ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML110

EPICS DRIVER FOR SIEMENS CP1616 COMMUNICATION MODULE*

Z. Huang, Y. Song, X. Sun, G. Liu†, NSRL, USTC, Hefei, Anhui 230029, China

Abstract

Siemens communication module CP1616 is a high-performance PROFINET controller, which can support both Real-time (RT) and Isochronous Real-Time (IRT) communication. Experimental Physics and Industrial Control System (EPICS) is a wildly used distributed control system in large scientific devices. In order to integrate PROFINET communication into EPICS environment, we develop this driver based on CP1616 and establish the prototype system. This paper will describe the design of EPICS driver for CP1616 and the test result of the prototype system.

INTRODUCTION

PROFINET is an Industrial Ethernet based on TCP/IP standard and IEC 61158 standard, designed for real-time data communication and equipment configuration [1]. Siemens CP1616 a PCI bus based PROFINET communication controller modules suitable for variable operation system, including windows and Linux. As CP1616 uses Enhanced Real-Time Ethernet Controller (ERTEC) 400 chip, it can support both PROFINET RT and IRT protocol.

As PROFINET has been used in accelerator control wildly [2–4], we develope this driver in order to integrate PROFINET into EPICS environment. The hardware controller is Siemens CP1616, the EPICS base is R3.14.12.6.

This paper focuses on the design of EPICS driver, and the details are organized as follows. Section I gives the general introduction of PROFINET and CP1616. Section II explains the IOC driver architecture by details. Finally, Section III shows the prototype system and the corresponding test.

OVERVIEW OF PROFINET

According to the different real-time performance, PROFINET communication can be divided into 3 classes : Non Real-Time (NRT) communication, RT communication and IRT communication. RT and IRT communications are in charge of real-time data transmission while NRT communication is for configuration and diagnostics data transmission. As IRT communication includes RT and NRT communication process, we will describe PROFINET based on IRT mainly. Figure 1 shows a PROFINET IRT bus cycle.

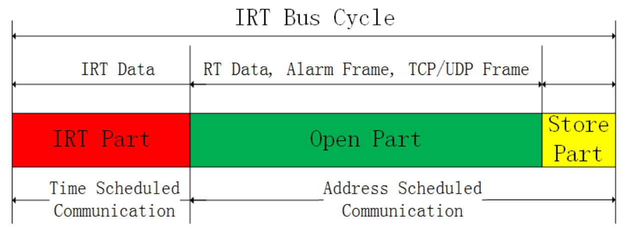

Figure 1: Profinet IRT Bus Cycle.

As shown in Figure 1, PROFINET IRT cycle is divided into IRT part (the red part), open part (the green part) and the store part (the yellow part). RT and Non Real-Time (NRT) frames are all in the open part and base on address scheduled communication. For RT frame, it is tagged with high priority VLAN bytes to ensure its transmission before NRT frames. The RT cycle times can achieve 1˜ 10ms.Thus RT protocol can be used for factory automation.IRT communication is more suitable for motion control applications. In IRT part, only time scheduled IRT frames can be processed [5, 6]. Because IRT supported devices use ERTEC 400 chip and Precision Transparent Clock Protocol (PTCP) for bus clock synchronization, real-time communication preprocessing can significantly improve performance, and the cycle time is less than 1ms [7].

If real-time data arrives switch beyond IRT part and Open part, it will be stored in the cache area in store part.

Figure 2 shows the PROFINET frame structure.

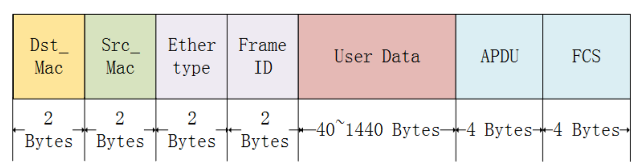

Figure 2: The Structure of PROFINET Frame.

According to Figure 2, one PROFINET frame can carry 40 ˜ 1440 bytes user data. Ethertype and Frame ID bytes identify the different real-time classes (NRT, RT or IRT communication) [8].

PROFINET network configuration is mainly done by Siemens software (such as Portal and STEP 7). Not as traditional Ethernet communication, CP1616 controller assigns bus addresses and configures network topology for all the IO devices in advance before communication.

EPICS DRIVER FOR CP1616 MODULE

Because IRT communication is compatible for RT communication, we develope this EPICS driver for CP1616 module based on IRT communication. It consists of 2 parts:devices support and driver support. Figure 3 is the software architecture of PROFINET driver. We develop this driver based on Siemens encapsulated library file libpnio.so.

During the development of EPICS driver, we use standard record types in record support. For the driver, record support process the real-time data by calling database access routines. Device support provides an interface between driver support and record support by specific driver Application Programming Interface (API). It can put real-time

* Work supported by National Natural Science Foundation of China (11375186)
† gfliu@ustc.edu.cn

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML110

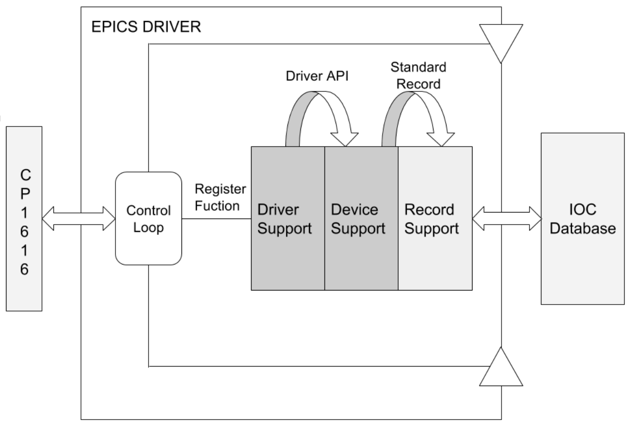

Figure 3: Software Architecture of PROFINET Driver.

data into raw value which is defined by Database (DB) file. Driver support communicates with CP1616 directly. Figure 4 shows the flow chart of driver support.

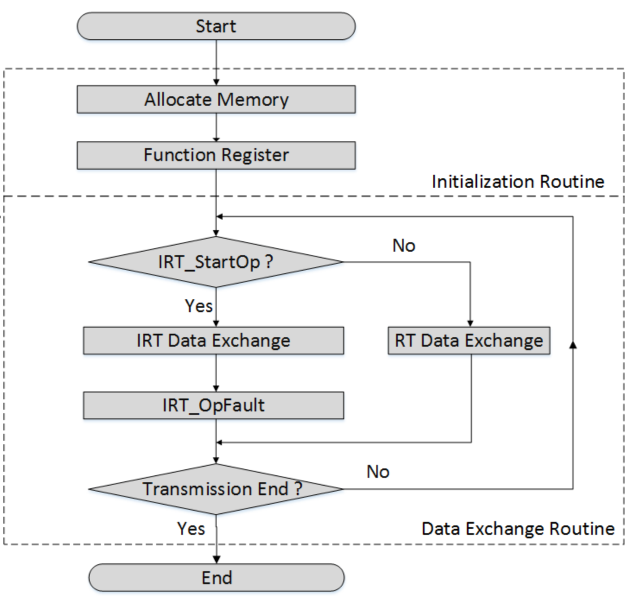

Figure 4: The Flow Chart of Driver Support.

For achieving time scheduled communication, we adopt asynchronous communication in PROFINET driver.

There are 2 main routines after calling epicsExportAddress to register the entry table at Database Definition (DBD) file: device initialization routine and data exchange routine.

By registering callback functions in the initialization routine and allocating memory, IRT data be exchanged only between IRT_startop function and IRT_opfault function, meanwhile it also defines the IRT part and open part. RT data and NRT frames are exchanged beside those two functions.

PROTOTYPE SYSTEM AND REAL-TIME PERFORMANCE TEST

For test the perfomance of EPICS driver, we establish a prototype system. EPCIS IOC is run on the Linux 2.6.32.20 operating system with rtai-3.9.1 real-time patch. IO device is ET 200SP IO station with digital input and output modules and analog input and output modules. Figure 5 is the hardware architecture of our prototype system. According to the architecture, EPICS mainly serves as a monitor, and CP1616 takes charge of processing input and output signal.

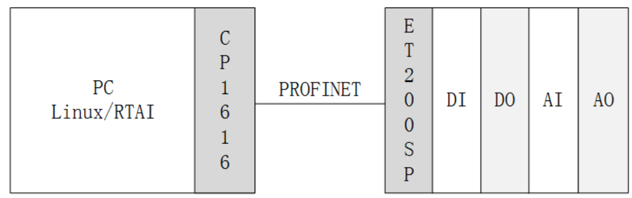

Figure 5: Hardware Architecture of Prototype System.

Figure 6 is the photo of prototype system. CP1616 controller is shown in the left of Figure 6, computer and ET 200SP IO station is shown in the right.

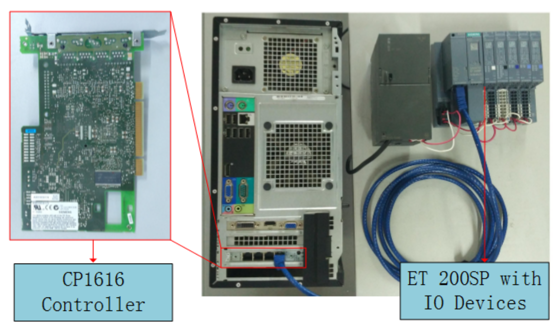

Figure 6: Photo of Prototype System.

In Protal V13 configuration process, the designed cycle time is 500μs, and calculated IRT part time is 16.970μs which takes up 3.394% of bandwidth, RT part time is 7.040μs which takes up 1.408% of bandwidth. Figure 7 is the configuration result.

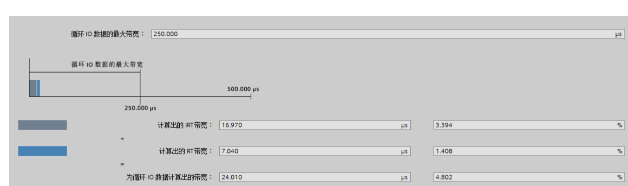

Figure 7: The Configuration Interface for IRT Communication.

During the test, we use Wireshark to capture the PROFINET IRT frames between CP1616 and the IO station. CP1616 receives the input frames and send the output frames to the IO station according to the algorithm. The

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML110

time differece between corresponding input frame and output frame can be seen as the response time. Figure 8 is the input frames of CP1616, and Figure 9 is the output frames of CP1616. In the Figure 8 and the Figure 9, *RTC3* represents the real-time class of PROFINET IRT communication. The response time is 1.001ms.

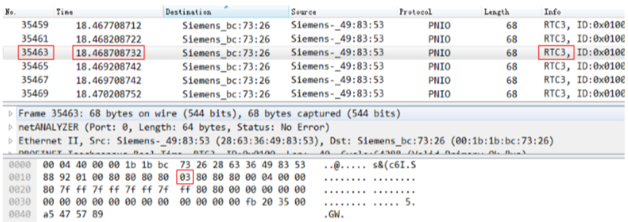

Figure 8: Input frames of CP1616.

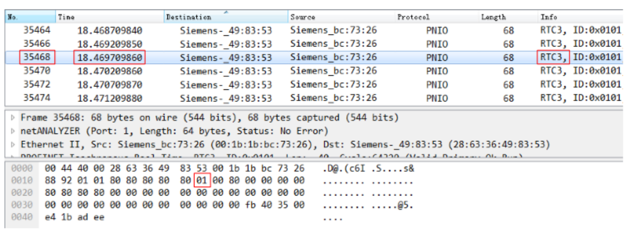

Figure 9: Output frames from CP1616.

EPICS is a soft real-time system, the process speed of the EPICS record can not catch up with the response speed of prototype system. In order to test the EPICS driver, a square wave signal with a period of 800ms is used as an input signal. A record is used to monitoring the square wave signal. Figure 10 is the screenshot of Databrowser for the record. The figure is consistent with the input signal.

Figure 10: The Sreenshot of Databrowser.

CONCLUSION

We integrate PROFINET protocol into EPICS environment by developing a driver based on the Siemens CP1616 communication module. The driver can support both RT communication and IRT communication. In the prototype system, the response time is 1.001ms. It reached the performance of PROFINET IRT communication.By the driver, we can monitor the status of input and output signals in EPICS environment.

REFERENCES

[1] PROFIBUS International, `http://us.profinet.com/technology/profinet/`

[2] Antolovic, Marco, et al. "PLC communication using PROFINET: experimental results and analysis", in *Emerging Technologies and Factory Automation, 2006. ETFA'06. IEEE Conference*, IEEE, 2006, pp. 1-4.

[3] Liu. G, et al. "Epics driver for phoenix contact redundant PLC", in *Proc of IPAC2013*, Shanghai, China, 2013, pp. 3176-3178.

[4] Huang. Ziyu, et al. "A Redundant EPICS Control System Based on PROFINET", in *15th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'15)*, Melbourne, Australia, 17-23 October 2015, pp.736-738.

[5] Wisniewski, Lukasz, Juergen Jasperneite, and Christian Diedrich. "Effective and fast approach to schedule communication in PROFINET IRT networks", in *Industrial Electronics (ISIE), 2013 IEEE International Symposium*, IEEE, 2013, pp. 1-6.

[6] Ferrari,Paolo, et al. "Evaluation of timing characteristics of a prototype system based on PROFINET IO RT_Class 3", in *Proc of IPAC2013*, Shanghai, China, 2013, pp. 3176-3178.

[7] Ferrari. P, A. Flammini, and S. Vitturi. "Response times evaluation of PROFINET networks", in *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium*, Vol. 4. IEEE, 2005, pp. 1371-1376.

[8] Wikipedia, `https://www.wikiwand.com/en/PROFINET`