

DC Motor Control through Remote using Embedded Linux

R.J.Sivakumar^{a,b}, N.P.Subramaniam^c, S.Ganesan^d

^a Post Graduate Student, Department of Electrical and Electronics Engineering

^c Lecturer, Department of Electrical and Electronics Engineer

^{b, c} College of Engineering, Anna University, Guindy, Chennai-25, India.

^d DYAnsys India Pvt Ltd, Chennai, India.

Abstract

Recently, the processor world has been growing with high speed advanced architecture compatible with any kind of source's. Equivalently, the Linux dominating the embedded market with the many features like real-time control, open source application, easy upgradability, compatibility with many processors etc. Which all add up the motor controllers with various functions not only real time control, communication capabilities but also wireless control from remote host computer and other motor controller for synchronization. This paper presents a brushless motor controller through remote (literally meaning wireless LAN control) by using the RTLinux. The proposed controller is composed of an embedded RTLinux board with high speed ARM11 microprocessor made by Freescale Semiconductor, capable of wireless (wifi) connection, a high speed current and position controller, and a BLDC motor driver for precision current control.

Keywords: LAN, WIFI, BLDC

1. Introduction

Nowadays, robotics requires various functions not only real time control or real time measurement of various external events but also image processing techniques, voice recognition techniques and wireless communication control techniques etc. But the processor should support all those features. To achieve this RTLinux plays an important role to overcome the complicated hardware system. Which also encourage the ubiquitous computing.

IEEE definition of real time is "a real-time system has deadline. For example: storing some data, or updati display. The task often is designed to repeat such activities ad infinitum, beginning each iteration at the occurrence of a hardware interrupt. This interrupt may be caused by a human clicking a button, a temperature reading, data from a visual sensor, or innumerable other possibilities. While time sharing OS like UNIX and Microsoft Windows strive to provide good average performance, for a real time OS, correct timing is the

key feature while throughput is of secondary concern. A system is considered hard real-time if it meets required maximum worst-case times. A soft real-time system is one in which its users are happier when the system responds optimally, but which is not considered to have "failed" when the system doesn't meet each and every desired response time. The real time Linux (RT Linux) is a hard real time variant of Linux that makes it possible to control robots, data acquisition systems, manufacturing plants and other time-sensitive instruments and machines. The RTLinux provides the capability of running special real time tasks and interrupt handlers on the same machine as standard Linux. These tasks and handlers execute when they need to execute no matter what Linux is doing. The worst case time between the moment when a hardware interrupt is detected by the processor and the moment when an interrupt handler starts to execute is under 15microseconds on the RTLinux running on a generic x86.A RTLinux periodic task runs within 35 microseconds of its scheduled time on the same hardware times are hardware limited and as hardware

improves RTLinux will also improve[1].

Chang-Hwan et al.[4] developed a brushless DC motor controller using embedded Realtime Linux utilizing the embedded realtime logic to control a BLDC motor and modified on removing the FPGA technique as high speed advanced ARM11 can also compensate that one. Hanamoto et al.[2] developed a sensor less speed controller for synchronous reluctance motor by using RTLinux utilizing the controller logic to interface with a PC, to condition the sensor signal and to generate pwm signals. This method gives an optimum design when resources are constrained by using a scheduling scheme. J.W.Dixon et al.[6] proposed an current control strategy for brushless DC motors based on a common DC signal ,which is implemented in current controller of BLDC motor.

This paper will concentrate on the wireless communication control handled by a high speed processor with minimal additional hardware for BLDC motor controller by using the RTLinux .The proposed controller is composed of an embedded RTLinux board with the ARM11 microprocessor made by Freescale Semiconductor, capable of wireless LAN connection, a high speed current and position controller and a two channel BLDC motor driver for precision current control.

The real time software module and communication programs are developed that are run in the embedded Linux board. Moreover, developing proportional integral (PI) controller to control the phase current ,pwm generation, sensor signal conditioning and a real time position controller is developed for the motor control.

2. System Configuration

Fig.1.Shows generic block diagram of a position controller. The position control is performed by the embedded Linux board operated by the RTLinux. The required reference phase current is transferred to the current controller . The advanced architecture and speed of the processor will compensate the burden on the embedded Linux board by combining the high bandwidth current control, and sensor interface logic to a same hardware. PI control is used for the current control and PID control is used for the position control.

Fig.2. shows the development environment for the RTLinux kernel, ramdisk , position controller, and IP

core. In development process consists of, developing the host control program for user interface, IPcore, real time motor controller, Ethernet communication program, and cross-compiler for embedded Linux board, which are operated by Linux. Moreover, the Linux host has various wireless networking capabilities such as bootp, tftp, nfs protocols.

Embedded RTLinux

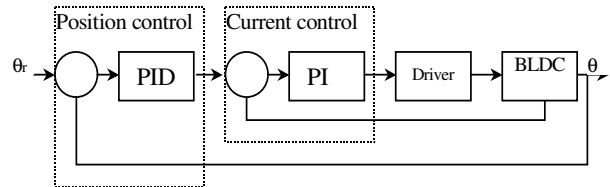


Fig. 1.Block diagram of a position controller.

The developing PCs and the embedded Linux board are connected through the intranet, Once the IP core (current control and position control module) is designed and synthesized ,it is included in the module as header file format of C language, which is compiled and downloaded to the embedded Linux board through nfs .

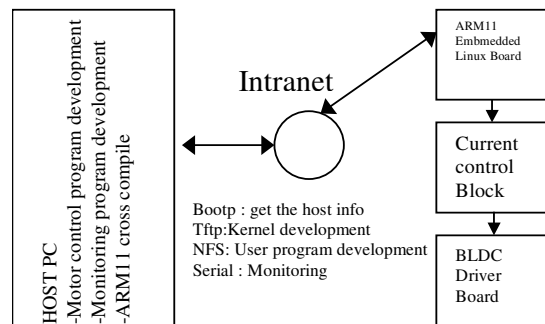


Fig. 2. The development environment for the RTLinux Kernel.

3. Real Time Control

3.1. Real-time controller architecture

Fig.3.show the block diagram of the controller, which is implemented within the embedded Linux board, operated by the RTLinux

There are two programs on the embedded Linux

board. One is an wireless networking configuration program that is run when the embedded Linux is booting, which is consist of a Linux kernel module program for accessing the wireless connection through Ethernet card of the strong ARM microprocessor, and an user program for managing the module program to initialize.

The other is the controller program that is run after booting is completed. Which consisted of three programs. One is the RTLinux kernel module program for real time control of the position which is PID control and other PI current controller program fetching the output of position control as input .The control loop is run every 200us.

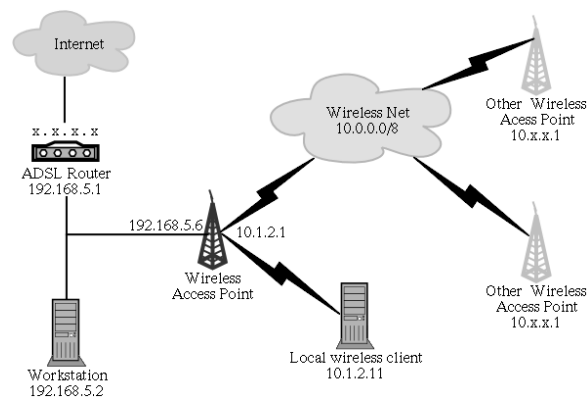


Fig.3. Wireless LAN Access point and router setup

The other is the Ethernet communication program to get the reference commands from the host PC and to send the sensor information to the host.The IEEE 802.11 wireless LAN protocol is used for wireless connection .The physical layer is modeled by transceivers operating at 2.472GHz with 1 Mbps data rate ,BPSK modulation ,15dBm transmit power and 100m coverage range. The kernel development takes place by TCP/IP and UDP protocol (minimum packet loss can be accepted without acknowledgment) is used for fast communication. The real time process and network client process are communicated by using the real time FIFO(RT-FIFO),an inter-process communication mechanism implemented on the RTLinux.

Fig.4 shows the detail description of the inter-process communication mechanism. There are five processes to run which are the network client, two

handler and two controller and four RT-FIFO , rtf1,rtf2 ,rtf3and rtf4 .Rtf1 and rtf2 is created in the network client program,Rtf3 and Rtf4 is created in the real time program. When network client program receives a command from the socket which is transferred from the host PC, it sends the command to rtf2.The event handler detects the command input, and forwards it to the rtf3.Then the position controller reads the command ,performs the control loop, and forwards it to the rtf4 ,then the current controller which get the input ,performs the control loop and returns the sensor information through rtf1.The network client reads the sensor information and sends the information to the socket, which will be transferred to the host PC.

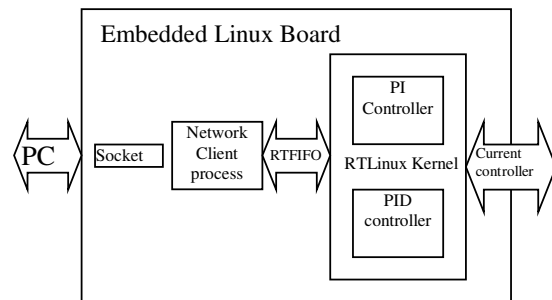


Fig.4 The block diagram of the position controller implemented in the RTLinux

3.2. IP Core Design

Fig.5.show The IP core is consist of two BLDC current controller which are same architecture including ADC interface for PI current controller and RDC interface for PID position controller, phase switching module, watchdog logic, sensor interfacing logic, bus interfacing logic.

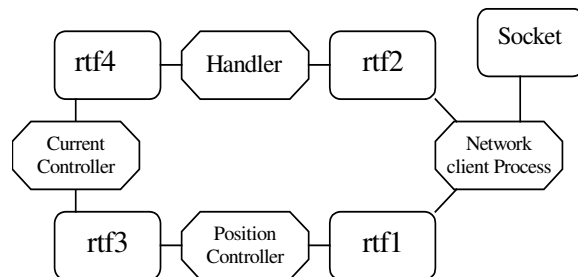


Fig.5. Inter-process communication mechanism of controller

3.3. Current Controllers

Various current control schemes are possible such as digital or analog implementation, and hysteresis or pwm implementation. But digital pwm control gives a better basis for converter design, acoustic noise prediction, and knowledge of the voltage. For these reasons this digital pwm control method is adopted in this paper.

The dynamics of the current loop will be limited by the sampling rate and a filter to smoothen out the current ripple caused by the switching frequency. There are several ways to improve the current control bandwidth such as increasing the sampling rate, increasing the switching frequency to reduce the current ripple and removing the current filter to reduce the time delay. If we try to increase the sampling rate, a high speed analog-to-digital converter and a microcontroller are necessary. Increasing the switching frequency also requires high speed power semiconductor devices, which is not a cost effective way. Therefore we try to remove the current filter to increase the bandwidth. Figure 6 shows the effect of the sampled current by using a zero-order-holder(ZOH) respectively. The filter is normally designed as an anti-aliasing filter with a corner frequency at least two times lower than the sampling frequency.

3.4. Sensor Interfacing Logic

The current controller has one Resolver-to-Digital converter and an analog-to-digital converter which are 12bit resolution and serial peripheral interface(SPI) logic.

4. BLDC Motor Driver For Current Control

A BLDC motor driver is developed by using the IGBTs. Fig.7 show the block diagram of the BLDC motor driver. Two channels of the motor driver is implemented in one board. Each channel is composed of 6 pieces of IGBTs made by International Rectifier co.ltd. and one gate driver.

A current sensor is attached as shown in Fig.8, which detects the bus current instead of measuring the phase current, which reduces the number of current sensor. However, the pwm signal. That is, when Q1 and Q5 are switched on, the current flows

through Q1-U-V-Q5 and flow out to the current sensor, then the measured current is positive. On the contrary, when Q1 and Q5 are off, the current flows through D4-U-V-D2 and flow back to the power supply, then the measured current is negative. Therefore, in order to get the correct motor phase current, the sampling of the current signal should be matched with the period that generates the positive current. The SPSS method in section IV considers this measurement constraint.

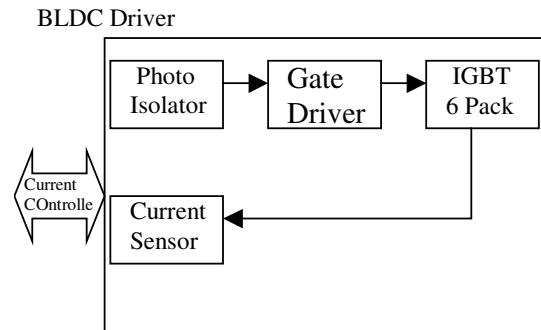


Fig.6. BLDC motor driver

Assume that the rotor angle is between 0 and 15°, when positive torque is required, pwm signal is applied to Q1 and Q5, which are switched on and off while regulating the phase current. When switched on, the current flows through Q1-U-V-Q5, and when switched off, the current flows through D4-U-V-D2. When negative torque is required, Q2 and Q4 are switched on and off, and the current path is Q2-V-U-Q4 when switched on, D5-V-U-D1 when switched off. This kind of switching method is called as 'hard chopping'. Since the voltage is switched at full range, the current is always controllable both at the motorizing and braking operations, therefore, it has good current control bandwidth. However, the voltage stress through the capacitor at the power supply line reduces the lifetime of the capacitor and the current ripple is large compare with 'soft chopping'. Since this paper considers servo control applications, the control bandwidth is more important than the others, so hard chopping is used, then the average voltage over one switching period gives (see Eq. 1).

$$V_{phase.avg} = D_{HC}V_{DC} + (1-D_{HC})(-V_{DC}) = (2D_{HC}-1)V_{DC}$$

Where D_{HC} is duty cycle of the pwm signal.

5.Conclusion

This paper presented a BLDC motor controller for wireless operation by using the RTLinux, embedded system using 32bit, high speed digital control microprocessor .The controllers are connected directly to the intranet through wireless card ,which is easily accessible not only in the laboratory but also in the whole world. Therefore ,this controller can be applied to any wireless applications.

Acknowledgment

Complete Embedded part of the project including porting Linux in ARM11 development board and tools has been sponsored and done in the R&D Laboratory of *DYAnsys India Pvt Ltd.*,Chennai , India.

References

- [1]. V.Yodaiken and M.Barabanov,"RTLinux version 2.2 Manual," VJY Associates LLC,<http://www.fsmlabs.com>
- [2]. T.Hanamoto,H.Ikeda, T.Tsuji and Y.Tanaka,"Sensorless speed control of synchronous reluctance motor using RTLinux." Proceedings of the power conversion conference ,vol 2,2002,pp,699-703
- [3] P.Kollig,B. Al-Hashimi and K.M.Abbott,"Design and implementation of digital systems for automatic control based on behavioral descriptions," IEE colloquium on Digital System Design Using Synthesis Techniques (Digest No:1996-029),1996,pp. 2/1-2/4.
- [4] F.Blaabjerg,P.C.Kjaer,P.O.Rasmussen and C.Cossar,"Improved digital current control methods in switched reluctance motor drives" IEEE transactions on power Electronics,vol 13, no 3,1999, pp.563-572
- [5] J.W.Dixon and I.A.Leal,"Current control strategy for brushless DC motors based on a common DC signal." IEEE transactions on Power Electronics,Vol 17,no 2,2002,pp 232-240.
- [6] D.Y.Ohm and J.H.Park,"About communication and current control methods for brushless motors" the 29th IMCSD symposium,1999,pp 1-11
- [7] Jaeseok kim,Linqing yang ,JenshanLin and Jonathan Liu "Analysis of a self-optimizing wireless