Proceedings of the 4th International Conference of Control, Dynamic Systems, and Robotics (CDSR'17)
Toronto, Canada – August 21 – 23, 2017
Paper No. 128
DOI: 10.11159/cdsr17.128

A Performance Evaluation of Open Source-based EtherCAT Master Systems

Soyeon Kim, Euncheol Shin
Korea Institute of Industrial Technology
143, Hanggaul-ro, Sangnok-gu, Ansan-si, Gyeonggi-do, Republic of Korea
nansy710@kitech.re.kr; unchol@kitech.re.kr

Abstract **–** In this study, real-time performance of open source-based EtherCAT master systems is examined. EtherCAT is widely used in the automation field because of the low system setup cost and excellent performance. There are various commercial products and open source projects because EtherCAT is an open field bus with an international standard. The EtherCAT master stacks are particularly being applied to the existing open source-based real-time linux platforms. In this study, the EtherCAT master stack of IgH, which is the most widely used open source EtherCAT master, was applied to an open source-based real-time linux platform. Then, a comparison experiment of real-time performance was conducted with the Beckhoff's TwinCAT, which is the standard of the EtherCAT master system.

Keywords**:** EtherCAT, Master, Performance, Real-Time

1. Introduction

EtherCAT is an Ethernet-based field bus system developed by Beckhoff. It is widely used in the automation field due to the low system setup cost and excellent performance. There are various commercial products and open source projects because EtherCAT is an open field bus with an international standard.[1] The EtherCAT master stack by IgH and open source EtherCAT master stacks like SOME(Simple Open EtherCAT Master) by OECS(Open EtherCAT Society) are particularly being applied to the existing open source-based real-time linux platforms, RT-Preempt, RTAI(Real Time Application Interface), and Xenomai.[2][3] The EtherCAT master stack by IgH is being used for open source projects for automation of systems such as ROS (Robot Operating System) and Linux CNC.[4] Applications of an open source-based EtherCAT master system not only reduces cost, but also makes application program development flexible. Application programs can be developed in PLC language and C/C++ language for Beckhoff's TwinCAT.[5] If application programs is developed in C/C++ language for Beckhoff's TwinCAT, it can only run on kernel space. TwinCAT has limitations regarding use of standard C/C++ math functions and memory allocation. Therefore, it is not convenient for development of applications that use complex functions such as dynamics-based control of manipulators. However, open source-based real-time linux platforms not only provide an interface on kernel space, but also user space, so EtherCAT-based application development in the user space is possible. In order to apply the open source-based EtherCAT master system, which offers cost reduction and flexible development of applications, to automated systems, performance must be guaranteed. Real-time performance is an important evaluation index for EtherCAT because data is transmitted and received through a polling. In this study, an EtherCAT master system is set up using the EtherCAT stack of IgH and a real-time linux platform, and the real-time performance is compared with the Beckhoff's TwinCAT, which is the standard of the EtherCAT master system, to confirm whether it can be applied to an actual system. The real-time linux platforms RT-Preempt and RTAI, Xenomai are used.

2. EtherCAT Master System

2.1. Open Source-Based EtherCAT Master System

The IgH EtherCAT master was used to set up an open source-based EtherCAT master system. The EtherCAT master of IgH consists of the EtherCAT master stack, EtherLab for a Matlab/Simulink-based real-time application, EtherCOS for a Scilab/Scicos-based real-time application, a test manager for a TCP/IP-based monitoring application, and a Data Logging Service (DLS) that is a graphics-based user interface.[4] In this study, only the EtherCAT master stack was applied. This master stack runs on RT-Preempt, RTAI, and Xenomai, so it can be used for all processors that port RT-Preempt, RTAI, and Xenomai. The Ethernet driver can be used in two ways. First, the device driver that operates in a general linux can be used. This method does not guarantee real-time performance. Therefore, a blocking problem can occur when reading data received from the slave due to processor performance or the load of another application. The second way is to use a device driver that provides EtherCAT master stack. In this case, it is divided from the linux network stack which reduces unnecessary overhead, and a polling is used instead of an interrupt, which secures real-time performance. If RTAI and Xenomai are used with a real-time platform in particular, a device driver that is RTDM(Real-Time Device Model)-based is used to get higher priority than other device drivers. Open source-based real-time platforms have a real-time patch applied to a general linux source. Then the configuration of kernel is used to change the power management and clock options that have negatively affect real-time performance. Figure 1 shows the structure of an Ethernet master system that has a real-time system. The IgH EtherCAT master stack runs on the kernel space of all modules of all systems, and if the application program of the user interface is developed in the user space, it works in the user space. If the application program is developed in the kernel space, the EtherCAT master's user interface runs on the kernel space. The Ethernet driver uses what is provided by the EtherCAT master stack.

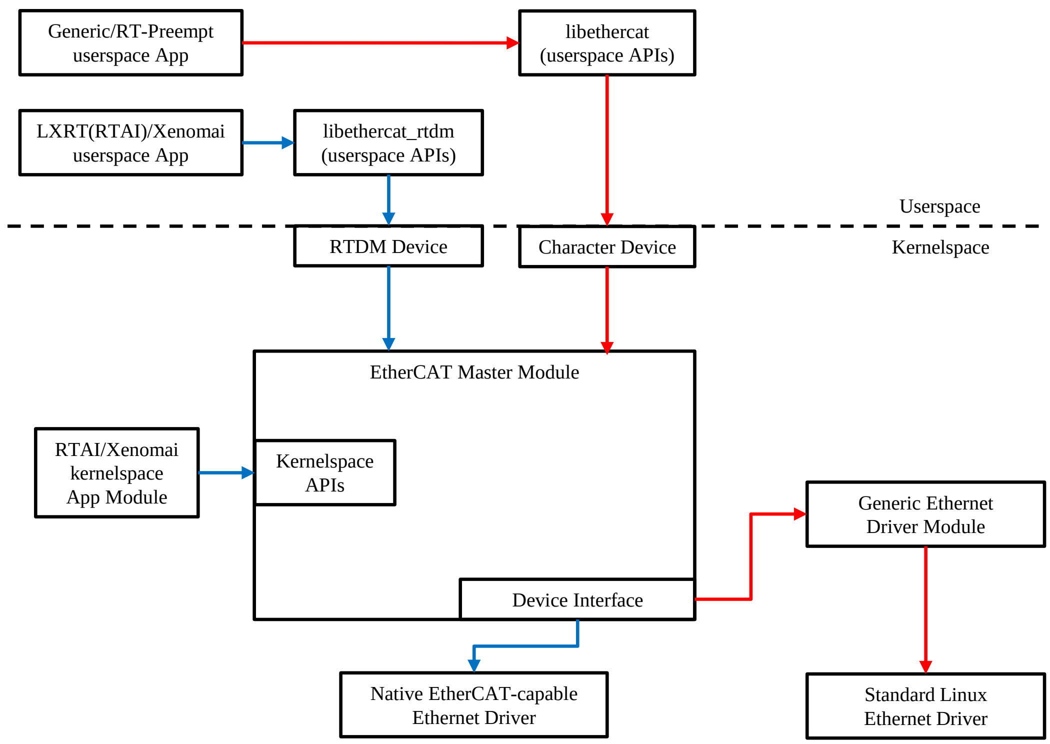

Fig. 1: Architecture of Open Source based EtherCAT Master System.

2.2. Beckhoff's TwinCAT

TwinCAT (The Windows Control and Automation Technology) is a PC-based real-time control system provided by Beckhoff, which developed EtherCAT. It includes a multi-PLC system, NC axis control, programing environment, and an operating station. It is used as a reference for the EtherCAT master system. TwinCAT services of TwinCAT 3 used in this study can be controlled and source codes managed in the Microsoft Visual Studio. TwinCAT runtime provides a software environment that can manage TwinCAT module, and a basic function that can use system resources (memory, field bus, etc.). TwinCAT3 can operate IEC61131, C/C++, and all control tasks in the Matlab/Simulink program language in the real-time kennel. The minimum cycle time of the real-time kennel is 50us, and tasks can be distributed for operation in a multi-core CPU.[5]

3. Performance Tests

3.1. Test Environment

Figure 2 shows the schematic diagram of the test environment set up for performance evaluation. A PC was used as EtherCAT master. For the EtherCAT slave, the Beckhoff EL2008 digital output device and 6 RSAutomation CSD7 sub-drives were used.

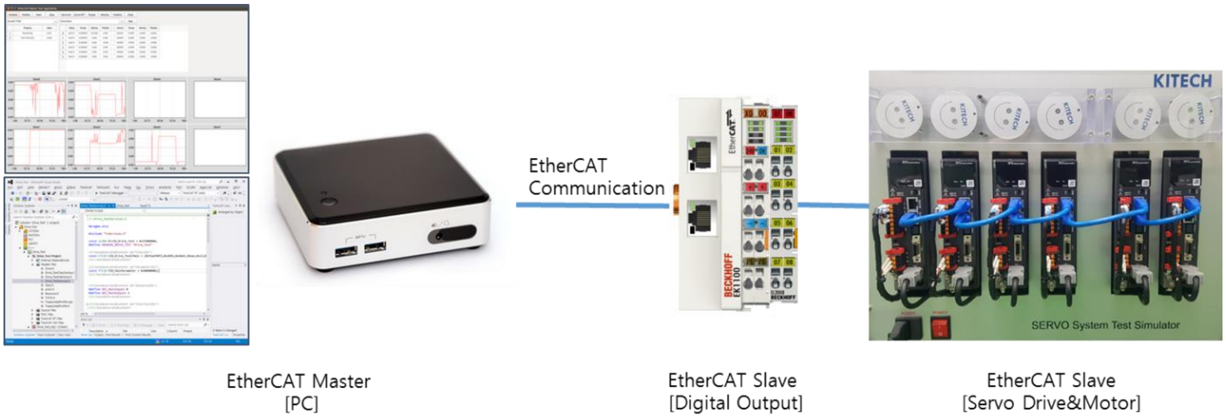

Fig. 2: Schematic diagram of experiment system.

The hardware of PC was composed of Intel i5-4250U(1.4GHZ), 12GBytes Memory and Intel I218-V Ethernet Chipset. As EtherCAT Master platform, we used the Windows7 & TwinCAT, RT-Preempt & IgH EtherCAT Master Stack, Xenomai & IgH EtherCAT Master Stack and RTAI & IgH EtherCAT Master Stack.

A DC(distributed clock) was used for the servo-drive, but not used for the digital output device. The PDO(Process Data Object) mapping information is as shown in Tables 1 and 2. The RxPDO(Receive Process Data Object) data is 37 bytes, and TxPDO(Transmit Process Data Object) data is 144 bytes. The total data is 181 bytes in size.

Table 1: EtherCAT Slave (Digital Output Terminal) PDO Mapping Data.

Index	Size	Name	Index	Size	Name
0x7000:01	0.1	Channel1	0x7040:01	0.1	Channe5
0x7010:01	0.1	Channel2	0x7050:01	0.1	Channe6
0x7020:01	0.1	Channe3	0x7060:01	0.1	Channe7
0x7030:01	0.1	Channe4	0x7070:01	0.1	Channe8

Table 2: EtherCAT Slave (Drive) PDO Mapping Data.

Index	Size	Name	Index	Size	Name
0x6040:00	2.0	Control Word	0x6077:00	2.0	Torque Actual Value
0x607A:00	4.0	Target Position	0x60F4:00	4.0	Following Error actual value
0x6040:00	2.0	Status Word	0x60B9:00	2.0	Touch probe status
0x6064:00	4.0	Position Actual Value	0x603F:00	2.0	Error Code
0x606C:00	4.0	Velocity Actual Value	0x60FD:00	4.0	Digital Input

Loads of 0%, 50%, and 80% of the periods of each master system were created, and the periods of the tasks were measured. The task periods were set as 1ms.

3.2. Test Results

Table 3 shows the results of measuring the task periods for 10 minutes according to each EtherCAT master system load. The average periods regardless of load was almost 1ms for each system. The maximum/minimum standard deviation showed that the open source-based EtherCAT master system had better performance than TwinCAT. The maximum jitter of the set up system was within ±50us. Among the Open Source based EtherCAT Master Systems, RT-Preempt and

Xenomai showed similar results. There was no significant difference in the experimental results, but performance of RT-Preempt and Xenomai platforms were better than RTAI. In the case of RT-Preempt and Xenomai, data of similar results were measured at 0%, 50% and 80% of different loads without being influenced by the load, and RTAI showed better results when the load was 80%. If a DC is used, shift time can be controlled. That means communication is possible between the master and slave with regular periods regardless of jitter.

Table 3: S/W experiment data.

Load	Data	RT	RTAI	Xenomai	Twincat
0%	Mean(ms)	1	1	1	1.000347682
	Min(ms)	0.995412	0.976246	0.994865	0.945
	Max(ms)	1.005423	1.03656	1.005627	1.058
	Dev(µs)	0.000257059	0.003722	0.000177	0.004272965
50%	Mean(ms)	1	1	1	1.000347857
	Min(ms)	0.996452	0.968854	0.993547	0.953
	Max(ms)	1.004538	1.031917	1.00757	1.057
	Dev(µs)	0.000222759	0.003105	0.000109	0.006022418
80%	Mean(ms)	1	1	1	1.000347
	Min(ms)	0.993451	0.996914	0.992808	0.9557
	Max(ms)	1.006178	1.002397	1.007268	1.0463
	Dev(µs)	0.000253937	0.000107	0.000148	0.002624

4. Conclusion

The method of setting up an open source-based EtherCAT master system, and real-time performance were examined in this study. Guaranteeing real-time performance of EtherCAT master system is very important because EtherCAT master transmits and receives data with the EtherCAT slave through a polling. Therefore, the real-time performance of Beckhoff's TwinCAT, which is the standard of EtherCAT master, and an open source-based EtherCAT master platform were compared in the experiment. Comparison results showed that the system set up was similar to or better than TwinCAT. Therefore, the proposed open source-based EtherCAT master system can be very useful for the field of EtherCAT-based robotics area because it can reduce cost and provide flexibility in application development.

Acknowledgements

This research was supported by part of the Core Industry Technology Development Project of the Ministry of Trade, Industry and Energy, "Next Generation Manufacturing Robot for Smart Factory and Work-Space Sharing with Human." Grant number 10063413.

References

[1] EtherCAT, [Online]. Available: http://www.ethercat.org

[2] I. Kim, S. Park, M. Sung, T. Kim, "Design and Implementation of a Real-Time Motion Controller using Open Source Software," *Journal of KIISE: Computer Systems and Theory*, vol. 39, no. 2, pp. 84-95, 2012.

[3] Y. Moon, N. Y. Ko, K. Lee, Y. Bae, J. K. Park, "Real-time EtherCAT Master Implementation on Xenomai for a Robot System," International Journal of Fuzzy Logic and Intelligent Systems, vol. 9, no. 3, pp. 244-248, 2009.

[4] EtherLab, [Online]. Available: www.etherlab.org

[5] TwinCAT, [Online]. Available: ttps://www.beckhoff.com