INTRODUCTION OF COACK TO THE HIBP CONTROL

H.Ogawa, T.Inoue, Q.Tsukada, M.Yokota, M.Kojima, J.Kodaira, A.Nishizawa, T.Ido, Y.Kawasumi,
S.Kato, K.Yamazaki, I.Abe*, T.Kosuge*, K.Nigorikawa*, M.Mutoh[†], and Y.Shibasaki[†]
National Institute for Fusion Science
*High Energy Accelerator Research Organization (KEK)
[†]Laboratory of Nuclear Science, Tohoku Univ.

Abstract

A PC based control system has newly developed for the Heavy Ion Beam Probe (HIBP) by using COACK (Component Oriented Advanced Control Kernel). The HIBP is important plasma diagnostic for the local measurement of the electric field and the turbulence of the density and the electric field. The HIBP consists of an Au^- ion source, a tandem type 6 MeV accelerator, beam line and an analyzer. We have implemented COACK for the upper layer of the control system and developed the operation console, the operation data logging function and the I/O functions. On the other hand, lower layer consists of PLCs, function generators and a CAMAC, etc. The total control points are over 400. As a result, we have completed the HIBP control system for short period with low cost, and the system is stably running now. The detail of the development of the HIBP control system is presented in this paper.

1 INTRODUCTION

We have a research project which is called Large Helical Device(LHD) project (Fig.1).

The mission of this project is to develop a scientific basis for stellarator fusion reactor. An important feature of the LHD design is a high magnetic field of 3 to 4 Tesla produced by one of the largest super-conducting magnet in the world.

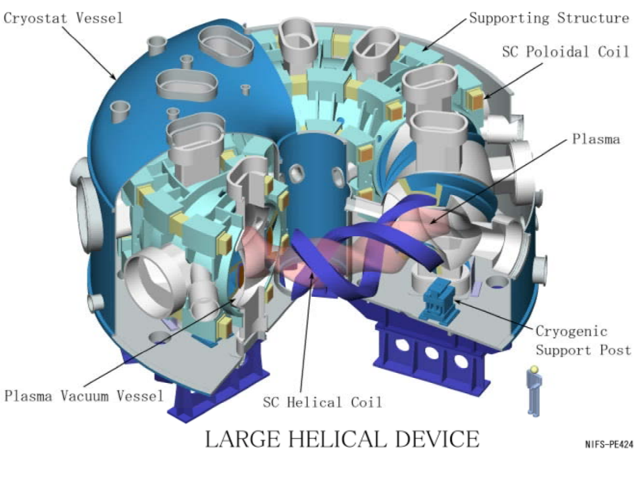

Fig.1 Bird view of LHD

2 CONSTRUCTION OF THE HIBP

The HIBP[1][2] is one of the about 50 diagnostic devices employed in LHD. The purpose of the HIBP is to analyze the local electric field inside the fusion plasma and to research the plasma turbulence of the density and the electric field. The whole view of the HIBP is shown in Fig.2.

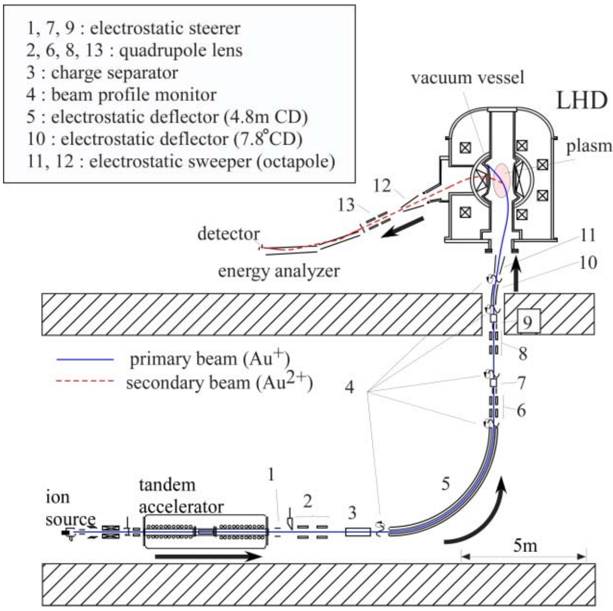
1, 7, 9 : electrostatic steerer
2, 6, 8, 13 : quadrupole lens
3 : charge separator
4 : beam profile monitor
5 : electrostatic deflector (4.8m CD)
10 : electrostatic deflector (7.8°CD)
11, 12 : electrostatic sweeper (octapole)
vacuum vessel
LHD
plasma
12
13
detector
energy analyzer
11
10
9
8
7
6
4
primary beam (Au+)
secondary beam (Au2+)
ion source
tandem accelerator
1
2
3
5
5m

Fig. 2 Heavy Ion Beam Probe

The dimension of the HIBP is 10m width and 15m height. The components of the HIBP are classified into four groups; an "Au^- ion source", a "6 MeV tandem type accelerator", a "beam line", and an "analyzer".

In the beginning, the Au^- ion is generated by the ion source. Then, it is changed into Au^+ and accelerated up to 6 MeV by the tandem type accelerator. After the ion is injected into LHD, Au^+ will be excited into Au^{2+} in the plasma. And the energy of the ion which comes out from the LHD is measured with the energy analyzer.

3 CHARACTERISTICS OF HIBP CONTROL

The fundamental HIBP,i.e.; the ON/OFF switching of the power supplies, the heater temperature, and the vacuum system are controlled by PLCs. There are beam scanning mechanisms, which are named "sweeper", at the port to LHD. The scanning waveform is set by a function generator. The quadrupole lenses and the steereres (beam alignment mechanism), etc are controlled by the high voltage power supplies with CAMAC DA converters. The pressures in the beam line are monitored by the vacuum gauge.

The HIBP control system consists of many controllers and the total number of I/O points exceeds 400. The previous HIBP control system is shown in Fig.3.

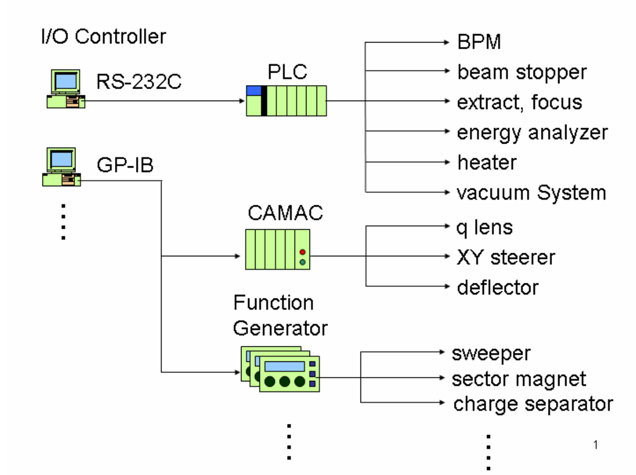

Fig. 3 Previous HIBP control system

We planned to control each controller by LabVIEW individually. Nevertheless, in the actual operation, only a single researcher has to operate all the HIBP components at the same time. Therefore, more efficient and user-friendly system is required. Furthermore, since the control parameters are closely related to the plasma diagnostic data, they should be recorded automatically and unitarily. It is forbidden during the plasma experiment to enter the LHD experiment hall where the HIBP device is located. Normally, the actual operation will be done at the Central Control Room which is 250m away from the LHD experiment hall. Therefore, the remote operation through the network is required.

To fulfil the above requirements, we decided to introduce COACK.

4 FEATURES AND FUNCTIONS OF COACK

COACK stands for "Component Oriented Advanced Control Kernel". Originally the COACK has been developed by KEK and Tohoku Univ. and NIFS has joined the project in 2000. The COACK has 3 remarkable features as follows;

1) PC based operation management system.
2) Component oriented system by using DCOM.
3) Easy development by using Visual Basic or LabVIEW.

The main functions of COACK are as follows;

1) ActiveX for data communication
   It enables to get values, set values, and receive interruptive events very easily from the server.
2) DCOM test tool.
   This tool performs powerfully during the communication check after the installation and the debugging.
3) Operational data logging and archiving.
4) Virtual machine.

The COACK has many other various functions[3]. "Virtual machine" function is explained in detail later.

5 STEPS TO INTRODUCE COACK

Before introducing COACK, we need a Visual Basic education only, because all the programming is done by using Visual Basic in COACK.

There are 3 steps to introduce COACK.

The first step is a "Server and client installation". It will be completed within about 15 minutes by using COACK installer. The second step is "Establishing a I/O table and register to the server". Actually, this step is the most important and it occupies about 50% of whole introduction work. The final step is "Client programming". We can use a basic sample program, so, the programming becomes simple and mechanical work.

Second step: Establishing a I/O table and register to the server

In order to deal with the HIBP system by COACK, it is necessary to list up the I/O points with the layered device tree. An example of the HIBP I/O table is shown in Table.1.

Table.1 A part of HIBP I/O table

Device Tree			Controller	Property
HIBP	IonSource	heater	PLC	voltage current temperature switch
		focus		
		vacuum system		
		BPM		
	BeamLine	XY Steerer	CAMAC	voltage
		vacuum gauge	gauge controller	pressure

For example, the heater temperature is expressed like “HIBP", "SOURCE" in the device tree, "HEATER” in the node, “PLC” and “temperature” in the property. All the other I/Os are also added same way.

The I/O table is registered into the COACK server as a XML file. Then, the HIBP devices are virtually created inside the server. And the clients can accesses to the controllers through the table indirectly. We call this function, “Virtual machine”. It enables that the client can be developed independently from the I/O interfaces like RS-232C and GPIB. The virtual machine is very efficient when we introduce COACK to the HIBP, because some of the HIBP components are under-construction. Even if the new controller is added or the specification is changed frequently, only we have to do at the COACK side is to change the I/O table. And the client programming is not so much influenced by the progress of the device layer. The COACK has a great extendibility and flexibility.

Third step: Client Programming

Thanks to the standard ActiveX components called "CustomComButton", "TransportControl", "LabInterface", the programmer doesn't need to notice about DCOM connection and can concentrate on the GUI and the user function programming.

Development window on Visual Basic is shown in Fig.4

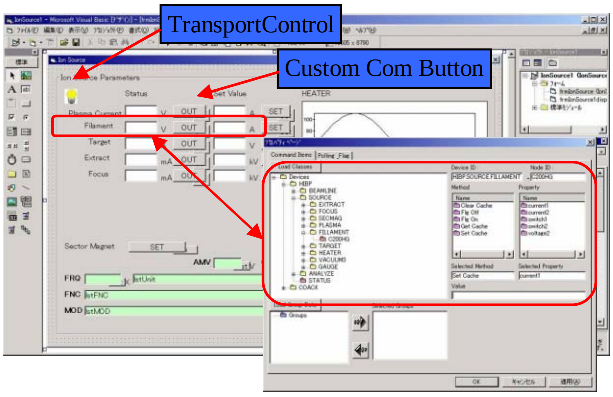

Fig.4 Development window on Visual Basic

A *TransportControl* and some *CustomComButtons* are pasted on the window. The first, what we have to do is to right click on the *CustomComButton* and make a relationship between the *CustomComButton* and device properties. The knowledge of DCOM communication is not necessary, because the source code is automatically generated if *TransportControl* is pasted on the window. For Example, the code for setup value to the server looks like;.

```
Private Sub cmdSetCache_Click()
    'Copy the input value to the property
    '   of the CustomButton
    ccbSetCache .PropertyValue = txtSetValue.Text

    'Execute set value
    ccbSetCache .cmdButton_Click
End Sub
```

Just 2 lines are specified in the function. The first line means copying the input value into the *CustomComButton* property. And the second line means executing set value. The programming for the "get current value" and "receive interruptive event" are also done easily.

COACK client can also be developed by LabVIEW by using *LabInterface* which is developed in Tohoku Univ..[4] Before introducing COACK into the HIBP control system, we planned to develop the I/O controller by using LabVIEW and control individually. Some of the I/O control programs exists, only we had to do was to add the COACK communication part in the existing I/O control program.

We made 10 of the operational windows and 9 of the I/O control windows which includes 2 virtual machine (Fig.5). Some of the functions are running stably in the actual operation.

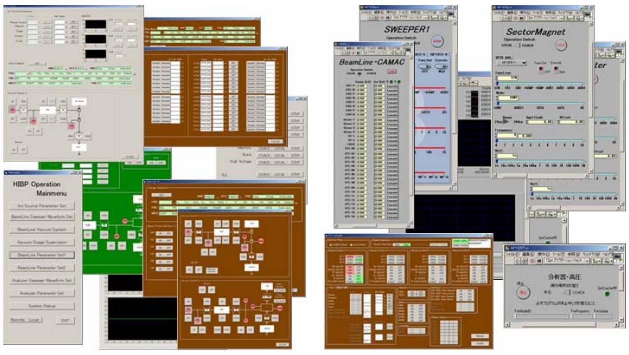

Fig. 5 HIBP operational and I/O control windows

6 CONCLUSION

We have developed the COACK system along with the construction of HIBP devices.

We have completed the development within 6 months and started the service for available components. Due to the introduction of COACK, we have achieved integrated control, unitary data management, remote control and improvement of extensibility and flexibility.

We recognize that the COACK is utilizable not only for an accelerator field but also a nuclear fusion field. The introduction to other control systems is also considered in the near future.

REFERENCES

[1] A.Fujisawa, et al, "A 6 MeV Heavy Ion Beam Probe for the Large Helical Device" IEEE TRANS. ON PLASMA SCIENCE Vol.22, No.4, Aug. 1994 P.395-402

[2] NIFS annual report Apl.2000-Mar.2001, P.155

[3] I. Abe, M. Mutoh, et al, "Recent status on COACK project", PCaPAC2000, Hamburg, October 2000.

[4] K. Nigorikawa, I. Abe, J. Kishiro, S. Kurokawa, T. Kosuge (KEK), M. Mutoh (LNS) "GUI AND I/O INTERFACE FOR COACK-II" ICALEPCS 1999, Trieste, Italy