



International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 3, Issue 1, July 2014)

Porting an EtherCAT Slave Stack for PLC I/O Module

Piyush N Malpe¹, Prof. Satyajit Pangaonkar²

¹PG Student, Department of Electronics & Telecommunication Engineering
Dhole Patil College of Engineering, University of Pune, Pune, India

²Assistant professor, Department of Electronics Engineering
MIT Academy of Engineering, University of Pune, Pune, India

¹piyush_etrx@yahoo.co.in
²sapangaonkar@entc.maepune.ac

Abstract—

This research work covers the various factors to be considered for the EtherCAT slave stack porting to PLC input/output module. It covers the different aspects of EtherCAT slave module implementation, EtherCAT functional principle and slave stack integration in the user application. It also details the role of EtherCAT in PLC backplane communication as well as real time capabilities of EtherCAT networks.

Ethernet for Control Automation Technology (EtherCAT) - is an open source, relatively high speed fieldbus system based on Ethernet basics. The motivation for EtherCAT development was to improve Ethernet capabilities so that it can be applied to automation applications which require real time operations and low component costs. Capability of EtherCAT to use Ethernet or EBUS/Low voltage differential signaling as a physical layer opens the door for EtherCAT to be used for backplane bus communication.

Index Terms— EtherCAT, CANopen Over EtherCAT (CoE), PLC, Backplane, Fieldbus, Protocol, Real Time Ethernet, I/O Module

I. INTRODUCTION

EtherCAT is currently a market leading technology in the Real-time Ethernet domain offering high real-time performance combined with a wide set of functionality. The initiator and main developer has been the company Beckhoff which have become a significant vendor of advanced equipment for industrial automation. The EtherCAT technology now has broad support and involvement by many companies.

The move towards Ethernet as the basic communication platform is based on the excellent price/performance relationship of the technology. However, there are areas where there is a need for the very highest performance that

Ethernet technology is able to deliver, using either hardware or software augmentation or both. This includes devices within robotics, high performance PLC systems, motor control systems and instrumentation and I/O systems. Common to all these are challenging communication requirements on low latency, high update rates and high throughput. It is mainly for these systems and applications EtherCAT has been developed. [5]

I/O modules play an important role in overall working of advanced control system such as PLCs (Programmable Logic Controllers), and keep them aware of external conditions for effective monitoring and control. Hence it is of utmost priority that they should communicate with central processor on real time basis. Adaptation of EtherCAT stack enables the I/O module for fast data transfer. Also the motivational factors for EtherCAT stack adaptation in I/O module are, it is matured standard, can co-exist with the other Ethernet standards, Short scan time and use of CANopen application layer protocol with minimum stack modifications.

II. LITERATURE SURVEY AND PREVIOUS WORK

No significant use of fieldbuses has observed in PLC I/O module development. Very less information is available regarding the protocols and buses used for PLC backplanes. Some of the backplanes employ simple serial communication methods and some are CAN protocol based. Since most of the backplane communication methods are proprietary and used for internal communication between modules, manufacturers are not keen to open these details.

Considering the cost as of prime importance, industry always demands for the deterministic and fast monitoring & control in industrial applications.

With EtherCAT, it is possible to work even at I/O level using Ethernet principle that too with limited financial budgets. EtherCAT presents full compatibility with Ethernet, along with maximum utilization of the bandwidth offered by Ethernet. Hard real-time capability at reduced cost is one of the important features of EtherCAT network. EtherCAT combined with the CANopen can be used to give solutions for wide range of industrial applications.

realized completely in hardware, minimizing the processing stages, and thus only several nano seconds delay is introduced. The effective utilization of communication datagram is over 90%.[2]

III. FUNCTIONAL PRINCIPLE AND BASIC OPERATION OF ETHERCAT

A. Protocol

EtherCAT is an IEEE802.3 Ethernet based fieldbus system. The implementation of EtherCAT is inexpensive to implement which allows the system to use fieldbus technology in applications which had to omit fieldbus use in the past. EtherCAT is an open technology which is standardized within the IEC (International Electro-technical Commission).

EtherCAT uses standard IEEE 802.3 Ethernet frames, thus a standard network controller can be used and no special hardware is required on master side.

EtherCAT Protocol is evolved from standard Ethernet, it uses Ethernet's physical layer, employing RJ45 connector and EBUS on which LVDS signal is used for transmitting the data. This research work proposes EBUS as a data transfer medium which can achieve maximum data rate of 100Mb/s for full duplex data transfer. Since EtherCAT uses IEEE 802.3 Ethernet frames, no special hardware at master side, other than network controller is required. To realize the safe and reliable operation, redundancy and hot swapping features can be added optionally. [4]. As shown in Fig. 1, An EtherCAT frame is subdivided into the frame header and one or more EtherCAT datagram, each datagram addresses a separate device or storage area, defined by FMMU and SM. Maximum 1486 bytes of data can be transferred in EtherCAT datagram. Data sequence does not rely on physical sequence of Ethernet terminal in network, can be addressable at will. [2]

B. Functional Principle

As per the EtherCAT slave controller (ET1200) data sheet, "The frames are processed by the ESC on the fly, i.e., they are not stored inside the ESC" as shown in fig. 2. It runs on the principle of master-slave communication, the master sends Ethernet frame to the first slave, and the slave extracts or inserts process data from or into the frame quickly, and then sends the frame to the next slave. The last slave sends the frame back to the master. This process is

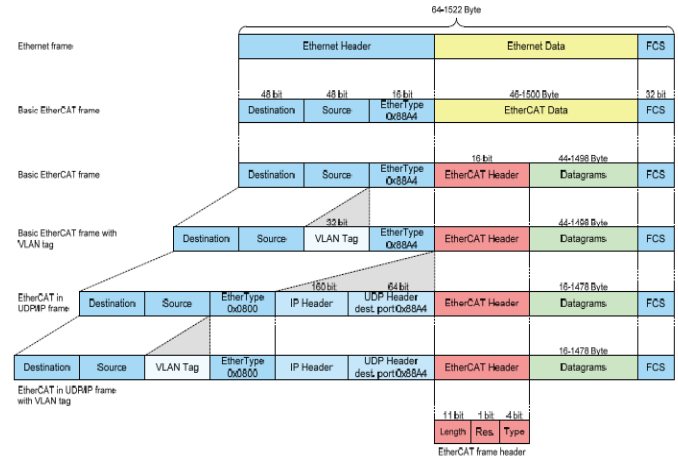


Fig. 1 Ethernet frame with EtherCAT Data [6]

The master uses embedded processor with standard network interface, the slave takes the special hardware as the ESC and common MCU as application controller. EtherCAT almost supports all the topology, including line, star, daisy chain, tree structure and so on. With the DC [10], EtherCAT allows precise synchronization between slaves. EtherCAT enables the network performance up to a new level. It can refresh 256 I/Os within 11 us, 1000 I/Os of 100 nodes within 30 us, and 100 servo axes within 100 us.

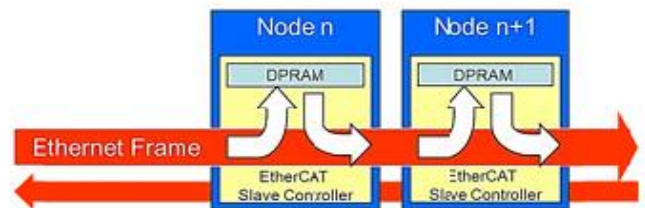


Fig. 2 Frame Processing On the Fly

C. EtherCAT Mailbox and CoE

It is shown that there is an Ethernet technology that retains CANopen to such a large extent, that even most of the CANopen communication protocol stack can be re-used. CANopen over EtherCAT features PDOs and SDOs as well as an Object Dictionary, the CANopen state machine and fully supports all CANopen device profiles. It is shown that EtherCAT provides a smooth migration path for CANopen devices towards the new world of Industrial Ethernet. [12]

CoE is one of mailbox types defined by EtherCAT. CoE defines a standard way to access a CANopen object dictionary and to exchange CANopen Emergency and PDO messages on an event driven path. EtherCAT can be realized easily for the devices with CANopen communication protocol.

IV. PROPOSED ARCHITECTURE

This research work intends to explore the design and development of EtherCAT based mixed digital input/output module for high end programmable logical controller (PLC). It also involves measurement of EtherCAT communication performance emphasizing on real time capabilities of EtherCAT networks. This work also explores the CANopen application layer protocol implementation over EtherCAT. CANopen is based on mailbox communication protocol implementation. Mailbox transfer is the standard way to exchange parameter data. This work implements CoE for data objects communication.

A. Functional Block Diagram

Figure 3 shows the block diagram of EtherCAT based I/O module. Here, CPU represents application controller; ET1200 is EtherCAT slave controller ASIC and 40 pin connector is of PLC backplane.

Interface between application controller and ET1200 is through SPI. Application controller acts as master and outs its data on falling edge of SPI clock and ET1200 (Slave) accepts its data on rising edge of SPI clock. Input circuit routes the signals from input channels to application controller. Interface between application controller and output switching IC is through SPI, application controller outs the data on falling edge of SPI clock.

Output switching IC (Slave) accepts the data on rising edge of SPI clock. EEPROM stores the EtherCAT specified parameters (mandatory and non mandatory) and non volatile application data as well. ET1200 accesses this EEPROM via I2C communication. LCD has inbuilt controller which implements SPI slave for external interface.

The EtherCAT Slave Controller, ET1200, processes the frames in hardware rather than application firmware. Thus, communication performance is independent from application processor power. So application controller has to deal with only user application requirements.

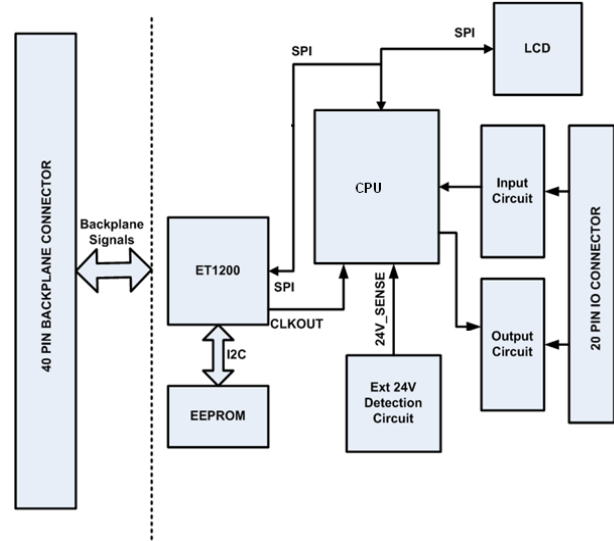


Fig. 3. Block Diagram of EtherCAT based digital I/O Module

B. Backplane Concept

The backplane must connect 4 to 32 modules. The EtherCAT is formed by two LVDS unidirectional buses, passing through all modules on the backplane.

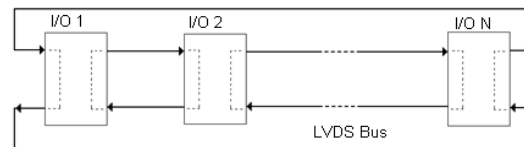


Fig. 4. EtherCAT Daisy Chain Topology

The Ether CAT is a dual, closed loop bus. The data flows from left to right in the upper part of the bus and from right to left in the lower part. This duality improves bus reliability. Also, each module can loopback the bus, in order to avoid missing paths due to a module failure.

The backplane must provide power to the EtherCAT slave chips and bus associated circuits.

C. EtherCAT Slave Device Structure

Figure 5 show typical EtherCAT slave device. An EtherCAT Slave Controller (ESC) takes care of the EtherCAT communication as an interface between the EtherCAT fieldbus and the slave application. EtherCAT slave controller implements the Data link layer (DLL). Service provider implements application layer protocol application. It uses LVDS (Low Voltage Differential Signaling) as physical layer within rack.

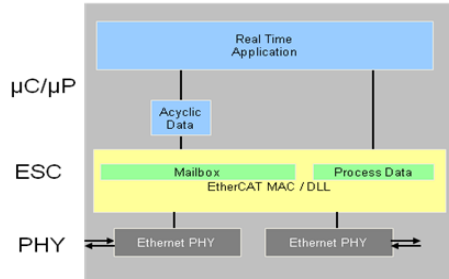


Fig. 5. EtherCAT Slave Device Organization

D. EtherCAT Stack Adaptation For Slave I/O Module

EtherCAT stack enables the IO module to be part of the PLC backplane network. Fig. 6 shows simple block diagram of the EtherCAT stack.

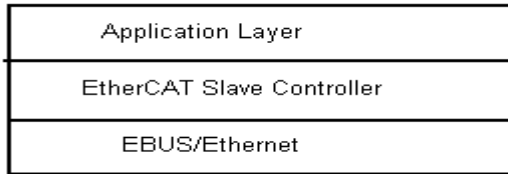


Fig. 6. Layered EtherCAT stack for I/O Module

Module application layer protocols are derived from EtherCAT application layers protocol that includes CANopen over EtherCAT (CoE).

The full duplex backplane bus is based on a point-to-point LVDS connection, running at 100Mbps.

E. CANopen Over EtherCAT

CANopen is a well proven, well established and very versatile Fieldbus technology, implemented in a large variety of devices. CAN unquestionably has distinct advantages just as low connection costs per device, true multi-master capability and outstanding error detection and handling features. However, for demanding applications like motion control or applications which require large network extension CAN is increasingly challenged by the upcoming industrial Ethernet technologies.

It is shown that there is an Ethernet technology that retains CANopen to such a large extent, that even most of the CANopen communication protocol stack can be re-used. CANopen over EtherCAT features Process Data Objects and Service Data Objects as well as an Object Dictionary, the CANopen state machine and fully supports all CANopen device profiles. It is shown that EtherCAT provides a smooth migration path for CANopen devices towards the brave new world of Industrial Ethernet [11].

Figure 7 shows typical CoE device structure.

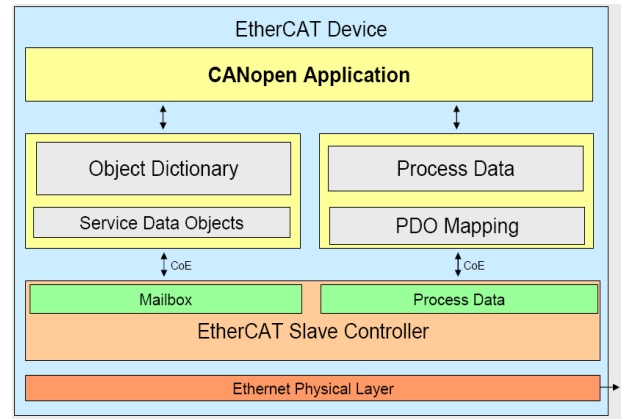


Fig. 7. CANopen over EtherCAT Device Architecture

V. RESULTS

EtherCAT performance can be calculated by measuring the cycle time. Performance is better if cycle time required for data communication is lower. Cycle time is calculated as time between packet forwarded by master and packet received by master. This broadly consists of the propagation delay along the cable and slave response time. In this research project, distributed clock feature of EtherCAT is not used which is responsible for the clock synchronization between the slaves (and the master) as well as for synchronous input/output events and signals. Disabling the distributed clock may increase the propagation delay slightly and hence the cycle time may vary/increase slightly.

In this project, Tcpdump is used to capture frames. Tcpdump is a common packet analyzer which is placed on master side. Wireshark protocol analyzer is used to display and analyze the EtherCAT frames captured by Tcpdump. Wireshark is a free and open-source packet analyzer. It is generally used for network troubleshooting, analysis, software and communications protocol development.

Frame snapshot in Fig.8 shows the frames used to read the inputs from modules (EtherCAT nodes) placed on PLC backplane. Logical Memory Read (LRD) command is used to read inputs. Here, slave puts read data into the EtherCAT datagram if received address matches with one of the configured FMMU areas for reading. Analysis shows that around 75 μs to 80 μs of cycle time is observed when master reads 72 input signals.

