

# PLC 编程器核心模块设计

任 伟 (桂林电子科技大学 计算机与控制学院 广西 桂林 541004)

蔡启仲 (广西工学院 电子信息与控制工程系 广西 柳州 545006)

**摘 要:** 阐述了一种以新型 PLC 装置为基础的手持编程器的设计,重点描述了编程器核心模块的结构和软件设计方法。编程器以 CAN 作为主通讯接口,ARM7 为控制器,构建了 uC/OS-II 内核的实时操作系统;实现了远距离编程、监控、调试 PLC 控制系统,脱机编程及直接与计算机交换数据的功能,适合工业现场使用。

**关键词:** CAN 总线; PLC; 编程器; ARM; uC/OS-II

## Design of the Core Modules of PLC Programmer

REN Wei<sup>1</sup>, CAI Qi-Zhong<sup>2</sup>

(1.School of Computer and Control,Guilin University of Electronic Technology, Guilin 541004, China; 2.Department of Electronic Information and Control Engineering, Guangxi University of Technology,Liuzhou 545006, China)

**Abstract:** The paper introduces the method for the design of the PLC hand-held programmer based on a new type of PLC device. It particularly describes the design method of the structure and software on core modules. The CAN bus is used as a main way of communication. The ARM7 chip is selected as the main controller. The RTOS on uC/II is built. The instrument realizes the function of remote monitoring, programing and adjusting the PLC device, and programming in off-line model. It can communicate with the computer directly and will be suitable for the industry field.

**Keywords:** CAN bus; PLC; programmer; ARM; uC/OS-II

在工业现场调试 PLC 设备时,编程装置只能连接 PLC 专用通讯端口。当监控点距离 PLC 主机较远时,常用的编程工具调试时很不方便。手持编程器多任务并行处理能力差、功能单一、用户空间小、外部接口不足,不能与计算机相互传递数据;需专用电缆与 PLC 连接通讯和获取电源,使用时严重依赖 PLC 设备<sup>[1]</sup>。利用 CAN 现场总线的远距离节点在工业现场的通讯和控制中的可靠性,将 CAN 总线技术用于 PLC 控制装置中,设计以 CAN 总线为主要通讯方式,以 ARM 嵌入式处理器为控制核心的便携式 PLC 编程器,实现了在监控现场编程和调试远距离 PLC 主机及直接与上位计算机传递 PLC 用户程序的功能。

## 1 系统总体设计

该编程器装置是一种基于 CAN 总线的通用可编程控制器的编程子系统。图 1 是 PLC 装置结构框图,该装置主要由 PLC 主机、编程器、现场监控人机界面模块、扩展模块等组成,各模块通过 CAN 总线连接;根据控制的需要,PLC 输入检测和输出控制分散在工业控制现场。编程器通过现场总线连接远距离的 PLC 主机,编辑和下载用户程序到 PLC 主机的程序存储器中。用户观察现场控制输出状态,修改和调试远距离主机中的 PLC 用户程序。编程器也可以通过 UART 异步串行通讯接口与上位计算机传递用户程序。利用计算机专用编程软件,在计算机上编辑处理用户程序后

基金项目:广西自然科学基金(桂科自 0991067)

收稿时间:2009-08-25;收到修改稿时间:2009-10-04

下载到编程器中，利用编程器完成现场调试。编程器能实现脱机编程，能通过 CAN 总线网络对 PLC 设备进行联机编程、调试及监控，且功耗低、便于携带、人机界面友好、方便与上位计算机通讯。

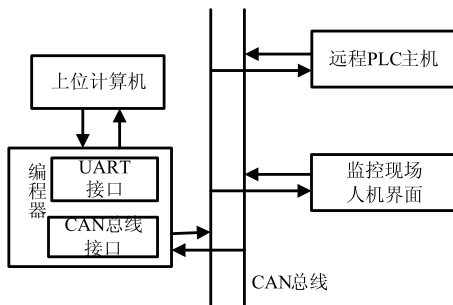


图 1 PLC 装置结构

nGCS1、nOE 通过逻辑门与 NAND FLASH 芯片的读写控制信号相连，该芯片的操作控制信号线与微处理器的 GPC1 ~ GPC3 相连，通过编写软件来实现对该芯片的读写。

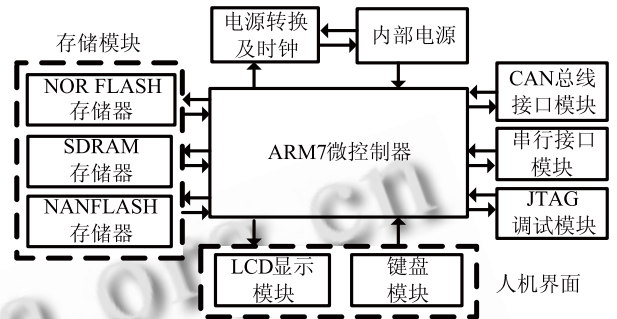


图 2 编程器结构

## 2 系统硬件设计

系统硬件主要由嵌入式微处理器、存储模块、CAN 接口模块、串口通讯模块、人机界面、JTAG 调试模块等组成。系统总体设计如图 2 所示。存储模块为操作系统软件 and 用户数据分别提供不同的存储空间。系统以嵌入式微处理器和系统存储器为基础构建嵌入式实时操作系统。CAN 总线接口模块连接现场总线，识别 CAN 总线上的设备，提供独立的 CAN 通讯控制，为应用程序实现联机模式下的各种功能提供通讯数据。串口通讯模块提供符合 RS-232C 标准的信号，满足编程器与计算机通讯。人机界面为人机交互提供 LCD 显示输出和键盘输入。LCD 显示模块显示编程器系统当前的运行状况，为用户操作提供系统菜单和具体操作选项。键盘提供用户命令输入接口，并根据操作性质不同分为系统键、功能键、复用键三部分。系统键可设定系统参数、运行状态等。功能键实现与编程器功能相关的文件管理、通讯控制、指令选择等操作。复用键与功能键一起使用完成对软件件和各种指令符号的输入和编辑。

系统存储模块由 NOR FLASH、SDRAM 和 NAND FLASH 组成。NOR FLASH 读取速度快，存放系统软件作为系统数据存储空间，采用 SST39VF1601 NOR FLASH 芯片，其标准总线接口可直接与处理器相连。SDRAM 为系统运行提供动态空间，采用 32MB HY57V561620 芯片，该芯片的控制信号线与 S3C44B0 有一一对应的接口。NAND FLASH 用作用户程序存储器，采用 K9F5608U0B 芯片，其存储模式为 32M × 8bit。S3C44B0 的控制与片选信号 nWE、

CAN 接口模块由 CAN 控制器、隔离光耦，CAN 收发器组成。电路如图 3 所示。采用独立 CAN 控制器 MCP2510 芯片。主控制器的 SIO 口与 MCP2510 的输入输出控制信号相连，直接控制 MCP2510 的内部寄存器。MCP2510 的复位信号与主控制器的复位信号直接相连，可以手动或自动复位。主控制器的 GPF2 引脚与芯片的片选信号相连并通过软件控制。MCP2510 的中断输出与微控制器的外部中断输入 EINT2 相连。为增强抗干扰性，CAN 控制器与 CAN 收发器的输入输出信号通过光耦隔离，两边电源独立供电，并通过 DC-DC 电压隔离模块 B0505S 隔开。采用 6N137 高速光耦隔离器，并在接收和发送数据线上各接一个隔离光耦。电路中使用 B0505S 输出的隔离电源，并在两根数据线与地之间并联小于 470pF 的电容用于高频滤波。终端电阻使用两个等值的电阻提高电磁兼容能力。

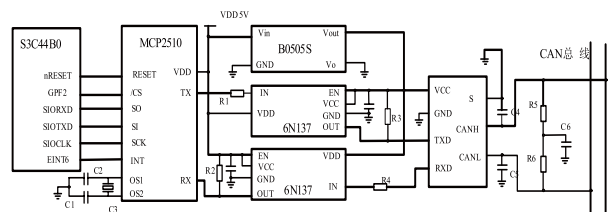


图 3 CAN 总线接口

## 3 软件系统设计

编程器软件系统采用 C 程序设计。由于编程器采用 ARM7 作为主控制器，采用了不同于计算机的 RISC 指令结构，必须使用交叉开发环境使在 PC 机上

编辑的 C 语言程序能在 ARM 处理器上运行。设计中使用 ADS1.2 集成开发工具,该工具是专门用于 ARM 开发和调试的综合性软件<sup>[2]</sup>。用户在 PC 机上用该工具编写和调试用户程序,最终生成符合 ARM 结构的程序代码。

### 3.1 系统软件结构

编程器有多个外围接口,要实现包括总线通讯、键盘输入、显示输出等多种功能,必须构建嵌入式实时操作系统对系统的软硬件资源进行管理和调度。在嵌入式实时内核的基础上构建文件系统,用户接口程序和应用程序组成实时操作系统来实现编程器的功能。软件系统结构如图 4 所示。设计中采用 uC/OS-II 实时内核,它是由 C 语言编写的容易移植剪裁的先占式实时多任务内核,适合不带 MMU 的 ARM7 结构的处理器。该内核已广泛使用于商业产品中稳定可靠<sup>[3]</sup>。



图 4 软件系统结构

在将 uC/OS-II 实时内核移植到 ARM7 的基础上,设计驱动程序、应用接口程序 API 和用户程序。系统文件主要存放在 FLASH 电子硬盘中,设计采用 FAT16 格式的文件系统,此文件格式与计算机中的文件系统兼容,便于与计算机传递文件。嵌入式操作系统是与特定硬件系统对应专用操作系统,其硬件驱动程序和应用接口程序都需要独立设计,主要有:与文件操作、PLC 指令编译相关的 API 函数,与键盘、LCD 显示、串口、CAN 总线接口等硬件相关的驱动程序和 API 函数的设计。

根据用户应用程序的功能和作用将其划分为几个不同的任务,各任务设定不同的优先级,由系统来调度。设计中将编程器要完成的功能划分为五个任务:键盘扫描任务、LCD 显示任务、串口扫描任务、系统主任务。系统主任务优先级最高,它通过消息系统接收其他任务发送的消息来调度应用接口函数 API 实现编程器的功能,实时响应用户的操作。各任务在由系统建立后即进入就绪态等待任务调度系统的调度运

行。以下是关于任务定义的 C 语言程序代码。

```

OS_STK Main_Stack[STACKSIZE*8]={0, };
//申请系统主任务的堆栈

void Main_Task(void *Id); //系统主任
务声明

#define Main_Task_Prio 12 //定义主任
务优先级

OS_STK Key_Scan_Stack[STACKSIZE]={0, };
//键盘扫描任务的堆栈

void Key_Scan_Task(void *Id); //键盘扫描任
务声明

#define Key_Scan_Task_Prio 50 //键盘扫描任
务优先级

OS_STK Lcd_Fresh_Stack[STACKSIZE]={0, };
//LCD 刷新任务的堆栈

void Lcd_Fresh_Task(void *Id); // LCD 刷新任
务声明

#define Lcd_Fresh_Task_Prio 53 //LCD 刷新任
务优先级

.....
    
```

系统启动时,先对硬件系统初始化,初始化处理器和外围的硬件接口。然后进行操作系统初始化,设置各种数据结构和变量并调用系统配置文件;创建用户任务并建立任务间通讯的邮箱或消息队列,分配任务优先级并将所有的任务置为就绪态。系统首先执行优先级最高的主任务,编程器进入运行状态。系统程序运行流程如图 5 所示。

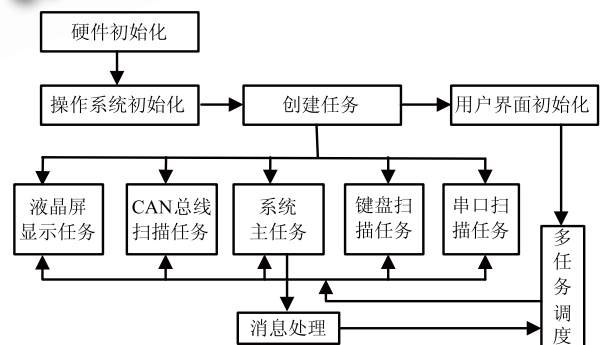


图 5 系统程序运行流程

### 3.2 应用程序结构

系统主任务实现编程器的主要功能。当系统启动进入用户界面后,进入主功能菜单选项。用户通过按

键选择不同的菜单功能。主任务在完成必要的初始化工作后,进入消息循环中等待用户操作。当有按键按下或 CAN 总线接收到数据时,其对应的监控扫描任务会向系统发送消息。主任务在等待消息时处于挂起状态。当收到消息后,主任务确认是需要的消息,随即进入就绪态并调用相关的 API 函数处理用户按键内容。完成用户指令任务后,主任务进入挂起态等待下一条消息。系统主任务的 C 语言程序代码如下:

```
void Main_Task(void *Id)
{POSMSG pMsg=0; // 定义一个消息变量
U32 mainstatus; // 定义菜单选择变量
.....
ClearScreen(); // 完成必须的初始化设置
for( ; ; ) // 进入消息循环
{ pMsg=WaitMessage(0); // 等待消息任务处于挂起状态
switch(pMsg->Message) // 收到消息,判断消息类型
{ case OSM_KEY: // 键盘扫描消息则主任务处于就绪态
..... ; break; // 调用相关函数处理用户按键操作
case OSM_CAN: // CAN 扫描消息主任务处于就绪态
..... ; break; // 主任务调用函数处理 CAN 总线操作
.....
DeleteMessage(pMsg); } // 删除消息,释放系统资源
}
```

编译器采用菜单式应用程序结构。当系统启动并进入用户界面后选择脱机或联机工作模式。在脱机模式下编译器有文件管理、参数设定、程序编译等多个功能选项。文件管理对编译器用户存储器中的数据文件进行读出、复制、删除、重命名等操作。程序编译实现用户编写、保存、编译指令表程序并形成二进制目标代码的功能<sup>[4]</sup>。联机模式下有文件管理、程序编译、设备调试、远距离监控、数据通讯等功能。设备监控是通过显示屏来监视和确认 PLC 设备的动作和控制状态。远距离调试则是通过 CAN 总线对 PLC 设备的位元件强制置位、对字操作元件数据进行修改等功

能。设计中采用函数指针数组来装载系统的各个功能函数,在数组中放置的是函数指针,指向对应的函数。函数指针数组结构与菜单结构对应。对于菜单上每一个选项都有一个状态,用 mainstatus 变量记住这个状态。在不同状态下时,高亮度显示当前选择的菜单项。当在某个状态下单击 OK 键时,系统便会执行相应的处理函数。例如,当 mainstatus 对应 OnlineMod (联机模式)时,单击 OK 键,进入联机模式选项。当菜单上选择条对应第二项 DevTest 时,按确定键后相应函数将得到执行。以下是函数指针数组结构 C 语言程序代码:

```
void *MainFunction[ ]={ (void *) OnlineMod,
(void *)OfflineMod };
void *OnlineMod[ ]={ (void *)DataComun, (void *)DevTest, (void *)TelMoni, (void *) FileManig}
void *OfflineMod[ ]={ (void *)ProgCop, (void *)ParaSet, (void *)FileManig}
```

可以通过如下方式调用指定序列的函数, MainListCtrl 是指向函数指针数组的指针。

```
((void*)(void)) (MainFunction[MainListCtrl
——>CurrentSelect] ) ( );
```

### 3.3 用户指令编译程序

编译器的一个主要功能是实现通过该设备完成编译处理用户指令表程序。用户代码编译模块由代码输入程序、词法分析程序、语法分析程序等三部分组成。代码输入程序将用户输入指令表程序进行分割,根据指令表程序的编程格式扫描用户输入程序,识别出操作码和操作数,并按照指令表程序的顺序放入二维数组<sup>[5]</sup>。词法分析程序实现对用户程序的操作码和操作数有效性的校验。词法分析程序通过对源程序的两次遍历检验错误,生成中间代码供语法分析程序调用。语法分析程序用堆栈存放语法分析的过程结果,分析过程构造两个变量,按照 PLC 编程规则分析中间代码,发现错误时返回错误信息。当无错误时,语法分析程序进行再次扫描并将中间代码转换为目标代码。目标代码编码占 32 位由操作码和操作数按一定的规则移位相加后得到。

### 3.4 CAN 通讯程序

在用户应用程序中将 CAN 总线接口扫描作为一个独立的任务,将 CAN 总线的通讯控制按功能不同编写对应的应用接口程序函数 API 供系统主任务来调用<sup>[6]</sup>。主要的 API 函数有: CAN\_INIT( )CAN 总线初始化函数、CAN\_SENDDATA( U8 \*Datapointer )报文发送

函数、CAN\_READDATA(U8 intvalue)报文接收函数等。初始化程序对 CAN 控制器初始化,设置各种寄存器的优先级。报文发送程序查询控制器的发送控制寄存器的发送请求位,当该位为 1 时启动相应的报文发送。有多个报文发送时,高优先级的报文先发送。报文发送成功后产生中断信号,并将数据缓冲器中的数据装入发送缓冲器。报文接收程序在报文缓冲器接收到传来的报文后,将 CAN 中断标志寄存器中断标志位置 1,产生中断请求。用户 CAN 扫描任务在单位时间内扫描中断标志寄存器,当有新的数据接收到时,创建并发送一个消息。通过消息传递参数供系统主任务调用相关的 CAN 总线 API 函数时使用。该任务随后产生一个延时让自身挂起,让出 CPU 的控制权。CAN 总线扫描任务的 C 语言程序代码如下:

```
void CAN_Scan_Task(void *Id)
{POSMSG pmsgCAN;          // 定义一个信号量
for( ; ; )                // 进入循环扫描
{InterruptValue=CAN_READ(CAN_IR);
//读中断寄存器
if(InterruptValue & 0x1f) //判断中断标志位
{pmsgCAN=OSCreateMessage(NULL,OSM_CAN,0
,InterruptValue);
//创建一个消息
if(pmsgCAN)SendMessage(pmsgCAN);} //消息创
建并发送
```

```
OSTimeDly(5);}          //延时挂起让出 CPU 控制权
}
```

## 4 结语

本设计将嵌入式技术与现场 CAN 总线技术结合,解决 PLC 编程设备在现场调试中的远距离通讯问题,构建嵌入式实时操作系统对通讯、控制、输入、显示等功能进行实时处理,符合当前 PLC 技术向网络化发展的趋势,实现了面向网络的,便携式,多功能,低功耗,高性价比的编程器设计,对当前工业领域的应用有推广价值。

## 参考文献

- 1 林小峰.基于 IEC61131-3 标准的控制系统及应用.北京:电子工业出版社,2007.230 - 250.
- 2 王田苗.嵌入式系统设计与实例开发(第二版).北京:清华大学出版社,2003.
- 3 张绮文,谢建雄.ARM 嵌入式模块与综合系统设计.北京:电子工业出版社,2007.198 - 203.
- 4 钟艳如,洪磊.基于 SIPN 的 PLC 通用编程器.计算机工程,2009,35(7):283 - 285.
- 5 周峰,王新华.软 PLC 编辑系统的设计与实现.计算机工程与应用,2005(7):111 - 117.
- 6 饶运涛.现场总线 CAN 原理与应用技术.北京:北京航空航天大学出版社,2007.102 - 108.