

Development of Realtime EtherCAT Master Library Using INtime

Yongseon Moon*, Tuan Anh Vo Trong*, Nak Yong Ko**,
Dong Jin Seo*** and Seung-Woo Lim****

* School of Information Communication, Sunchon National University, Korea

** Dept. Control, Instrumentation, and Robot Engineering, Chosun University, Korea,

*** Robotics Institute, Redone Technologies, Ltd., Korea

**** Dept. of Information and Communication, DongKang College, Korea

Abstract

This paper proposes an architecture of a real-time EtherCAT master library called RtEML. The controls EtherCAT slaves under EtherCAT protocol in real-time. It provides a simple programming interface which is useful in developing robot application in C/C++ or C#. To achieve deterministic, hard real-time control in Microsoft Windows environment without additional hardware, INtime is used. Since INtime is designed specifically to take advantage of the powerful capabilities of the x86 processor architecture, the proposed RtEML achieves micro-seconds of real-time performance.

Key Words : EtherCAT; INtime; Real-time.

1. Introduction

Integrating multiple heterogeneous components in a system through a unified framework is vital to achieve an intelligent robotic system. Especially it is needed to get real time performance as well as using as many components as possible in the system. Middleware technology plays a key role in integrating components[1-4].

Hardware independent programming with real-time capability is one of the challenging new demand for complicated and highly dynamic systems such as robotics system[5]. The RtEML proposed in this paper is dedicated to support users and developers of robotics system. Especially, it provides flexible real-time Ethernet solutions instead of the solution using the traditional field bus protocols[6].

EtherCAT is an open real-time Ethernet network. EtherCAT slave system has flexible topology and simple configuration. EtherCAT master is optimized for processing data and can be transported directly within the Ethernet frame based on the special EtherType[7].

A robot system consists of several kinds of hardware. The inter connection between them is quite complex and it is required to control the system in real time. An RTOS can be used for this purpose. There are a lot of real time operating system(RTOS)s. In this paper, the authors introduce INtime based EtherCAT master. INtime is an RTOS developed by TenAsys[8].

The paper consists of 5 sections. Section 2 introduces INtime

environment. The communication between Windows and INtime is presented in section 3. Section 4 details the real-time EtherCAT master in the multi-tiers architecture. The result and the evaluation of RtEML are given in section 5. Lastly, section 6 concludes the paper.

2. INtime RTOS

Many applications require Windows as a component of the solution, but also require real-time or deterministic elements. The Windows operating system is a reliable, full-featured, and general purpose operating system designed from the ground up with features such as a pre-emptive, multitasking kernel, built-in protection, and security mechanisms. However, Windows is neither deterministic nor real time. To take all advantages of Windows and real-time performance, INtime is adopted in a robot system.

INtime software adds real time performance to Windows. INtime software consists of the following environments.

- Development environment: tools used to create real time applications that run in the runtime environment under Windows.
- Runtime environment: additions to your Windows system that provide RT performance for INtime applications.

Using INtime, developers can develop their real-time applications with Visual Studio or .NET which are familiar and convenient to use. Moreover, INtime supports some tools to debug and evaluate the performance.

3. Windows-INtime Communication

The typical INtime robot solution consists of two executable parts. The first one is a standard Windows process that provides

Manuscript received Jun. 1. 2009; revised Jun. 8. 2009.

This work is financially supported by the Ministry of Education, Science and Technology(MEST), the Ministry Knowledge Economy(MKE) and the Ministry of Labor (MOLAB) through the Hub University for Industrial Collaboration(HUNIC).

access to the graphical user interface and other Windows-specific functions. The second one is a real-time INtime process containing time-critical threads. Communication between the two parts is managed by the INtime Windows extension(NTX) library. Figure 1 shows the basic architecture of the connection between Windows and INtime environment which uses mailbox.

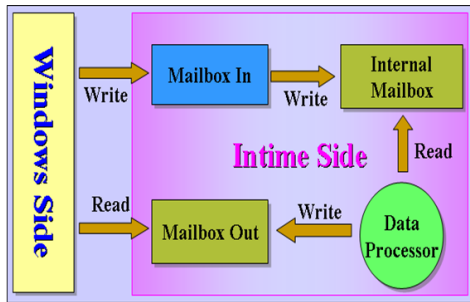


Fig. 1. Windows-INtime communication using Mailbox

Windows application threads communicate with the real time counterparts via the NTX API which allows them to share the contents of real-time objects and coordinate activities. This feature has the advantage that enables users to develop their robot applications in both Windows and INtime environments.

In the architecture shown on the Figure 1, the program in Windows uses NTX library, supported by INtime software, to connect to INtime process. The programs in INtime require not only 3 mailboxes for communication with windows but also a real-time thread to process data.

The followings depict the function of the elements shown in the Figure 1.

- (1) Mailbox In: Receives data from windows, passes the data to Internal Mailbox.
- (2) Internal Mailbox: Receives data from Mailbox In and waits for the Data Processor to get the data.
- (3) Data Processor: Reads data from Internal Mailbox, processes the data, and sends the response to Mailbox Out.
- (4) Mailbox Out: Stores the response data, waits for the command from Windows.

4. EtherCAT Master Architecture

4.1. EtherCAT master in multi-tier architecture

Nowadays, most of the robot systems aim to modularize both the hardware and software[9]. The authors focus to decompose the robot system into functional or logical components with well-defined interfaces adequate for communication across the components.

Basically, EtherCAT based robot system is divided into three layers. The top layer is robot application, in which all business requirements are implemented. The top layer communicates with EtherCAT devices through the second layer. The second layer is a real-time EtherCAT master library which is

represented as RtEML in the paper. RtEML connects the robot application and hardware systems which are called the slave applications. Slave application is the third layer. It receives commands from RtEML and controls robot directly.

RtEML is designed and developed to enable the system to be extended and integrated easily to a new system or new application. In addition, it enables programmers to code the functions using the languages C/C++ or C#. RtEML enables reusability of the components. Reusability is a crucial characteristic of a high quality software component. The structure of the EtherCAT based robot system is shown in Figure 2.

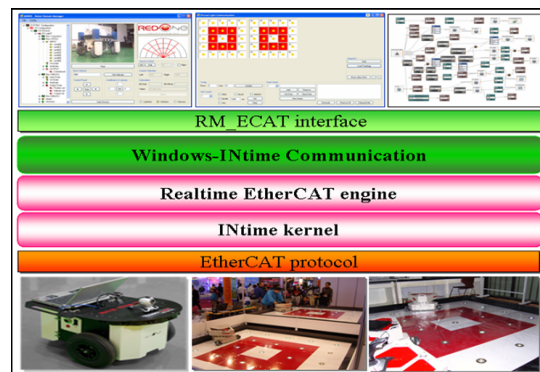


Fig. 2. The RtEML is a three layer model

RtEML is developed as a library based on EtherCAT. RtEML communicates with hardware devices through RJ45 port. The library encapsulates a set of functions. It connects robot applications to robot or other network services such as internet and CANopen. It resides between application layer and physical data layer in a multi-layered model. So it can be reused in other programs.

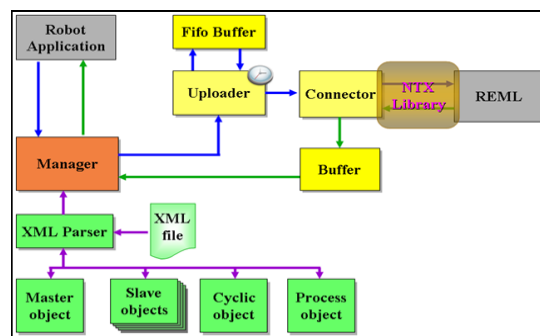


Fig. 3. Parsing XML data and passing to REML

RtEML is configured using XML configuration file. All information about EtherCAT slave is described in this file. The XML configuration file is parsed in Windows by a XML object. The information is stored in a FIFO buffer and passed to REML by an uploader object. The detail of this task is shown in Figure 3.

RtEML provides a simple C/C++ interface or an object interface which enables programmers to control the robots.

- (1) “startDevices” to create virtual objects and initialize the input information for these objects.
- (2) “addCommand” to add a command to queue.
- (3) “execute” to execute all commands in queue.
- (4) “read” to get information from robot or slave device (sensor, velocity)
- (5) “clear” to remove all commands in buffer.

Calling “startDevices” method to make robot ready to control is the first step for using RtEML to control an EtherCAT robot. The code below shows the sample to start device.

```

RtEML engine = new RtEML();
void StartDevice(){
int result = engine.startDevice();
if (result != 1)
ShowError("Could not start device: Error code = " +
result.ToString());
}
    
```

After starting device successfully, we can control robot by using “addCommand” method to add command to queue and “execute” method to execute all commands in queue. A sample for adding and executing commands is shown in the following code.

```

void SetWheelVelocity(int left, int right){
engine.addCommand(32, 240, left);
engine.addCommand(32, 344, right);
engine.execute();
}
    
```

4.2 Realtime EtherCAT master core design

In INtime master core, EtherCAT Master Controller (EMC) manages all EtherCAT Master business requirements. An IO Controller with low priority is used to get commands from Windows side. Another IO Controller is used to get data, which needs to update as soon as possible. Because the cyclic frame is sent in every time cycle, it needs to be cached to improve the system performance. The IO Controllers and other components will update data to cyclic frame if they want to change the data, which will be sent to robot in the next time cycle.

Besides, a Response Data Processor is used to process response data. This data is updated to data buffer and sent to Windows application if necessary. Figure 4 shows the architecture of a whole real-time robot system and some main objects inside the real-time EtherCAT master core.

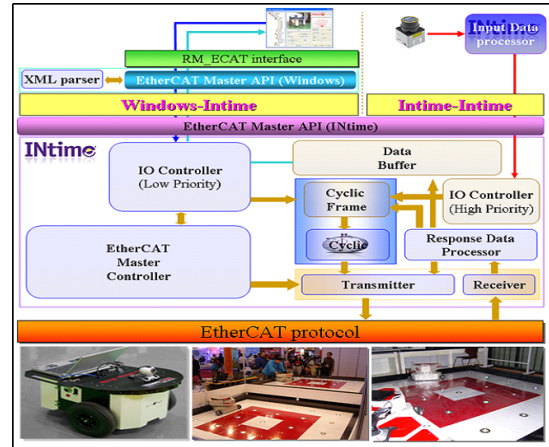


Fig. 4. Inside the Realtime EtherCAT master core

5. Performance Evaluation Results

The performance of RtEML is verified through experiments. The experiment uses Pentium(R) D CPU 3.40GHz, 1GB of RAM, Realtek RTL8139 Family PCI Fast Ethernet NIC, Windows XP SP2, INtime 3.0, 2 EtherCAT slaves, and EL9800.2 from Beckhoff. The Package size is 60 bytes.

In the experiment, all packets sent are captured using Ethereal. There are 5702 packets. The packets are divided into 7 groups according to the time period captured by the Ethereal. The groups are shown in the Table 1. As shown on Figure 5 through 8, average of 200 micro seconds of time period is needed for packet transfer. It means 5 KHz of frequency for the transfer of 60 bytes. In [10], IEEE 1394 was used for control of humanoids. It took about 90 micro seconds for the IEEE 1394 to transfer 64 bytes to a device if Xenomai is used. So, 180 micro seconds is needed for data transfer to and from two joints of a robot. In case of [10], as the number of joints increase, the time required for data transfer increases proportionally. In our development the data transfer rate is not so much affected by the number of joints. Therefore, as the number of joints increases, the proposed method exhibits better performance than the other methods.

Table 1. packet group

Delta t (µs)		Packets	Label
From	To		
0	150	1260	< 150
150	250	3136	150-250
250	400	1295	250-400
400	650	7	400-650
650	900	0	650-900
900	1100	0	900-1100
1100		4	>1100

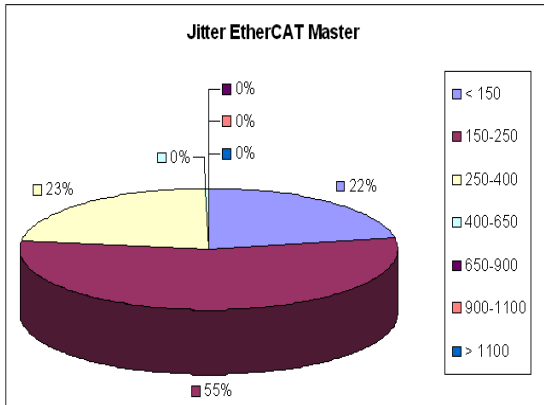


Fig. 5 Pie chart showing the update time in percentage

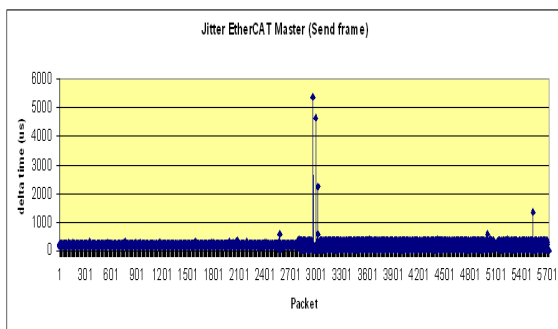


Fig. 6. A line chart with 5702 packets

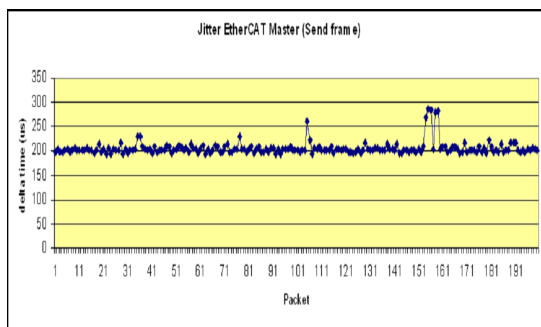


Fig. 7. A line chart with the first 200 packets

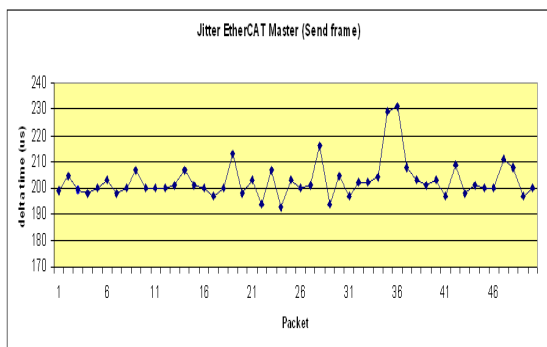


Fig. 8. A line chart with the first 50 packets

6. Conclusions

This paper proposes a Real-time EtherCAT master library(RtEML)implemented by INtime to control a robot system which consists of EtherCAT devices. The performance of the proposed system is evaluated. RtEML wraps EtherCAT protocol and enables the users to develop their application easier and faster. The users don't need to know the low-level architecture of the robot.

Robot middleware is a part of a robotic system. In order to incorporate new features into existing systems consistently, a robot middleware should be considered with multi-layered model of the robotic system.

RtEML has the following features.

- (1) RtEML can be used in any robot platform developed with EtherCAT.
- (2) RtEML can drive any devices with EtherCAT interface.
- (3) RtEML provides a simple interface.
- (4) RtEML has real time performance.

Besides, the modularization of RtEML promotes reusability of the robotic components

References

- [1] Sang Chul Ahn, Jin Hak Kim, Kiwoong Lim, Heedong Ko, Yong-Moo Kwon, Hyoung-Gon Kim, "UPnP Approach for Robot Middleware," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 18-22, pp.1959- 1963, 2005.
- [2] Hyun Kim, Young-Jo Cho, Sang-Rok Oh, "CAMUS: a middleware supporting context-aware services for network-based robots," *Proceedings 2005 IEEE Workshop on Advanced Robotics and its Social Impacts*, 12-15, pp. 237- 242, 2005.
- [3] Wei Hongxing, Li Shiyi, Zou Ying, Yang Liang, Wang Tianmiao, "A Middleware Based Control Architecture for Modular Robot Systems," *2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, 12-15, pp. 327-332, 2008.
- [4] Pablo Gil, Ivan Maza, Anibal Ollero, and Pedro Jose Marron, "Data centric middleware for the integration of wireless sensor networks and mobile robots," *Proc. of 7th Conference on Mobile Robots and Competitions*, , 2007.
- [5] Michael Barabanov, "A Linux-based Real-Time Operating System," *Thesis paper for Degree of Master of Science in Computer Science*, New Mexico Institute of Mining and Technology, New Mexico, 1997
- [6] D. Jansen, H. Buttner, "Real-time ethernet the EtherCAT solution," *Computing and Control Engineering Journal*, vol. 15, no. 1, pp.16-21. 2004
- [7] www.ethercat.org
- [8] www.tenAsys.com
- [9] Alessandro Farinelli, Giorgio Grisetti, and Luca Iocchi,

"Design and implementation of modular software for programming mobile robots," *International Journal of Advanced Robotic Systems*, vol. 3, no. 1, pp. 37-42, 2006.

- [10] M. Omar Faruque Sarker, ChangHwan Kim, Seungheon Baek and Bum-Jae You, "An IEEE-1394 Based Real-time Robot Control System for Efficient Controlling of Humanoids," *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.1416-1421, 2006.
-



Yongseon Moon received the B.S. degree, M.S. degree, and Ph.D. degree from the Division of Electronics Engineering, Chosun University, Korea. He is Professor of the school of information communication, Sunchon National University, Korea, from 1992. His research interests include industry communication, robot internal network

technology by EtherCAT, middleware and nerves network.

Phone : +82-61-753-1133
Fax : +82-62-373-2022
E-mail : moon@sunchon.ac.kr



Tuan Anh Vo Trong received the B.S. degree from the Information Technology Faculty, University of Natural Sciences, Vietnam. He is a graduate student of Sunchon National University and his major is Electronics Engineering. His research interests include industry communication, robot internal network technology by

EtherCAT, middleware and nerves network.

Phone : +82-61-750-3572
E-mail : votrongtuananh@yahoo.com



Nak Yong Ko received the B.S. degree, M.S. degree, and Ph.D. degree from the Department of Control and Instrumentation Engineering, Seoul National University, Korea, in the field of robotics. He is Professor of the department of Control, Instrumentation, and Robot Engineering, Chosun University, Korea, from 1992.

During 1996~1997 and 2004~2005, he worked as a visiting research scientist at the Robotics Institute of Carnegie Mellon University. His research interests include autonomous motion of mobile robots (collision avoidance, localization, map building, navigation, and planning), manipulator force/torque control, and incorporation of mobile robot technology into GIS.

Phone : +82-62-230-7108
Fax : +82-62-233-6896
E-mail : nyko@chosun.ac.kr



Dong Jin Seo is a Research Engineer in Robotics Institute at REDONE Tech. He earned B.A degree, M.S. degree and Ph.D. degree from the Department of Control and Instrumentation Engineering, Chosun University, Korea in 2000, 2002 and 2006.

During 2004~2005, he worked as a visiting student scholar at the Robotics Institute of Carnegie Mellon University, USA. His research interests are multi-robot cooperation, localization, navigation and modeling robot simulation systems with uncertainty.

Phone : +82-62-373-2024
Fax : +82-62-373-2022
E-mail : commidi@gmail.com



Seung-Woo Lim received the B.S. degree, M.S. degree, and Ph.D. degree from the Division of Electronics Engineering, Chosun University, Korea. He is Professor of the department of Information and Communication, DongKang College, Korea, from 1991. His research interests are embedded system, task planning for

intelligent robot, industry communication.

Phone : +82-62-520-2261
Fax : +82-62-520-2394
E-mail : lsw5802@dkc.ac.kr