

Development of Advanced RISC Microprocessor Based Embedded System with Embedded Operating System for Robot Applications

Vaishak N L

Dept. of Mechanical Engineering
S.C.E.M Mangalore

Shilpa B

Dept. of Information Science and Engineering
C.E.C Mangalore

ABSTRACT

In this work, we introduce and developed the configuration of embedded system, and then present a robot control system based on an embedded operating system and Advanced RISC Machines using Linux network. By combining advanced RISC machines and ARM-Linux, this work involves development of embedded robot control systems for industrial utilization. In the implementation of this system, some features are taken into account such as hierarchy structure, modular hardware, and structured software, to make the system suitable for a variety of robots applications through some hardware adjustment and software customization only. The effectiveness of proposed work has been verified by a straight line Motion Demonstration of a 6-DOF series manipulator. The main purpose of this design is to meet the requirements of controlling the multi- robots system, meanwhile, to pay attention to some characteristics, such as: size, weight and power consumption, etc. to which the robot is sensitive. The design of embedded control system includes four aspects, i.e., system structure, functions, hardware, and software design. The project considers configuration techniques which can help to ensure that single-processor embedded systems are reliable.

Keywords

ARM Linux, Embedded system, RISC.

1. INTRODUCTION

With the development of the science and technology, the application of robots will be significantly increased in covering a number of fields, such as outer space exploration, ocean resource exploitation, etc. However, no matter what purpose for adopting the robots, almost everything made up of main two parts, i.e the mechanical body and the control system¹. The mechanical body not only represents the basic function of a robot, but also it determines that it is an automatic system. Dealing with works in dangerous and difficult environments, the robot should be facilitated with the abilities of thinking and making decisions to some extents, moreover, it needs

multi motors work together coordinately for the motion control of a robot. All of these require complex algorithms including motion control algorithm and pattern recognition algorithm³. As far as we know, it is difficult for robots we discussed above to carry a computer with them as a motion control system, meanwhile, it is also impossible for the robot control system with only low-performance microprocessors to deal with so many complex calculations. Fortunately, it is just a possible solution to overcome these difficulties through developing embedded systems using 32-bit microprocessors.

An embedded system is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function. A good example is the microwave oven. Almost every household has one, and tens of millions of them are used every day, but very few people realize that a processor and software are involved in the preparation of their lunch or dinner. This is in direct contrast to the personal computer in the family room⁸. It too is comprised of computer hardware and software and mechanical components. However, a personal computer is not designed to perform a specific function. Rather, it is able to do many different things. Many people use the term general-purpose computer to make this distinction clear. As shipped, a general-purpose computer is a blank slate; the manufacturer does not know what the customer will do with it. One customer may use it for a network file server, another may use it exclusively for playing games, and a third may use it to write the next great American novel.

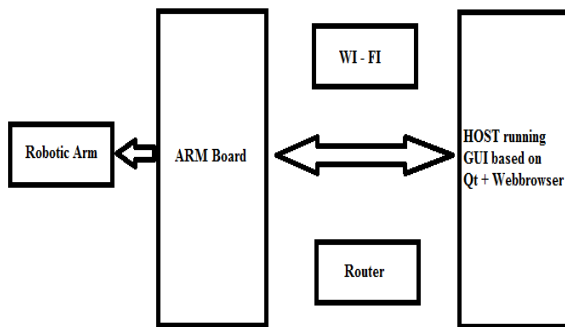


Fig 1: Functional Diagram

2. ADVANCED RISC MACHINES

Advanced RISC Machine - (ARM, Originally Acorn RISC Machine). A series of low-cost, power-efficient 32-bit RISC microprocessors for embedded control, computing, digital signal processing, games, consumer multimedia and portable applications. It was the first commercial RISC microprocessor and was licensed for production by Asahi Kasei Microsystems, Cirrus Logic, GEC Plessey Semiconductors, Samsung, Sharp, Texas Instruments and VLSI Technology. The ARM has a small and highly orthogonal instruction set, as do most RISC processors. Every instruction includes a four-bit code which specifies a condition (of the processor status register) which must be satisfied for the instruction to be executed. Unconditional execution is specified with a condition true.

Instructions are split into load and store which access memory and arithmetic and logic instructions which work on registers (two sources and one destination). The ARM has 27 registers of which 16 are accessible in any particular processor mode. R15 combines the program counter and processor status byte, the other registers are general purpose except that R14 holds the return address after a subroutine call and R13 is conventionally used as a stack pointer. There are four processor modes: user, interrupt (with a private copy of R13 and R14), fast interrupt (private copies of R8 to R14) and supervisor (private copies of R13 and R14).

2.1 Software Requirements

This section describes software requirements to our work. These are as follows:

- Programming language: C
- Linux kernel version 2.6
- GTK (GIMP Toolkit)
- GDB (GNU Debugger)
- Qt creator

2.2 Hardware Requirements

This section defines hardware requirement that are to be supported by the software, including logical structure, physical addresses, expected behavior, etc. The minimum hardware requirements are as follows:

- PC(Intel Pentium Processor with more than 400 MHz)
- S3C2440 Embedded Board
- Ethernet Cable
- Serial Cable
- USB cable
- USB network adapter
- Wireless Router
- Robotic Arm

3. SYSTEM ANALYSIS

3.1 Existing system

With the development of the science and technology, the utilization of robots will be significantly increased in a number of fields, such as

- Outer space exploration,
- Ocean resource exploitation, etc.

However, no matter what purpose for utilizing the robots, almost all of them are made up of two parts,

- The mechanical body and
- The control system.

The mechanical body not only represents the basic functionality of a robot, but also determines that it is an automatic system. Dealing with tasks in dangerous and difficult environments, the robot should be facilitated with the abilities of thinking and making decisions to some extents, moreover, it needs multi motors work together co-coordinately for the motion control of a robot.

3.2 Problems in existing system

Because of the limitation from the lower performance such as low running speed, low addressing capability and high power consumption, etc, the 8- or 16-bit microprocessors cannot fulfill the demands of some complex embedded applications. Complex algorithms including motion control algorithm and pattern recognition algorithm. It is also not possible for the robot control system with only low-performance microprocessors to deal with so many difficult calculations.

3.3 Proposed system

It is just a possible solution to overcome the difficulties occur in existing system, through Developing embedded system, 32-bit microprocessors. An embedded system is a special-purpose computer system, which is fully encapsulated by the controlling device, so there are some specific demands for each system, such as functions, reliability, cost, size, and power consumption, etc. Based on the computer technology, an embedded system is designed for specific-application with hardware and software that could be shaped to adapt the system requirements. As the core device of an embedded system, the embedded microprocessor can be an 8-,16- or 32-bit microprocessor.

3.4 Advantages of Proposed System

To overcome the disadvantages or limitations occurs in the present system, a trust based technique is developed. In the field of the 32-bit embedded system application, ARM (Advanced RISC Machine) grabs huge success. The ARM-kernel microprocessors present as lesser size, less power consumption, minimum cost, and relatively higher performance, etc. For instance, with a plenty of register and high instruction executing speed, most of data operations are completed in registers.

4. DESIGN

The main purpose of this design is to meet the requirements of controlling the multi- robots system, meanwhile, to pay concentration to some properties, such as: size, weight and power consumption, etc. to which the robot is sensitive. The design of embedded control system involves four main parts, like system structure, functions, hardware, and software design.

The embedded robot control system designed in this paper is applied to a 6-DOF serial robot and done fine characteristics of reliability, real-time and general-purpose capability. Moreover its small size and less power consumption fulfill the demands. With the absence of computer, the control system can carries out the control arithmetic on the server controller (ARM) and control the client controllers (DSP) through bus. Thus the robot is entitled the capacity of making decision individually to some extent. So far, there are some limitations in this developed embedded control system as well, such as the design of control arithmetic and improvement of functions, on which further study should be conducted. With the development of the science and technology, the application of robots will be significantly increased in a number of fields, such as outer space exploration, ocean resource exploitation, etc.

4.1 Hardware Design

The server controller is suggested to adopt S3C2410 supplied by Samsung as its CPU. The S3C2410 is a 16/32-bit, 266MHz, low power consumption, and high performance RISC microprocessor with ARM920T as its kernel, which is particularly suitable for real-time control. Meanwhile, it supports Window CE Palm OS Symbian OS, Linux and real-time operating system, etc.

Moreover, S3C2410 supplies abundant equipment inside to bring down the cost of the whole system and there is no need of collocating excess equipment any more. The function of integrated circuit include: separate 16KB instruction Cache and 16KB data Cache, MMU virtual memory management, LCD controller, supporting NAND Flash system induction, outside memory controller, 3-channels UART, four channels DMA, four-channels PWM timer, 117 currency I/O interface, 24-channels outer interrupt recourse, RTC with calendar, 8-channels 10bit ADC and touch screen interface, IIC-BUS interface, IIS-BUS interface, USB mainframe, USB equipment, two-channels SPI, and interior PLL clock times frequent count. The peripheral inside the chip is connected with the bus inside of the chip as well.

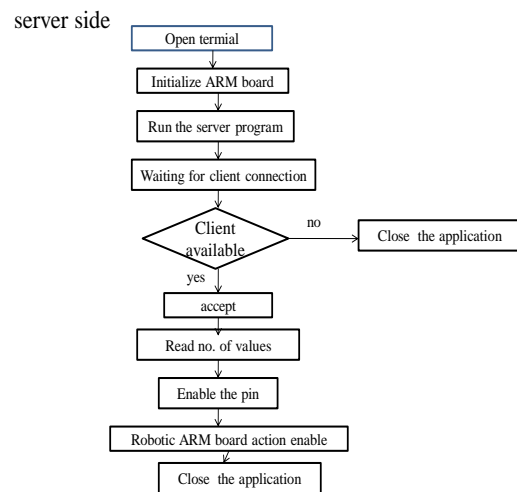


Fig 2: Flow chart of connection establishment at server side

There are differences in control algorithms for unlike type of robots, the difference is mainly shown by their server algorithms. The server algorithms are based on the specific tasks that the robot will carry out. Generally, a robot system needs only a server controller as the brain; the numbers of several client controllers are dependent on the specific task requirements. Through simply adjusting a

few interfaces or to add/delete some sensors will make the hardware suitable for various tasks.

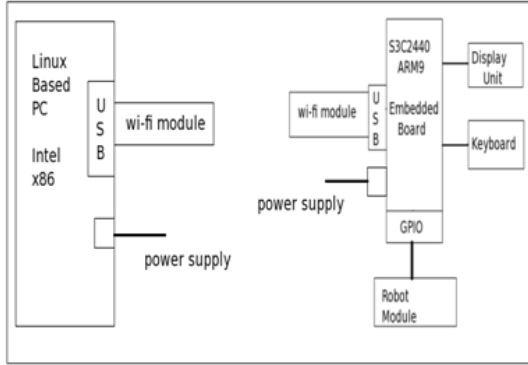


Fig 3: Architecture of hardware design

4.2 Software design

The operating system is one of the most essential parts in the total system. It is in charge of maintaining all available resource and distributing them among different tasks in order. According to different resource distribution strategies, there are two kinds of the operating system: one is Real-Time Operating System; the other is Time-Sharing Operating System. The operating system is one of the most important parts in the whole system. It is in charge of managing all available resource and distributing them among different tasks in order. According to different resource distribution strategies, there are two kinds of the operating system: one is Real-Time Operating System; the other is Time-Sharing Operating System.

This system software is mainly based on the RT Linux which is Real-Time and multi-tasks operating system and it is differentiated from the kernel of Linux by adding RT Linux kernel control between the process of Linux and hardware interrupt. It carries out a virtual interrupt mechanism in the kernel of RT Linux.

Client side

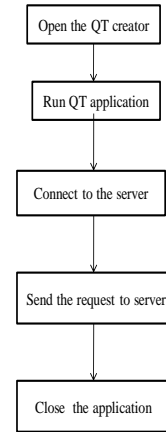


Fig 4: Flow chart of connection establishment at server side

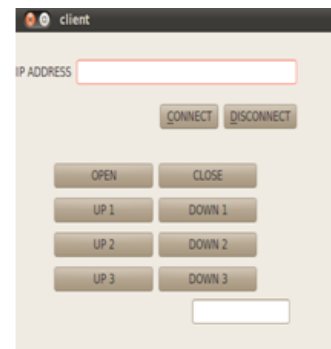


Fig 5: Qt software

MODULE1	GIVEN INPUT	MODULE2	GIVEN INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	REMARKS
Sender(QT)	Ip address	Receiver(GPIO)	Ip address	Connection must be established	Connection must be established	Tested ok
Sender (QT)	Click on open button	Receiver(GPIO)	Enable pin gpiog0	Gripper Open	Gripper open	Tested ok
Sender (QT)	Click on close button	Receiver(GPIO)	Enable pin gpiog1	Motor gripper close	Motor gripper close	Tested ok
Sender(QT)	Click on that up1 button	Receiver(GPIO)	Enable pin gpiog3	Motor wrist up1	Motor wrist up	Tested ok
Sender (QT)	Click on that down1	Receiver(GPIO)	Enable pin gpiog5	Motor wrist down	Motor wrist down	Tested ok
Sender (QT)	Click on that up2	Receiver(GPIO)	Enable pin gpiog6	Motor ankle up	Motor ankle up	Tested ok
Sender (QT)	Click on that down2	Receiver(GPIO)	Enable pin gpiog7	Motor ankle down	Motor ankle down	Tested ok

7. CONCLUSION

The embedded robot control system designed in this work is adapted to a 6-DOF serial robot and performs fine characteristics of reliability, real-time and general-purpose capability. Moreover its small size and less power consumption fulfill the requirements. Without computer, the control system can carries out the control arithmetic on the server controller (ARM) and control the client controllers (DSP) through bus. Thus the robot is entitled the ability of making decision independently to some extent.

So far, there are some limitations in this developed embedded control system as well, such as the design of control arithmetic and improvement of functions, on which further study should be conducted.

8. REFERENCE

[1] R. M. Murray, Z. X. Li, S. S. Sastry, "A Mathematical Introduction to Robotic Manipulation", America: CRC Press, 1994.

[2] W. H. Chen, G. L. Yang, Edwin H. L. Ho, and I. M. Chen, "Interactive Motion Control of Modular Reconfigurable Manipulators", Proceeding of the 2003 IEEE/RSJ Inter. Conference on Intelligent Robots and Systems Las Vegas, Nevada. October 2003.

[3] L. Romdhance, "Design and analysis of a hybrid serial parallel manipulator", Mechanism and Machine Theory, 1999, Vol. 34: 1037-1055.

[4] K. Yaghmour, "Building Embedded Linux Systems", America, O'Reilly, 2004.

[5] Michael Barr, "Programming Embedded Systems in C and C++", America: O'Reilly, 2001

[6] J. Catsoulis. "Designing Embedded Hardware". America: O'Reilly, 2001

[7] L. P. Yan, H. Zeng, K. Song, "Analysis and Design for Embedded Intelligent Family Gateways", Control and Automation, 2005, 2: 14-16.

[8] T. Z. Sun, W. J. Yuan, and H. F. Zhang, "Manual for Embedded Design and Linux Driver Development: Based on ARM9 Microprocessors", Beijing: Publishing House of Electronics Industry, 2005

[9] Y. Chen, M. Li, Y. Yang. "Analysis and Practice of Open-Code Embedded System Software: Based on SKYEYE and ARM Development Platform", Beijing: Beijing University of Aeronautics and Astronautics Press, 2003

[10] L. G. Zhou, Foundation and Practice for ARM Microprocessors, Beijing: Beijing University of Aeronautics and Astronautics Press, 2003

[11] J. G. Li, X. Y. Nie, and Z. M. Jiang, "Development Specifications for ARM Application Systems", Beijing: Tsinghua University Press, 2003

[12] H. P., Liu, W. J Wang, and Y. Jiang, "Application and Development on C Language for TMS320LF240x DSP", Beijing: Beijing University of Aeronautics and Astronautics Press, 2003

[13] S. Wang, J. J. Li, and J. H. Liang, "Embedded Autopilot Design of Micro-Robots", Journal of Beijing University of Aeronautics and Astronautics, 2005, 31(7): 775-779

9. AUTHOR'S PROFILE

Vaishak N L is working in Sahyadri College of Engineering and Management as Assistant Professor in the department of Mechanical Engineering. He received the M-Tech degree in Industrial Automation and Robotics in Srinivas Institute of Technology, Mangalore from Visvesvaraya Technological University and B.E degree in Vivekananda College of Engineering and Technology Puttur in 2008. He had presented three national conference and one international conference papers in the field of robotics. He also published five international journal papers in the field of Automation and Robotics. He is an active member of ISTE. His area of interest is total quality management in the preventive maintenance (in processing industry).

SHILPA B is working in Canara Engineering College, Benjanapadavu as Assistant Professor in the department of Information Science and Engineering. She received the M-Tech degree in Industrial Automation and Robotics in Srinivas Institute of Technology, Mangalore from Visvesvaraya Technological University and B.E from Vivekananda College of Engineering and Technology

Puttur in 2010 under Visvesvaraya Technological University. She had presented 3 national and international conference papers in various fields. And she also published two international journal papers in the field of Automation and Robotics. She is an active member of ISTE. She has done the project in the area of Matlab.

IJSHRE