连的器件，而从站微处理器是实现控制与通信的最重要的部分，设计符合功能要求的高实时性、高可靠性，以及具有可移植性的 CANopen 从站协议栈对于 CANopen 的组网应用起着尤为重要作用。本章根据 CANopen 协议设计了从站微处理器 LPC1754 的驱动程序。详细介绍了从站协议栈的分类结构设计和 CANopen 协议中对象字典的设计与实现，分析了通信实现的各个功能模块和从站节点的中断处理。

## 第 4 章 模块测试

一个 EtherCAT 网络系统的原型起码应支持测试及验证。此外，另一个重点是协议原型里的各个单独组件可以被真实的节点取代。利用这种方式，在系统开发的过程中，就可以比较容易地测试各个节点的功能及性能。本章介绍协议程序实现的效果，在搭建的 EtherCAT 网络上进行协议软件的运行，上位机主节点发送各种命令和数据，从节点根据协议软件处理和输出数据。

### 4.1 从站通信测试

结合主站的控制界面以及超级终端，程序运行平台 TKstudio 上对上一章设计的 CANopen 从站程序进行了测试。

图 4-1 是主节点的控制界面，从节点当前的工作状态处于运行状态，因为在运行状态下既可以进行 SDO 通信，也可以进行 PDO 通信。也可以通过点击相应的状态切换按钮来进行从节点的状态切换控制。在图中可以看到对象字典的索引，子索引，数据长度以及值。配置或者访问对象字典时，利用 SDO 服务。传输实时数据时，进行 PDO 数据通信。

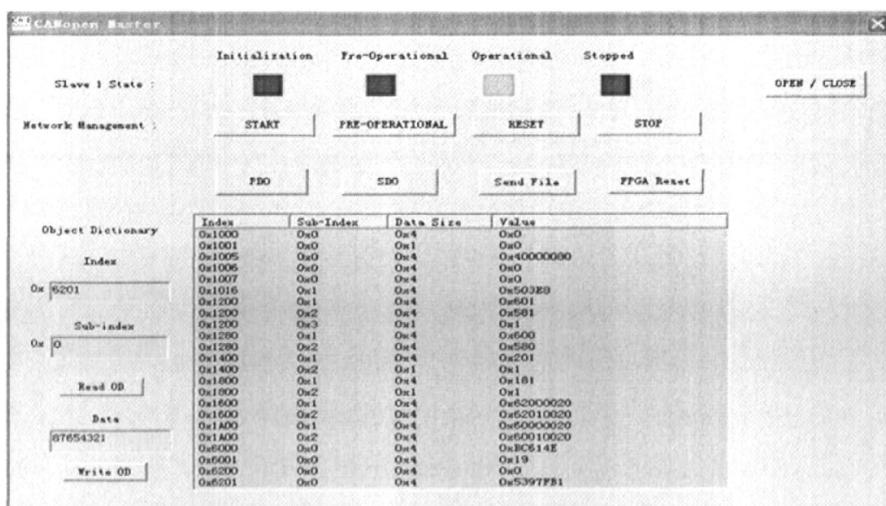


图 4-1 主控制器 Master 界面

PDO 测试中，主节点向从节点发送 PDO，从节点 Node-ID 为 5。PDO Number 为 1。

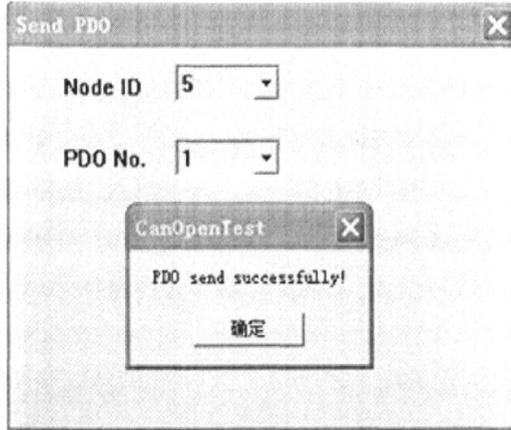


图 4-2 PDO 发送

图 4-3 所示，测试中用到的从节点 Node-ID 为 5，每 2 个时间戳发送一个心跳报文。当接收到一个 PDO，报文中的 COB-ID 首先被识别，然后从对象字典中提取对应的映射参数，PDO 数据最后存储到对象字典的索引 0X6200 中。

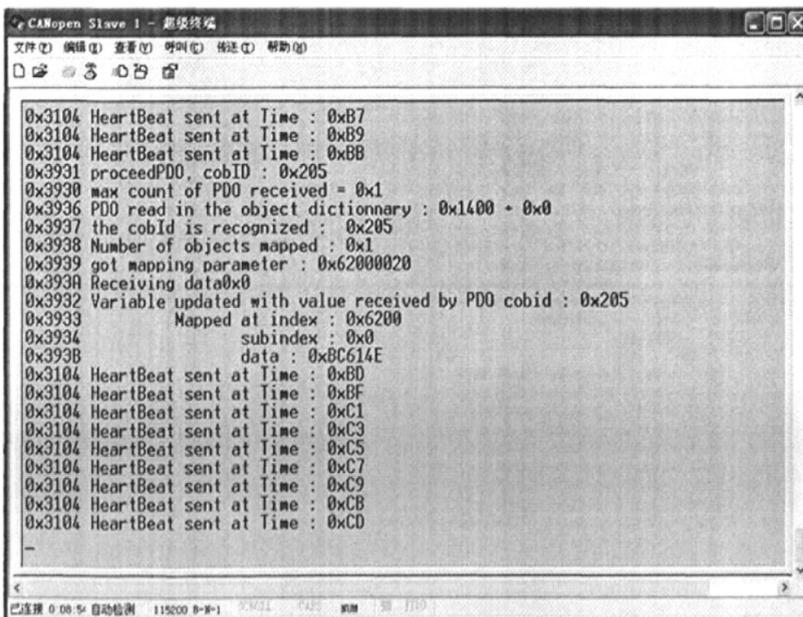


图 4-3 PDO 处理

图 4-4 为 SDO 服务。对从节点 5 的对象字典索引 0X6201 进行访问。

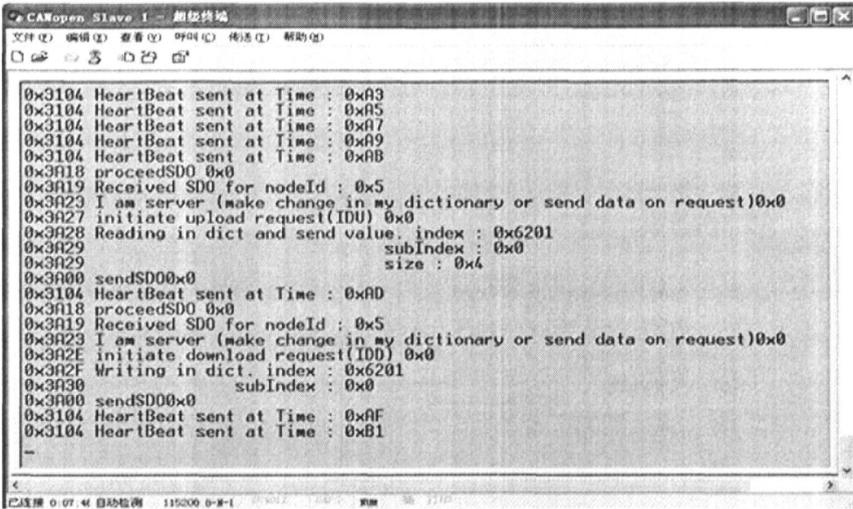


图 4-4 SDO 处理

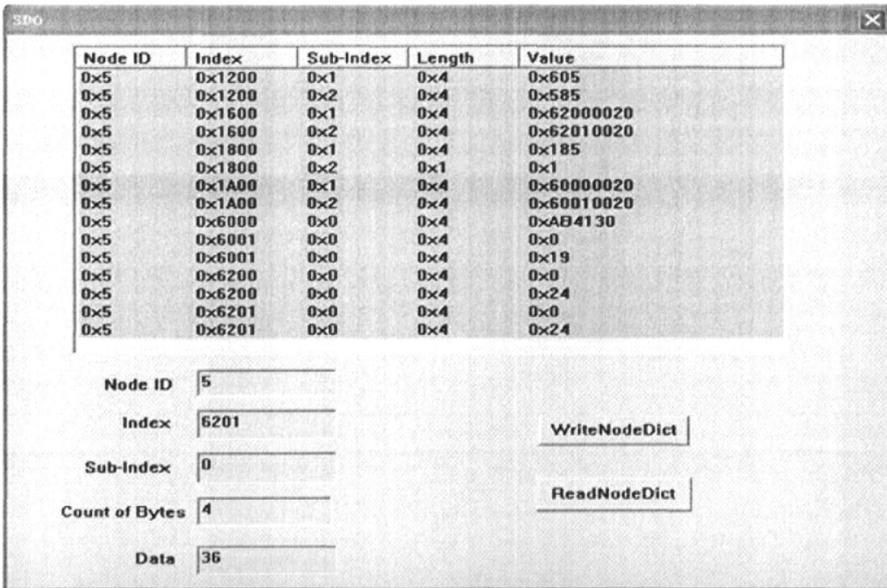


图 4-5 SDO 服务

图 4-5 为从节点 5 的相关对象字典内容。

从上面的测试结果中可以看到，所设计的从站软件可以简单的实现 SDO 以及 PDO 数据通信功能，可以正确地对对象字典进行读写。

## 4.2 利用 TwinCAT 软件测试 DA 模块

### 4.2.1 对象字典的测试

在装有 TwinCAT 软件的上位机上我们可以读取从站 LPC1754 的对象字典如图 4-6 所示。

Index	Name	Flags	Value
1000	Device type	RO	0x00000191 (401)
1008	Device name	RO	DAC
1009	Hardware version	RO	01
100A	Software version	RO	01
- 1011:0	Restore default parameters	RO	> 1 <
1011:01	SubIndex 001	RW	0x00000000 (0)
- 1018:0	Identity	RO	> 4 <
1018:01	Vendor ID	RO	0x00000002 (2)
1018:02	Product code	RO	0x0FA43052 (262418514)
1018:03	Revision	RO	0x00100000 (1048576)
1018:04	Serial number	RO	0x00000000 (0)
- 1601:0	RxPDO-Map OutputsCh.2	RO	> 1 <
1601:01	SubIndex 001	RO	0x6411:01, 16
- 1C00:0	Sync manager type	RO	> 1 <
1C00:01	SubIndex 001	RO	0x01 (1)

图 4-6 从站 LPC1754 的对象字典

通过上图，我们很清晰的看到从站定义的各个对象字典的索引、子索引、名称、读写属性以及对应数值。这样任何熟悉 CANopen 协议的工作人员便可以对整个 CANopen 从站有了一个很清晰的了解，具有很强的通用性。另外，我们还可以根据我们的需要将某些对象字典的值改成我们想要的值，这样会更加容易满足不同用户的需求。

### 4.2.2 数据解析的正确性验证

本文进行了三组实验，检测 CANopen 节点对 CoE 报文的处理正确性。具体步骤为：

- (1) 分别将 1V, 5V, 9V 的直流电经 AD 采样过后，发给上位机；
- (2) 上位机将数据在 TwinCAT 环境下封装之后通过 EtherCAT 发给 ET1100；
- (3) 由 ET1100 将 CoE 帧提取出来发给 LPC1754；
- (4) LPC1754 对 CoE 帧处理后提取数字量接到由 TwinCAT 软件控制的端口上，检测解析数据是否正确。

实验结果见图 4-7, 4-8, 4-9。

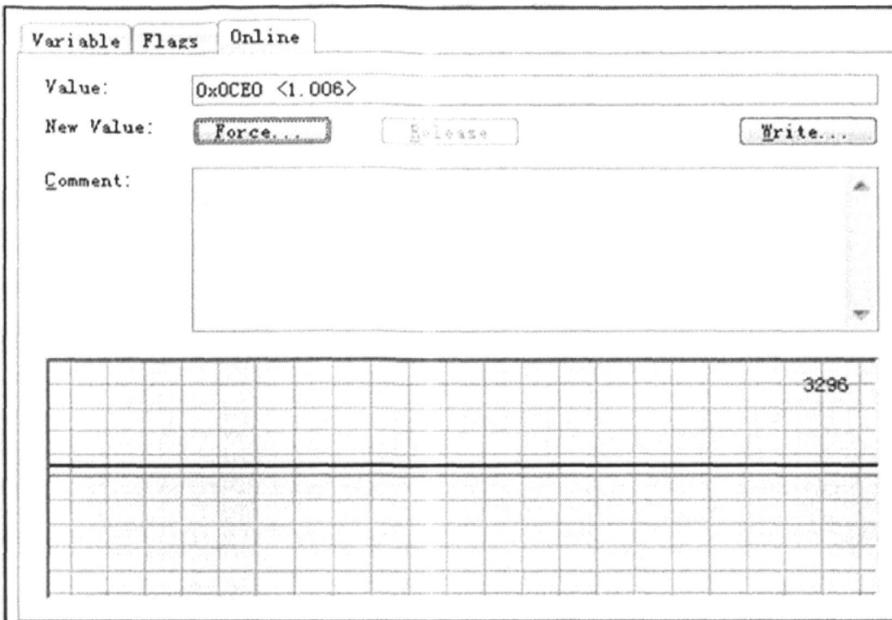


图 4-7 1V 直流电采样值经解析后测试数据

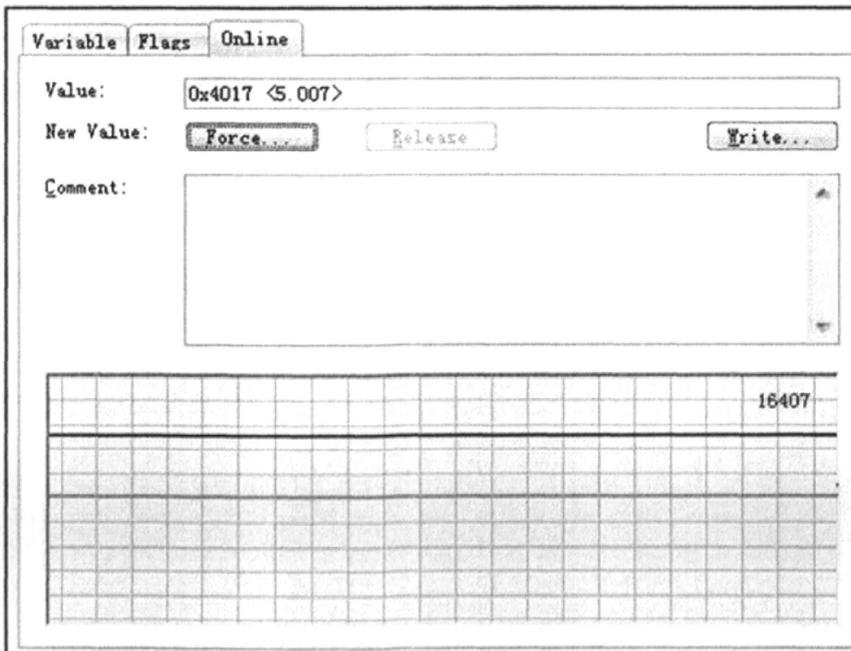


图 4-8 5V 直流电采样值经解析后测试数据

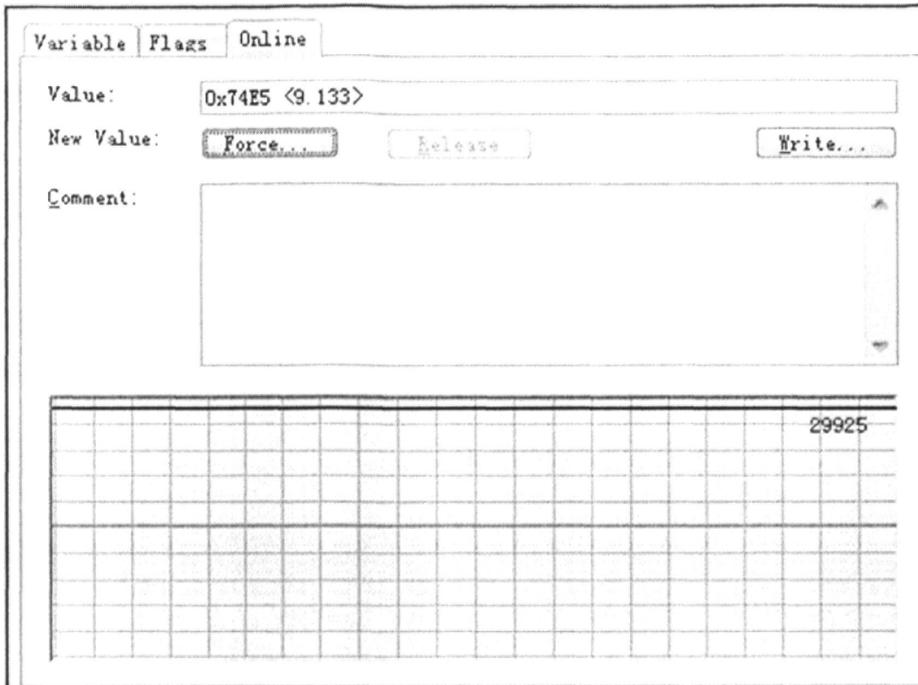


图 4-9 9V 直流电采样值经解析后测试数据

由以上三幅图可以看出，LPC1754 对 CoE 报文的解析是完全正确的，以上图形中的偏差产生是由 AD 采样时产生的。

### 4.2.3 DA 转换精度的测试

上个实验，我们已经验证了 LPC1754 能够对 CoE 报文进行正确的解析。除此之外，DA 转换的精度也是本课题的一个重要指标。因此，本文又对 DA 转换的精度进行测试。同样，本文进行三次实验，分别测试 DA 的转换精度。实验步骤如下：

- (1) 分别将峰值为 8V，频率为 50Hz、500Hz 和 1kHz 的正弦电压由 AD 采样；
- (2) 将采样值发给 TwinCAT 软件，TwinCAT 将数据封装成 EtherCAT 帧发给 ET1100；
- (3) ET1100 将 CoE 帧从 EtherCAT 帧中抽取出来，发给 LPC1754；
- (4) LPC1754 再对 CoE 帧进行解析，将数字量提取出来，按 CiA DS401 协议处理之后发给 DA 转换器；
- (5) 将 DA 转换后的模拟电压接到示波器上观察转换效果。

实验结果分别见图 4-10, 4-11, 4-12。

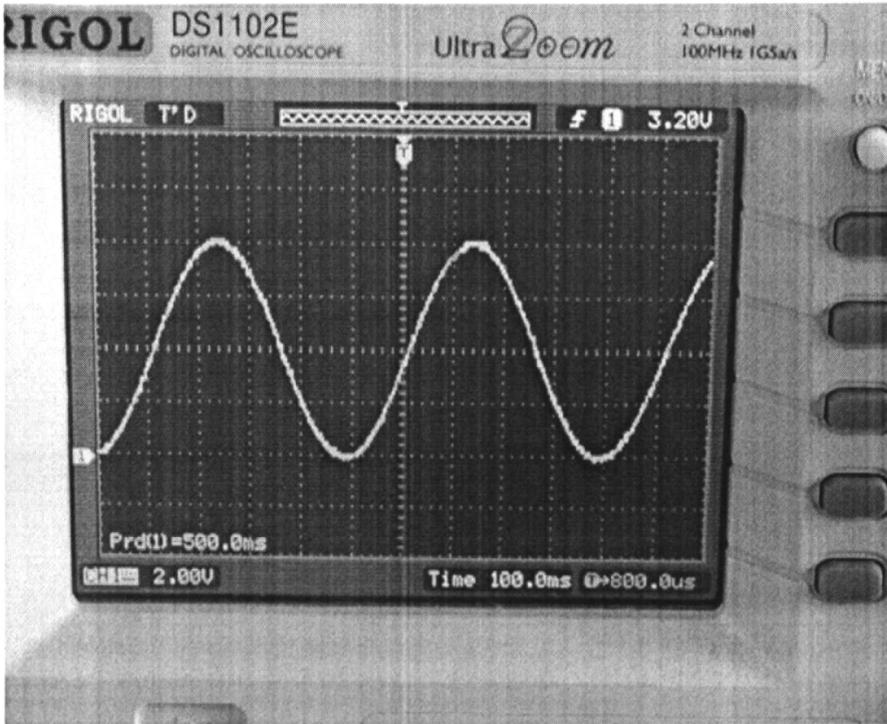


图 4-10 50Hz 正弦电压 DA 转换结果

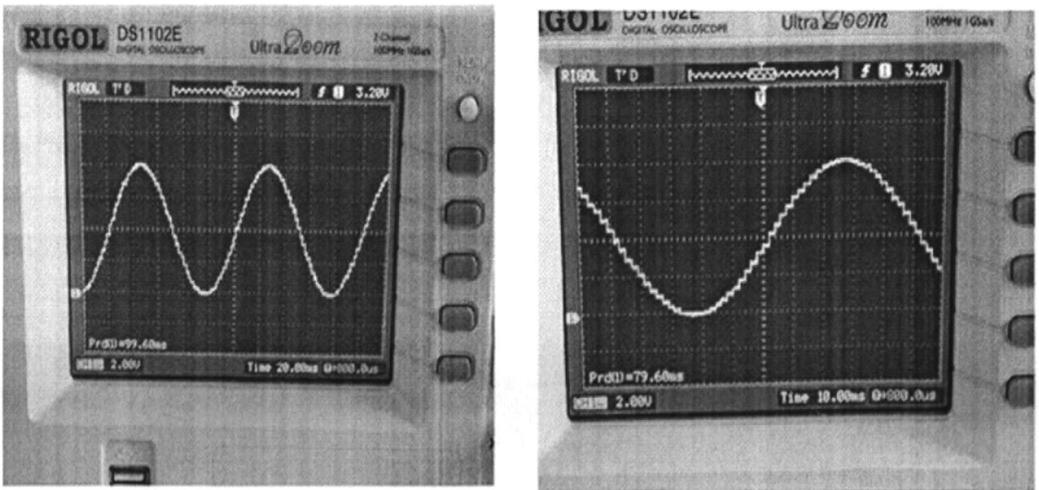


图 4-11 500Hz 正弦电压 DA 转换结果

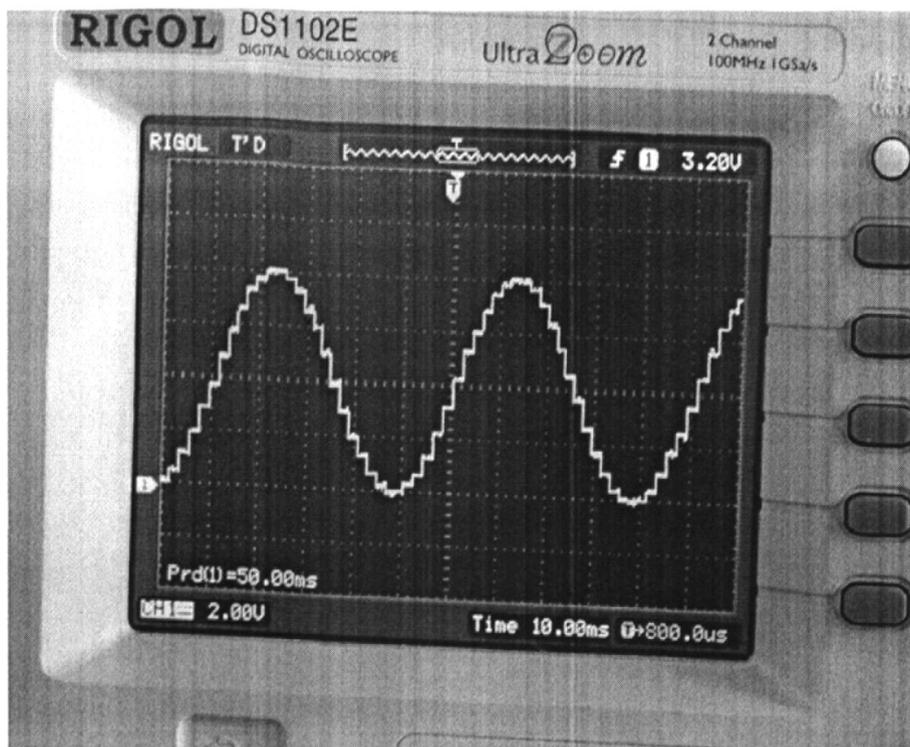


图 4-12 1kHz 正弦电压 DA 转换结果

通过以上三幅图，我们可以看出：转换电压峰峰值基本和 8V 相等，波形基本接近正弦波，再次验证了解析的结果是正确的。

- (1) 50Hz 正弦电压经 AD 采样后再有 DA 转换成模拟量输出，其失真很小，这是因为 AD 采样频率和 DA 的转换频率远远大于 50Hz，外界虽然有一些干扰，但对 50Hz 的电压影响不大；
- (2) 500Hz 正弦电压经 AD 采样后再有 DA 转换成模拟量输出，有一定的失真，由 4-11 右图放大之后可以看到明显的阶梯痕迹，这说明 AD 采样频率和 DA 转换频率并不是远远大于 500Hz，再加上外界的干扰，造成其转换后结果如图 4-11 所示。但由左图可以看出，其失真还不是很大；
- (3) 1kHz 正弦电压经 AD 采样后再有 DA 转换成模拟量输出，其失真较大，由图 4-12 明显地可以看到阶梯状的电压波形，这说明 AD 采样频率和 DA 转换频率并不是远远大于 1kHz，可以说已不足以用到工业生产中去了，再加上外界的干扰，造成其转换后结果失真较大。

综上，通过 TwinCAT 监控从站的实现，验证了本文设计的 EtherCAT 从站能够正常运行，并且 CANopen 从站协议栈的运行性能稳定且可靠，具有良好的通

用性，此外 DA 转换的结果具有较好的精度，整个从站系统基本符合设计要求，初步达到了工业控制系统的要求。尽管目前该系统还有待完善，但是通过进一步的测试和改进设计，将完全能达到工业应用的要求，是一套可行的 DA 转换系统。

### 4.3 本章小结

本章主要对所设计的 LPC1754 驱动程序做了测试。测试结果表明所设计的驱动程序能在 LPC1754 上面正常运行，并且能够对 CoE 格式的数据帧进行处理。所设计的 LPC1754 芯片能够正确实现主站与从站之间的数据交换功能。利用 TwinCAT 软件检测从站系统的对象字典无误，并对其报文解析过程进行了测试发现完全正确。利用示波器检测了 DA 转换结果的精度并分析其结果。利用以上实验检验设计的从站微处理器驱动程序是否达到设计要求，运行是否稳定、准确、可靠。

## 第5章 总结与展望

### 5.1 文章总结

本文调研了 EtherCAT 这种实时工业以太网和 CANopen 协议的研究与应用现状, 在深入研究基于现场总线的网络控制系统、EtherCAT 网络和 CANopen 协议原理的基础上, 构建了基于 CANopen 协议的 EtherCAT 网络控制系统, 重点设计和实现了基于 EtherCAT 网络的从站微处理器, 该从站运行稳定、准确、可靠、实时性高。

论文主要完成的工作如下:

1、针对数据传输与应用, 选择先进的 EtherCAT 工业以太网及其成熟的应用层协议 CANopen over EtherCAT 协议在工业控制中的应用作为研究方向, 建立了基于 CANopen 协议的 EtherCAT 网络控制系统的基本框架, 充分保证了系统的兼容性, 很好地推动了 CANopen 协议在工业以太网中的应用, 具有重要的现实意义。

2、深入研究 EtherCAT 网络协议及 CANopen 协议原理。本文首先研究了 EtherCAT 网络的原理, 而且在研究理论的基础上在 ARM7 单片机 LPC1754 上实现从站节点通讯。这部分的实际开发工作是在执行机构 EtherCAT 从站硬件平台上进行的, 熟悉了从站控制器 ET1100、设计了微控制器 LPC1754 周围的硬件电路以及 SPI 通信方式的使用, 因此为实现应用层协议的实际开发打下了基础。其次, 深入研究了 CANopen 协议原理, 主要研究的是 CiA DS301 和 CiA DS401 协议, 前者是实现 CANopen 通讯的基础, 后者是 GPIO (通用 I/O 口) 的标准规范, 为从站驱动程序的开发奠定了基础。

3、EtherCAT 从站协议栈的设计与实现。本文的研究重点和难点就是着重编程实现 EtherCAT 从站协议栈, 实现网络通讯。首先设计和实现了从站的驱动程序, 然后实现了 CANopen 的核心部分从站协议栈的对象字典, 又分别实现通信的各个功能模块, 重点是开发 PDO 报文的封装与解析。另外, 在分析 CANopen 从站的特点基础上, 研究了 CANopen 从站的启动过程。具有可移植性、高可靠性、高稳定性和高实时性的 EtherCAT 从站协议栈对于工业应用要求较高的场合

具有重要的意义。

4、对所设计的LPC1754芯片以及驱动程序进行了测试。本文使用了TwinCAT软件对EtherCAT从站进行监控和管理，在Windows XP操作系统上测试运行，实现DA转换的功能，利用测试到的结果对所设计的驱动程序进行了分析。

## 5.2 未来工作展望

本文着重设计了EtherCAT从站微处理器驱动程序以及从站微处理器硬件的开发，整个EtherCAT从站系统中所用到的通信协议为CANopen协议，由于CANopen协议较为复杂，而且开发时间紧、任务重，在上面的工作成果中尚有不少问题有待进一步研究和探讨，现总结如下：

1、CANopen网络管理方面。作为网络中的一个节点，其运行状态控制和错误检测机制是很重要的一个方面，是作为主站进行网络管理的部分之一。但是对于从站错误的反馈，以及从站状态机方面没有进行深入研究。

2、CANopen通讯方面。由于SDO通讯复杂，有加速传送和分段传送以及块传送，故对SDO没有深入研究。由于本文重点在于实时PDO数据的获取和传输，对于通过配置管理配置网络中从站的对象字典以及本地对象字典的管理方面没有深入研究。

3、硬件方面。由于16位高精度的AD转换器对干扰的敏感度较高，所以在电路板设计中，EMC、EMI方面的内容显得尤为重要，但是这方面所涉及的内容较多，只是研究了一些常涉及的方面。

虽然本次设计存在许多不足之处有待改进，但是它为后续的开发奠定了很好的基础，有一定的实际意义，通过实验证明了本文提出的设想对于将现场总线技术与工业以太网进行结合可以取得良好的控制效果，进一步促进了CANopen通讯方式在工业上的应用。

具备以太网物理层的高速现场总线将是未来的发展趋势，同时微处理技术的发展使得工业自动化技术的各个应用领域如PLC、HMI、MC、CNC等整合在一起，即All-In-One模式，这样一个完善的网络化控制系统在未来的工业生产中将起到极大的作用。主机将通过EtherCAT网络连接着驱动（伺服、变频、液压等）、通用IO（DA、AD、DIO、PWM等），网关（CAN、Profibus、Devicenet等）等应用部分，并对其进行控制，这将是未来工业控制的一个趋势，而本课题的研究就是基于EtherCAT的DA模块开发，它是通用IO部分的一个子部分，其研

究意义不言而喻。由于其具有良好的可移植性，故其在工业生产中的应用领域比较广而且其未来的应用将会更加广泛。

## 致 谢

在此论文完成之际，谨向所有给予我指导、关心、帮助和鼓励的老师、同学和亲人，致以衷心的感谢！

衷心地感谢刘可文老师，刘老师在我在外做毕设期间给予了许多的指导和帮助，正是这些指导我才顺利的完成此次毕业设计。在生活方面也给了我很多帮助，借此，谨向刘可文老师表示最深切的谢意！

感谢张隽硕士在 CANopen 协议上的指导以及在项目开发和研究中给予的帮助，以及在生活上的关心！感谢刘效祥硕士在驱动程序以及 CANopen 协议方面的指导！感谢胡炯培学士在硬件方面，包括电路原理图以及 PCB 板绘制等方面的帮助与指导！

本论文工作是在中科院顺德工业与信息育成中心完成的，期间得到了公司其他领导和工程技术人员的配合和指导，感谢公司提供了良好的生活和工作环境，感谢所有领导、工作人员的关心和帮助！

感谢刘雨涵等同窗好友，感谢你们对于我学习和生活的帮助，三年里大家共同学习成长，结下了深厚的友谊，真心的感谢你们！

最后，我要感谢含辛茹苦培养我的父母，是你们的关心让我走的更远，是我前进的动力。在我即将离开大学校园，再次感谢所有教育我，关心我，给我支持和前进动力的老师、同学、朋友和亲人们，感谢你们为我做的一切！

## 参考文献

- [1] Olaf Pfeiffer, Andrew Ayre, and Christian Keydel. Embedded Networking with CAN and CANopen. America: RTC Books [M], 2003. Page 1-2
- [2] 刘喆, 郇极, 刘艳强. 基于 XML 的 EtherCAT 工业以太网协议解析技术[J]. 北京航空航天大学学报, 2011/09
- [3] H. Boterenbrood. CANopen: high-level protocol for CAN-bus Version 3.0[S], March 2000.
- [4] 郇极, 刘艳强. 工业以太网现场总线 EtherCAT 驱动程序设计与应用[M]. 北京: 北京航空航天大学出版社, 2010: 3, 6-8, 25-26, 96-97
- [5] 赵飞. 基于 STM32 的 CANopen 运动控制主从站开发[D]. 武汉: 华中科技大学, 2011
- [6] 高大朋. 基于现场总线的电机智能控制器的研究与开发[D]. 北京: 华北电力大学, 2011
- [7] 任作新. 网络化监督与控制系统[M]. 北京: 国防工业出版社, 2007: 285-286
- [8] 饶运涛, 邹湘军, 郑勇芸. 现场总线 CAN 原理与应用技术[M]. 北京: 北京航空航天大学出版社, 2003: 14-20
- [9] 宋晓强. CAN bus 高层协议 CANopen 的研究以及在模块化 CAN 控制器上的实现[D]. 天津: 天津大学, 2008
- [10] 刘思捷. CANopen 协议在伺服系统中的软件实现与植入研究[D]. 武汉: 华中科技大学, 2011
- [11] 田乐. 基于 FPGA 和 CAN 总线的八通道数字频率测量系统的研制[D]. 天津: 天津大学, 2008
- [12] 谢朝文. 基模拟赛车运动控制平台及控制策略研究[D]. 武汉: 武汉理工大学, 2011
- [13] Chen Yongming, Chen Hua, Zhang Mingzhong. The Relevant Research Of CoE Protocol In EtherCAT Industrial Ethernet. IEEE Intelligent Computing and Intelligent Systems, December, 2010, 10(1109) : 67-70
- [14] H. Boterenbrood. CANopen high-level protocol for CAN-bus[S]. Version 3.0 NIKHEF, Amsterdam, 2000: 3-7
- [15] 汪雅楠. 工业以太网 EtherCAT 通信系统的研究与仿真[D]. 北京: 北京交通大学, 2011
- [16] CiA. CANopen Application Layer and Communication Profile (CiA Draft Standard 301)[S]. Version 4.02. February 2002: 29-32, 77-78, 79-80

- [17] CiA. CANopen Device Profile for Generic I/O Modules (DS401) [S]. Version 2. 1. May 2002: 14, 26-27
- [18] 王晓亮. 基于 EtherCAT 协议的网络控制系统研究[D]. 大连: 大连理工大学, 2010
- [19] Mohammad Farsi, Manuel Bernardo Martins Barbosa. CANopen Implementation: applications to industrial networks[M]. England, RESEARCH STUDIES PRESS LTD, 2000: 88-98
- [20] 修建竹. 基于 EtherCAT 网络的实时控制技术研究[D]. 大连: 大连理工大学, 2010
- [21] 向乾亮. 实时以太网 EtherCAT 系统设计及在电力系统中的应用[D]. 北京: 华北电力大学, 2008
- [22] 谢香林. EtherCAT 网络及其伺服运动控制系统研究[D]. 大连: 大连理工大学, 2010
- [23] 孔丽丽. 基于 EtherCAT 波高数据采集系统的研究[D]. 大连: 大连理工大学, 2009
- [24] CAN in Automation Device Profile for Generic I/O Modules[C], 2002.
- [25] 周润景, 袁伟亭. Cadence 高速电路板设计与仿真[M]. 北京: 电子工业出版社, 2007: 1-720
- [26] 吕文红, 郭银景, 唐富华等. 电磁兼容原理及应用教程[M]. 北京: 清华大学出版社, 2008: 214-248
- [27] 张厚林. CANopen 通信协议设计与实现 [D] . 武汉: 华中科技大学. 2009
- [28] 饶怡新. 基于 CANopen 协议的智能电动执行机构监控系统主站的研究 [D]. 广东: 华南理工大学. 2010.
- [29] 喻文冲. 基于 CANopen 总线的列车控制网络系统研究及应用 [D]. 北京: 北京交通大学. 2008.
- [30] 王瑞鹏. 基于 CANopen+uC/OS\_II 平台的主从节点通信 [D]. 天津: 天津大学. 2007.
- [31] 赵斌. 基于 CANopen 的模拟量测量模块 [D]. 天津: 天津大学. 2007.
- [32] 张晓阳. 基于 CANopen 协议的分布式控制系统智能从站设计 [D] . 河北: 河北工业大学. 2010.
- [33] 卢珞先; 陈元. 基于 CANopen 协议的电动车车载网络设计[J]. 武汉理工大学学报, 信息与管理工程版, 2007/12
- [34] Beckoff Automation GmbH EtherCAT Slave Controoller datesheet V1.4. 2008[S].
- [35] EtherCAT technology group. <http://www.ethercat.org>.
- [36] EtherCAT technical Introduction and Overview[S]. EtherCAT technology group-up, 2007.

- [37] CANopen Object Dictionary (Software Manual). SYS TEC electronic GmbH[S], 62
- [38] MAX FELSER. Real-time Ethernet—Industry Prospective[J]. Proceeding of the IEEE. 2005, 1118-1129.
- [39] 叶浩峰. CANopen 总线原理以及实现[D]. 广东, 华南理工大学. 2005.
- [40] CAN in Automation e. v. CANopen-application Layer and communication Profil -e. CIA Draft Standard 301[S].