ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 5, May 2015

OPEN SOURCE MOTOR CONTROL USING XILINX SPARTAN-6

J.S.VARUNA VIVEKA, S.NAVANEETHAN

***Abstract*—Communication networks have been traditionally employed in motion control applications, especially within factory automation systems. While in the past they were merely used to exchange non time critical data (e.g. parameters and configuration data) nowadays they allow for much more powerful performance. Recently introduced Real-time Ethernet (RTE) networks, have been explicitly designed in order to cope with very tight timing constraints in terms of both determinism and real-time. In this project the focus is on Ethernet openPOWERLINK along with an example of its employment for a motion control application. In particular, the motor control application along with openPOWERLINK v1.8 is used to transmit the control parameters to the motor which includes the speed, position, voltage control, direction and also the feedback from the motor via quadrature encoder and hall sensor for implementing a complete control by means of the RTE network. It is a master-slave concept where the POWERLINK master is been implemented in both Linux and Windows platform with an upgraded version of POWERLINK stack v2.0. Initially Speed calculation based on QEP timer count and rate of change of angular position is been implemented and tested. The critical timing parameters in a POWERLINK communication cycle, the SoC Jitter is also monitored such that the delay generated in the network driver from receiving the packet until it is sent out to the network which determines the cycle quality is low. Then the effectiveness of the solution proposed will be checked by means of numerous test cases which take into consideration possible error scenarios deriving from the adoption of the communication networks such as transmission errors and communication delays to obtain the synchronization of the drives.**

***Index Terms*—openPOWERLINK,Quadrature Encoder,RTE, SoC Jitter.**

I. Introduction

Industrial Ethernet (IE) refers to the use of standard Ethernet protocols with rugged connectors and extended temperature switches in an industrial environment, for automation or process control. Components used in plant process areas must be designed to work in harsh environments of temperature extremes, humidity, and vibration that exceed the ranges for information technology equipment intended for installation in controlled environments.

Manuscript received May 02, 2015.

J.S.VARUNA VIVEKA*[1]*,PG Scholar, M.E. Embedded SystemTechnologies, Department of ECE, Anna University, Sri Shakthi Institute of Engineering and Technology Coimbatore, India

S.NAVANEETHAN*[2]*, Assistant Professor, Department of Electronics and Communication, Sri Shakthi Institute of Engineering and Technology Coimbatore, India.

Ethernet POWERLINK is a communication profile for Real-Time Ethernet (RTE). It extends Ethernet according to the IEEE 802.3 standard with mechanisms to transfer data with predictable timing and precise synchronization. The communication profile meets timing demands typical for high performance automation and motion applications.

It does not change basic principles of the Fast Ethernet Standard IEEE 802.3 but extends it towards RTE. Thus it is possible to leverage and continue to use any standard Ethernet silicon, infrastructure component or test and measurement equipment like a network analyzer.

openPOWERLINK is an Open Source Industrial Ethernet stack implementing the POWERLINK protocol for Managing Node (MN, POWERLINK Master) and Controlled Node (CN, POWERLINK Slave).

It implements all important features required by modern POWERLINK devices such as Standard, Multiplexed and PollResponse Chaining mode of operation, dynamic and static PDO mapping, SDO via ASnd and SDO via UDP, as well as asynchronous communication via a Virtual Ethernet interface.

II. Overview

A. OpenPOWERLINK Stack

The openPOWERLINK stack is a POWERLINK stack developed by SYS TEC electronic. SYS TEC published the POWERLINK stack under the Open- Source BSD license. openPOWERLINK contains all functionalities and services required for implementing a POWERLINK MN and CN.

It runs on Linux and other operating systems and platforms. Although there are Linux solutions available for other Ethernet based field busses, these are mostly Linux drivers for proprietary hardware. With the openPOWERLINK stack a pure software based solution is available which runs on a standard PC and no proprietary hardware is needed.

B. XILINX Spartan-6 FPGA

The low-cost Spartan-6 FPGA LX9 MicroBoard is the perfect solution for exploring the MicroBlaze soft processor or Spartan-6 FPGAs. It has several pre-built MicroBlaze "systems" allowing to start software development just like any standard off-the-shelf microprocessor. The Software Development Kit (SDK) provides a Eclipse-based environment for writing and debugging code. It is used as a general purpose prototyping and testing board.

The included peripherals and expansion interfaces make it ideal for a wide variety of applications. From a system running RTOS to a Linux-based web server, the Spartan-6 LX9 MicroBoard help to validate the next design idea.

C. Qt Creator

Qt Creator is a cross-platform C++, JavaScript and QML integrated development environment which is part of the SDK for the Qt GUI Application development framework. It includes a visual debugger and an integrated GUI layout and forms designer. The editor's features include syntax highlighting and autocompletion, but not tabs.

Qt Creator uses the C++ compiler from the GNU Compiler Collection on Linux and FreeBSD. On Windows it can use MinGW or MSVC with the default install and can also use cdb when compiled from source code. Clang is also supported.

D. Brushless DC Motor

The BLY17 Series Brushless DC Motors come in a compact package with high power density. These motors are cost effective solutions to many velocity control applications. They come in four different stack lengths to provide you with just the right torque for your application. A number of windings are available off the-shelf and all motors can be customized to fit your machine requirements.

The motors come in a standard 8-lead configuration with three wires for the phases and five wires for the hall sensors. We can also customize the winding to perfectly match your voltage, current, and maximum operating speed. Special shaft modifications, cables and connectors are also available upon request. The low voltage driver board is used along with this motor.

III. PROPOSED SYSTEM ANALYSIS

A. Qt PLK Master

Thehuman machine interface is designed for controlling the motor where the control parameters are mirrored as GUI buttons and knobs. It includes separate functions handing the speed data along with rotating direction and the start and stop of the motor. In POWERLINK terms two slaves with node ID's configured as 1 and 2 can be controlled using this GUI.

Fig 1: Qt – GUI

openPOWERLINK can be used to implement both, Managing Node (MN) and Controlled Node (CN) functionality. The following protocol features are supported by the stack.

- Modes of operation:
  Standard Node, Multiplexed Node (only CN), PollResponse Chaining Node, Async-only node
- SDO protocol:
  SDO/ASnd, SDO/UDP
- PDO mapping:
  static mapping, dynamic mapping
- Object Dictionary:
  User-configurable (at compile-time**)**

B. POWERLINK Slave

A DC Servo Drive that can be controlled over an Ethernet Network with limited motor control parameters acts as a Slave. The Slave includes

- Power electronics circuit to drive the motor.
- FPGA Spartan 6 unit for localized closed loop feedback control.
- Feedback Decoder IP core to calculate the speed and detect the direction of rotation.
- A control algorithm to control the motor speed and positon under various loads and randomly induced variations.

These together form the CN in the Slave. The Node ID's are fixed with respect to the number of slaves in the entire setup.The openPOWERLINK is running on a standard Microblaze CPU inside the FPGA. A POWERLINK Slave can be realized in an S6LX9 or larger FPGA.The MN waits for the response from the CN. This prevents collisions on the network and enables deterministic timing.

A fixed time is reserved in the network cycle for asynchronous data. Asynchronous data differs from cyclic data in that it need not be configured in advance**.** One asynchronous frame can be sent per POWERLINK cycle. The CNs can signal the MN in the poll response frame that they would like to send asynchronous data.

The MN determines which station is allowed to send, and shares this information in the SoA frame. Any Ethernet frame can be sent as an asynchronous frame (ARP, IP, etc.). However, a maximum length (MTU = Maximum Transfer Unit) must not be exceeded.

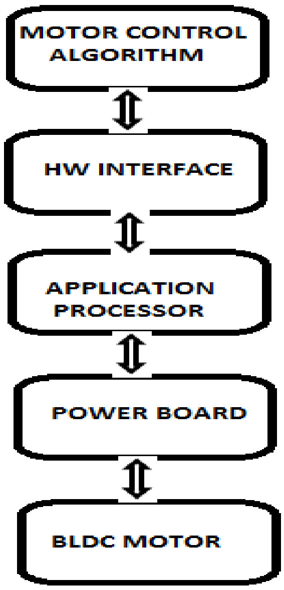

Fig 2: Flow chart of Complete Control

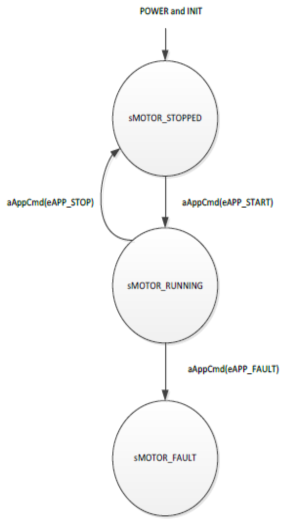

Fig 3: State Machine

The Algorithm functions on three basic genetic operators of selection, crossover and mutation. Based on the types of these operators GA has many variants like Real coded GA, Binary coded GA, These parameters have a great influence on the stability and performance of the control system. It focuses the Binary coded GA & finds the value of crossover, mutation of PID controller.GA is a stochastic global adaptive search optimization technique.

Based on themechanisms of natural selection. Recently, GA has been recognized as an effective and efficient technique to solve optimization problems. Compared with other optimization techniques GA starts with an initial population containing a number of chromosomes where each one represents a solution of the problem which performance is evaluated by a fitness function. Basically, GA consists of three main stages: Selection, Crossover and Mutation.

The application of these three basic operations allows the creation of new individuals which may be better than their parents. This algorithm is repeated for many generations and finally stops when reaching individuals that represent the optimum solution to the problem.

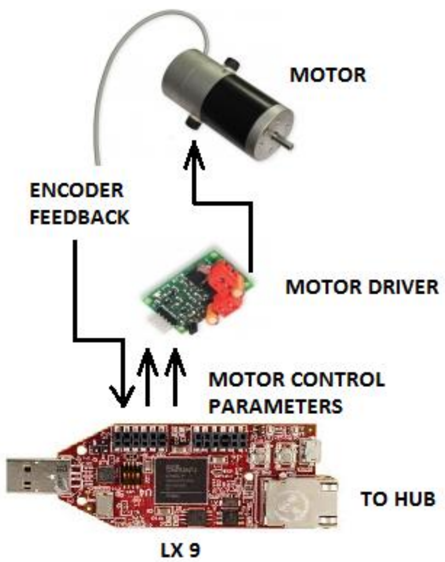

Fig 4: Connection Model

IV. Results and Discussions

High-End System:

The following test parameters were applied
Reference Cycle Time: 500 µs
Measured Cycles: 10 • 106

StressTests	Min Cycle	Max Cycle	Deviation
Idle	460.3 µs	548.8 µs	48.8 µs
CPU	474.6 µs	525.9 µs	25.9 µs
Network	438.1 µs	560.4 µs	61.9 µs

Table 1: Performance Measurement

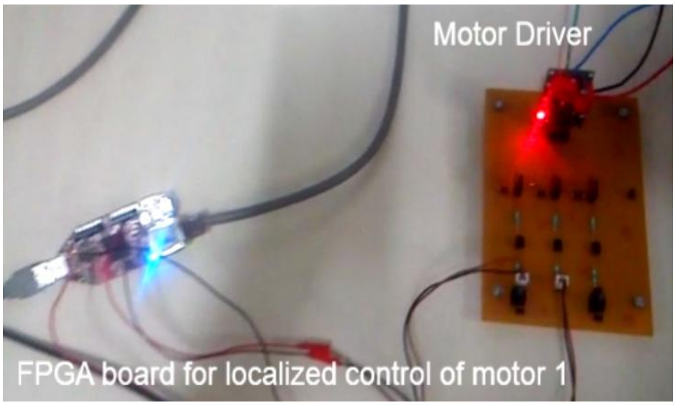

Fig 5: Application Block 1

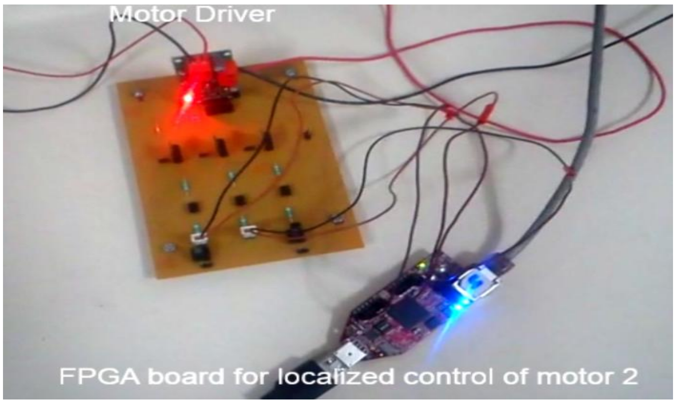

Fig 6: Application Block 2

Superloop kernel with high priority PWM-synchronised interrupts driving the scheduling of ADC and SINC current feedback measurement. Secondary priority level interrupts implement a basic scheduler that calls the main application state machine, from which the background 'state tasks' of each code module are called.

V. CONCLUSION

The performance evaluation showed that the Qt - GUI is an ideal platform for implementing a high-quality POWERLINK MN HMI. The high-resolution timers ensure a very high cycle time accuracy which is sufficient for many industrial applications. However, it is not clear why the results are best for the SoC jitter measurement when the CPU is heavily loaded. This needs further investigation. The cycle time could be lowered down to 250 µs which allows the implementation of industrial systems which require very low cycle times, such as motion control systems. The generated system load is low enough to implement small systems using an embedded platform or medium systems by using high end industrial PCs. Due to the measured values there is evidence that even larger networks could be realized without problems. However with respect to the motor control application the cycle time at which the network parameter are passed is tested with 500 µs.

REFERENCES

[1] Josef Baumgartner and Stefan Schoenegger, 'POWERLINK and Real-Time Linux: A Perfect Match for Highest Performance in Real Applications', *Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.*

[2] *openPOWERLINK Protocol Stack Source*, http://openpowerlink.sourceforge.net/

[3] *The RT Wiki, CONFIG PREEMPT RT Patch*, https://rt.wiki.kernel.org/index.php/CONFIG PREEMPT RT Patch

[4] Megha Jaiswal , Mohna Phadnis(2013), ' Speed Control of DC Motor Using Genetic Algorithm Based PID Controller', *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 3, Issue 7.

[5] Nagaraj.B, Murugananth.N, 'A Comparative Study of PID Controller Thing Using GA, EP, PSO and ACO' *ICCCCT-10*, IEEE Trans.

[6] Phillip David Power (2001), 'Non Linear Multi Layer Perceptron Channel Equalisation', Chapter 4:Genetic Algorithm Optimisation, *IEEE Transactions*, The Queen University of Belfast.

[7] EPSG Working Draft Proposal 302-C, Ethernet POWERLINK, Part C, 2009, *Ethernet POWERLINK Standardisation GroupV 0.0.3*.

[8] openPOWERLINK Protocol Stack Source, http://openpowerlink.sourceforge.net

[9] Mahony.T.O, Downing.C.J and Fatla. K (2000), 'Genetic Algorithm for PID Parameter Optimization: Minimizing Error Criteria', *Process Control and Instrumentation*, University of Stracthclyde, pg 148- 153