

文章编号:1671-7848(2010)03-0367-05

基于 CANopen 协议的电动执行机构设计

胥布工, 程 俊, 匡付华

(华南理工大学 自动化科学与工程学院, 广东 广州 510640)



摘 要: 针对常规型电动执行机构的现场接线方式存在着布线工作量大、操作维护困难、可靠性不高、网络通信能力差等方面的问题, 提出了一种基于 CANopen 协议的总线型电动执行机构设计方案。在简要介绍电动执行机构架构的基础上, 首先给出了电动执行机构控制器的硬件设计方法, 其中主要介绍了电动执行机构控制单元电路和 CAN 总线通讯接口电路的设计; 接着给出了软件设计的思路, 重点介绍了对象字典的设计及 CANopen 通讯程序的设计, 针对通讯程序设计中的难点, 提出了一种基于队列方式的报文处理模式。最后, 在实际工程中成功地应用了所提出的设计方案, 应用结果表明这种总线型电动执行机构具有智能化、信息化、数字化等特点, 能够满足工业控制系统的网络化发展需求。

关 键 词: CANopen 协议; 电动执行机构; 对象字典

中图分类号: TP 273

文献标识码: A

Design of Electric Actuator Based on CANopen Protocol

XU Bu-gong, CHENG Jun, KUANG Fu-hua

(College of Automatic Science and Engineering, South China University of Technology, Guangzhou 510640, China)

Abstract: To the problem that the traditional field wiring of conventional electric actuator has certain disadvantages, such as heavy wiring work, operation and maintenance problems, poor reliability and network communication etc., a design scheme of intelligent electric actuator based on CANopen protocol is presented. A brief description of the structure of electric actuator is introduced. The electric actuator controller hardware design methods is presented. The electric actuator control unit circuit and CAN bus communication interface circuit are designed. The idea of software design focusing on the object dictionary and CANopen communication program is proposed. To the difficulty of communication program design, a method based on the queue processing mode is given. The proposed design scheme is applied to an actual project, and the results indicate that electric actuator with fieldbus interface has the characteristics of intellectualization, informationization and digitization, etc, meeting the needs of networking development of industrial control systems.

Key words: CANopen protocol; electric actuator; object dictionary

1 引 言

电动执行机构是工业过程控制系统不可缺少的部分, 被广泛用于石油化工、电厂等领域。近年来, 伴随着微电子技术和控制技术的迅速发展, 电动执行机构也获得了快速的发展, 国外一些生产厂商在这几年中相继推出了常规的、带现场总线通信协议的总线型电动执行机构。而国内起步晚, 目前生产的主要是常规的电动执行机构, 它采取传统的现场接线方式, 如每台电动执行机构至少需要通过 7~14 根控制线才能将其参数传递到控制系统的控制模板上。这种方式安装、维护代价高, 系统的可扩展性差, 且易受现场操作人员的技术水平、精神状态及工作情绪等人为因素的影响, 误操作较多。另外,

对于一些有着特殊要求的环境, 如易燃易爆、辐射、或人们不易到达的地方, 这种传统控制方式难以胜任。为解决上述问题就要求执行机构向数字化、智能化、网络化方向发展^[1]。鉴于此, 本文提出了一种基于 CANopen 协议的电动执行机构设计方案。

2 设计方案概述

CANopen 协议是一种在欧洲非常流行的 CAN 高层协议, 该协议清晰、透明、精炼, 便于系统配置和功能重构, 在工业控制中有着良好的应用前景。本文提出的设计方案是在常规电动执行机构设计中加入 CAN 总线通讯模块^[2], 使其成为一种具备总线通讯能力的智能电动执行机构。这种智能电动执行机构采取总线控制方式, 只需通过 2 根信号

收稿日期: 2009-07-24; 收修定稿日期: 2009-11-30

基金项目: 国家自然科学基金资助项目(U0735003, 07006470)

作者简介: 胥布工(1956-), 男, 江苏盐城人, 教授, 博士生导师, 主要从事时滞和不确定系统的分析和综合以及网络化控制系统理论及应用等方面的教学与科研工作。

线就能完成所有信息的传输,更重要的是由于其具备了总线通讯功能,操作人员只需在控制室就可对执行机构实行远程控制。

3 电动执行机构硬件设计

电动执行机构硬件设计是建立在了解其基本架构的基础之上的,为此,在硬件设计前先简要介绍一下电动执行机构的架构。

1) 电动执行机构的架构 电动执行机构是根据调节器(或上位机)给出的控制信号产生推力或位移的一种装置,主要由控制器、伺服电机、位置检测装置和减速器四部分组成,其构成如图 1 所示。

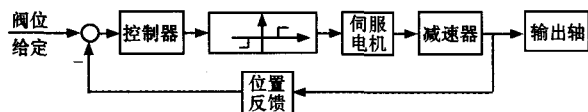


图 1 电动执行机构的架构
Fig. 1 Electric actuator structure

控制器采集阀位给定信号和位置反馈信号,阀位给定信号由上位调节装置经总线系统发送,由现场总线接口接收。减速输出的信号经位置检测装置后形成位置反馈信号送给控制器,控制器输出经过一个采用晶闸管作为开关元件的继电器非线性环节去控制电机的运转,电机带动减速器来驱动输出轴产生相应的位移。

2) 控制器的硬件设计 控制器系统硬件由电动执行机构控制单元和 CAN 总线接口单元组成,如图 2 所示。

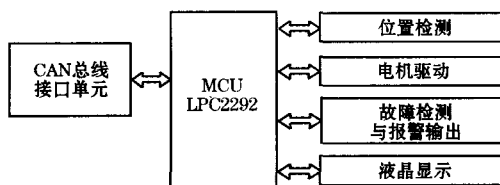


图 2 控制器的硬件结构
Fig. 2 Hardware structure of controller

整个电动执行机构设计采用模块化设计,选用 LPC2292 作为这两个单元模块的中央处理器。LPC2292 是 Philip 公司的内嵌有 CAN 控制器的高性能单片机,工作频率高达 30 MHz,非常适合用于总线控制系统。电动执行机构控制单元含位置检测电路、电机驱动电路、故障检测/报警电路及液晶显示电路。位置检测电路用导电塑料精密旋转电位器作为位置传感器,电机驱动电路由固态继电器和正反互锁逻辑电路组成,完成电机的正反转启动、停止等控制。故障检测/报警输出电路经光电隔离电路采集限位、过热等报警信号,输出报警信息;液晶显示采用 128 * 128, 4 级灰度的点阵液晶屏,液晶界面为现场操作人员提供了良好的操作平台。

CAN 通讯接口单元^[3]是总线型智能电动执行机构硬件上所特有的,它提供了总线访问的接口,其电路,如图 3 所示。

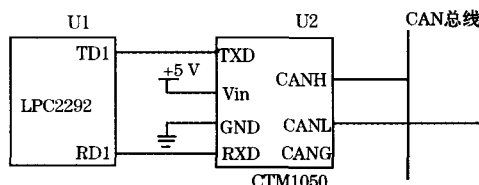


图 3 CAN 通讯接口电路

Fig. 3 CAN communication circuit

电路中收发器模块选用带隔离功能的 CTM1050 芯片,通过它可以很好地实现 CAN 总线上各节点电气、电源之间的完全隔离和独立,极大地提高了 CAN 节点的稳定性和抗干扰性。

4 电动执行机构的软件设计

电动执行机构软件设计主要包括两方面:执行机构的控制程序和基于 CANopen 协议的 CAN 总线通讯程序。其中,控制程序主要实现阀门的位置控制功能,包含初始化、数据采集、处理及显示等几部分,且在常规型电动执行机构中都已实现,不再赘述。本部分将重点介绍基于 CANopen 协议的 CAN 总线通讯程序设计,根据 CANopen 协议描述^[4-5],把这部分程序设计分为对象字典的设计和 CANopen 通讯程序设计两部分,而后者是难点。

1) CANopen 协议通信机理 CAN 协议只定义了物理层和数据链路层两个底层协议,而 CANopen 协议在其基础上规定了应用层协议,其通信模型,如图 4 所示。

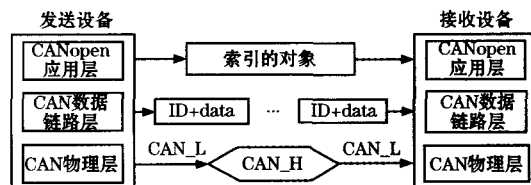


图 4 CANopen 通讯模型

Fig. 4 CANopen communication model

在 CANopen 的应用层,设备间通过相互交换通信对象进行通信。考虑到工业自动化系统中数据流量的不同需要,CANopen 定义了 4 类标准通信对象,分别为过程数据对象(PDO)、服务数据对象(SDO)、管理对象(NMT)和特殊功能对象;其中,PDO 对象主要用于传输实时数据,SDO 对象用于传送配置信息,NMT 对象用于实现对 CANopen 网络的管理,而特殊功能对象提供一些特殊服务。对这些通信对象的访问都是通过对象字典来实现。

2) 对象字典的设计 对象字典(Object Dictionary)是 CANopen 协议的核心,它是一个有序的对象

集合，包含了 CANopen 网络中设备的所有信息，每个设备的对象字典具有结构相同、内容不同的特点。CANopen 设备访问对象字典时，是通过一个 16 位索引和一个 8 位子索引的方式来寻址。由于对象字典包含条目众多，不可能一一列举，下面以使用最频繁的 PDO 对象为例，从其报文属性和数据封装两方面来介绍 PDO 对象的建立。根据电动执行机构的控制需求，定义了 PDO 报文属性，见表 1。

表 1 PDO 报文属性
Table 1 PDO message properties

对象	通信参数索引	COB-ID	传输类型	映射参数索引	映射参数的子索引的值
TPDO1	1800h	181h	255	1A00h	0x60000108(实际阀位) 0x60000208(实际力矩)
RPDO1	1400h	201h	255	1600h	0x62000108(阀位给定) 0x62000208(力矩给定)

对于在 CANopen 网络中作为从节点的电动执行机构来说，TPDO 是其发送给主节点的实时数据，而 RPDO 则是接收来自于主节点发送的数据或命令信息。由表 1 可知，每个 TPDO 和 RPDO 都包含 2 个对象字典条目：通信参数和映射参数。下面以 TPDO1 为例说明通信参数和映射参数的索引、子索引之间的关系，以及 PDO 对象的数据封装过程，详细流程，如图 5 所示。



图 5 映射封装 PDO

Fig. 5 PDO mapping

由图 5 知，通信参数索引 1800h 下有 5 个子索引，分别规定了该 PDO 使用的 COB_ID、传输类型、禁止时间及定时器周期等参数；而映射参数索引 1A00h 下有 2 个子索引，它们表示映射到该 PDO 中的 2 个对象，例如子索引 1 中的存放的是 60000108h，表示把索引 6000h，子索引 01h，长度为 8 位的数据映射到 TPDO1 的第 1 个字节，而

60000108h 里存放的是用户规定的实际阀位值，当满足通讯参数中规定的传输类型时，实际阀位值就能通过 TPDO1 发送给 CANopen 网络主节点，从而实现主从节点之间的信息交互。

3) CANopen 通讯程序设计 电动执行机构的 CANopen 通讯程序流程，如图 6 所示。

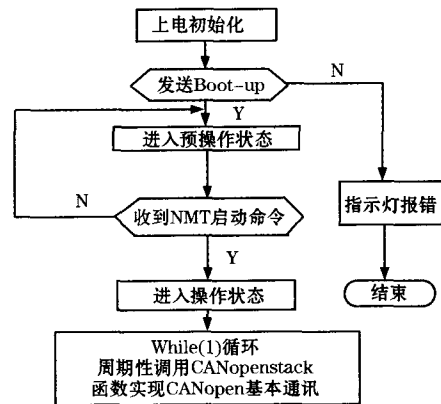


图 6 CANopen 通讯流程图

Fig. 6 Flow chart of CANopen communication

首先，执行机构上电初始化，待初始化完成后，它发送 Boot-up 报文，同时其通信状态由初始化状态切换到预操作状态，该状态下仅能进行 SDO 通信；在收到 NMT 主节点的启动报文后，节点通信状态由预操作切换到操作状态，它是系统的正常工作状态，在该状态下执行机构应用程序循环调用 CANopenstack 函数实现 CANopen 基本通讯。

CANopen 基本通讯主要包含以下三方面：实现 CANopen 状态转换机制、调用 CAN 报文收发函数处理 NMT，PDO，SDO 等报文，根据报文处理结果刷新实际输入和输出。其中，第二个方面是难点，因为 CANopen 规定的通讯报文种类较多，如何快速有效地对这些报文进行分类，并调用相应函数处理是保证系统实时性的关键。为此，在程序设计时采取了一种基于队列的分块存取报文处理模式，其基本思想，如图 7 所示。

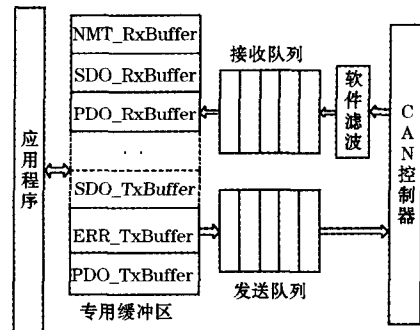


图 7 基于队列的报文处理模式

Fig. 7 Message handling mode based on queue

程序中为在 CANopen 协议里规定的每种报文分配一个专用缓冲区，该缓冲区是应用程序与队列

之间的公用接口；当需要发送报文时，应用程序先把待发送的报文按 CANopen 协议规定封装好，再写入相应缓冲区，然后依次入队，最后启动发送命令，报文出队并发送到总线上。接收报文时采用中断接收方式，为了减轻总线负载，设置了一个软件滤波环节，若收到的 CAN 报文满足软件滤波条件时才放入接收队列中。同时程序在不断扫描接收队列，若其中有报文则执行出队，出队后的报文根据其 COB-ID，写入到对应的专用缓冲区中，应用程序从缓冲区中取出报文并进行后续处理。程序中定义了 2 个队列：接收队列和发送队列，队列采用顺序存储结构的循环队列^[6]。

具体代码如下：

```
#define MAXsize 16 //循环队列长度
typedef struct _CAN_MSG
{ uCANTFI FrameInfo; //CAN 报文帧信息
  uCANID msgCANID; //CAN 报文 ID
  uint8 bDb[8]; //CAN 报文数据字节
} CAN_MSG; //CAN 报文格式定义
typedef struct _Queue
{ CAN_MSG Msg[ MAXsize ]; //存储 CAN 报文
  int front, rear; //队头、队尾指示器
  bool RLock, WLock; //读、写队列允许标志
} Queue; //循环队列定义
Queue TxQueue, RxQueue; //定义发送、接收队列
```

为了增强程序的健壮性，在定义循环队列时，增加了队列读、写的允许标志位。队列的初始状态为 $front = rear = 0$ ，表示队列为空，当有新报文入队时， $rear$ 加 1；同时若有报文出队，则 $front$ 加 1。初始时刻 $RLock = WLock = 0$ ，表示队列处于可读可写状态，报文在入队过程中保持 $WLock = 1$ ，入队完成后重置 $WLock = 0$ ，这样就避免了 2 条报文同时入队。 $RLock$ 用于出队操作中，使用方法与 $WLock$ 类似。

报文的发送和接收都有入队和出队操作，下面以接收报文的入队和出队为例，说明应用程序是如何对队列进行操作的。当通过中断方式接收到的报文满足滤波条件时，将调用函数 $gCB_CanBufferMsg$ 执行入队操作，其部分代码如下：

```
void gCB_CanBufferMsg( CAN_MSG * ptMsg)
{ CAN_MSG * pEntry;
  pEntry = gQue_PeekFree((Queue *) RxQueue);
  /* 获取报文在队列中的存储地址 */
  if (pEntry) { //队列未满
    memcpy(pEntry, ptMsg, sizeof(CAN_MSG));
    /* 入队操作 */
  }
}
```

接收报文入队的详细操作流程，如图 8 所示。

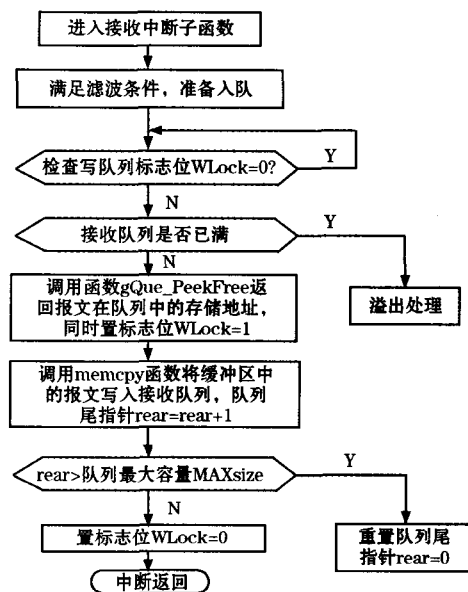


图 8 接收报文入队流程图

Fig. 8 Flow chart of putting receive message into queue

相对于接收报文入队流程，其出队过程要简单一些。接收报文出队的详细操作流程如图 9 所示。

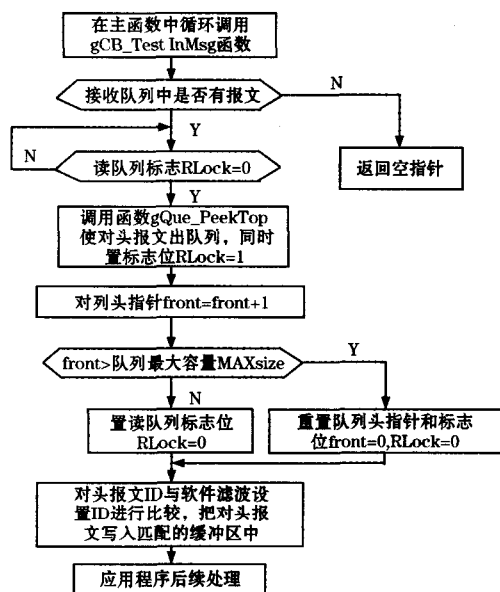


图 9 接收报文出队流程图

Fig. 9 Flow chart of removing receive message from queue

程序中通过函数 $gQue_IsElement$ 检测接收队列，若其中有报文则队头元素出队，其部分代码如下：

```
void gCB_TestInMsg (void)
{ CAN_MSG * ptMsg;
  if (! gQue_IsElement((Queue *) RxQueue))
  { return NO_MESSAGE; }
  else {
    ptMsg = gQue_PeekTop((Queue *) RxQueue);
    /* 队头报文出队 */
  }
}
```

采用这种基于队列分块存储报文处理模式有以下两方面优点：首先，通过队列缓冲，避免了由于节点收发报文过于频繁，应用程序来不及处理而产生的丢包现象；其次，存储区按报文种类来分类，可以让应用程序快速存取，保证了系统的实时性。

5 应用实例

根据以上设计方案，构建了一个基于 CANopen 协议的电动执行机构现场总线控制系统，系统实物，如图 10 所示。

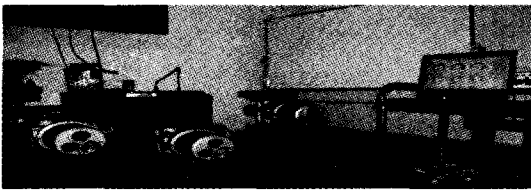


图 10 系统实物图

Fig. 10 Chart of physical system

该系统主要由电动执行机构、上位机、PCI_CAN 智能通讯接口卡等几部分组成。其中，上位机和 PCI_CAN 通讯卡组合在一起作为 CANopen 网络的主节点，执行机构作为从节点，主从节点之间通过 PDO 来交换实时数据。为实现人机对话，利用 VC++6.0 设计了人机交互界面，如图 11 所示。

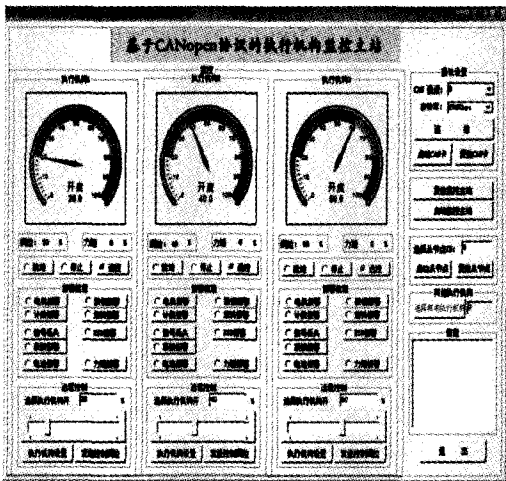


图 11 人机界面

Fig. 11 Human machine interface

通过该人机界面不仅可以对每台电动执行机构的状态进行实时监控，而且还能对执行机构进行远程控制、参数标定、功能组态和故障诊断。

在系统运行过程中，对远程控制精度进行测试。初始时刻，把 3 台执行机构的零点和行程分别调整为 0 和 5115，其对应阀门的 $0^\circ \sim 90^\circ$ 。上位机通过 PDO 给 3 台执行机构分别发送控制阀位 0 %，25 %，50 %，75 % 和 100 %，其预期的角

度值分别为 0° ， 22.5° ， 45° ， 67.5° ， 90° ，而测试结果，见表 2。

表 2 控制精度测试结果

Table 2 Result of control precision test

阀位 给定值	给定值对应 角度的理论值	执行机构 1 角度的实际值	执行机构 2 角度的实际值	执行机构 3 角度的实际值
0 %	0°	0°	0°	0°
25 %	22.5°	22.2°	22.3°	22.4°
50 %	45°	45.7°	45.4°	45.3°
75 %	67.5°	67.5°	67.6°	67.4°
100 %	90°	90°	90°	90°

测试结果表明，3 台执行机构的阀位给定值和实际值之间虽有一点小偏差，但经计算后发现它们的相对误差都在 1 % 以下，可见远程控制精度还是很高的，能够满足实际控制精度需求。同时，在上述测试过程中，通过测试软件 ZLGCANTest 测得发送每条 PDO 报文的平均时间为 2.283 ms，表明系统实时性较好。另外，经初步成本核算，采用这种控制方式时，硬件成本节约了 3.57 %。通过这些数据表明本文提出的设计方案是合理可行的。

6 结 语

本文针对常规型电动执行机构传统点对点连接方式存在的弊端，提出了基于 CANopen 协议的总线型电动执行机构设计方案。该方案通过把 CANopen 协议应用到执行机构的设计当中，使其具备总线通讯功能，合理有效地解决执行机构数据传输可靠性、远程网络通讯和操作维护困难等方面的问题。通过实际工程应用表明，这种总线型电动执行机构具有智能化、信息化、数字化等特点，实现了高可靠性、低成本的目标，其应用前景一片光明。

参考文献 (References) :

- [1] Pfeiffer O, Ayre A, Keydel C. Embedded Networking with CAN and CANopen [M]. New York: Press of RTC Books Scan Clemente, 2003.
- [2] 周纯杰, 宋明权, 王永骥. 基 CAN 总线网络通信的一种方法及实现 [J]. 控制工程, 2005, 12(1): 50-53. (Zhou Chunjie, Song Mingquan, Wang Yongji. Implementation of communication for nodes of CAN filedbus [J]. Control Engineering of China, 2005, 12(1): 50-53.)
- [3] 张颖超, 杨宇峰, 叶小岭, 等. 基于 CAN 总线的温室监控系统的通讯设计 [J]. 控制工程, 2009, 16(1): 104-106. (Zhang Yingchao, Yang Yufeng, Ye Xiaoling, et al. Design of communication for greenhouse monitoring system based on CAN Bus [J]. Control Engineering of China, 2009, 16(1): 104-106.)
- [4] CiA Draft Standard 301 V4.02, CANopen Application Layer and Communication Profile [S]. Stuttgart: CiA, 2002.
- [5] CiA Draft Standard 401 V4.02, CANopen Device Profile for Generic I/O Modules [S]. Stuttgart: CiA, 2002.
- [6] 严蔚敏, 吴伟民. 数据结构 [M]. 北京: 清华大学出版社, 2002. (Yan Weimin, Wu Weimin. Data structure [M]. Beijing: Press of Tsinghua University, 2002.)