TUTORIAL

EtherCAT Tutorial

An Introduction for Real-Time Hardware Communication on Windows

By Kevin Langlois, Tom van der Hoeven, David Rodriguez Cianca, Tom Verstraten, Tomislav Bacek, Bryan Convens, Carlos Rodriguez–Guerrero, Victor Grosu, Dirk Lefeber, and Bram Vanderborght

Setting up real-time hardware communication for applications such as precise motion control can be time consuming and confusing. Therefore, this tutorial introduces the deployment of an Ethernet for control automation technology (EtherCAT) protocol. We situate EtherCAT, briefly discuss the origins and working principles, and mention advantages over other widely used protocols. Additionally, the main objectives of the tutorial and the required software to complete it are presented. Online supplements are included, explaining all steps to run a Simulink model in real time on a Windows machine within a few hours.

Background

Particular application domains in the field of robotics and industrial automation require precise motion control capabilities. To achieve this, real-time control systems are the key to developing highly dynamic and intelligent robotic systems, such as prosthetic devices [1], [2], exoskeletons [3], legged robots [4], humanoids [5], and so on. In these high-end motion control applications, there is a need for very fast, low-level control loops with data rates that range from 250 μs to 1 ms [6]. Furthermore, these systems require accurate time synchronization and high data throughput for which real-time Ethernet (RTE) protocols have recently emerged as the leading solution in industry.

EtherCAT is a widely used RTE protocol that has shown excellent performance at a relatively low cost [7]–[9]. The simplicity with which it can be deployed and run a Simulink model in real time makes EtherCAT a practical solution for robotic prototyping. This is an introduction on how to deploy an EtherCAT master running a Simulink model in real time on a Windows machine. The general working principles and the origins of EtherCAT are summarized to situate EtherCAT in the vast family of RTE protocols and show its advantages relative to other RTE systems and controller area networks (CANs). In an ideal case, the reader should be able to have a simple test setup running in a matter of hours and with limited hardware costs. The fully detailed step-by-step tutorial

Digital Object Identifier 10.1109/MRA.2017.2787224
Date of publication: 12 March 2018

1070-9932/18©2018IEEE

accompanies this article in IEEE *Xplore*. Additionally, the tutorial contains successful implementations on several systems, such as the Cybernetic Lower-Limb Cognitive Ortho-Prosthesis (CYBERLEGS) Beta-Prosthesis (Figure 1).

EtherCAT in Layman's Terms

The functionality of the EtherCAT protocol's general concepts and origins of industrial communication are briefly recalled, and the general working principles that apply specifically to EtherCAT are discussed here with a certain level of abstraction, as the goal of this tutorial is only to give the reader a basic understanding of the system to successfully deploy the protocol. Specific resources are conveyed to the reader interested in delving more deeply into the field.

EtherCAT Origins

Technologies such as programable microcontrollers and digital signal processors allowed for the replacement of purely analog control loops with digital controllers, such as programmable logic controllers (PLCs). This led to the core of industrial networking, termed *fieldbus protocols*. Many fieldbuses were developed in parallel, and, essentially, every company designed its own protocols, which naturally led to confusion for consumers and developers [11]. As a reaction to this, there was the fieldbus war, a web of company politics and marketing interests that, in 2000, led to the establishment of an international fieldbus standardization, the International Electrotechnical Commission Standard 61158 [12]–[15]. In the meantime, Ethernet protocols were already well established in the office world. The increased data rates of newer Ethernet standards (for example, 802.3u fast Ethernet) made it easier to create real-time Ethernet protocols, as the transmission and retransmission times are significantly shorter.

While fieldbus systems such as Profibus, Serial Real-Time Communication System (SERCOS), CANs, and many others have allowed for distributed industrial automation systems, the performance of these protocols was considered to be too limited when compared to the available performance (mainly in terms of data throughput) of nonreal-time networks such as Ethernet [16]. In addition, Ethernet bandwidth enables bus cycle times in the microsecond range instead of the millisecond range, which, together with the superior performance of modern personal-computer-based control systems, allows one to close the control loops over the fieldbus that previously had to be closed locally in the peripheral systems [17]. Finally, the large amount of research that has gone into developing Ethernet as a standard, as well as inexpensive and readily available hardware, and the Internet of Things [18], which enables the connection of almost anything with everything, anywhere, illustrates the wish to develop RTE communication protocols. Nonetheless, the many challenges that presented themselves as the requirements between those two domains—the office world and industry—are very different, as discussed in [11] and [19]. The two main differences in requirements are related to communication time determinism and precise clock synchronization.

First, *time determinism* is the requirement for the transmission times of data packets to be known. This means that the latency of a signal must be bounded and have a low variance. The variance of the response time of a signal is often referred to as *jitter*. A low jitter is required because the variance in time has a negative effect on control loops (the derivative and integral portions of a control loop are affected by time variation). The second difference is the precise clock synchronization of the network. Real clocks drift and will differ in time over long periods. As pointed out in [20], unsynchronized networks usually suffer from nonnegligible jitters. In conclusion, a family of RTE protocols has, in recent years, evolved into a large number of solutions and standards. Two of the most prominent representatives of this group are EtherCAT and Profinet, the former of which is believed to offer the best performance in terms of communication efficiency and short cycle times [8].

Working Principle

Today, there are more than 25 different RTE solutions on the market, and they are offered by diverse manufacturers and the academic community [19], [9], [16]. These solutions integrate different working principles, such as the method for encapsulation of process data into Ethernet frames, the limitations on network topology, synchronization of the network, implementation costs, and so on. Primarily, EtherCAT is industrial Ethernet and utilizes standard frames and the physical layer as defined in the Ethernet IEEE Standard 802.3 [21]. Of the fast Ethernet standards, 100BASE-TX is the most common and the one used in EtherCAT networks. By utilizing the full duplex (data transmission in two directions), features of fast Ethernet allow effective data rates of 100 Mb/s [17]. Furthermore, the EtherCAT protocol employs the master/slave principle to control access to the medium. The master node (typically the control system) sends the Ethernet frames to the slave nodes (such as sensors and actuators), which can extract data from and insert data into these frames. These process data (of all the network devices) are carried together in one

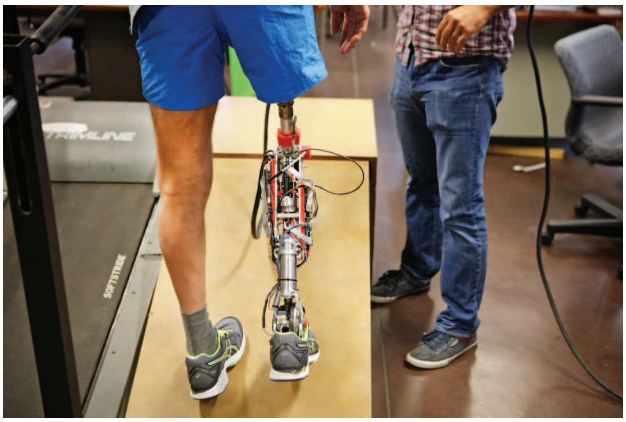

Figure 1. The CYBERLEGS Beta-Prosthesis, which actuates two degrees of freedom (the knee and ankle), is controlled via an EtherCAT protocol. The prosthesis integrates series elastic actuators controlled by custom-made electronics boards.

or more Ethernet frames. This is the so-called summation frame principle, as opposed to the individual frame approach in which every frame carries process data for only one device [8].

With EtherCAT, the standard Ethernet packet is no longer received, interpreted, and copied at every node. Instead, slave devices process frames on the fly, reading and inserting data while the frame is passing through the device. This is handled by hardware-integrated EtherCAT slave controllers. The process data in industrial networks are relatively small in quantity (only a few bytes) so that the summation frame method, combined with the processing-on-the-fly feature of the EtherCAT slaves, offers strong system performance [8], [22]. Moreover, network topology plays an important role when the performance of a system is evaluated [22]. Crucial aspects are not only cycle time or efficiency but also cabling effort, diagnostic features, redundancy options, and plug-and-play features. EtherCAT networks have no practical limitations regarding the topology, line, star, tree, ring, and all those combined with up to 65,535 nodes per segment. Then, for synchronization, EtherCAT relies on a clock synchronization mechanism that is known as a *distributed clock* (*DC*). Essentially, all the DC-enabled slaves in the network are synchronized with a common timing reference under direct control of the master [7]. Despite being simple and straightforward, the DC mechanism enables accurate synchronization (in small-to-medium systems, clock deviations are well below 1 μs).

Ethernet-Based Solutions

There are more than 25 Ethernet-based industrial protocols on the market, but the list of protocols that have a considerable impact on industry is much shorter [23]. They are

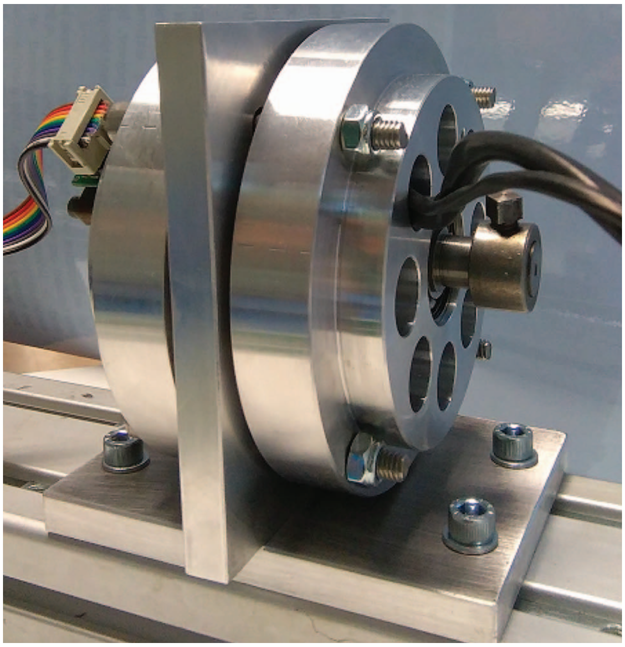

Figure 2. A Kollmorgen RBE-type 12-pole frameless motor as used in the test setup. The sensors in the motor include hall sensors for the commutation of the motor and an optical incremental encoder for its position. A small ring is clamped to the main axle to be able to turn the motor axle manually.

- EtherCAT
- Powerlink
- Ethernet/Internet Protocol (IP)
- Modbus/Transmission Control Protocol (TCP)
- ProfiNet.

A multitude of performance evaluations of the remaining systems are reported in the literature. The conclusions of these studies are that EtherCAT is an overall highly performing real-time protocol when compared to the aforementioned protocols [24], [25], [8]. ProfiNet has advantages over EtherCAT in specific conditions, such as efficiency in asynchronous communication [26]. Neither Ethernet/IP nor Modbus/TCP are deterministic and, by consequence, not suited for hard real-time control. The major advantages of EtherCAT over ProfiNet and Powerlink are the costs of implementation and the commercial diffusion of the technology [27]. Studies also predict the future pervasiveness of EtherCAT in the industrial automation and robotics fields [23]. This suggests that EtherCAT is the leading communication protocol for these applications. Another major communication protocol that is not part of Ethernet-based systems are serial protocols, of which CAN is widely used in the robotics field.

CAN

CAN is still an adequate choice for low-cost industrial embedded networking. However, Ethernet-based protocols are now able to overcome the shortcomings of CAN, such as limited baud rate and limited network length (1 Mb/s at 120 ft). Although the very low-cost implementation as well as the low resource requirements of CAN protocols still make it an adequate choice in certain applications (such as the automotive industry, small embedded solutions, and aerospace), the overall advantages of EtherCAT over CAN are

- the data throughput (currently 100 times higher)
- the unlimited network length
- increased system performance
- the use of established hardware components.

EtherCAT Deployment

An EtherCAT master runs the EtherCAT network and communicates with all slaves. This master needs to be implemented on a real-time operating system. For this case, several solutions have been developed. The one demonstrated in this tutorial is the Windows control and automation technology (TwinCAT). Beckhoff provides the TwinCAT program, whose essential functionality is to reserve a number of physical cores on a user's personal Windows computer and run the EtherCAT network from these cores. The Windows operating system does not run on these cores anymore and only operates on the cores specified by the user. As an example, in this tutorial, a quad-core laptop was used in which two cores are reserved for the EtherCAT protocol and two cores are used for running Windows.

The driver running on the EtherCAT reserved cores is a compiled version of a program that can be written in either

PLC language or C/C++ code. Beckhoff provides functionality to run compiled Simulink models (in C++) as drivers in the kernel space. Following is a list of the required software to successfully turn a Windows computer into a real-time target running an EtherCAT network:

- Microsoft Visual Studio 2010 or higher; required if programing is directly done in C++, otherwise the shell provided with TwinCAT can be used
- MATLAB 2011 (or newer) including MATLAB Coder
- TwinCAT 3.1 (free for noncommercial use)
- a Beckhoff TE1400 module (free for Simulink models with five inputs, five outputs, and 100 blocks; larger models require a license)
- Microsoft Windows Driver Kit 7 (free) or higher.

Conclusion

In this article, the main outline of a broader hands-on tutorial on the EtherCAT communication protocol for real-time hardware communication in robotics and automation applications is presented. This includes an introduction to fieldbus systems and a general description of the EtherCAT features. The tutorial guides the reader through the implementation of a real-time control loop developed in Simulink and compatible with widely used hardware components (Maxon drivers and motors). Debugging tools for TwinCAT and Simulink as well as instructions on the activation of Beckhoff licenses are included. Additionally, the reader will find experimental implementations such as those depicted in Figures 1 and 2. Furthermore, the literature concerning the assessment of the protocol's performance was conveyed, and the advantages of the protocol were discussed.

The technical features of EtherCAT make it an outstanding communication protocol for applications requiring precise motion control, while keeping the implementation cost to a minimum. Additionally, it provides compatibility with common hardware and software. With the appended tutorial, the reader should be able to deploy a reliable real-time communication system and run a Simulink control loop within a matter of hours.

Acknowledgment

This work was supported in part by the European Research Council grant Series-Parallel Elastic Actuation for Robotics 337596 and the Strategic Basic Research project Yves of Flanders Make.

References

[1] L. Flynn, J. Geeroms, R. Jimenez-Fabian, B. Vanderborght, N. Vitiello, and D. Lefeber, "Ankle–knee prosthesis with active ankle and energy transfer: Development of the CYBERLEGS Alpha-Prosthesis," *Robot. Autom. Syst.*, vol. 73, pp. 4–15, Nov. 2015.

[2] P. Cherelle, V. Grosu, A. Matthys, B. Vanderborght, and D. Lefeber, "Design and validation of the ankle mimicking prosthetic (AMP-) Foot 2.0," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, pp. 138–148, Jan. 2014.

[3] S. Wang, L. Wang, C. Meijneke, E. Van Asseldonk, T. Hoellinger, G. Cheron, Y. Ivanenko, V. La Scaleia, F. Sylos-Labini, M. Molinari, F. Tamburella, I. Pisotta, F. Thorsteinsson, M. Ilzkovitz, J. Gancet, Y. Nevatia, R. Hauffe, F. Zanow, and H. van der Kooij, "Design and control of the Mindwalker exoskeleton," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 2, pp. 277–286, 2015.

[4] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, and D. G. Caldwell, "Design of the hydraulically actuated, torque-controlled quadruped robot HyQ2max," *IEEE/ASME Trans. Mechatronics*, vol. 22, pp. 635–646, Apr. 2017.

[5] F. Negrello, A. Settimi, D. Caporale, G. Lentini, M. Poggiani, D. Kanoulas, L. Muratore, E. Luberto, G. Santaera, L. Ciarleglio, L. Pallottino, D. Caldwell, N. Tsagarakis, A. Bicchi, M. Garabini, and M. Catalano, "WALK-MAN humanoid robot: Field experiments in a post-earthquake scenario," *IEEE Robot. Autom. Mag.*, to be published.

[6] P. Neumann, "Communication in industrial automation—What is going on?," *Control Eng. Practice*, vol. 15, pp. 1332–1347, Nov. 2007.

[7] G. Cena, I. C. Bertolotti, S. Scanzio, A. Valenzano, and C. Zunino, "Evaluation of EtherCAT distributed clock performance," *IEEE Trans. Ind. Informat.*, vol. 8, pp. 20–29, Feb. 2012.

[8] G. Prytz, "A performance analysis of EtherCAT and PROFINET IRT," in *Proc. IEEE Int. Conf. Emerging Technologies and Factory Automation,* 2008, pp. 408–415.

[9] J. Jasperneite, M. Schumacher, and K. Weber, "Limits of increasing the performance of industrial Ethernet protocols," in *Proc. IEEE Conf. Emerging Technologies and Factory Automation,* 2007, pp. 17–24.

[10] V. Grosu, C. Rodriguez Guerrero, B. Brackx, S. Grosu, B. Vanderborght, and D. Lefeber, "Instrumenting complex exoskeletons for improved human-robot interaction," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 5, pp. 5–10, Oct. 2015.

[11] B. Galloway and G. P. Hancke, "Introduction to industrial control networks," *IEEE Commun. Surveys Tutorials*, vol. 15, no. 2, pp. 860–880, 2013.

[12] *Industrial Communication Networks—Fieldbus Specifications—Part 1: Overview and Guidance for the IEC 61158 and IEC 61784 Series*, IEC-61158-1, 2014.

[13] M. Felser and T. Sauter, "The fieldbus war: History or short break between battles?," in *Proc. 4th Int. Workshop Factory Communication Systems,* 2002, pp. 73–80.

[14] J.-P. Thomesse, "Fieldbus technology in industrial automation," *Proc. IEEE*, vol. 93, no. 6, pp. 1073–1101, June 2005.

[15] J. R. Moyne and D. M. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data," *Proc. IEEE*, vol. 95, no. 1, pp. 29–47, Jan. 2007.

[16] J.-D. Decotignie, "Ethernet-based real-time and industrial communications," *Proc. IEEE*, vol. 93, no. 6, pp. 1102–1117, June 2005.

[17] M. Rostan, J. E. Stubbs, and D. Dzilno, "EtherCAT enabled advanced control architecture," in *Proc. IEEE/SEMI Advanced Semiconductor Manufacturing Conf. (ASMC),* 2010, pp. 39–44.

[18] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of Things," *Int. J. Commun. Syst.*, vol. 25, no. 9, pp. 1101–1102, Sept. 2012.

[19] M. Felser, "Real-time Ethernet—Industry prospective," *Proc. IEEE*, vol. 93, no. 6, pp. 1118–1129, June 2005.

[20] P. Ferrari, A. Flammini, D. Marioli, A. Taroni, and F. Venturini, "Experimental analysis to estimate jitter in PROFINET IO Class 1 networks," in *Proc. ETFA '06, IEEE Conf. Emerging Technologies and Factory Automation,* 2006, pp. 429–432.

(continued on page 122)

Quantum School of Technology, Roorkee), M.S. Gupta (academic dean, Quantum School of Technology, Roorkee), Raunak Gupta (IEEE Student Branch counselor), Mohd. Furqan Khan (manager, KVCH, Noida, India), and Uma Shanker Yadav (senior research engineer, KVCH, Noida, India) (Figure 5).

To start the program, Raunak Gupta welcomed the 31 dignitaries and participants, informing them that this was the Student Branch's third event this year. He said that robotics, being an interdisciplinary field, has a large number of applications in every walk of life, such as manufacturing and the medical, aerospace, and automotive industries.

During the TEP-R, the participants learned basic robot design, function, and control. Yadav explained the components used in making a robot. Students learned fabrication techniques through a hands-on, practical session (Figure 6). They worked on various algorithms, such as Line Follower Robot and Edge Follower Robot. During the last TEP-R segment, the organizers sprang a quiz to assess attendees' knowledge, and the top six participants were awarded a certificate of merit.

At the conclusion of the program, Chauhan expressed his thanks to the audience. He also assured everyone that the Student Branch would provide the best of opportunities to all of the institute's budding engineers, researchers, and professors who are keen to pursue future explorations in the field of robotics.

Tutorial *(continued from page 25)*

[21] *IEEE Standard for Ethernet*, IEEE 802.3-2015 (Revision of IEEE 802.3-2012), Mar. 2016.

[22] M. Knezic, B. Dokic, and Z. Ivanovic, "Topology aspects in EtherCAT networks," in *Proc. 14th Int. Power Electronics and Motion Control Conf. (EPE/PEMC),* 2010, pp. T1-1–T1-6.

[23] W. Voss. (2011). The future of CAN/CAN open and the industrial Ethernet challenge. Electronics, Inc. Greenfield, MA. [Online]. Available: http://www.esd-electronics-usa.com/Shared/Library/OutsideArticles/TheFutureofCAN.pdf

[24] X. Wu, L. Xie, and F. Lim, "Network delay analysis of EtherCAT and PROFINET IRT protocols," in *Proc. 40th Annu. Conf. IEEE Industrial Electronics Society*, 2014, pp. 2597–2603.

[25] J. Robert, J.-P. Georges, E. Rondeau, and T. Divoux, "Minimum cycle time analysis of Ethernet-based real-time protocols," *Int. J. Computers, Commun. Control*, vol. 7, no. 4, pp. 743–757, 2012.

[26] R. Schlesinger, A. Springer, and T. Sauter, "New approach for improvements and comparison of high performance real-time Ethernet networks," in *Proc. IEEE Emerging Technology and Factory Automation (ETFA),* 2014, pp. 1–4.

[27] D. Orfanus, R. Indergaard, G. Prytz, and T. Wien, "EtherCAT-based platform for distributed control in high-performance industrial applications," in *Proc. IEEE 18th Conf. Emerging Technologies Factory Automation (ETFA),* 2013, pp. 1–8.

Kevin Langlois, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium. E-mail: kevin.langlois@vub.be.

Tom van der Hoeven, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium. E-mail: tomvdhoeven89@gmail.com.

David Rodriguez Cianca, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium. E-mail: david.rodriguez.cianca@vub.be.

Tom Verstraten, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium. E-mail: tom.verstraten@vub.be.

Tomislav Bacek, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium. E-mail: tbacek@vub.be.

Bryan Convens, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium. E-mail: bryan.convens@vub.be.

Carlos Rodriguez–Guerrero, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium. E-mail: carlos.rodriguez.guerrero@vub.be.

Victor Grosu, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium. E-mail: vgrosu@vub.ac.be.

Dirk Lefeber, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium. E-mail: dirk.lefeber@vub.be

Bram Vanderborght, Robotics and Multibody Mechanics Research Group, Vrije Universiteit Brussel, Belgium. E-mail: bram.vanderborght@vub.be.