TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.3, March 2014, pp. 2133 ~ 2137
DOI: http://dx.doi.org/10.11591/telkomnika.v12i3.4424

Research and Implementation of PLC Editor System

Qingchao Wei, Qizhong Cai*, Congse Xie
College of Electric and Information Engineering, Guangxi University of Science and Technology
East ring road no. 268, city central district, Liuzhou city, Guangxi, China,
Ph./ Fax: /86-772-2687009/ 86-772-2687009
Corresponding author, e-mail: caiqzh@163.com, wqckhd@163.com, mail2011xcs@163.com

Abstract

According to the new type of portable PLC programming device adopting the own invention technology, a PLC editor system with ARM microprocessor and μC/OS-II system as the core was constructed. Logic operation instructions with multiple operands and their encodes were designed. The data storage capacity was expanded. A CAN extension protocol was formulated. Two-way chain table was adopted to edit PLC source instruction file dynamicly, and the file was compiled to a binary object code file. According to CAN extension protocol format the object code file was encoded and sent to PLC host via CAN Bus. The results show that the system improves editting efficiency and the running speed of PLC host. Many sets of source instruction files' storage, PLC source instruction file compiled accuratly, the highly reliable communication and monitoring PLC host are implemented.

Keywords: *ARM, source instruction file, edit, CAN bus, monitor*

Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Nowadays small scale PLC programming device [1-3] mainly has two kinds. One kind is a computer with 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 [4], a small PLC editor system is designed. It adopts the embedded ARM microprocessor as the core, can edit and compile PLC programs in online or offline mode. Many sets of source instruction files are stored. CAN bus nodes are set at the right place in the industrial field. PLC programming device is connected via CAN bus nodes, and can remotly monitor PLC host. Multiple operation logic operation instructions are designed. FPGA is applied to implement the parallel execution of logic operation. So the speed of PLC is improved.

2. Overall Design Thought

LPC2478 is adopted as the control core of the hardware platform in charge of running the whole system. The hardware framework is shown in Figure 1, and is composed of by main control module, storage module, man-machine interface module, communication module and JTAG debug module. 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.

Received August 15, 2013; Revised October 5, 2013; Accepted October 14, 2013

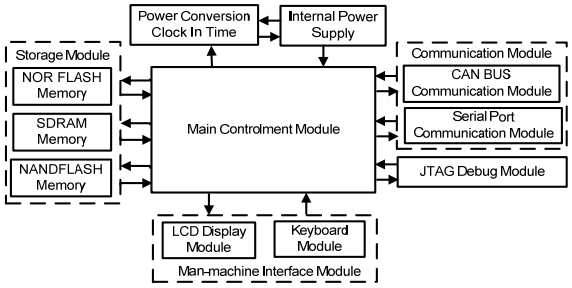

Figure 1. The Hardware Framework of PLC Editor System

The Software framework is composed of PLC program edited module, PLC program compiled module, data storage module, man-machine interface module, communication module. First, uC/OS-II system is transplanted. Its drives, application programs, and the task scheduling are achieved [5]. 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.

3. Instructions System design

3.1. Multiple Operands Logic Operation Instructions' Characteristics

In-depth studying the characteristics of PLC ladder diagram program [6-8], 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. Compared with mitsubishi PLC instructions, LDR instruction is added for multiple soft components in parallel connected to the bus, If LD is followed by OR tightly, LDR repalces LD and LDR. So LD, LDR, AND and OR logical operation instructions have at least one operand, up to 26 operands.

A logical operation instruction has one more operands, which makes full use of parallel handling operands' feature.In the process of performing user programs the identification number reduces. If more operands appear, logical operation instructions' advantages are more obvious.

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.

3.2. 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.

4. PLC Editor System Implementation

PLC editor system mainly achieves to create, write, insert, modify, delete, read,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. Editing implementation flow diagram is shown in Figure 2.

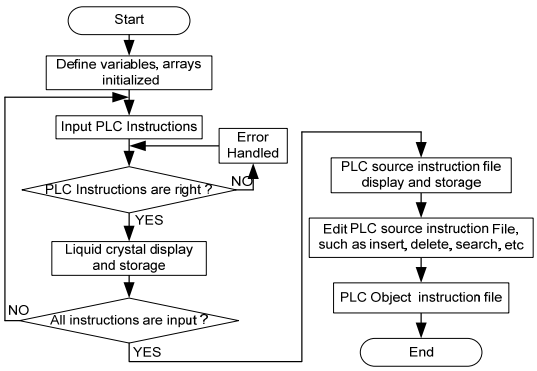

Figure 2. Editing Implementation Flow Diagram

4.1. Human-machine Interface Plan

Reuse design is adopted by the keyboard, including function keys, command keys, alphabetic keys, numeric keys, shift keys, and so on. Amog them, command and alphabetic keys reuse, command and numeric keys reuse, function reuse keys include read and write, insert and delete. LCD screen has scrolling up and down, rinsing the screen.

4.2. Create Files

Create enough space for PLC source instruction file, and set up PLC source instruction file, including PLC source instruction file and PLC object code file. PLC object code file is generated 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 representations. A operator and a operand take up a word. Every two operands take up a word.

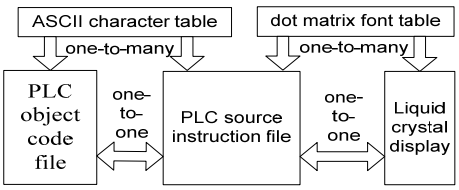

Figure 3. The Mapping Relation among PLC Source Instruction File, Object Code File, the ASCII Character Table and Dot Matrix Font Table

In ASCII character values table with 128 ASCII values, each ASCII value takes up 8 bit binary number, said as an address. The mapping relationship among PLC source instruction file, object code file, the ASCII character table and dot matrix font table is shown in figure 3. If modifying the error or changing orders, 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.

4.3. Data Storage

The storage module is composed of NOR FLASH, SDRAM and NAND FLASH memory. PLC editor system has large storage. NOR FLASH is used to store system programs, user program and datas. SDRAM is used to provide dynamic storage space for system running

communication, datas and object code file. NAND FLASH is used to store many sets of PLC source instruction files.

Two-way chain table is adopted to store all the files. Each node has precursor point and subsequent point. Data domain of sach node is used to store PLC instructions.

4.4. User Program Compiled Process

User program compiled mainly achieves to lexical analysis, syntax analysis, semantic analysis, code optimization [9, 10] for PLC source instruction file. If there were 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.

5. Communications Design

For remotly monitoring PLC host, in on-line mode CAN Bus [11-12] 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. 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.

During 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 decoded 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.

6. Example Test

PLC ladder diagram is adopted as an example to illustrate the system's reliability and effection. The diagram is shown in Figure 4.

If mitsubishi PLC instructions were used to write the program, at least 12 instructions would be needed, but only 7 instructions are needed with the new instructions. The binary object code are displayed in the form of hexadecimal number. PLC source instruction file is located in the upper, and the binary object codes are in lower side, as shown in Figure 5. Compiled compared with the expected results, the results confirm the system's correctness.

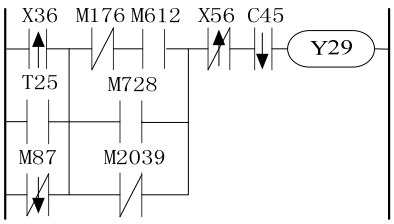

Figure 4. PLC Ladder Diagram

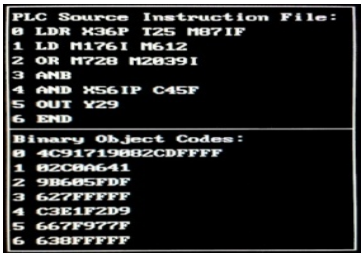

Figure 5. PLC Source Files and Binary Object Codes

7. Conclusion

In this paper a small PLC editor system overcomes the drawbacks in the existing programming device. The system can be used independent power supply under the off-line mode.It can store many sets of PLC programs. Operands logic operation instructions are designed to improve the efficiency of editing and compiling and the speed of PLC host. CAN bus expansion protocol solves the communication problem with PLC host. Via CAN Bus the system remotly monitors the state of PLC host and the informations of soft components.

Acknowledgements

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

References

[1] TIAN Wenqi. *Design and implementation of soft PLC system of the development based on IEC61131-3 standard.* Shen Yang: Graduate University of Chinese Academy of Sciences. 2011.
[2] SUI Tao, SUN Fanchao, LI Lijun, etc. Design of the Temperature Control System with the MicroLogix. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(8): 4651-4655.
[3] FANG Chengyuan, ZHANG Zhenguo. The factory electrical control technology. Bei Jing: Mechanical Industry press. 2006: 179-190.
[4] CAI Qizhong, GUO Yifeng, CHEN Wenhui etal. *General small programmable controller and its control method.* China Patent: 200710052941.X. 2009.08.19.
[5] ZHOU Hangci. Programming technology based on embedded real-time operating system. Beijing: Beijing university of aeronautics and astronautics press. 2011.
[6] BAI Jiang, WANG Yuhan, JIN Yongqiao. *Develop and study of numerical control system soft PLC module.* Machinery Design and Manufacture. 2011; (02): 138-140.
[7] Yan Yi, Zhang Hangping. *Compiling Ladder Diagram into Instruction List to comply with IEC 61131-3.*Computers in industry. 2010; 61(5):448-463.
[8] ZHU Huabing, LIANG Benlei, DONG Bolin, etc. Research on FPGA-based Programmable Logic Controllers' Technology. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(12): 4782-4786
[9] IANG Zongli, JIANG Shouxu. Compilation principle. Beijing: Higher education press. 2011.
[10] LU Lin, BAI Ruilin. *Design and implementation of a kind of PLC compiler.* Micro computer information. 2008; 24(12): 17-19.
[11] WANG Liming, XIA Li, SHAO Ying. Design and application of CAN Field Bus system. Beijing: Electronic industry press. 2008.3.
[12] YAO Xiaofeng, ZHAO Rui, XU Huipu. City Heating Network Dispatching and Management Information System. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(4): 851-857.