International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 4, No. 1, March 2015, pp. 22~27
ISSN: 2089-4864

The Experimental Platform Design for NCS Based on CAN Bus

Zhao Weiquan, Yuan Huaqiang, Wei Xiaorui
Computer College, Dongguan University of Technology, Dongguan 523808, China

Article Info

Article history:

Received Aug 1, 2014
Revised Nov 2, 2014
Accepted Dec 3, 2014

Keyword:

Networked control system
Experimental platform
CAN bus
Scheduling algorithm
Component

ABSTRACT

In order to facilitate research on the performance of Networked Control System, overcome the defects of unable to reflect the system performance, based on the CAN bus a kind of NCS experimental platform is designed. The system is composed of a main node and several slave nodes and Dc motor as the actuator. The master node is designed with ARM + U-boot + Linux structure, connecting with PC via Ethernet. For this experiment platform, using c + + Builder a scheduling algorithm performance testing software is developed, to further research on the effects of different scheduling algorithm on the NCS performance.

Corresponding Author:

Zhao Weiquan,
Computer College, Dongguan University of Technology, Dongguan 523808, China.
Email: zhaowq@dgut.edu.cn

1. INTRODUCTION

Networked Control Systems (NCS), with networked structure and decentralized management and network node intelligent, overcome the complex problem of traditional control system, such as reducing the amount of information transmission, difficult to extension, high cost etc. On the other hand, due to the communication network is introduced into the closed-loop control system, many problem has caused such as network induced delay, network congestion, packet loss, nonuniversal protocol. The discussion around these issues is becoming a research hotspot of scholars both at home and abroad. The study of network control system should reflect not only the control features (transition process, stability, etc.), but also the network features (bus type, network speed, etc.). The study method of networked control system can be divided into: simulation, experimental platform for validation, etc. [1-3]

In order to carry out the research and application of networked control systems effectively, the typical experiment platform is necessary. With CAN (Control Area Network) as control network, DC motor as actuator structures, a typical experiment platform is built [4-5]. Experimental platform consists of two kinds of node which connected with CAN, the master node and slave node.

The master node functions as: receive and process the information from slave node and network information, implement network resource scheduling by performing a control strategy and scheduling algorithm. The slave nodes (including motor driving node, the signal acquisition node) function as follows: realize driving motor, collecting control information, etc. The node diagram of experimental platform is shown in figure 1.

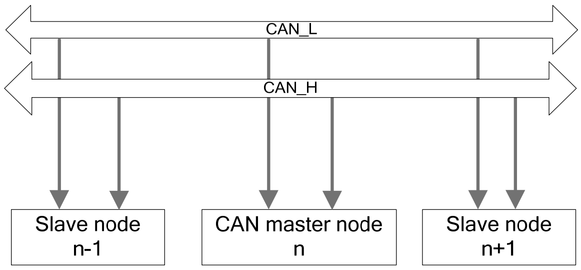


Figure 1. Node Diagram of Experimental Platform

2. THE CAN MASTER NODE DESIGN

A. The CAN Master Node Hardware Design

Combined with the advantages of Linux and LPC2294 with CAN bus controller integrated, transplanting Linux system into the ARM platform, the CAN bus node prototype is implemented [6]. The specific design of node includes: hardware selection, software platform choice, hardware circuit design, PCB (Printed circuit board) design, welding components and debugging, transplanting software platform, overall system test, etc.

B. The CAN Master Node Software Design

Based on PCB design of master node, software design begins. The specific work includes: transplanting Boot loader program U-Boot, transplanting Linux OS. The next job after completion of system boot also include: loading the root file system, storing external application file system, transplanting the underlying hardware driver (including Nandflash driver, CAN bus driver, Ethernet card driver, etc) [7]. System transplanting scheme shown in figure 3, which uCLinux is uCLinux200808 based on Linux2.6.25. After completing the kernel transplantation successfully, the root file system made of busybox1.0, YAFFS2 as NandFlash storage file system. The frame of software system porting is shown in figure 2.

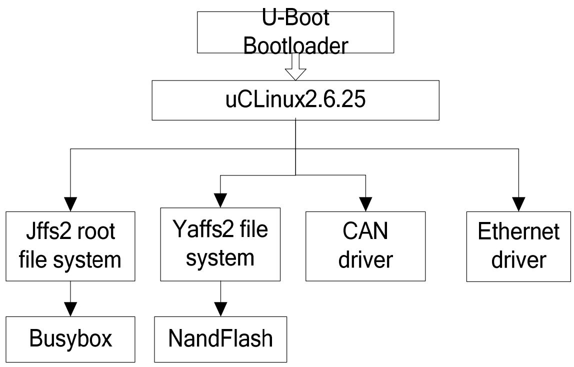


Figure 2. The frame of software system porting

C. U-boot configuration

Given U-Boot configuration does not support visual interface, to achieve the required functions needing modifying the macro definition contained in the header file [8]. U-Boot configuration shown in the figure 3, including the following five parts:

- Processor architecture: choose ARM7 as processor, supporting 16-bit THUMB instruction set, LPC2294.
- Ethernet card: select RTL8019, LAN IP and NFS server IP needs to be configured.
- NandFlash storage: select base operating range of the NandFlash, choose yaffs2 as NandFlash file system format.
- Command: set U-Boot environment variables and the supported Command for U-Boot.
- Linux startup: RAM starting address choice, the required transmission parameter selection for Linux starts.

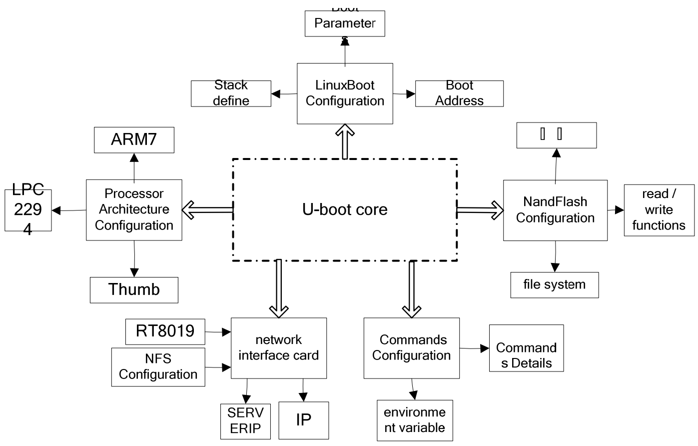


Figure 3. Configuration scheme of U-boot

D. Transplanting Linux OS

Because the embedded system is resource-constrained system, the generic Linux cannot be directly used. The embedded Linux, which Linux is cut and simplified, solidified in embedded system memory. The arch of the embedded Linux/subdirectory contains all the kernel code associated with the hardware. The core of Migration is reconfigured and compilation to the file in the arch/subdirectory. Because the uCLinux 2.6.25 support CAN drive, from the view of safety and reliability point, choose uCLinux2.6.25 as driver code. The transplanting process needs to complete the following:

- Installing patch for LPC2294;
- Configuration NandFlash driver;
- Configuration Linux core;
- Cross-compiling the Linux kernel.

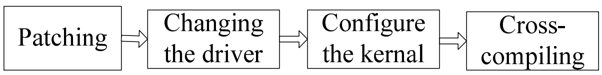


Figure 4. Linux Porting Procedure Scheme

Because there is not the MMU (Memory Management Unit) Unit in LPC2294, it is unable to run Linux. To run uCLinux, uCLinux patches have to be load. The required documents and patches as shown in table I.

Table 1. Node function description table

Name	Patch File Name
uClinux core	*uClinux-dist-20080808.tar.gz*
Linux core	*linux-2.6.25.tar.bz2*
Cross compile tools	*arm-uclinux-elf-tools-base-gcc3.4.3-20050221.sh*
Patch1	*linux-2.6.25-uc0-big.patch.gz*
Patch2	*linux-2.6.21-uc0-lpc2468.diff*
Patch3	*uClinux-dist-20070130-nxp-lpc2468.tgz*

3. PLATFORM TESTING (THE COMMUNICATION BETWEEN A TYPICAL SENSOR SLAVE NODE WITH THE MASTER NODE)

Use the CAN node as master node, with corresponding slave nodes (CAN-StartII from Guangzhou zhiyuan Co.). With stepping motor and motor driver and other auxiliary circuit structures, a typical networked control system experimental platform is built; system structure is shown in figure 5. Node function description table is shown in table II.

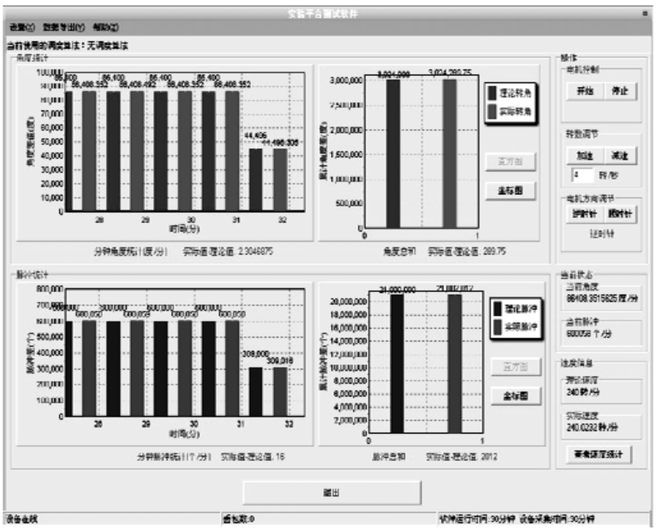
Figure 5. Whole structure diagram of experimental platform

Table 2. Node function description table

Node Name	Functional Description
Master node	*Control algorithm (motor speed PID control and collect network information to realize network scheduling algorithm*
Slave node1	*Receive control value from the master node, driving stepper motor rotation*
Slave node2	*Collect motor encoder pulse, sent to the master node, for motor speed control algorithm calculation*
Interference node	*The node has nothing to do with the control system functions, send periodic data transmission task, used to imitate the other nodes in the network to the influence of the network data transmission*
PC	*Running testing software of the experiment platform, display the current network status*

The communication baud rate is set to 1 MB/s, the CAN bus packet transmission time is 1 us in experiment. Assume that the packet from sensor to controller is the same size as the packet from controller to trigger, using common transmission 8 bytes data frame format, not considering a filling cases, need 111 bits, so the transmission time of a frame is 111 us.

Stepper motor is the control unit that turns the electrical power pulse signal into the angular displacement or displacement. As long as the number of control input electrical pulse frequency and the phase sequence of electric motor winding is in control, the required rotation speed and the direction of rotation can be obtained. The mechanical displacement and rotating speed is respectively proportional with the pulse number and frequency of the input motor winding. Rotating Angle is determined by the number of pulses, and the speed of the motor is determined by the pulse frequency.



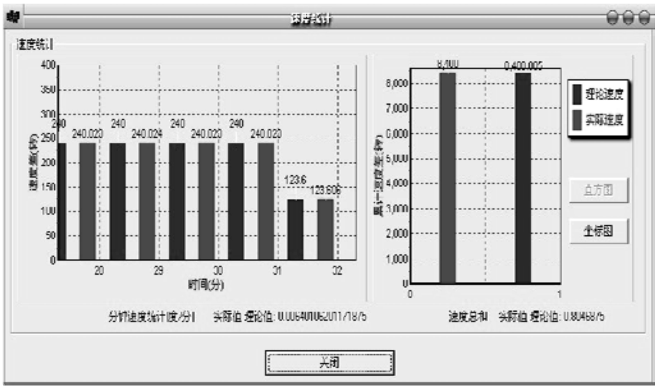


Figure 6. Test results figure without scheduling algorithm

Upon completion of the experimental platform hardware and driver design, using c++ Builder, the performance testing scheduling algorithm software for the experimental platform is developed. According to different scheduling algorithm, observe the deviation of the actual value with the given value for the stepper motor actual rotation angle and the speed, and draw the corresponding curve, to test the effects of different scheduling algorithm on system performance. The part of test software interface as shown in figure 6 [9-10].

For example, we used the design of experimental platform, respectively for RM and EDF classic real-time scheduling algorithm, dynamic bandwidth allocation algorithm based on feedback control 1 and 2, and fixed bandwidth allocation algorithm for performance testing. Observation system under different scheduling algorithm, the actual rotation Angle and speed stepper motor and the deviation of a given value, and combined with Matlab tool drawing performance curve, according to different scheduling algorithm to test the actual effect the performance of the system, and part of the test results as shown in the table III [11].

Table 3. Part scheduling algorithm test results under the experimental platform

Scheduling Algorithm	Result Analysis
No Scheduling Algorithm	*With the increase of nodes on the network bandwidth utilization ratio, the packet cannot timely access to transmission, deviation increases, system become unstable*
RM Scheduling Algorithm	*For the same interference node to the bandwidth utilization ratio, high priority node is always able to obtain network access to transmit data, ensure the control performance, but low priority nodes may be unable to get net access for a long time.*
EDF Scheduling Algorithm	*Relative to the RM scheduling algorithm, more fair to the allocation of the resources*
Dynamic Bandwidth Allocation Algorithm 1	*The system output curve shows that the overall performance is better. Adjust for a long time, but the system requires more system resources.*
Dynamic Bandwidth Allocation Algorithm 2	*System adjustment time is short, fast stable; But compared with the dynamic bandwidth allocation method 1, overall system performance*

4. CONCLUSION

Based on CAN bus a typical NCS experiment platform is designed. The CAN bus master node hardware and driver design process is highlighting introduced. Choose LPC2294 microprocessor hardware, through transplanting U-boot as a Bootloader and ucLiux as OS, a software framework is established. Combined with the hardware design of the experimental platform, using c++ Builder, a scheduling algorithm performance testing software is developed. For no scheduling algorithm, RM, EDF and two feedback control dynamic scheduling algorithm, the actual performance test is performed, and the preliminary conclusions are drawn. To the no scheduling algorithm, the system tends to be unstable. To the EDF scheduling algorithm, the allocation of resources is better than that of the RM scheduling algorithm. Feedback control dynamic scheduling algorithm can achieve better system performance. To further prove that the establishment of the experimental platform on all kinds of scheduling algorithm research on the performance of network control system has a high application value.

ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation of China (NSFC No. 61300197), Supported by Production Project of University-Industry cooperation of Guangdong Province and Ministry of Education of China (No. 2012B091100295) .

REFERENCES

[1] X.S. Li, *et al.*, "Analysis and Simplification of Three-Dimensional Space Vector PWM for Three-Phase Four-Leg Inverters", *IEEE Transactions on Industrial Electronics,* vol. 58, pp. 450-464, Feb 2011.
[2] Antsaklis P, Baillieul J, (Guest Editors). *Special issue on technology of networked control systems*. Proceedings of the IEEE, 2007, 95(1): 5-8.
[3] Baillieul J, Antsaklis P. *Control and communication challenges in networked real-time systems*. Proceedings of the IEEE, 2007, 95(1): 9-28.
[4] Velasco M, Fuertes JM, Lin C Xetal. A control approach to bandwidth management in networked control systems. Proceedings of the Industrial Electronics Conference. Piscataway NJ. USA, 2004: 2343-2348.
[5] Hong S. H., Kim W.-H., Bandwidth allocation scheme in CAN protocol, IEE Proceedings on Control Theory and Applications, 2000, 147(1): 37-44.
[6] Thomas Nolte, Hans Hansson, and Christer Norstrom, Probabilistic Worst-Case Response-Time Analysis for the Controller Area Network, http://www.mric.mdh.se.
[7] PHILIPS Semiconductors. LPC2119/2129/2292/2294 USER MANUAL [EB/OL]. [2008]. http://www.die.uniroma1.it/personale/ balsi/didattica/testi/testi SE-9.pdf.
[8] Wang tiegang, Wang Zhongqing. Design of Intelligent CAN Bus Node Based on LPC2294. *Microcomputer Information.* 2008 20 81-82 44.
[9] Yu Hui, Wen Yangdong, Liu Weiping. Ethernet Communication Embedded System Base on LPC2292. *Instrument Standardization & Metrology*, 2007 (01) 24-26.
[10] Thomas Nolte, Hans Hansson and Christer Norstrom, Minimizing CAN response-time jitter by message manipulation, Real-Time and Embedded Technology and Applications Symposium, 2003. Proceedings. The 9th IEEE, May 6, 27-30, 2003.
[11] Thomas Nolte, Mikael Sjodin, and Hans Hansson, Server-based Scheduling of the CAN Bus, http://www.mrtc.mdh.se.
[12] Zhao Weiquan. Study on Resource Scheduling of Networked Motion Control Systems. Guangzhou: South China University of Technology 2010.