

A Multi-core Heterogeneous Programmable Automation Controller System of Construction Machine

WANG Guoqing*, YE Hong, Qu Jin, YANG Huaxin, LIU Chenghuan

Key Laboratory of Road Construction Technology and Equipment, Ministry of Education
Chang'an University, Xi'an 710064, Shaanxi, China

*Corresponding author, e-mail: wang_gq@chd.edu.cn

Abstract

There are two main issues, high cost and interrupt response delay for current PACs (programmable automation controllers) which are using high frequency single-chip microprocessors and commercial RTOS (real-time operating system) working by switching the tasks and interrupts. To solve these problems, a new low-cost multi-core heterogeneous PAC structure was proposed. Based on the traditional front-end of interrupt service without interrupt switching delay and the division & deployment of multi real-time tasks. We implemented a new programming language Engineer C including its syntax, compiler and an integrated development environment, which can dynamically divide the multi tasks and deploy the division to single low-cost microcontroller, then built the hardware prototype, which consists of a digital signal board, an analog signal board and a motion control board, every board has its microcontroller and communicates each other by an enhanced SPI bus. Testing results show the hardware prototype and IDE can be provided as a low-cost PAC solution for construction machine.

Keywords: *programmable automation controller, multi-core heterogeneous, integrated development environment, engineer C programming language*

Copyright © 2013 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

As the new generation of the industrial controllers, the programmable automation controller (PAC) is the core of the industrial automation platform. At present, the NI LabVIEW integrated development environment and its advanced PAC System-CompactRIO are the leading products in the PAC field [1]. In contrast, Chinese manufacturers have introduced PAC products, but most of them are based on the PLC model (such as Codesys, Isgraf, KW) or use WinCE, VxWorks and other foreign development system as the platform. In the case that the hardware IC manufacturers are increasingly inclined to provide a complete solution, the difficulty of hardware design is gradually reduced. Due to the lack of independent "PAC integrated development environment"-centered hardware and software development platform, the Chinese PAC manufacturers are still in the stage of fighting in hardware and price, can't provide more added knowledge value as NI or other products. With the development of the PAC, PAC platform mode for a particular purpose stands out from the initial generic PAC modes [2]. The lack of independent PAC integrated development platform has become a bottleneck restricting the development of Chinese PAC.

The embedded operating system (OS)-based PAC inevitably needs to process interrupts and switch time frequently. If a stack overflow or other circumstances happened, unpredictable errors would occur. And in low-cost occasions, single microcontroller (MCU) is widely used to complete the system development [3]. To this end, using a front-end mode based on the division of tasks for multi-microcontroller, we proposed a multi-core heterogeneous programmable automation controller system which has completely independent intellectual property rights. This system consists of a programming and debugging integrated development environment (IDE) for the multi-microcontroller, and an IDE-supported PAC system's hardware architecture in which the multi-MCU can be master-slave switched and flexibly configured.

2. Design of the Intergrated Development Enviroment

First of all, compared the characteristics of current mainstream programming languages, based on the core researching thought of "easy to use, practical to use, and sufficient to use", we independently designed Engineer C language, which is based on C language, combined with the characteristics of C + + / C #, simplified the syntax rules and improved the programming usability [4]. Then, we designed an ordinary text input environment and a graphical input environment. In order to make users without programming experience to use, an interactive automatic code generating system was designed. Finally, in order to facilitate the user to manage codes, we also set up a document automatic generating system.

2.1. Ordinary Text Input Environment

The following figure is the text input environment being constructed. From the figure, we can see that the Engineer C language has the same grammatical features with the commonly used high level programming languages. The text input environment implemented some functions which are not supported in Keil uVision such as variable tips, member hints, indent of code block, and so on [5].

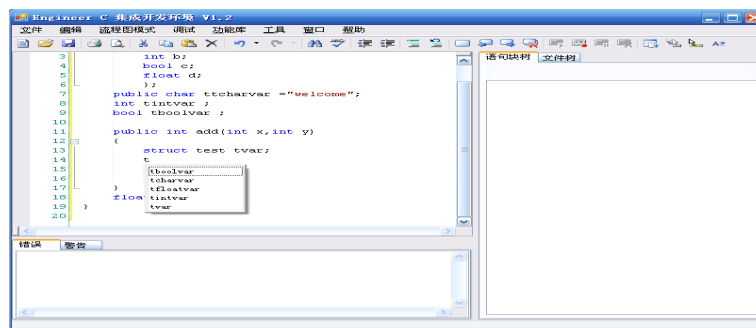


Figure 1. Screenshot of Engineer C integrated development environment

2.2. Graphical Input Environment

The several international widely used pieces of software have the function of graphical input, such as Matlab/Simulink, Labview, etc, and the graphical input mode is good enough to decrease the using difficulty for the users. In view of this, we also designed a graphical input environment of flow chart model [6, 7], as shown in the figure below:

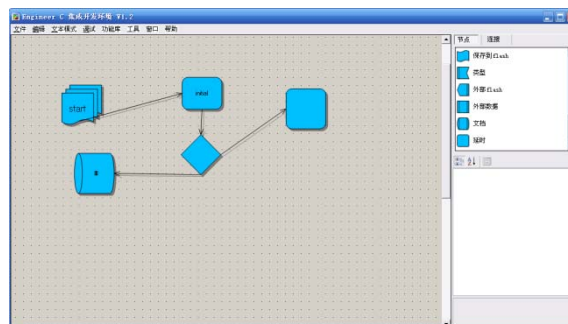


Figure 2. Engineer C Flow Chat Input Mode

2.3. Interactive Code Automatic Generating

Taking "Easy to use" as the central design idea, combined with the actual situation that the novices lack programming abilities, in order to facilitate the users to use, the IDE integrated an interactive code automatic generating system. This system can analyze the

information submitted by users, and automatically converts it into Engineer C codes. Combined with the above flow chart input mode, a simple system can be constructed without writing one line of codes. Overall, the system consists of two parts: (1) Using the interactive prompt mode, the system gets code parameters from specific input forms, and then automatically generates Engineer C codes; (2) Setting for the useful hardware modules: the entire software system services for hardware, it is not easy for the beginners to master the large amount of hardware functions quickly. To this end, hardware modules setup guiding program was designed. As long as the user determines which functions to be used and completed the setting of the modules' parameters, Engineer C codes will be easily attained.



Figure 3. Engineer C Automatic Code Generating System

2.4. Document Automatic Generating

For the situation that writing software document is a heavy workload, in order to facilitate the users to use, a software document automatic generating system was designed. This system can automatically extract the comments from the code lines in the text, combining with the analysis of the user's program, to generate the diagram of calling relationships between functions [8], as shown in Figure 4. Meanwhile, a function annotation generating system was provided. In this system, a text file will be generated as the output to form the user's software document.

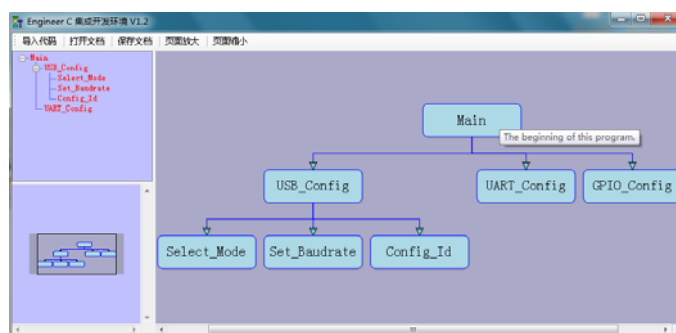


Figure 4. User's Software Document Automatic Generating System

3. Design of the Hardware Modules

The hardware part of this multi-core heterogeneous programmable automation controller system is mainly composed of six parts of a base board, three pieces of mother board, five pieces of son board, a diagnosis module, a synchronous debugger and an enhanced SPI bus. Among them, the base board contains power-supply facilities, communication devices and synchronous debugger buses, etc. The three pieces of mother board are analog signal motherboard, digital signal motherboard and motion control motherboard. They are connected to each other through the custom enhanced SPI bus and installed on the base board by pins. The diagnosis module is a level signal collecting circuit for

the logic analyzer. One of its ends is connected to the three pieces of mother board to realize the level signal collecting for the mother boards, and the other end is connected to a PC to achieve the analysis for the acquired level signal. The five pieces of son board include: DAC interface extended circuit, ADC interface extended circuit, I/O interface extended circuit, stepper motor controlling interface extended circuit and encoder interface extended circuit. They are installed on the respective mother board, to extend the functions of the three pieces of mother board. The two ends of the synchronous debugger are respectively connected to a PC and the three pieces of mother board, used to achieve the synchronous debugging for the motherboards [9, 10].

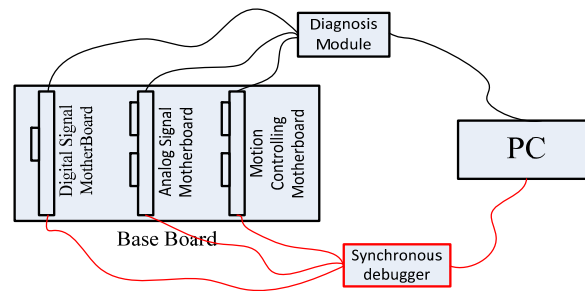


Figure 5. The Connection Diagram of the System's Hardware Modules

3.1. Mother Boards

Each motherboard has its own function modules and separate commutation modules: distributed control-supported field bus-CAN, RS485 interface, USB serial bus, downloading and debugging interface-JTAG, asynchronous communication-used universal serial data bus-UART and SPI bus interface, and so on. These modules can ensure the independent and asynchronous work of each motherboard. The three pieces of motherboard have the same level and they achieve the master-slave switch of interrupt response mode by the custom enhanced SPI bus. At any time in the running process, only one piece of mother board can be the master and the others should be slaves.

The three pieces of motherboard can be run separately and also can be combined to accomplish complex applications when needed. The analog signal motherboard is used to convert the collected digital signal to analog voltage signal by DAC and then convert the output of 0 to 2.5V voltages to $\pm 10V$ through the backward channel circuit, used in the analog control for the servo motor. The DAC interface extended circuit and the ADC interface extended circuit are connected to the analog signal motherboard. The digital signal motherboard is used for the input and output of the controller's switching values. The I/O interface extended circuit is connected to the digital signal motherboard. The motion control board is used to control the stepper motor and used for the input of the special signals (origin signal, positive limit, negative spacing, etc.). The stepper motor controlling interface extended circuit and encoder interface extended circuit are connected to the motion control motherboard.

3.2. The Diagnosis Module and Synchronous Debugger

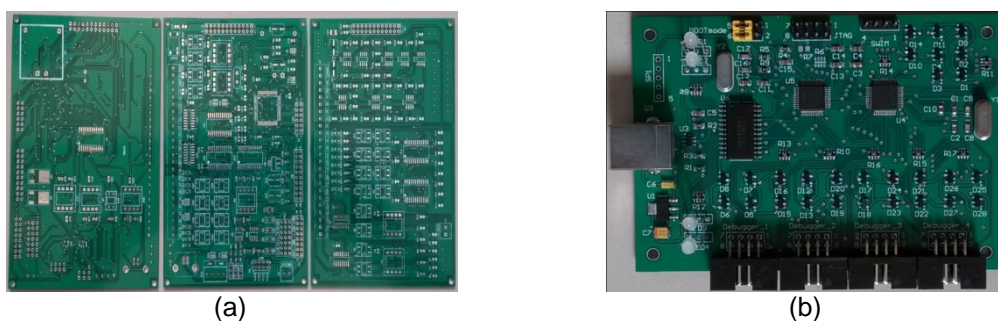


Figure 6. (a) The three pieces of mother board; (b) The synchronous debugger

The diagnosis module is a level signal collecting circuit for the logic analyzer, used to collect the level signal for the three pieces of motherboard, then analyzing and showing the collected level signal on the PC. The synchronous debugger and the three pieces of motherboard constitute a self-organizing reconstruction system, which can well ensure the reconfigurability and programmability of this multi-core heterogeneous programmable automation controller.

3.3. The Enhanced SPI Bus

The custom enhanced SPI bus includes 12 signal lines, which are data lines and clock lines (SPI_SCK, SPI_MISO and SPI_MOSI), chip-selecting signal lines (NSS0, NSS1, NSS2, NSS3 and SPI_NSS) and interrupt signal lines (IT0, IT1, IT2 and IT3), and up to four MCUs' synchronous communication can be supported. Among them, the data and clock lines are used for data transmitting among the host and the slaves, the chip-selected lines are used for the host to select target slaves and the interrupt signal lines are used to send interrupt signal from the slaves to the host, so as to interrupt the signal of the host and set itself to be the host and the others to be the slaves. This custom enhanced SPI bus can achieve the responding connection of interrupt mode among the host and slaves.

4. Conclusion

The multi-core heterogeneous programmable automation controller put forward in this paper, using a front-end mode based on the division of tasks for multi-CPU mode, can effectively solve the problems of heavier system load, poorer real-time responding ability for multi tasks and relatively lower processing efficiency, which exist in the traditional programmable automation controller system based on embedded operating system. The Engineer C integrated development environment provided a programming language of easy learning and use and set up the modules of text input programming, graphical programming, code automatic generating and software document automatic generating, etc. It provided an efficient way for the developing and debugging for the multi-core heterogeneous system.

Acknowledgement

This work was supported by the Chinese central financial scientific research fund for university (CHD2010JC091). We would like to thank the anonymous reviewers for their helpful comments.

References

- [1] Xiao SUN, Hao ZHOU. Research of Control System of Packaging Machine on PLC. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(6): 3060-3065.
- [2] James Truchard. *Letter from Dr. James Truchard, CEO and Cofounder of National Instrument*. <http://www.ni.com/eso/>
- [3] Jiejia LI, Rui QU, Yang CHEN. Construction Equipment Control Research Based on Predictive Technology. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(5): 960-967.
- [4] Wang Guoqing, Wang Yongxin. *Configurable Integrated Development Environment Software for the Controller of Construction Machinery*. Copyright of Computer Software, No. 0352306. 2011-11-29.
- [5] Yu Jiquan, Qian Wantao. *Study on an Integrated Development Environment for PAC with Complete Intellectual Property*. Applied Mechanics and Materials. Trans Tech Publications. Switzerland. 2012; 135-136: 944-949.
- [6] Rui Monteiro, Luís Paulo Reis, António Correia Pereira. Humanoid Dynamic Controller. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(8): 1936-1947.
- [7] Wang Guoqing, Li Feng. *Graphical Integrated Development Environment System for the Robot Brain*. Copyright of Computer Software. 2011; No. 0362147.
- [8] Wang Guoqing, Ye Hong. *Engineer C Graphical Programming System V1.0*. Copyright of Computer Software. 2013; 0522382.
- [9] Zhang Yunbo, *Study on the key technologies of intelligent control algorithm accomplished by PAC*. Journal of Changchun Institute of Technology (Natural Sciences Edition). 2010; 3: 51-54.
- [10] Wang Guoqing, Yang Huaxin. *Asynchronous Multi-core Programmable Automation Controller*. Chinese Patent. 2013; CN102929194A.