An Open CNC System based on EtherCAT Network

Beibei Li[1,2] , Hu Lin[2] , Liaomo Zheng[2] , Shujie Sun[1,2] , Zhenyu Yin[2]
1. Shenyang Institute of Computing Technology, University of Chinese Academy of Sciences
2.National Engineering Research Center for High-end CNC
Shenyang, China
li5566189@126.com, {linhu & zhengliaomo & sunshuji & yinzhenyu}@sict.ac.cn

***Abstract*—The introduction of Fieldbus technology makes the performance of the CNC system greatly improved. However with the development of mechanical engineering technology, equipment manufacturing, aerospace and defense industry and other industries for high speed and high precision machining requirements is greatly improved. Limited by transmission speed and bandwidth, traditional Fieldbus has been difficult to meet those demands. In order to solve this problem and improve the machining speed and accuracy of the open CNC system, EtherCAT network is used in this research. In this paper, the architecture of the open CNC system is given first, its hardware is designed based on two processors architecture and its software is designed adopting the idea of hierarchy and modularization. Then EtherCAT is introduced in the open CNC system and its related design is described in detail. Finally, a complicated workpiece is machined by a five-axis machining center to showcase the machining precision of the prototype CNC system.**

Keywords—CNC system; open architecture; EtherCAT; hierarchy; modularization

I. INTRODUCTION

With the development of electronic technology, information technology and control technology, diversified and personalized products processing has become the main feature of the new manufacturing system, and it puts forward a higher requirements, such as high speed and precision, integration, intelligent, etc, for the core components of the manufacturing system—CNC system. In this situation, open CNC system comes into being, open CNC system is dedicated to solve the contradiction between complex and diverse machining demand and special and closed system structure of the traditional CNC system. Its software and hardware has the features of interoperability, interaction, portability, and scalability [1]-[2].

The openness of the open CNC system is mainly manifested in two aspects: the open architecture of the system and the Fieldbus communication mode between the controller and the servo driver. The former reflects in the software level, which involves the functional modules classification and the relationship among the modules. The latter reflects in the hardware level, which uses Fieldbus as a mean of communication between the controller and servo driver. In this communication mode, each device (servo driver, I/O and senior, etc) is considered as a node of the bus, CNC system exchanges data with them through the Fieldbus [3]-[5].

EtherCAT, as a new type of Fieldbus technology, compared to the traditional Fieldbus, has higher transmission speed and greater data transfer bandwidth. It offers a promising solution in the development of an open CNC system. The rest of this paper is organized as follows: Section 2 presents the hardware and software architecture of the open CNC system. Section 3 describes the EtherCAT related design. Section 4 performs a workpiece machining experiment based on the prototype system with a five-axis machining center. Finally, section 5 concludes the paper and points out further work.

II. OPEN ARCHITECTURE CNC SYSTEM

A. The Hardware Platform of the Open CNC System

The hardware platform of the open CNC system adopts two processors architecture (Fig.1), which includes human-machine interface unit (HMU) and machine-tool control unit (MCU). The two processors architecture makes user's operation separate from machine-tool control, in order to reduce the workload of the MCU so that it can focus on the processing of machine-tool related control tasks. In addition, the connection method of Ethernet and Fieldbus is used among those functional units. HMU and MCU are connected via Ethernet and SSB-III (super serial bus) and each controlled unit, such as servo driver, I/O, etc, can be considered as a node on the network, connected with MCU through EtherCAT network in a daisy chain.

978-1-4673-9613-4/16/$31.00 ©2016 IEEE

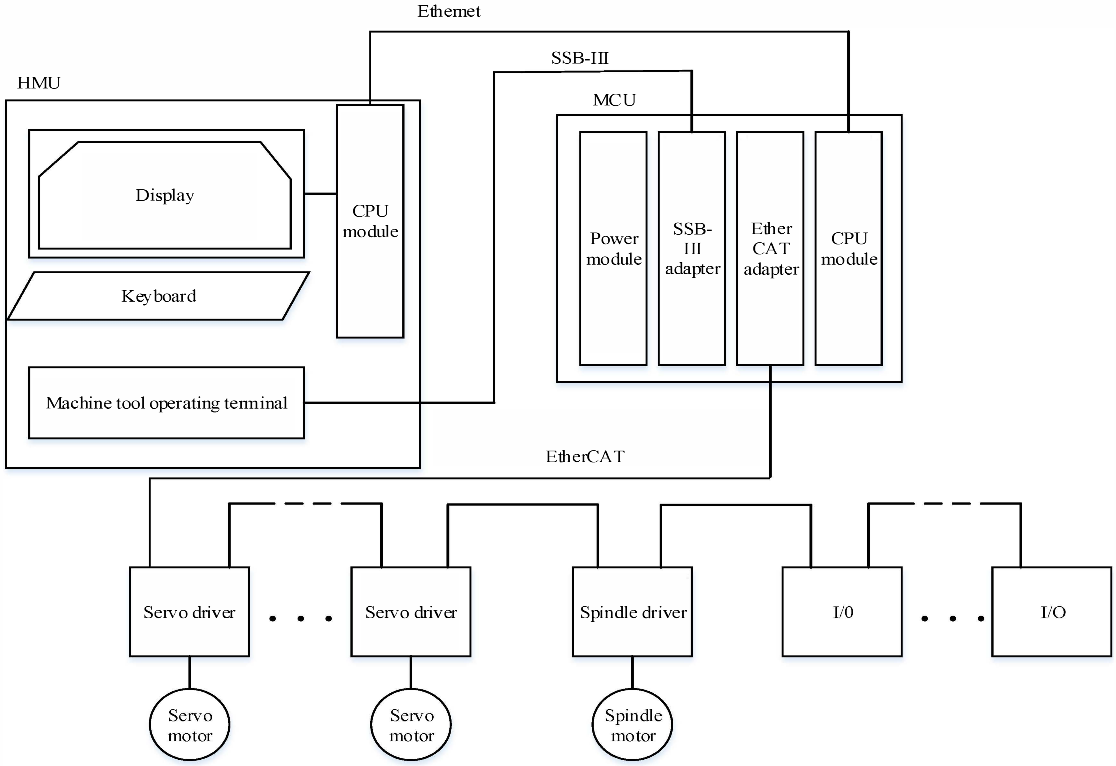

Fig. 1. The hardware platform of the open CNC system

1) MCU: MCU is the main unit of the system. It adopts the Compact PCI (CPCI) chassis structure. There are a number of slots inside the case. Those slots are connected with high speed back-plane bus based on Compact PCI standard. One control card with processor, one power card and several communication cards have been inserted in the slots. In addition, The MCU case also has a reserved slot, which can support function extension.

2) HMU: HMU is configured with LCD display and operation panel of machine-tool. It supplies the hardware platform for user's command input, interactive programming, information management and status display of machine-tool. It uses non real-time operation systems, in order to make better use of the software and hardware resources on the existing platform.

B. The Architecture of the Open CNC System

The basic architecture of open CNC system (Fig.2) is composed of two main parts. One is system software and the other is system platform. the system software is designed adopting modular structure. It contains several functional modules which realize special functional requirements of CNC. The system platform is designed adopting layered structure. It provides supporting for those functional modules. The system platform is divided into four layers, including hardware layer, real-time operating system layer, communication management layer and application program layer.

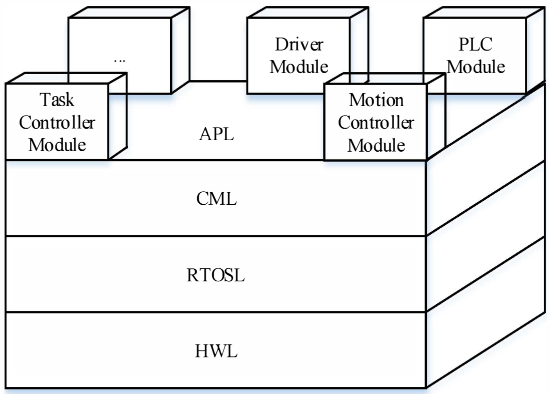

Fig. 2. The architecture of the open CNC system

1) Hardware Layer (HWL): It is the basic component on which the software platform and application software run. It is a physical entity and located in the bottom of the basic architecture. It composed of computer platform and interface modules (such as network and Fieldbus interfaces, etc).

2) Real-Time Operating System Layer (RTOSL): It is located in the bottom of the software platform and is the basis of the software platform. CNC system is a typical real-time control system, it needs not only to complete the specified control operation, but also to return the results of the operation in the specified time. So the operating system must meet the requirement of real-time and predictability. In the course of research, we adopt RT-Linux operating system, it can meet the real-time constrains of CNC system.

3) Communication Management Layer (CML): The purpose of communication management is to shield the details of the implementation of the communication mechanism in bottom level. It provides the required communication service and the mutual operation method for the function module via a standard interface. CML can add the required functional modules and generate the relationship of modules' topology architecture, which can make the CNC more re-configurable.

4) Application Program Layer (APL): It mainly transfers application programs into functional modules based on the functional desire of CNC system, such as task controller module, motion controller module, etc. And the APL can provide function interface through CML.

C. The Functional Modules of the Open CNC System

The functional modules of the open CNC system is shown in Fig.3, which can be divided into two parts: remote module and local modules, according to the different running platform. Remote module runs on the HMU with a non real-time operating system, including HMI module. It does not demand a very high real-time, so is designed to be a common task in user space. Local modules run on the HMU with a non real-time operating system, including task controller module, motion controller module, PLC Module and Driver module. Task controller module, same as HMI module, is also designed to be a common task in user space. As they are in the different hardware platform, they exchange data through the Ethernet. The other modules demand a very high real-time, in order to ensure the continuous and smooth machining of CNC system, they are designed to be real-time task in kernel space.

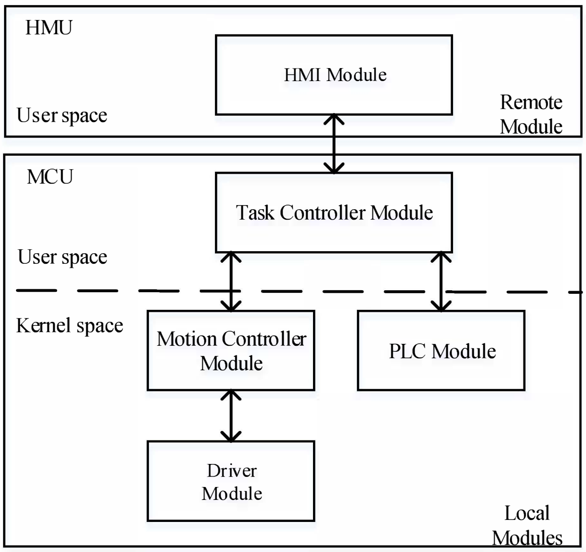

Fig. 3. The functional modules of the open CNC system

1) HMI Module: HMI is the most direct present of a CNC system, it provides users with a convenient and friendly interface. HMI collects user's input control commands and sends these commands to the task controller through network. At the same time, it gets and displays the state and error information feed back from task controller.

2) Task Controller Module: Task controller gets user operations from HMI and controls the motion controller and PLC to perform a complete machining task, according to the machining program. The machining tasks include interpretation and execution of G/M code program, logic sequential control, error detection, diagnosis and treatment.

3) Motion Controller Module: Motion controller receives the motion data of each axis from driver, then makes the trajectory plan and calculates the corresponding axis's positions using different interpolation algorithms, finally send them to the driver. In addition, it also responsible for alarm processing and servo driver parameters' reading and writing and so on.

4) PLC Module: PLC is mainly responsible for the control of discrete I/O, therefore, it is used to control the emergency stop switch, cooling switch, exchanging cutting tool switch and so on.

5) Driver Module: Driver module is responsible for receiving the control commands from motion controller, encapsulating these commands into data frames according to the bus communication protocol and sending them to servos. At the same time, getting the feedback data from servos and sending them to motion controller.

III. ETHERCAT RELATED DESIGN

A. EtherCAT Master Protocol Stack

The main tasks of an EtherCAT master are the network initialization and the handling of the state machines of all devices, the process data communication and providing acyclic access for parameter data exchange between master application and slave. The EtherCAT master stack consists of several modules which implement base services as shown in Fig. 4.

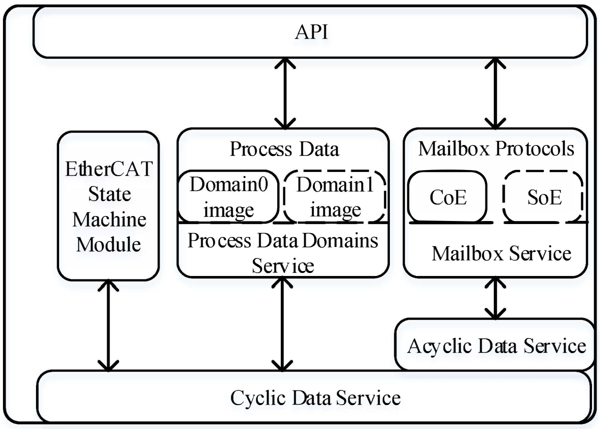

Fig. 4. EtherCAT master stack architecture

1) Cyclic Data Service: This service implement the cyclic input and output data exchange of EtherCAT frames. Cyclic process data is immediately updated with the input/output process data image in every cycle. Acyclic data is buffered for processing by the acyclic data service.

2) Acyclic Data Service: The service implement all acyclic communication which run in parallel to the cyclic data exchange to initialize and control the EtherCAT network. The EtherCAT frames are not sent directly. Instead they are buffered and sent by the cyclic data services after the cyclic data is transmitted.

3) Mailbox Service: Mailbox transfer is a non real-time service used to access application configuration data(CoE, SoE). One or more mailbox protocols should be support by every master. The mailbox service provides the common base functionality for several EtherCAT mailbox protocols implementations. It works on the top of the acyclic service.

4) Process Data: To exchange the process data the EtherCAT Master manages separate images for input and output and implement service to access the process data.

a) Process Data Image: Slaves offer their inputs and outputs by presenting the master so-called "Process Data Objects" (PDOs). The application can register the PDOs' entries for exchange during cyclic operation. The sum of all registered PDO entries defines the "process data image", which is exchanged via datagrams with "logical" memory access (like LWR, LRD or LRW).

b) Process Data Domains Service: Process data domains service is responsible for managing the process data image by creating so-called " domain image " , which allow grouped PDO exchange. They also take care of managing the datagram structures needed to exchange the PDOs. Domains are mandatory for process data exchange, so there has to be at least one. There is no upper limit for the number of domains, but each domain occupies one FMMU in each slave involved, so the maximum number of domains is limited by the slave.

5) EtherCAT State Machine Module: The EtherCAT state machine defines the network behavior of the master and of the slaves. The EtherCAT master implement an individual virtual state machine for each slave in the configuration and for itself. The EtherCAT state machine, which can be seen in Fig.5, leads through the process of configuring a slave and bringing it to a certain application-layer state.

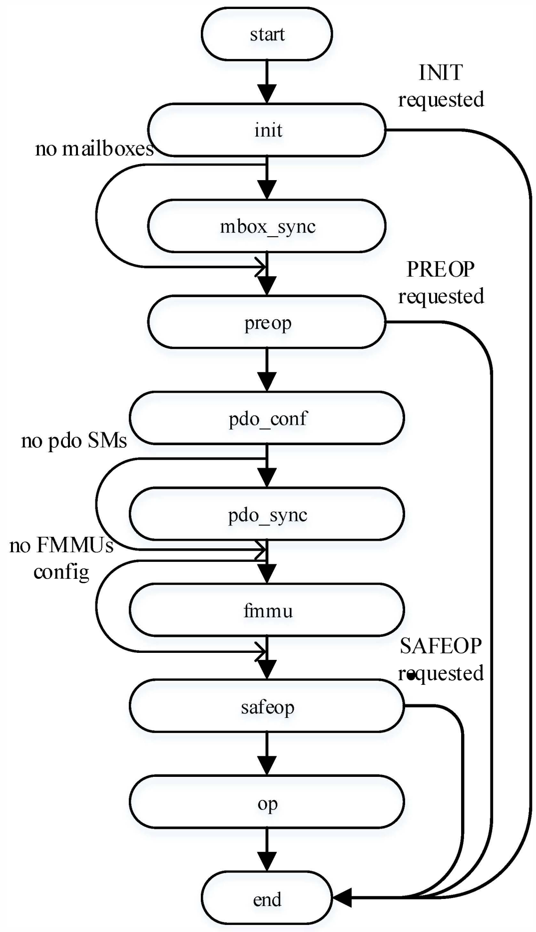

Fig. 5. The EtherCAT state machine

B. The Bottom Structure of the Open CNC System

The bottom structure of the open CNC system based on EhterCAT, which can be seen in Fig.6, has added a parameter configuration module on the basis of the original structure in user space, used to transfer configuration information (axis number, axis address, cycle time, etc) to EtherCAT master driver module. Configuration information is stored as a file in .INI format and transferred using shared-memory mechanism.

To solve the problem of data exchange between motion controller module and EtherCAT master driver module, we have constructed a series of interface functions, which generally can be divided into A, B two categories, as shown in Fig.7. Functions of A are used for periodic data exchange, implemented by using process data mechanism of EtherCAT protocol, which include functions associated with position control of servo axis and with velocity control of spindle axis and so on. Functions of B are used for non periodic data exchange, implemented by using mailbox mechanism of EtherCAT protocol, which include functions of alarm processing and driver parameters reading and writing, etc.

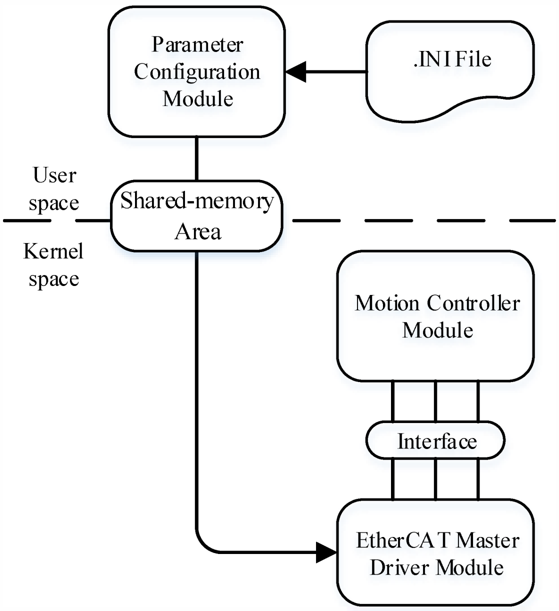

Fig. 6. The bottom structure of the open CNC system

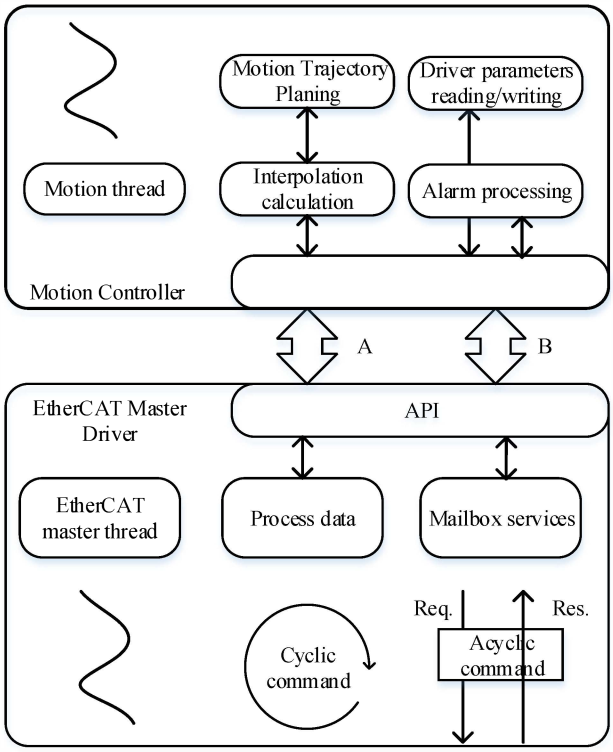

Fig. 7. The Interface structure of two modules

C. *Method of Thread synchronization*

After the EtherCAT master driver module and the motion controller module are initialized, they will create a motion thread and a EtherCAT thread respectively, as shown in Fig. 7, in the background. In each real-time cycle (interpolation period), EtherCAT master thread will get the interpolation data of motion thread from the interface functions. If the two threads can not maintain accurate synchronization, it is possible that the value to the EtherCAT master thread reading from the interface is not the current value calculated by the motion thread, resulting in inconsistent data.

To solve this problem, a method of thread synchronization is proposed, which motion thread follows EtherCAT master thread. As shown in Fig. 8, the motion thread is designed as a kernel real-time thread and EtherCAT master thread is designed as a read-time periodic thread. EtherCAT master thread is responsible for receiving and sending of frame , assembly and dissection of frame, and exchanging data with motion thread. Motion thread is responsible for data processing. In each real-time cycle, EtherCAT master thread is executed first. It encapsulates the output data stored on the previous cycle into EtherCAT frames and then exchanges data with motion thread though the interfaces. EtherCAT master thread can not get motion thread's output data of this cycle in that it is followed by the motion thread. So it stores the motion thread's output data of last cycle and provides motion thread its input data of last cycle as well. After exchanging data, it sends frame to the network and polls the hardware. If the frame has been received in the meantime, it will be dissected. At last, it stores the input data dissected from the frame and wakes up motion thread. When motion thread is awakened, it performs data processing operations and after that it suspends itself and waits for the next cycle to be awakened.

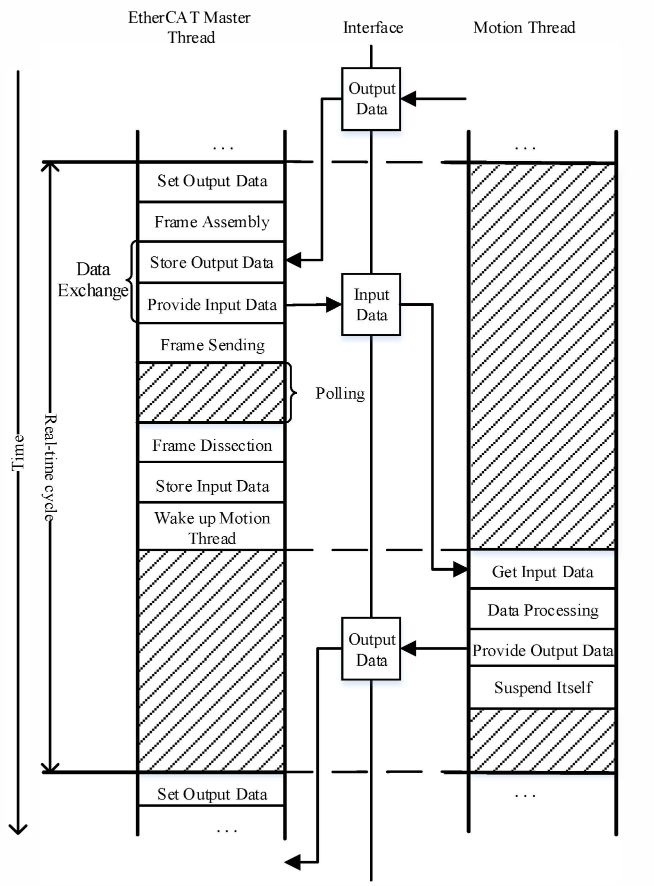

Fig. 8. Method of Thread synchronization

IV. EXPERIMENT AND ANALYSIS

A. The Prototype System and Experimental Environment

The prototype open CNC system has been developed using EtherCAT network (Fig. 9). The MCU is an IPC with a dual core 1.67-GHZ x86 CPU, a 2-Gbytes RAM, a 4-Gbyte hard drive and a EtherCAT communication card. The HMU is single-board computers with a single core 1.67-GHZ x86 CPU, a 512-Mbytes RAM, a 4-Gbyte hard drive. Five ServoOne drivers and matching motors have been selected as the slave stations, which are developed by LT-i Electric Corporation. Four of them are used to control servo axises and the other is used to control spindle axis. The software environment includes Red Hat Linux 9.0, Linux kernel 2.4.20, RTLinux 3.1 and a CNC software system. The architecture of the open CNC software system has been modified accordingly. The connections between the MCU and the drivers are realized through Ethernet cables.

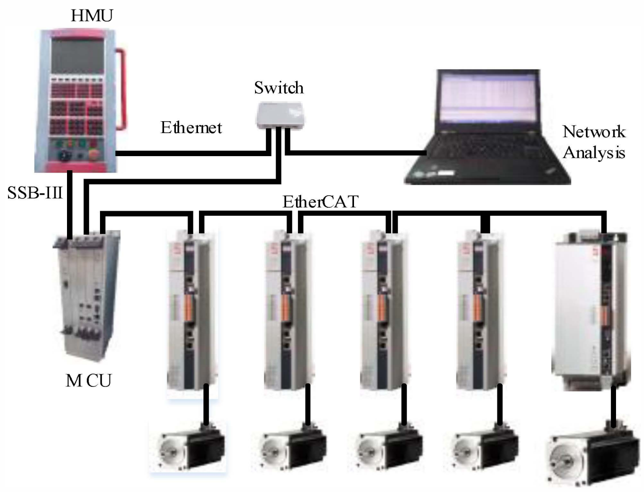

Fig. 9. Connection of the open CNC system based on EtherCAT network

At present, the prototype CNC system has been applied in the field of aviation manufacturing, used to machine the Aircraft Structure Parts. The testing platform consists of the prototype CNC system and a matching bridge type gantry five-axis machining center, as shown in Fig. 10. its spindle speed can reach to 24000r/min and the maximum feed rate is 24m/min. In this paper, the machining precision of the prototype CNC system is tested by using “S” specimen, which can reflect the comprehensive precision of moving parts and dynamic response characteristics in multi-axis synchronous CNC machining.

Fig. 10. The five-axis machining center

B. The Experimental Results and Analysis

The “S” specimen processed by the testing platform, as shown in Fig. 11. The specimen surface is smooth, without obvious tool marks. After testing, the processing accuracy is achieved by 0.08mm, there is only one point exceeds the tolerance and its maximum deviation is 0.065mm. The specific parameters are shown in Table 1. The result is indicated that the machining accuracy and dynamic characteristics can meet the processing requirements of typical aircraft structure. The main indicators of the CNC system has reached the requirements of high-grade CNC system.

Fig. 11. The “S” specimen

TABLE I. THE TESTING RESULT OF “S” SPECIMEN

edge strip thickness tolerance	permissible tolerance	$3mm \pm 0.1$
	max measuring deviation	$-0.08mm$
profile tolerance	permissible tolerance	$\pm 0.05mm$
	max measuring deviation	$0.065mm$
	bad points	1

V. CONCLUSIONS

As a new generation of CNC system, the open CNC system based on real-time Ethernet and digital servos has become a hot research topic and also become the development trend of CNC system in the future. In this paper, we put forward the hardware and software architecture of the open CNC system firstly. And then we apply EtherCAT technology into the open CNC system and describe EtherCAT related design in detail. Finally, we verify the performance of the prototype CNC system through a workpiece machining experiment. The future research work includes the network platform for the open CNC system and its information security.

ACKNOWLEDGMENT

This work was supported by “High-end CNC Machine Tools and Basic Manufacturing equipment”, the Major National R&D Projects of Research on Functional Safety Technology of CNC System under Grant No.2014ZX04009-031.

REFERENCES

[1] D. Yu, H. Lin, R. Guo, J. Yang, and P. Xiao, “The research on realtime middleware for open architecture controller,” in Proc. 11th IEE Int. Conf. Embedded and Real-Time Comput. Syst. Appl. (RTCSA’05), 2005, vol. 00, pp. 80–83.

[2] Du S, Yu D, Hu Y, et al. A component model and layered system architecture for reconfigurable CNC systems[C]//Information and Automation, 2009. ICIA'09. International Conference on. IEEE, 2009: 922-927..

[3] Hu Y, Yu D, Du S, et al. Design and implementation of reconfigurable CNC system based on Fieldbus[C]//Information and Automation, 2008. ICIA 2008. International Conference on. IEEE, 2008: 794-799.

[4] Yu D, Hu Y, Xu X W, et al. An open CNC system based on component technology[J]. IEEE Transactions on Automation Science and Engineering, 2009, 6(2): 302-310.

[5] Hu Yi, Yu Dong, Li Peinan, et al. Design and implementation of an open CNC system based on field bus [J]. Journal of Chinese Computer System, 2008, 29 (9): 1745-1749.