Journal of Computer Hardware Engineering(2018)

Original Research Article

Design and Application of Motion Control System Based on Xscale and Embedded

Yunfei Zhao,Xiong Chen,Yueming Zheng

Computer Engineering College, Xining University of Technology, Qinghai, China

ABSTRACT

With the rapid development of computer technology, microelectronics and related technologies, embedded systems are being applied to more and more fields. In industrial control, especially in motion control, the complex control and processing, in the past more use of industrial computers and PLC as a processor. However, there are many disadvantages to the use of industrial computers and PLCs in smaller applications such as mobile robots. The embedded motion control system based on microprocessors and embedded Linux is small in size, has strong processing power and flexibility. More and more applications. In this paper, we study the motion control system based on embedded Linux. The host computer built by Intel Xscale architecture microprocessor PXA255 adopts the embedded Linux operating system. The DC motor digital servo control board constructed by AT89S52 is used as the lower motion control unit, which communicates through the serial interface to form a movement Control System.

KEYWORDS: Embedded system Xscale 51 single chip motion control serial communication

1. Introduction

1.1. Embedded System Overview

In everyday life, embedded systems are everywhere, such as mobile phones on every day, electronic watches on wrist, microwave ovens for cooking, office printers, car fuel injection systems, ABS systems and popular digital cameras, digital Cameras, PDA and so are the application of embedded systems. At present, with the rapid development of computer technology and information technology, as well as 3C (computer: communication: Communication, Consumer Electronics: Consumer Electries) the popularity of human into the post-PC era, a variety of information is very rich, digital information technology and Network technology is highly developed, people's requirements for the product is also improving, digital, intelligent products become the direction of development, a large number of information processing and digital and intelligent requirements make the application of embedded systems more widely. Embedded system to application as the center, emphasizing the size and function of the cut, is to complete the control, monitoring and other functions as the goal of the dedicated system. In the embedded application system, the system to perform the task of hardware and software are embedded in the actual device environment, through a dedicated I / O interface to exchange information with the outside world, the general implementation of their task program is not prepared by the user. Embedded systems are mainly used for a variety of signal processing and control, has been used in national defense, national economy and social life in various fields. Embedded systems in the number of applications far more than a variety of general-purpose computer.

1.1.1 Definition of embedded system

According to the definition of the Institute of Electrical Engineers, an embedded system is a device used to control, monitor, or assist a device, a machine, or a plant or plant. China is generally considered that the embedded system is application-centric, based on computer technology, hardware and software can be reduced, which can adapt to the practical application of the function, reliability, cost, size, power and other strict requirements of the special computer system.

1.1.2 Composition of the embedded system

Embedded systems are usually composed of embedded processors, embedded peripherals, embedded operating systems and embedded applications, most of the composition.

1. Embedded processor is the core of embedded systems.

The biggest difference between an embedded processor and a general purpose processor is that it works mostly in a system designed for a particular user group. It usually integrates many of the tasks completed by the board in the general computer into the chip, which is conducive to the miniaturization of embedded system design, but also with high efficiency, high reliability and other characteristics. International large hardware manufacturers almost have their own embedded processor, from the low-end 51 single-chip to now widely used ARM, MIPS, PowerPC, MC6800 and so on. In this paper, the host computer is used by Intel's Xscale architecture PXA255 processor.

2. Embedded peripherals

Embedded peripherals are other components that are used in an embedded system to perform ancillary functions such as storage, communication, debugging, and display, in addition to embedded processors. According to the functions of peripherals can be divided into the following three categories: (1) memory: mainly static nonvolatile memory (RAM / SDRAM), dynamic memory (DRAM) and Fash memory. Among them, Flash has a number of rewritable times, storage speed, large capacity and low price advantages in the embedded field has been widely used. (2) interface: the most widely used, including parallel port, RS-232 serial port, IrDA infrared interface, SPI serial peripheral interface, bus interface, USB universal serial bus interface, Ethernet. Network interface and so on. (3) human-computer interaction: LCD display, keyboard and touch screen and other human-computer interaction equipment.

3. Embedded operating system

In large and complex embedded applications, in order to make embedded development more convenient and fast, need to have a stable, secure software module set to manage the memory allocation, interrupt handling, inter-task communication and timer response , As well as the provision of multi-tasking, that is, embedded operating system.

4. Embedded application software

The embedded system application software is aimed at the specific practical field, based on the corresponding embedded hardware platform, and can complete the user's expected task computer software. The user's task may have time and precision requirements. Some applications require embedded operating system support, but in simple applications can also need no special operating system. Embedded application software is the focus of user development, the user's system development usually need to do the software work is mainly application software.

2. Xscale host computer

2.1. System overall design

The motion control system studied in this paper will be used as a development platform for mobile robots, requiring precise motion control. DC servo motor can be used as a power battery, suitable for mobile platforms. So the design of the motion control system using DC servo motor. Servo control unit using the main processor AT89S52, servo control chip LM629, H bridge power driver LMDl8200, MAXON DC servo motor RE25,500 line incremental photoelectric encoder HEDL-5540 constitute the servo system hardware platform. System by the powerful processing Xscale architecture processor PXA255 constitute the host computer, AT89S52 as the main processor and servo control chip LM629 constitute the next crew to serial communication. Constitute an all-digital servo control system platform. As shown in Figure 3 below. The servo control chip LM629 is a programmable all-digital servo control dedicated motion control processor. With 8-bit resolution of the PWM output, the internal trapezoidal speed graph generator, can be position and speed control. And its speed, position and digital PID controller parameters can be changed during the control process. This system has two kinds of control modes such as position and speed, high integration, less peripheral components, easy debugging, and simplified high-precision motion control system design. The sensor signal input in the following figure can be used for system expansion. In this paper, the sensor signal of the host computer is used instead of the sensor signal.**Xscale host computer**

Xscale architecture microprocessor is based on ARM V5TE architecture solution, is a full performance, cost-effective, low-power processor architecture. It supports 32-bit ARM instructions and 16-bit Thumb instructions and DSP instruction set, has been used in digital mobile phones, personal digital assistants, network products and industrial control and other occasions. Xscale architecture processor is Intel's main promotion of an ARM microprocessor. Motion control system of the host computer using Xscale architecture PXA255 processor to build. Figure 4 shows the PXA255 processor structure. PXA255 uses high-performance, low-power Xscale core, RISC technology and O. 18 micron process, and it also uses Intel's advanced media processing technology, including 40-bit accumulator and 16-bit SIMD to enhance processing power and audio / video decoding capabilities. PXA255 also has a rich expansion interface, in addition to SDRAM, DRAM and Flash and other system memory interface, but also support PCMCIA, Compact Flash, MMC / SD card expansion card, UART, Blue Tooth, IC, SSP, USB Slave and other serial interface As well as a large number of GPIO interfaces.

With PXA255 as a mobile robot processing platform, to carry out the subsequent visual and complex application development. In order to facilitate the future development of the system, the need for Boot Loader (boot loader) transplantation, where the use of open source U-Boot as a Boot Loader for transplantation. The following is a description of the overall configuration of the host computer:

1, The microprocessor: Intel Xscale PXA255, clock speed: 400Mhz; 32bit RISC architecture, 32K instruction cache.

2, Memory: SDRAM 64MByte, bus speed 100MHz; Intel NOR FLASH 32MByte.

3, The display system: LCD interface, support a variety of STN, TFT LCD, the maximum support 800X600 TFT LCD; 4 +2 programmable LED indicator.

4, Input device: 4 programmable keys; 4-wire resistive touch screen interface; SPI interface, expandable keyboard.

5, The audio system: stereo output, the output impedance of 32 ohms; stereo mix input, LINE IN, mono microphone input.

6, Communication interface: 10 / 100M adaptive Ethernet interface; IRDA infrared interface, support SIR and FIR; standard RS232 interface, support a complete control signal; standard UART interface, one can support high-speed Bluetooth transmission; USB DEVICE interface One, support USB protocol 1.1.

7, Debugging interface: a standard JTAG port, support for simulator debugging; CPLD JTAG one, support CPLD content updates.

8, Expansion card slot: CF card socket one, support memory card, WLAN, MODEM, ETHERNET; SD card socket one, support the memory card;

9, Other: 4 programmable IO port; 4 10bitADC interface; 1 PWM output port;

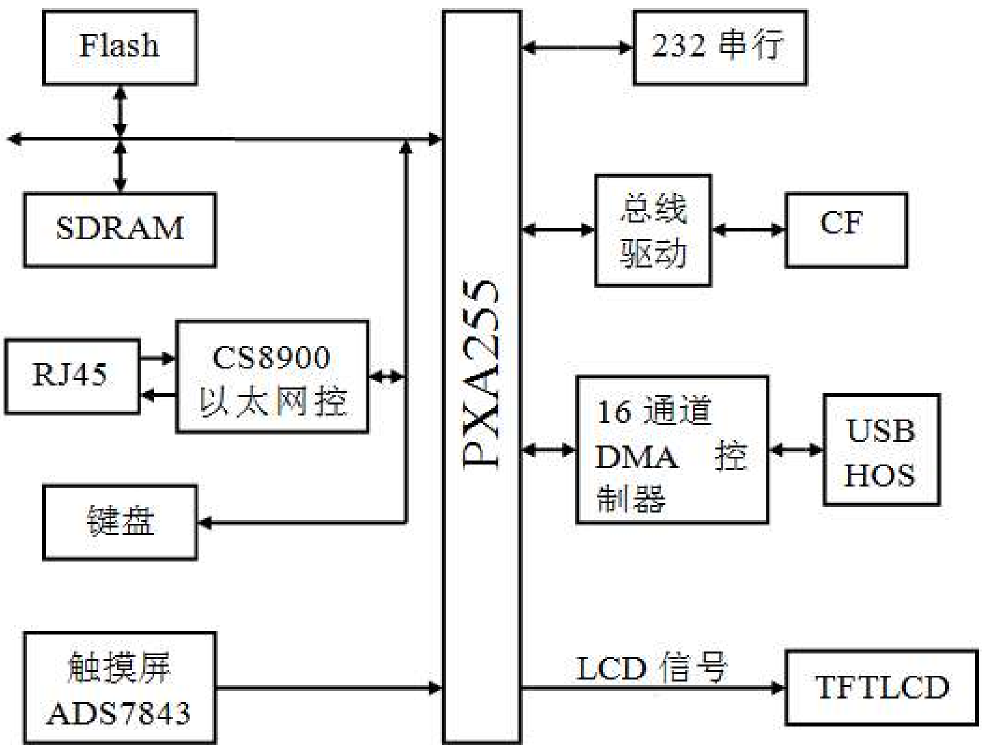


Figure 1.

3. Linux device drivers

Embedded system development, a lot of work is for a variety of equipment to write the driver, unless the system does not use the operating system, the program directly manipulate the hardware. Linux system, the kernel provides a protection mechanism, the user space process is generally not directly access the hardware. Linux devices are abstracted and all devices are treated as files. Equipment read and write and ordinary files. The user process accesses the device driver through the standard interface of the file system. The device driver performs these functions primarily:

(1) Detection equipment and initialization equipment;

(2) Transfer data from the kernel to the hardware and read data from the hardware;

(3) Reading the data sent by the application to the device file and the data requested by the echo application;

(4) Detection and processing equipment errors.

Linux systems divide the device into three types: character devices, block devices, and networks. Character device read and write in bytes, access is not cached. Block devices are read and write in blocks and have cache support to improve efficiency when accessed. Typical character devices include a mouse, a keyboard, and a serial port. Block device mainly includes hard disk, floppy disk and CD-ROM, a system to install the operating system must use the block device. Network equipment for communication, network equipment in Linux to do a special deal. Linux network system is mainly based on BSD Unix socket mechanism. Between the system and the driver defined a special data structure (sk buff) for data transfer. The system supports the buffering of sending and receiving data, providing flow control mechanisms and support for multiple protocols. All devices have a number of common drivers, the preparation of all types of drivers are common, the operating system to provide support for the driver is also roughly the same. These features include;

(1) Read and write

Almost all devices have inputs and outputs. Each driver is responsible for reading and writing the device.

(2) Interrupted

The interruption has an important position in the modern computer architecture. The operating system must provide the driver with the ability to respond to interrupts.

(3) Clock

In the realization of the driver, many places will use the clock. Such as some protocols in the timeout processing, there is no interrupt mechanism of the hardware polling and so on. The operating system should provide a timing mechanism for the driver, usually after a predetermined time has passed the callback registered clock function.

3.1 The structure of the device driver

Linux device drivers can be divided into the following three important components:

(1) Automatic configuration and initialization subroutine, responsible for testing the hardware equipment to be driven and whether it can work properly. If the device is normal, the software status required for this device and its associated device drivers is initialized. This part of the driver is only called once at initialization time.

(2) Services and I / O request subroutine, also known as the upper part of the driver. Calling this part of the program is due to the result of the system call. This part of the program in the implementation of the system is still considered to be the process of calling the same process, but by the user state into a core state, with the system call to the user program operating environment, which can be called in some Process running environment related functions.

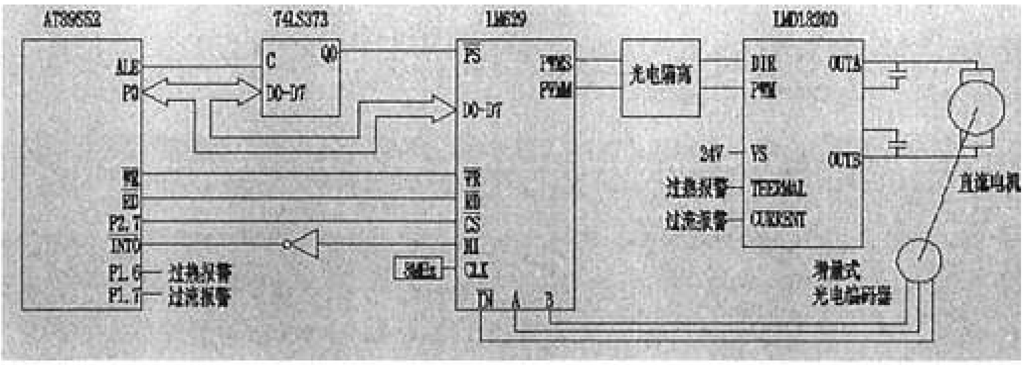
(3) Interrupt service subroutine, also known as the lower part of the driver. In the Linux system, not directly from the interrupt vector table call device driver interrupt service subroutine, but by the Linux system to receive hardware interrupt, and then by the system call interrupt service subroutine.

4. Servo motion control system program

DC servo systems are ubiquitous in modern production and life, especially in battery-powered mobile devices such as AGV, LGV, and robotic robots. DC system has an irreplaceable role, DC servo technology development has been very mature. By a number of dedicated integrated chips, such as position control chip LM629, pulse width modulation chip 637 UC, and power driver chip IR2110, LMDl8200, using them can constitute different power levels of DC servo system. DC servo motor servo control system consists of two: speed servo system and position servo system (also known as the follow-up system). In order to achieve precise control, the servo motion control system in the subject has two modes of speed mode and position mode.

As the platform will be used as a mobile robot development platform, requires a precise motion control, so the use of the Swiss company MAXON DC servo motor, battery as a driving force. Selection of motor model: RE2520W, parameters: rated voltage 24V, no-load speed 9550r / min, stall current 10400mA, the maximum continuous torque 26.1mNm. Equipped with planetary gear reducer: speed ratio 157 / l. Servo control board with the main

Processor AT89s52, servo control chip LM629, H bridge power driver LMDl8200, MAXON DC motor, 500 line incremental photoelectric encoder HEDL-5540 constitute the servo system hardware platform, the system has two kinds of position and speed control mode, high integration , Requires less peripheral components, easy to debug, simplifying the design of high-precision motion control system. The structure of the system is shown in Figure 4:



4.1. Servo system hardware interface analysis

1. The processor and LM629, LMDl8200 interface

The design uses AT89S52 microcontroller as the main processor for servo motor motion control. AT89S52 microcontroller's main job is to send LM629 motion data and PID data; LM629, LMDl8200 on the motor to monitor the operation: communication with the host computer and display the motor and the system working status. LM629 I / 0 port Do ~ D7 connected with the microcontroller's PO port, used to transfer data from the microcontroller and control instructions, from the LM629 motor transmission status and motion information. The P2.7 pin of the microcontroller is connected to the LM629 chip select as the address line for the selected LM629. After the D0 = 0, the microcontroller can write instructions to the LM629 or read the state from the LM629. When DO = 1, the microcontroller can be connected to the LM629, and the LM629 can be connected to the LM629. Write data, or read information from LM629. LM629 interrupt the lead HI by a non-gate and the microcontroller interrupt 0 connected, LM629 has six interrupt sources through the application for the interrupt, once the interrupt request, the microcontroller must read the LM629 state word to identify which interrupt occurred The Processor AT89S52 P1.7, P1.6 feet are LMDl8200 overheating and over-current signal input pin, the system work in the AT89S52 but also non-stop query LMDl82000 state, according to over-current and overheating signal conditions for system protection.

2. LM629, LMDl8200 and motor interface

LM629 output PWM direction and size of the signal PWMS and PWMM LM629 is built according to the built-in trapezoidal speed map generator, after optoelectronic isolation and LMD1820 D direction, PWM input pin connection. Increased photoelectric isolation is to prevent the motor drive circuit current and its changes and noise on the front-end control circuit. As the LM629 clock frequency is 8MHz, so here the photoelectric isolator requires a special choice, cannot choose the commonly used in the low-frequency devices, or optoelectronic isolation device from the input PWM signal rectangular wave, the output of the optocoupler device can only get trapezoidal wave, and cannot achieve accurate PWM control. So the selection of high-speed optical isolator 6N137. High-speed optocoupler 6N137 consists of phosphorus arsenide light-emitting diode and photosensitive integrated detection circuit. Received via photodiode

Signal and amplified by the internal high gain linear amplifier signal, the output by the collector open the door. The photoelectric isolator is short and low in the transmission delay time is short, typically only 48 ns, is close to the TTL circuit transmission delay time level, and thus the transmission speed can fully meet the requirements. In addition, 6N137 also has a control side, through the end of the control, can make the optocoupler output high resistance state. The LMDl8200 in accordance with the input PWM direction and size, the output voltage to pin 2, 10 directly drives the motor. DC motor installed incremental photoelectric encoder, the output IN, A, B three signals to detect the motor speed and location and access LM629 1 ~ 3 photoelectric encoder feedback pin, LM629 according to the feedback signal every other sampling Time to carry out the dynamic trapezoidal velocity diagram calculation, and the formation of a new PWM signal output provided to the LMDl8200, constitute a closed-loop control system, the motor position and speed to implement precise control.

4.2 Software design of servo motion control board

Main program design: the main program is the function of the system initialization, including the processor itself and LM629 initialization; monitoring and maintenance system operation; LM629 parameter transmission; status display. The key to the main program design is the following:

1. The main processor initialization

The initialization of the main processor AT89S52 includes: system environment and stack settings, timer initialization, serial communication initialization and so on.

1, Stack settings

The stack is the necessary resource for the processor to run, and the stack is the most commonly used when a function call, interrupt handling, and the need to save intermediate variables occur. MCS51 single-chip stack is the form of upward growth, the instruction system has a special set of stack pointer SP, initialization need to set the stack pointer and empty a memory as a stack to use, the specific size can be based on the size of the program, function call level and interrupt Processing to protect the number of field parameters to decide.

2, Timer initialization

AT89S52 chip contains three timers To, T1 and T2 and a Watchdog timer. To ~ T2 has three ways of working: way 0, way l, way 2 and way 3. Here the timer is used at 50ms timer, and timer T1 is used to generate the serial communication baud rate.

Timer settings are mainly on the AT89S52 several special function registers to operate, AT89S52 special function registers and the timer is related to the timer / counter control register TCON, working mode control register TMOD, interrupt enable register IE and interrupt priority register IP.

(1) TMOD settings

TMOD In the special function register, the byte address is 89H, no bit address. The high 4 bits of TMOD are used for T1, the lower 4 bits are for TO, and the four symbols have the meaning: GATE: gate control bit. GATE and the software control bit TR, the status of the external pin signal INT, the common control timer, the counter on or off, GATE = 0 is the software control bit TR to control the timer status.

(2) TCON setting

TCON In the special function register, the bit address (from low to high) is 88H ~ 8FI-I, because bit address, it is very easy to carry out bit operation. TFl, TR1 for timer T1; TF0, TR0 for timer TO. Two sets of symbols have the same meaning; TF: Timer / Counter interrupt request flag. When the timer is full of zero, TF = 1, and can apply for interruption; when the CPU interrupt and enter the interrupt service routine, TF automatically cleared. If the TF query, the timer back to zero, use the instructions to TF cleared. TR: Timer / Counter on / off control bit. IE1, ITl for external interrupt 1 (1NTl); IE0, IT0 for external interrupt 0. The meaning of the two sets of symbols is the same. IT: Falling edge / low level causes the external interrupt request to be selected. IT = 1, caused by the falling edge of the interruption; IT = 0, caused by the low level interrupt. IE: External interrupt request flag. IE = 0, indicating no external interrupt request; IE = 1, indicating that there is an external interrupt request. When there is an external interrupt request, IE automatically for l, the microcontroller CPU response to this interrupt, IE automatically cleared. In this case, timer 0 is used to interrupt the timer, reset TCON to 0X00 to clear the setting, and then set TCON = 0X01 (external interrupt 0 is set to falling edge trigger) for the LM629 interrupt pin HI input. After the initial value is set, the TCON bit operation is performed: Set TR0, TR1, start timer 0 and 1.

(3) The initial value set

It is important to set the timer initial value, need to set two special function registers. TH and TL. The initial setting of timer T0 is TH0 and TL0, which are the high byte and low byte of timer 0 respectively. The timing time is 50ms and the crystal frequency is 12MHz. The formula according to the timing time is: Timer initial value) × crystal period 12

5. Conclusion

The main task of this paper is to use Intel Xscale host computer and AT89S52 constructed by the motion control board together to form a motion control system, and the development of related content design.

1. The definition of embedded system is given, and the structure, principle, current situation and development prospect of embedded system are discussed. Analyzes the origin, development and characteristics of Linux operating system, and discusses the advantages and disadvantages of Linux in embedded system. Embedded Linux system with open source, the kernel can be cut, the advantages of low development costs, are getting more and more applications.

2. The analysis of Linux to the embedded operating system transformation, introduced how to generate from Linux for this article Xscale architecture CPU PXA255 ARM-Linux system. The establishment of the embedded Linux embedded development environment; introduced ARM-Liniux kernel reduction and compilation to ARM-Liniux file system to establish the completion of the ARM-Linux kernel to Xscale host computer transplant.

3. This paper discusses the realization and development process of the device driver under Linux, introduces the principle and implementation of the Linux module programming technology, and compiles the serial communication program of Xscale host computer and servo motion control board, and realizes the Intel Xscale Machine and servo motion control board composed of the lower computer communication between.

4. The hardware and software design of the servo control board are introduced. The hardware and software implementation methods and procedures of the servo control board are analyzed. The technical problems in the hardware and software design process are discussed. With the development of computer technology and network technology, Linux operating system as a representative of open source software has been more and more widely used in our country has been the support of many governments and enterprises are in a booming stage. Due to the special requirements of the performance and conditions of the motion control system, the research and development of the embedded system based on Linux, especially the adaptability and flexibility of embedded motion control system research has very good theoretical and practical significance The

References

1. Mao Dehua, Linux kernel source code scenario analysis (upper and lower), Zhejiang: Zhejiang University Press, 2004
2. Chen Zhanglong, Embedded technology and systems - InIel Xscale structure and development, Beijing: Beijing University of Aeronautics and Astronautics Press, 2004
3. Wang Xingzhi, AT89 series of single-chip principle and interface technology, Beijing; Beijing University of Aeronautics and Astronautics Press, 2004
4. Dai Mei calyx et al., Micro-computer technology and application - from 16 to 32: Tsinghua University Press, 1998
5. Liu Zhengrong et al., Embedded Linux application development Xiangjie, Beijing: Machinery Industry Press, 2004
6. Li Yubo, Linux C programming: Tsinghua University Press, 2005
7. Ma Zhongmei et al., ARM Linux Tutorial, Beijing: Beijing University of Aeronautics and Astronautics Press, 2004
8. Hu Hancai, Single-chip principle and interface technology, Beijing: Tsinghua University Press, 2004
9. Qin Jirong, Modern DC servo control technology and its system design, Beijing: Mechanical Industry Press, 1993
10. Alessandro Bubini et al., Linux DeviceDrivers, Beijing: China Electric Power Press, 2000