Design and Realization of MiniPLC Programming System

QingChao Wei[1, a], QiZhong Cai[1,b], CongSe Xie[1,c], ShaoMing Pan[1,d]

[1]College of Electric and Information Engineering, Guangxi University of Science and Technology,

Liu Zhou city, Guangxi, China

[a]wqckhd@163.com,[b]caiqzh@163.com,[c]512315178@qq.com,[d]panshaoming@sohu.com

Keywords: LPC2478, PLC Instruction File, Edit, Compile, CAN Bus

Abstract According to the independent research small Programmable Logic Controller based on ARM and FPGA, a miniPLC programming system is designed. LPC2478 and μC/OS-II system are adopted as the platform's core. Instruction List is adopted as the programming language. A new instruction system and its encodes are designed. PLC source instruction file is editted and compiled. The mapping relation among PLC source instruction file, PLC object instruction file, PLC object code file, ASCII character table, dot matrix table is established. LCD screen display is planned.Two-way chain tables are adopted to edit and store PLC source instruction file.After analysis, optimization and error handling, PLC source instruction file is compiled to create binary object code file PLC host can identify. The object code file is sent to PLC host via CAN Bus.Through the test, it is demonstrated that the system can edit and compile PLC instruction list correctly and effectively, and monitor PLC host via CAN Bus.

Introduction

Nowadays MiniPLC programming device mainly has two kinds. One kind is a computer with a special programming software and the other kind is a dedicated portable handheld programmer. Both must connect PLC equipment fixed communication port through the special communication cable in on-line mode in order to operate. The latter cannot be used in offline mode, because its working power supply must be provided by special PLC interface. The latter's storage capacity is limited in which only a set of programs stored. Although they can respectively communicate with PLC communications equipment, but one cannot directly communicate with another between them, and they can't remotly monitoring PLC host.

In a small programmable controller based on ARM and FPGA[1], a MiniPLC programming system is designed. It adopts the embedded ARM microprocessor, and builds an embedded real-time operating system for multitasking real-time scheduling, with large storage capacity. In online or offline mode it can independently supply working power, edit, compile, and debug PLC programs, remotly monitor PLC host via CAN field bus. It has a quick computing speed, low power consumption, good real-time performance and so on, and is suitable for the operation of the industrial field.

Overall Design

LPC2478 is adopted as the control core of the hardware platform in charge of running the whole system. Its hardware framework is composed of by main control module, storage module, man-machine interface module, communication module and JTAG debug module. The storage module is composed of NOR FLASH, SDRAM and NAND FLASH memory, used to store user program and datas. Communication module is composed of CAN Bus communication module and serial interface communication module. In power supply module DC power supply can be got via an external power supply from the PLC host, also through the USB transceiver access between USB interface and PC, also

by the internal rechargeable battery 5v. When an external power source is used ,it can charge the battery, and improve the applicability of the programming device.

The Software framework is composed of PLC program edited module, PLC program compiled module, data storage module, man-machine interface module, communication module. The software framework is shown in figure 1. First, uC/OS-II system is transplanted. Its drives, application programs, and the task scheduling are achieved[2]. In uC/OS-II system environment, data storage module is used to store PLC instruction file, binary object codes and so on. Communication module is used to communicate the system with PLC host through CAN Bus.

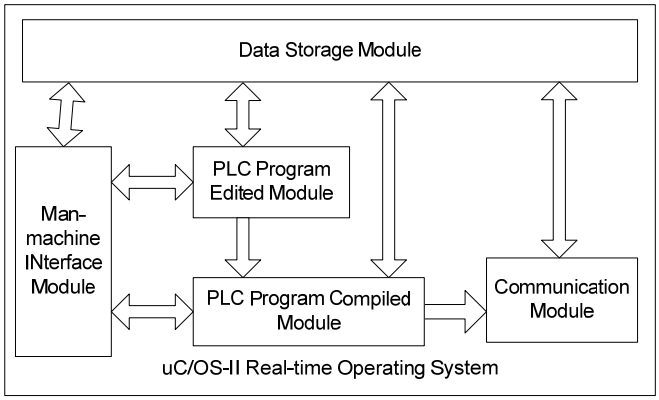

Fig. 1 The Software framework of PLC editing system

Instructions System Design

The characteristics of multiple operands logic operation instructions.In-depth studying the characteristics of PLC ladder diagram program[3,4], it is found that multiple close or moving off contact are in series or parallel arrangement. In this case, In the application of FPGA, a logic arithmetic controller is designed and can be performed simultaneously with 27 operands. So that a new LDR instruction is used for multiple soft components in parallel connected to the bus, so LD, LDR, AND and OR logical operation instructions have at least one operand, up to 26 operands. The soft components adopts direct addressing or indexed addressing mode.

The emergence of each instruction has a certain regularity. For example, AND can only appear in before the instruction is OR or ORB, ANB, MPP and so on, after it the instruction is LD. Under the condition of LD instruction followed the OR after, LDR repalces LD and OR. Compared with mitsubishi instructions, the instruction system's improvement focuses on the logical operation instructions.

In multiple operands logic operation instructions, contact types of operands have normally open, normally open up with the differential, normally open down and the differential, normally closed, normally open up with the differential, normally open down with the differential, the normally closed up with the differential. Among them the latter five types respectively is shown by P , F, I, IP, IF. The rising and falling edge differential of the signal show after a scan cycle delays the signal is outputed in the rising or falling edge.

Instruction Classification.PLC instructions are divided into general instructions, step instructions and application instructions. In PLC instruction system some instructions have no operands, some have multiple operands. Depending on the number of operands, PLC instruction is divided into three categories, the first is multiple operands instruction, the second is instructions without operands, single operand in the basic instruction and step instructions, the third type is application instructions. High four binaries are used to encode instruction types. The second and third types of instruction are respectively 0110, 0111, the first kind is corresponding to 0000 to 0101 and 1000 to 1101.

PLC Program Edited Module Design

PLC source program editing module mainly achieves to create, read, write, insert, modify, delete, search PLC source instruction file, and so on.

During editing PLC instruction file, to modify files, such as insert or delete orders or some part of the order, so the whole editing process is a process of dynamic storage. when editing commands, storage structures is used. Each instruction is stored in the data domain of a node. Two-way linked list is used to store the entire file. As long as operating the pointers, reading, writing, inserting, deleting, searching and other functions can be finished.

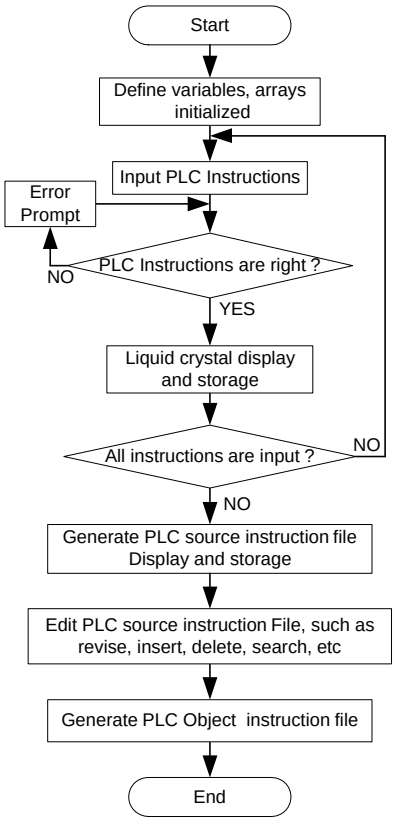

Fig. 2 Editing implementation flow diagram

Editing implementation flow diagram is shown in figure 2.First of all, Creating enough space for instruction file, seting up instruction file, including PLC source instruction file, PLC object instruction file and PLC object code file. PLC source instruction file stores source program. PLC object instruction file stores PLC source file that is editted correctly, PLC object code file stores the generated binary codes after it is compiled.Three files adopt two-way linked list, The nodes are corresponding in three files.In PLC instruction file a storage unit is a 32-bit word. A node stores a instruction, including the order number, operator and operands. Instruction number takes up a word, and uses four decimal representation. A operator and a operand take up a word. Every two operands take up a word.

In ASCII character values table with 128 ASCII values, each ASCII value takes up 8 bit binary number, said as an address. Dot matrix font table contains two types, one is 8*16 lattice of Numbers and Letters, the other is 16*16 dot matrix Chinese characters. The mapping relationship among PLC source instruction file, PLC object instruction file, PLC object code file, the ASCII character table and dot matrix font table is shown in figure 3. If modifying the error or changing orders (e.g.insert or delete an operand), the corresponding memory address will be changed too, and the number and the content of the data domain of the nodes in PLC object instruction file, PLC object code file also changes accordingly.

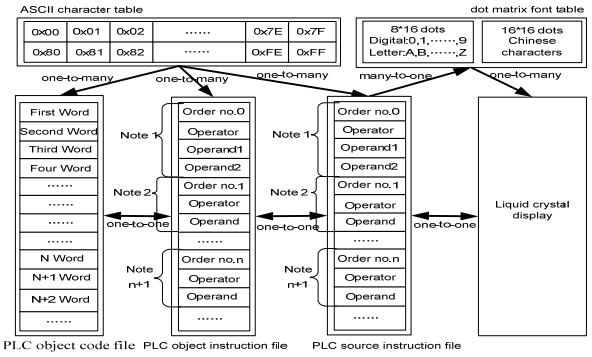

Fig. 3 The mapping relation among PLC source instruction file, PLC object instruction file, PLC object code file, the ASCII character table and dot matrix font table

PLC Program Compiled Module Design

PLC program compiled module mainly achieves to lexical analysis, syntax analysis, semantic analysis, code optimization [5,6] for PLC source instruction file. If there ware errors, errors would be handled. Binary object code file is generated. Static compilation is combined with dynamic compilation adopted to build PLC instruction file. The instructions without the operands are compiled in the static compilation mode, and the ones with operands that addresses are changing are compiled in the dynamic compilation mode.

After above stages, completely correct PLC source instruction file on structure and logic is compiled into binary object code.

Communication Module Design

For remotly monitoring PLC host, in on-line mode CAN bus[7] is adopted by the programming device to realize the communication between the programming device and PLC host. Via CAN bus, PLC host's running state is monitoring, the programs are debugged. PLC object code file is sent to PLC host, and user programs that PLC host sends are received by the device.All kinds of information requests of soft components from PLC host are monitored and detected.

In the communication large quantities of complex datas appear, the extension of CAN protocol is designed. the communication data types between the device and PLC host include PLC program and the informations of soft components. So that the custom format is start bit+device address+data type+ data number+data length+ data+CRC check code+terminator, named as CAN expand protocol format that is shown in table 1. According to CAN extend protocol, the communication is completed between the programming device and PLC host. Before the datas sended, fill the datas to finish encoding in accordance with the expand protocol, and decoding according to the format after receiving the datas. When CRC check code received by PLC host and the CRC check code sent by the programming device matches, it shows that the datas received are right, or datas continue to be sent again. According to the decoding contents and the state of PLC host, each task executions are controlled.

Table 1. CAN Expand Protocol Format

Format	Length	Implication
Start bit	1 Byte	0x68
Device Address	1 Byte	0x00：PLC HOST 0x01：Programming Device
Data Type	1 Byte	0x00：PLC User Program 0x01：X 0x02：Y 0x04：M 0x05：D 0x06：C 0x03：S 0x07：T
Data Number	2 Bytes	For PLC program, the data number refers to the NORFLASH address PLC program stored in, For soft components, the Data number refers to its following number
Command	1 Byte	0x00:Read, 0x01:Write, 0x02:Changing PLC host state, 0x03:Power on the query 0x04:Response on Power, 0x05:Editing State, 0x06:Running State 0x10:Interrupt to read, 0x11:Interrupt to write；
Length	2 Bytes	Bytes of datas read, writen, and deleted
Data	Length Bytes	The datas stored
CRC check code	2 Bytes	Check Code
Terminator	1 Byte	0x7E(End Flag)

PLC programming device is connected to the PLC host. When PLC host is in editing state, the programs can be sent to PLC host, and are solidified to PLC host's Norflash. When PLC host in running state, PLC programs can be read by the device, and the device can monitor and test the values of soft components.

Example Test

A section of PLC source program are adopted as an example to illustrate the system's reliability and effection. PLC source program is as follows:

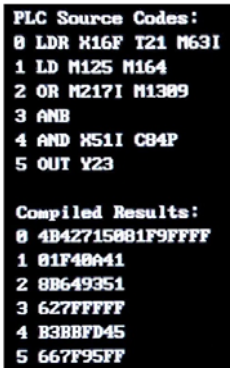

Fig. 4 PLC source program and the compiled results

```
0 LDR X16F T21 M63I
1 LD M125 M164
2 OR M217I M309
3 ANB
4 AND X51I C84P
5 OUT Y23
```

If mitsubishi PLC instructions ware used to write the program, at least 11 instructions would be needed, but only 6 instructions are needed with the new instructions. The binary object code are displayed in the form of hexadecimal number. PLC source program are located in the upper, and the results are in lower side, as shown in figure 4. Compiled compared with the expected results, the results confirm the system's correctness.

Summary

In this paper a PLC programming system is designed. In uC/OS-II system environment the system realizes editing, compiling PLC instruction file, and communication with PLC host. Pointer two-way chain table is used to complete editing. PLC instruction file is analysed correctly, and compiled into binary codes. The test showed that multiple operands logic operation instruction have improved the efficiency of editing and compiling PLC instruction file. Via CAN Bus the programming system communicates with PLC host, and remotly monitors the state of PLC host and the informations of soft components.

References

[1] CAI Qizhong，GUO Yifeng, CHEN Wenhui etal.General small programmable controller and its control method[P]. China Patent：200710052941.X，2009.08.19.

[2] ZHOU Hangci. Programming technology based on embedded real-time operating system[M]. Beijing: Beijing university of aeronautics and astronautics press,2011.

[3] BAI Jiang, WANG Yuhan, JIN Yongqiao. Develop and study of numerical control system soft PLC module[J].Machinery Design and Manufacture, 2011,(02):138-140.

[4] Yan Yi, Zhang Hangping. Compiling Ladder Diagram into Instruction List to comply with IEC 61131-3[J].Computers in industry, 2010,61(5):448-463.

[5] JIANG Zongli, JIANG Shouxu.Compilation principle[M]. Beijing: Higher education press, 2011.

[6] LU Lin, BAI Ruilin. Design and implementation of a kind of PLC compiler[J].Micro computer information, 2008,24(12)：17-19.

[7] WANG Liming, XIA Li, SHAO Ying. Design and application of CAN Field Bus system[M]. Beijing: Electronic industry press, 2008.3.

Foundation Project

Guangxi Graduate Education Innovation Project(2013105940811M01);
Guangxi Science Foundation Project(Guangxi Science Foundation from 0991067)