J Fusion Energ (2014) 33:412–416
DOI 10.1007/s10894-014-9689-2

ORIGINAL RESEARCH

Application Software of Main Control Server for Neutral Beam Injection Control System

Yuanzhe Zhao · Chundong Hu · Peng Sheng · Xiaodan Zhang · Deyun Wu · Qinglong Cui · Rui Zhang · Yulian Lin

Published online: 10 April 2014

Abstract As the center of neutral beam injector (NBI) control system, main control server controls the whole NBI system's input and output services, and it verifies whether the experimental parameters are reasonable, stores the real-time data, monitors the statuses of the experimental equipments in real time and makes it easy to perform self-protection and alarm automatically. In order to realize these functions, MYSQL technology, client/server communication mode, multithreading and cyclic redundancy check technology are used in this paper.

Keywords NBI · MCS · MYSQL · CRC

Introduction

The neutral beam injector (NBI) is one of the main plasma heating methods for Experimental Advanced Superconducting Tokamak (EAST). Every NBI consists of two neutral beam ion sources, arc power supply, filament power supply, high voltage supply, vacuum system, water system, cryogenic system, diagnostic system and control system [1]. NBI control system provides the remote timing, remote monitoring and command reference control for all power supplies. It has four levels that are shown in Fig. 1: remote monitoring level, main control level, site measurement level and video monitoring level [2]. MCS works at the main control level, the center of the NBI control system.

Requirement and Analysis of MCS Software

The MCS software designed by C++ language runs on the Linux operating system. As the foundation of remote monitoring of NBI control system, it is links the site measurement level and remote monitoring level. To ensure the reliability and stability of NBI system, MCS should be furnished with the following specific functions:

a. User identity validation function. For NBI control system safety operation and avoiding network attacks, only legally authorized users can carry out experiment operations. MCS sends read or write authorities to different terminals. Then the different users can read or write different data according to the authorities getting from MCS.
b. Initialization of parameters of the whole NBI control system. Once MCS software runs, it gets configuration from profiles that store last experimental configuration and initializes the tables of MYSQL database. Through this function MCS software can reset all parameters of NBI system.
c. Parameter storage and validity check. Once MCS receives configuration from a terminal, it will store and send these data to terminals who need these data after it confirms the configuration is reasonable.
d. Monitoring the statuses of vacuum system and cryogenic system. MCS gets vacuum data and cryogenic data from vacuum system and cryogenic system. If the data is out of the normal range, MCS software will send an error message to monitoring terminal to display. At the same time, MCS carries out self-protection to protect NBI system.

National Natural Science Foundation, No. 11075183.

Y. Zhao (✉) · C. Hu · P. Sheng · X. Zhang · D. Wu · Q. Cui · R. Zhang · Y. Lin
Institute of Plasma Physics, Chinese Academy of Sciences, Hefei 230031, China
e-mail: zyz@ipp.ac.cn

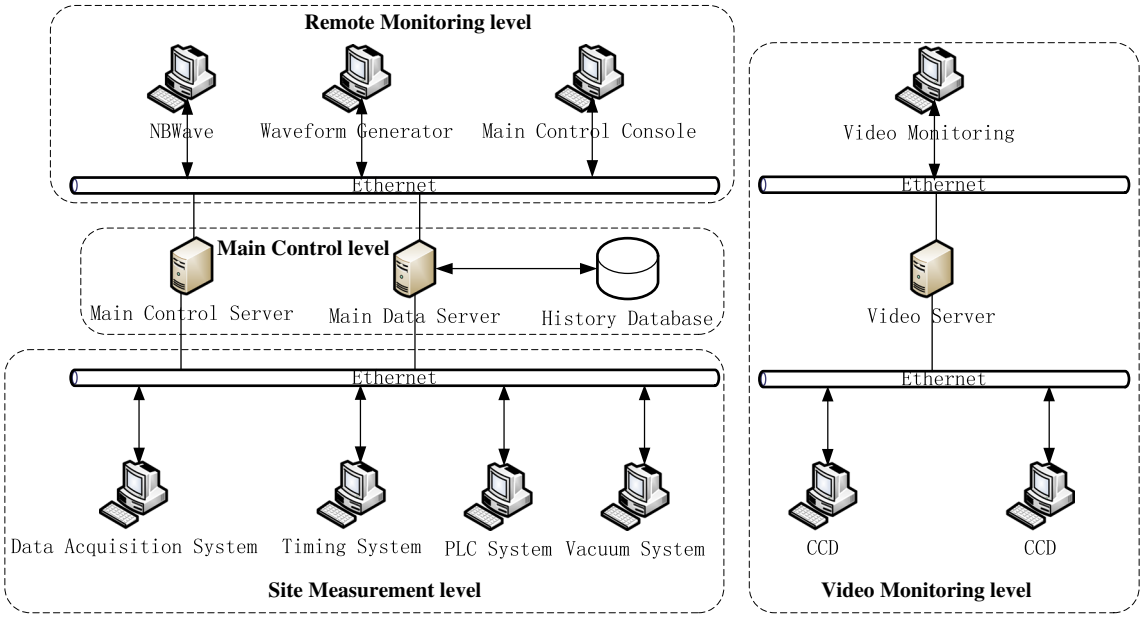

Fig. 1 Structure of NBI control system

e. Monitoring the statuses of power systems. MCS gets the statuses of arc power system, filament power system, high voltage power system, snubber power system, magnetic power system and suppressor power system from timing system (TS). MCS sends these statuses to the monitor terminal in real-time. Once the status equals to zero, MCS sends an error message to monitor terminal and carries out alarm and self-protection automatically.
f. Data synchronization. There are many terminals in the site measurement level and remote monitoring level that send and get data from MCS. MCS uses different threads to communicate with different terminals. With the help of multithreading technology, MCS can send the same data to different terminals at the same time.
g. Real-time monitoring of power amplitude. MCS software gets arc power value, filament power value, high voltage power value, snubber power value, magnetic power value and suppressor power value from TS in real-time. If the received value is higher than the setting value, MCS sends an error message to the remote monitoring terminal and stops power supply.

Implementation

MYSQL Technology

MCS works under Linux operating system using C++ language, its core technology is MYSQL technology. Before the MCS software runs, the database and tables used in the MCS software have been created. Table 1 shows a few examples of MYSQL tables. MCS inserts record to database through insert statement, updates record of database using update statement and deletes record from database by delete statement.

There are 19 tables in the NBICS database, including UserInfoTab, RegTab, TsDAChInfoTab, TsDOChInfoTab, ModeTab, LogTab, ShotNoTab, DaqTab, HvsTab, DAQTab, McsRecvStatusTab, TsStatusTab, VacTab, TsDAValueTab, TsDAGainTab, TsDIChConTab, EastTimeTab, PrbModeTab, TagTab. UserInfoTab is used to store all users' usernames, passwords and authorities. RegTab is used to store all terminals' IPs, device IDs, service No, Socket and command IDs. TsDAChInfoTab is used to store amplitude settings of power systems, including channel name, channel No, channel gain, and channel amplitude value. TsDOChInfoTab is used to store timing settings of power systems, including channel name, channel no, and channel timing value. ModeTab is used to store experimental model and countdown time. LogTab is used to store log message and makes it easy to failure checking. ShotNoTab is used to store current shot No, next shot No and EAST shot No. DaqTab is used to store the configuration of data acquisition system, including channel name, channel No, channel gain, and channel unit. HvsTab is used to store value and timing of high voltage system. DAQTab is used to store IP and acquisition time of all data acquisition system. McsRecvStatusTab is used to store error message of every ion source. TsStatusTab is used to store the status of TS. VacTab is used to store the value of vacuum system. TsDAValueTab is used to store the amplitude feedback of power systems. TsDAGainTab is used to store the gains of analog channels. TsDIChConTab is used to store the statuses of power systems. EastTimeTab is used to store the

time of NBI which is gotten from EAST control system. PrbModeTab is used to store the mode of probe feedback system. TagTab is used to store the tags that are used for internal communication of MCS.

TCP Data Structure Uniform

The working model of MCS is based on client/server (C/S) model and communicates with terminals through TCP/IP protocols. In view of the processing data between MCS and terminals, the unified data structure must be harmonized. The NBI TCP communication data structure is defined as follows, shown in Fig. 2.

The part of NBI TCP header information occupies 60 bytes, which contains information that is necessary to a reliable and effective communication between equipment of NBI control system [3]. The CRC is the check value (or data verification), the value of it is based on custom cyclic redundancy check (CRC) algorithm. CRC can verify the legality of the custom data and find the wrong data due to network failures. The numbers at the top of figure express the size of bytes in different fields [4].

Control and Data Process

MCS is the server side of NBI control system. Once MCS software runs, it will create three threads: thread A, thread B and thread C. Figure 3 is the working and data flow chart of MCS. For the security of network communication, all terminals (expect for MCS) take steps of registering to MCS as one of its client when the terminal starts and applies for logout when it exits [5].

As shown in Fig. 3, thread A is used to listen to clients' connection. As soon as terminal's registration process, thread A will create thread A' to listen to other terminals' connections and the old thread A is used to deal with current terminal's information. If the terminal's information has existed in the RegTab table, thread A will update

Table 1 A few examples of MYSQL tables in MCS

RegTab						
UserName	DevID	IP	SrvNo	Cmd	Active	Timer
WG_1L	1,001	192.168.20.11	1	3	1	3,600
TS_1L	101	192.168.2.6	2	−1	1	3,600
nbi1 l	10	192.168.20.141	3	1	0	1,985
…	…	…	…	…	…	…
TagTab						
TagName	UserNameString	Num	Status	flag	UserName	
SHOT_NO	nbi1 l_mt*WG_1L*	2	0*0*	0	MDS	
1L_TS_CONFIG	TS_1L*WG_1L*	2	0*0*	0	nbi1 l	
1L_EXP_RESET	TS_1L*	1	0*	0	nbi1l	
…	…	…	…	…	…	
HvsTab						
HvTagName	CONVERTHV_V	STRIFIRETIMES	DISCHARGETIME			
NBI_1L_ACC_SV	30	1	2,000,000			
NBI_1R_ACC_SV	40	2	3,000,000			
NBI_2L_ACC_SV	50	1	5,000,000			
NBI_2R_ACC_SV	60	1	1,000,000			
TsDOChInfoTab						
ChName	ChNo	Gain	ChData			
NBI_1L_ACC_START	28	1	(17,000,000,2,000,000)			
NBI_1L_ACC_CMD	0	1	(16,900,000,2,200,000)			
NBI_1L_FIL_START	6	1	(2,000,000,18,050,000)			
…	…	…	…			

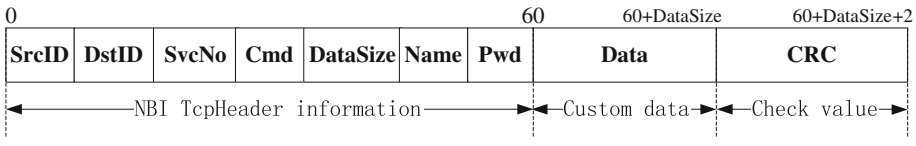

Fig. 2 Definition of the NBI TCP communication data structure

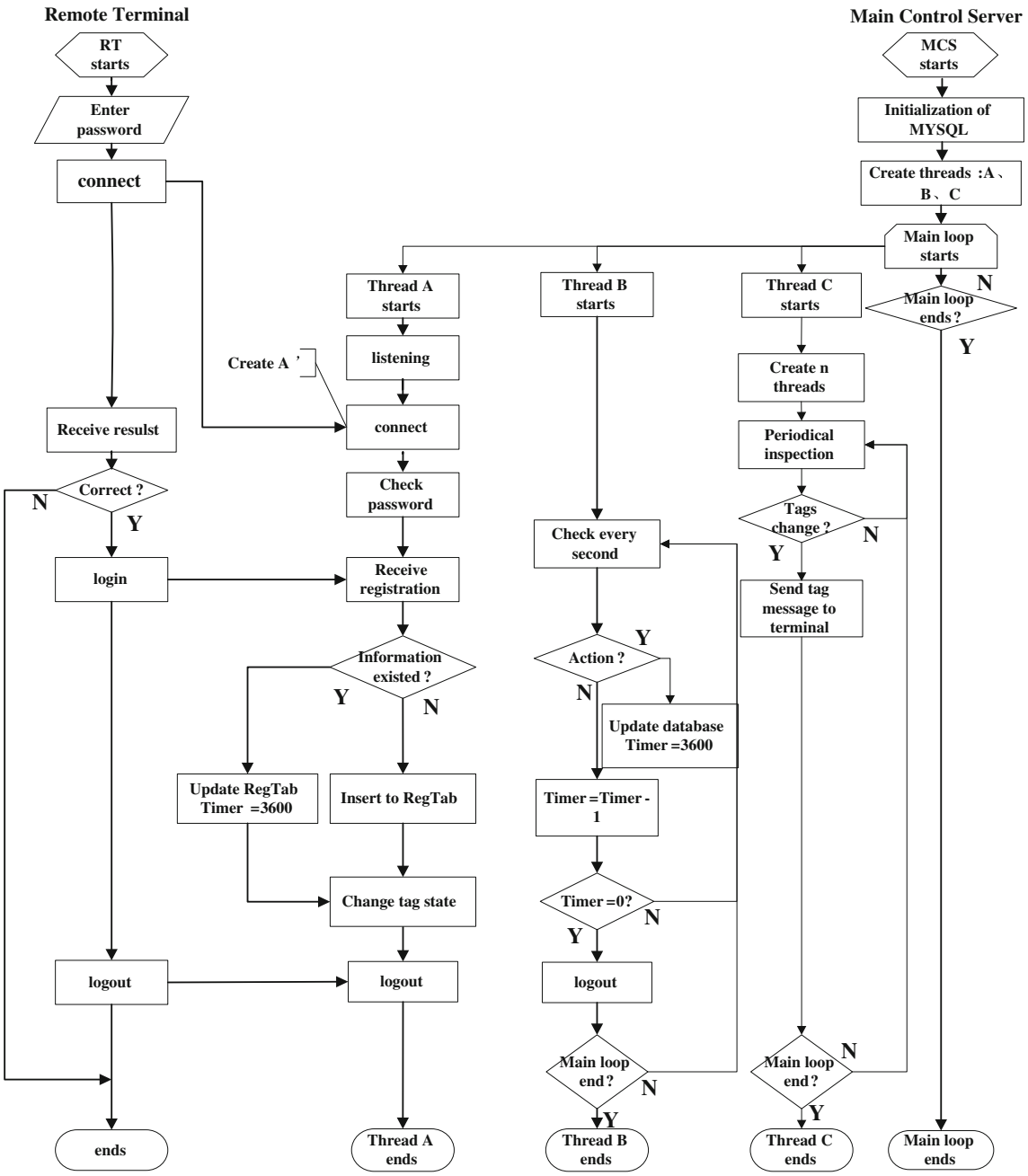

Fig. 3 The working and data flow chart of MCS

terminal's information. If there is no information of current terminal, thread A will insert current terminal's information to RegTab table. At the same time thread A collects and analysis data receiving from terminal, saves data to the right table and changes the state of the data's tag. For example, if thread A gets ShotNo message from main data server (MDS), it will save current shot NO to ShotNoTab table and update the state of SHOT_NO tag of TagTab table.

Thread B checks action of the existed users every second. If one user has no action, thread B subtracts counter with one and if the counter is equal to zero, thread B deletes this user's information from RegTab table. Else if the user has an action, thread B sets the value of counter to 3,600.

Once thread C is created, it will get tags' information from TagTab table, and then create multiple threads according to the number of tags in the TagTab table. One thread corresponds with one tag. These threads monitor the state of every label in real-time. If the state of one tag is changed, current thread will send the message of that tag to the terminals whose information is saved in the UserNameString field of TagTab table.

Discussion and Conclusions

The design and implementation of MCS for NBI control system is described in this paper. MCS application software is very well to meet the requirements of the experimental process of NBI control system. With the help of MYSQL technology, MCS software achieves the storage of real-time data and synchronous transmission. At the same time, the multithreading technology reduces the time of program operation and TCP protocol ensures reliable packet delivery through the network. Experimental results show that all the specified functions of MCS are reachable and MCS can perform well as a link between the site measurement level and remote monitoring level, with the characteristics of security and stability.

Acknowledgments This work is supported by National Natural Science Foundation (No. 11075183), the Knowledge Innovation Program of the Chinese Academy of Science (the study of neutral beam steady-state operation of the key technical and physical problems).

References

1. C. Hu, NBI Team, Plasma Sci. Technol **13**(5), 541–545 (2011)
2. P. Sheng, C. Hu, S. Song, et al., Control plan of EAST-NBI ion source test bench. Asia pacific conference on plasma science and technology, Jeju, Korea, 4–8 July 2010
3. X. Zhang, Z. Liu, P. Sheng et al., Research of data server application software for NBI mechanical control on EAST. Appl. Mech. Mat. **214**, 538–542 (2012)
4. D. Wu, C. Hu, P. Sheng et al., Design of main control console software in EAST neutral beam injector's control system for the first beam line. J. Fusion Energ. **32**(5), 552–556 (2013)
5. Y. Zhao, C. Hu, P. Sheng et al., Design of timing system software on EAST-NBI. J. Fusion Energ. **32**, 557–560 (2013)