

基于 VB 的 CAN 总线测试软件设计

孙 兵 , 赵 斌 , 曹 莹

(南通纺织职业技术学院机电系, 江苏南通 226007)

摘 要: 结合 CAN 总线的应用实例, 阐述了 CAN 总线控制系统开发过程中测试软件的设计与实现方法, 介绍了相关 VB 程序的结构和功能, 并对其中的数据通信部分程序代码进行了深入分析。

关键词: Visual Basic; CAN 总线; 测试

引 言

CAN(Controller Area Network)总线是现场总线的一个分支,因其具有很高的可靠性和性能价格比,已经成为国际标准,在工业过程监控设备的互连方面得到广泛应用,受到工业界的广泛重视,并已被公认为几种最有前途的现场总线之一。在我院“物料搬运机械手”项目的开发设计过程中,我们就将 CAN 总线引入机械手的控制系统,形成了基于 CAN 总线的分布式控制系统结构。该控制系统涉及众多软、硬件模块,给程序的设计和调试带来一定难度。尤其是作为上、下位机间联系纽带的 CAN 总线通信部分,一旦在整个系统运行期间发生问题,若没有良好的人机界面和测试手段,将很难及时准确地找到并排除故障。同样,在控制系统的研制过程中,为了尽可能地减少故障和缩小故障范围,也应设计相应的测试软件来具体负责 CAN 总线通信及接口部分的调试、运行任务。

Visual Basic 6.0 以其强大的功能、使用简单、能在短时间内开发出高效的应用程序而成为 Windows 系统开发的主要编程语言,它所见即所得的可视化界面设计风格和 32 位面向对象的程序设计方法等特点,非常适合用于开发 CAN 总线测试软件。本文将结合笔者参与开发的物料搬运机械手 CAN 总线控制系统,对如何利用 VB 设计 CAN 总线测试软件进行介绍。

1 CAN 总线控制系统概述

物料搬运机械手是由物料分拣和气动机械手两部分装置组成的,其中包含能够完成分拣物料的 3 个普通气缸以及完成机械手圆柱坐标运动的一个滑块气缸、一个机械手爪、一个气动伺服缸和一个步进电机等执行部件。如图 1 所示,这些执行部件分别通过四个 CAN 下位机节点接至 CAN 总线。下位机的 CAN 总线网络接口使用 Philips 公司的 SJA1000 芯片和 PCA82C250 芯片组成。SJA1000 具有完成 CAN 总线通信协议所要求的全部特性,它与独立 CAN 总线控制的 PCA82C200 完全兼容,并有支持 CAN2.0B 协议、扩展接收缓冲器、增强错误处理能力和增强验收滤波功能等新增功能。SJA1000 可以直接进行 CAN 总线互联,为了增强驱动能力,使用总线收发器 82C250 与之配合。而 PC 机作为上位机,是通过安装在其 PCI 总线插槽中的 CAN 适配卡连接到 CAN 总线上的,这里我们使用的是瑞隆德公司的 CANPCA 单口 CAN 总线适配卡。该适配卡中 CAN 总线数据收发也是由 SJA1000CAN 控制器和 82C250CAN 收发器完成的,主机通过 PCI 桥和接口控制电路来访问 CAN 控制器,从而实现数据通信。

CAN 总线测试软件的主要功能,一方面是测试 CAN 适配卡与 PC 机间的通信是否正常,另一方面是通过 CAN 适配卡与总线网络测试各 CAN 节点的通信功能和接口方法是否正确。设计该测

实践与经验

试软件的目的,是为机械手控制系统的进一步研制以及日后的软硬件维护工作打下基础并提供便利的手段。只有在确保 CAN 总线接口及通信无故障的前提下才能继续进行控制系统的扩展以及控制策略的实现工作。

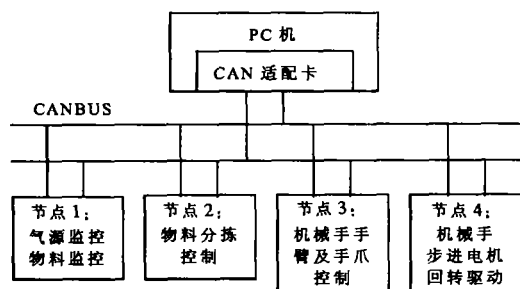


图1 物料搬运机械手 CAN 总线控制系统示意图

2 CAN 总线测试界面设计

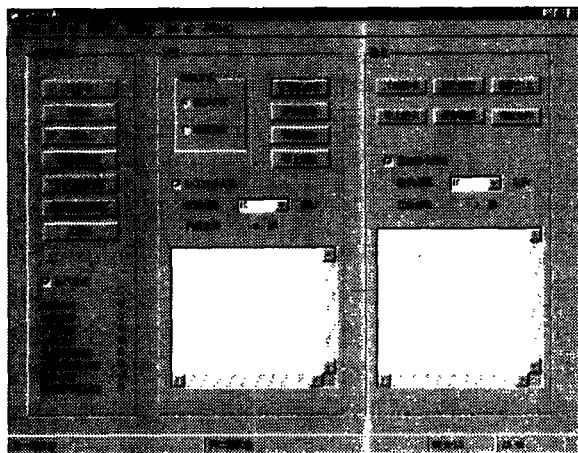


图2 CAN 总线测试软件主界面

CAN 总线测试软件的用户界面主要是由一组基于自定义对话框生成的界面共同组成的。图2所示的是其中的主操作界面,其他的界面大多是该主界面的子窗体,用于完成诸如初始化、帧内容设定等任务。该主操作界面主要包括“状态及操作”、“发送”和“接收”三个测试栏,通过它们可以对 CAN 总线的运行状态和通信能力进行有效的测试。CAN 总线的运行状态是通过八个形状(Shape)控件来反映的,每个形状控件对应一种运行状态。CAN 总线运行状态一旦发生改变,其对应的形状控件的 FillColor 属性也随之变化,从而将 CAN 总线的运行状态通过该控件的颜色变化形象地表现

出来。另外,在图2界面中除了已显示的各控件外,还在三个测试栏内分别放置了一个时钟(Timer)控件(程序运行时不显示),用于根据各自文本编辑框中设定的间隔时间或系统默认的时间值来产生计时器事件,从而驱动相关程序完成设定的任务(主要是 CAN 总线数据通信任务)。

以上位机接收下位机 CAN 节点的数据为例,其基本操作步骤如下:

(1)单击“打开卡”按钮,在随后出现的对话框中完成对 CAN 适配卡的调用及初始化;

(2)单击“初始化”按钮,在随后出现的对话框中完成对 CAN 总线通信的帧格式、波特率、接收滤波方式、接收码、屏蔽字的设置以及中断设置;

(3)单击“设置帧内容”按钮,在随后出现的对话框中完成对帧类型、帧数量、地址标识符、数据长度、数据内容等的设置;

(4)输入接收间隔时间值;

(5)单击“接收允许”按钮;

(6)单击“开始发送”按钮,将第(3)步中设定的内容向下位机 CAN 节点发送,以命令其向上位机传送采集到的数据,同时该设定内容也将在发送文本框中显示出来;

(7)单击“开始接收”按钮,将接收到的下位机上传的数据在接收文本框中显示出来。

若在“初始化”界面中将中断设置为允许接收中断,则只要外部某一 CAN 节点上有新的输入状态,该节点就会自动地向上位机传送这一新状态所对应的数据,测试软件也就能自动地将其显示在接收文本框中。

3 CAN 总线测试程序设计

(1)测试程序的基本组成和功能

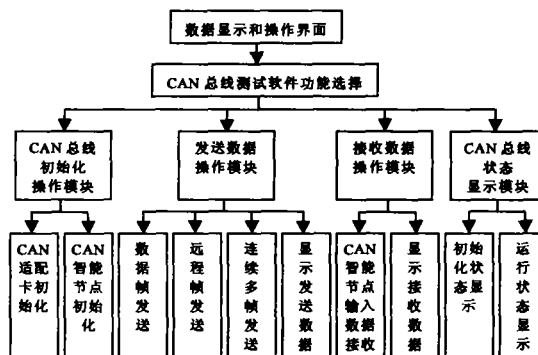


图3 CAN 总线测试软件功能框图

按照图3所示的软件基本功能,结合图2的操作界面,我们设计了相应的VB程序。由于篇幅有限,不可能一一介绍,仅就CAN总线“接收数据”的部分程序在下面进行分析。

(2)实现CAN总线“接收数据”功能的部分程序分析

```
Private Sub comInit_Click()
'CAN 控制器初始化部分程序
Dim retval As Integer
comBaud.ListIndex = 0 '波特率待输入
comAFM.ListIndex = 0 '接收滤波方式待输入
comFF.ListIndex = 0 '帧格式待输入
ptrconfig(cardno).workmode = comFF.ListIndex
'设定标准帧/扩展帧
ptrconfig(cardno).filterMode = comAFM.ListIndex
'设定 CAN 控制器采用单/双滤波方式
ptrconfig(cardno).accCode = CHex(txtCode(0).Text) * 256
* 256 * 256 + CHex(txtCode(1).Text) * 256 * 256 + CHex
(txtCode(2).Text) * 256 + CHex(txtCode(3).Text)
'设定与 CAN 节点 ID 标识符对应的接收码
ptrconfig(cardno).accMask = CHex(txtMask(0).Text) *
256 * 256 * 256 + CHex(txtMask(1).Text) * 256 * 256 + CHex
(txtMask(2).Text) * 256 + CHex(txtMask(3).Text)
'设定与 CAN 节点 ID 标识符对应的屏蔽码
ptrconfig(cardno).timer0 = CanTime0(comBaud.ListIndex)
'设定 CAN 控制器 time0、time1 寄存器内容 (用于设定
波特率)
ptrconfig(cardno).timer1 = CanTime1(comBaud.ListIndex)
ptrconfig(cardno).Control = chkInt(0).value
'设定是否允许中断
ptrStruct(cardno).card = cardno
retval = (CAN_Init(ptrStruct(cardno), ptrconfig(cardno)))
If retval = 1 Then
Unload Me
'初始化成功则退出初始化对话框,返回主操作界面
Else
MsgBox "初始化失败", vbInformation, ""
End If
End Sub
Private Sub comCF_Click() '帧内容设置部分程序
Dim n As Byte
comData.ListIndex = 0 '帧内容设定为标准帧
Trans_ptrpacket (cardno, frameno).length = CInt(txtLen.
Text)
Trans_ptrpacket(cardno, frameno).rtr = 0
```

```
'帧格式设定为数据帧
datalen = CInt(txtLen.Text) '取数据长度
check_id0 = CHex(txtID(0).Text) * (2 ^ 3)
check_id1 = (CHex(txtID(1).Text) And &HE0) / 32
Trans_ptrpacket(cardno, frameno).CAN_ID = check_id0 +
check_id1
'将输入的 CAN 节点地址按帧格式重组后形成真正的
CAN_ID 标识符,用于 CAN 数据通信中
Trans_ptrpacket (cardno, frameno).length = CInt(txtLen.
Text)
For n = 0 To datalen - 1
Trans_ptrpacket(cardno, frameno).data(n) = CInt(CHex
(txtData(n).Text))
'从相应文本编辑框中取其余数据部分
Next n
End Sub
Private Sub comBRec_Click() '开始接收部分程序
timRec.Enabled = True '开启接收定时器
recn=0
ptrStruct(cardno).card = cardno
i = CAN_Rece(ptrStruct(cardno), Rece_ptrpacket(cardno,
0)) '调接收数据函数(一次接收一帧)
If i = 1 Then
'若接收数据成功则返回值为 1
recn = recn + 1
'以下为按设定格式显示接收数据
labRN.Caption = recn
intY = recn
tempStr = "N" & intY '对接收到的数据进行编号
tempStr = tempStr & "A" & Hex ((Rece_ptrpacket (cardno,
0).CAN_ID) * 2 ^ 5))
'对应的节点地址
number = Rece_ptrpacket(cardno, 0).length And &HF
'接收到的数据长度
tempStr = tempStr & "L" & number & "D"
For intX = 0 To number - 1
tempStr = tempStr & Hex (Rece_ptrpacket (cardno, 0).
data(intX)) & " "
Next intX
txtRec.Text = txtRec.Text & tempStr & vbCrLf
'先显示数据编号及对应的节点地址,后显示数据长度
和内容
End If
End Sub
(注:在 VB 开发环境中包含有随 CAN 适配卡一起提
供的 pcican.dll 动态链接库)
```

实践与经验

程序中针对本机械手控制系统收发数据的类型特点,在CAN帧标准数据格式的选取上,采用CAN2.0B标准帧数据格式,以收发数据帧为主(远程帧的收发方法与数据帧类似,在此不再累述),其CAN_ID标识符由11位二进制数组成,包含了下位机CAN节点的地址信息。

结 语

测试软件要求具有较高的可靠性、高效率、较好的可移植性和通用性。基于VB的CAN总线测试软件通过在物料搬运机械手控制系统中的试验和调试,已经表明:数据传送实时可靠、操作直观方便、通信正常、运行状态显示及时准确。该测试软件稍加改造后已经作为机械手控制系统的有效组

成部分之一,对进一步提升CAN总线的通信能力,提高系统的可靠性和稳定性,扩展CAN总线的应用范围具有积极的意义。随着CAN总线在工业控制领域的应用不断深入,其测试软件的作用将得以更充分地发挥。

参考文献

- [1] 邹宽明. CAN总线原理和应用系统设计. 北京:北京航空航天大学出版社, 1996
- [2] 龚沛曾, 陆慰民, 杨志强. Visual Basic 程序设计教程. 北京:高等教育出版社, 2000
- [3] 孙占辉等. CAN总线在现场总线控制系统中的应用. 微计算机信息, 2002(7)
- [4] 瑞隆德公司. PCI 非智能卡使用说明书, 2002

(收稿日期:2003-04-23)

Design for the CANbus Test Software based on Visual Basic

SUN Bing , ZHAO Bin , CAO Ying

(Department of Machinery and Electronics, Nantong Textile Vocational Technology College, Nantong 226007 China)

Abstract: With an example of CANbus application, this paper introduces the test software design of CANbus control system, the construction and function of VB program. Furthermore, the paper has analysed the major program of data communication.

Key words: Visual Basic; CANbus; Test