

## Research of the Real-time Database in Embedded Configuration Software

Wang Xiujuan<sup>1,a</sup>, Li Xiaobing<sup>2,b</sup>, Cheng Meng<sup>3,c</sup>, Chen Yutang<sup>4,d</sup>, Zhang Zhongxin<sup>5,e</sup>, Cai Xiuyun<sup>6,f</sup>

<sup>1,2,3,5</sup>Mechanical and Electrical Engineering College, University of Electronic Science and Technology of China, Chengdu, China

<sup>4,6</sup>Dongguan Yuefeng Electronic Technology Limited Company, Dongguan, China

\*Corresponding author, e-mail: wxj2799@126.com<sup>a</sup>, 1014270681@qq.com<sup>b</sup>, chengmeng125@163.com<sup>c</sup>, leo@cables.com.tw<sup>d</sup>, 876156954@qq.com<sup>e</sup>, certificate@yfc-china.com<sup>f</sup>

### Abstract

*In recent years, the application of embedded technology and configuration technology in industrial control is more and more widely. The embedded configuration software which is combined of embedded and configuration has become the inevitable trend in industrial control field. Real-time database system as the core of embedded configuration software, the organizational structure whether reasonable and effective is directly related to the performance of the whole system, affecting field devices real-time communication and data transmission in graphic display interface. Based on a large number of configuration-related papers, this paper deeply researched the real-time database and using three layer storage structures which consist of shared memory, file system and general database. It improves the access efficiency of real-time database and data reliability in a timely manner.*

**Keywords:** embedded system, configuration software, real-time database, data structure, storage structure, modular design, and interfaces mechanism

**Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.**

### 1. Introduction

Configuration software develops quickly because of the widely application of PC. It specializes in the industrial-control field [1]. And embedded system has strong played an increasingly important role in industrial-control field. There have been many manufacturers and available embedded OS like Linux, WinCE, VxWorks and so on [2]. At present, the most popular international Commercial-embedded-configuration-software are Movicon X from PROGEA (Italy), WinCC from Siemens (Germany), InduSoft-CE1500 and InduSoft-300 both come InduSoft Web Studio. There are also some good domestic ones like kingview and Beijing Kunlun MCGS. Though the existing embedded configuration software have good man-machine interface, abundant drawing function, lifelike graphics display interface and flexible configuration, there are some certain deficiencies. Based on a large number of configuration-related papers, this paper deeply researched the real-time database which is the core of the embedded configuration operation.

### 2. Research Method

#### 2.1. Embedded System

Embedded system is the dedicated computer system which centers on application, bases on computer technology, software and hardware can be cut, adapts application system to strict with function, reliability, cost, volume, power consumption [3, 4]. It is the electronic equipment or devices which composed of microprocessor, peripheral equipment and related support hardware, embedded operating system and application software, to realize the functions such as control, monitoring and management for the other equipment [5], and its system structure as shown in Figure 1.

The embedded system's main features [6] are:

(1) Real-time: Can rapidly responses in the system response time limit to the foreseeable events or user intervention, have very strong real time;

- (2) Reliability: Usually work in unattended specific occasions, such as harsh environment or run continuously for a long time, has the high requirement of reliability;
- (3) Specificity: General is geared to the particular specific application, has certain specificity;
- (4) Diversity: Specificity determines its application in the field of different special, specific hardware and software needs to selects and develops according to the actual situation, reflects the diversity of embedded systems;
- (5) Cutting: In order to meet the specificity and the control of system cost, according to the practical application to cut developing, to achieve the most reasonable configuration;
- (6) Low power consumption: embedded products mainly in the face of some small application system which does not have large power, has a strict requirements on power consumption.

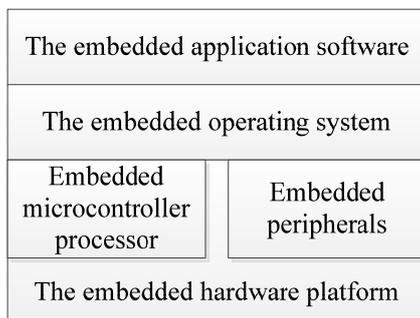


Figure 1. Embedded System Structure

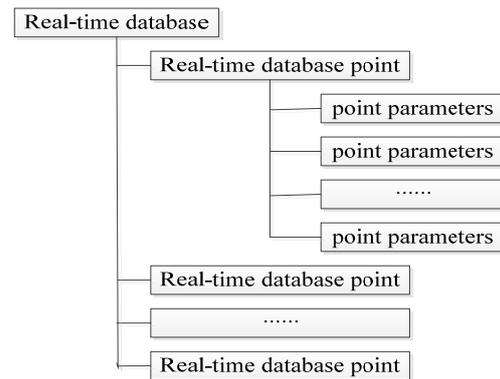


Figure 2. The Relationship between Point and Point Parameter Diagram

Table 1. Point Parameter Table

Type	Main parameters
Analog quantity	Point name, number, type, unit, connected devices, upper and lower bounds of initial value, offset, data, storage, marking, alarm tags, alarm level, etc.
Digital quantity	Point name, number, type, unit, connected devices, upper and lower bounds of initial value, offset, data, storage, marking, alarm tags, alarm level, etc.
Memory variables	Point name, number, type.

**2.2. Configuration Software**

Configuration software is the specialized software development environment works on the automatic control system monitoring layer to complete the data acquisition and process control [7], it provides a friendly graphical development interface and easy operation method, provided use of its various components can easily develop satisfying the various needs of monitoring application function, at the same time to control and management layer provides a variety of hardware and software interface, easy to integrate with other systems or programs [8]. Configuration software's main purpose is to make the automation engineer convenient to generate application system to satisfy his needs in don't need to modify the software program source code.

The development of embedded system in the field of industrial control necessarily promotes configuration software production and the development. Embedded configuration software running on hardware system as the core of the embedded processor and its supporting environment is mostly the embedded real-time multitask embedded operating system.

**2.3. Real-time Databases**

Real-time database (RTDB) is the core of the configuration. It is responsible for the equipment on site production process data acquisition and processing, and data organization

and management work, provides an important safeguard for the normal operation of the whole system. It not only gives the user interface the whole system running status data, convenient user for the corresponding control operation; but also has other functions such as the preservation and statistical analysis of the non real-time data, alarm processing and I/O data connection.

The transaction and data of real-time database both have timed featured or explicit time limit [9]. The correctness of real-time database system depends on both the data logical results and the time when it produced, that is to say, the system can accept inaccurate data within the time limit, but can't accept the accurate data more than the time limit [10]. And it should meet the requirement of data real-time and consistency, support a large amount of data sharing, maintain the consistency and integrity of data, and support the time limit of data and transaction [11]. The main purpose of real-time database transaction scheduling is to process transactions as much as possible within the stipulated time limit.

Real-time database abstracts each data object into a point (Tag) which contains several parameters. Every I/O device in the industrial site is associated with the corresponding Tag in database. The composition of point parameters as shown in Table 1 and the structure relationship between point and point parameters in real-time database as shown in Figure 2.

### 3. Results and Analysis

#### 3.1. Design of Real-time Database

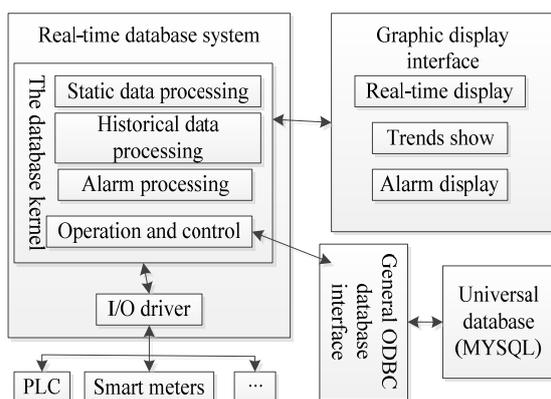


Figure 3. Real-time Database Data Flow Diagrams

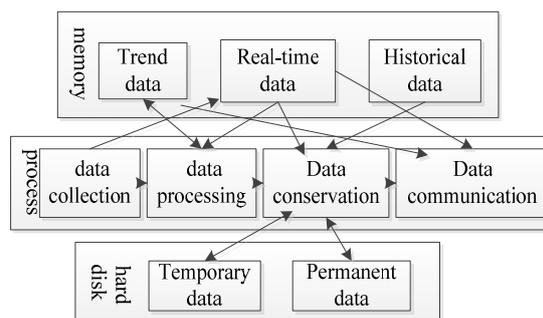


Figure 4. Memory Data Diagram

The data processing process of real-time database as shown in Figure 3. Display interface accesses the needed data from real-time database regularly to adjust the pixel in interface, with the intuitive images to show the running situation of the whole system; In addition, take the control command into field devices according to the operation of the user interface.

Real-time database as the data server, to provide data sources to graphical interface; and graphical interface as the client of the data, to get the data from the server side and display in real time, both constitute C/S mode. In order to satisfy the independence, real-time and consistency of the data in the system, this subject adopted three layers storage structure which composed of memory database, file system and general relational database. As shown in Figure 4.

(1) Storing the dynamic data which needs to update each sampling period in memory to guarantee the real-time response speed of real-time database.

(2) Storing the static data which does not require high real-time response speed or the unshared data which does not require long-term preservation in a file system.

(3) Storing the outdated production data which need to save long time and the shared data which has no special requirements in general relational database (MYSQL) to query and statistical analysis later.

Inter-process communication mechanism of Linux platform has main pipeline and named pipe, signal, message queues, shared memory, semaphores and cover interfaces [12]. Shared memory can provide strong support to real-time dynamic data interaction between real-time database and graphic display interface. Named pipe is the better choices to achieve static data interaction between real-time database and graphic display interface. It's convenient to realize data exchange between real-time database and universal database through the interface function provided by ODBC and database access API interface function provided by Linux. The storage structure designed in this paper which consists of shared memory, named pipe and ODBC interface communication mechanism as shown in Figure 5.

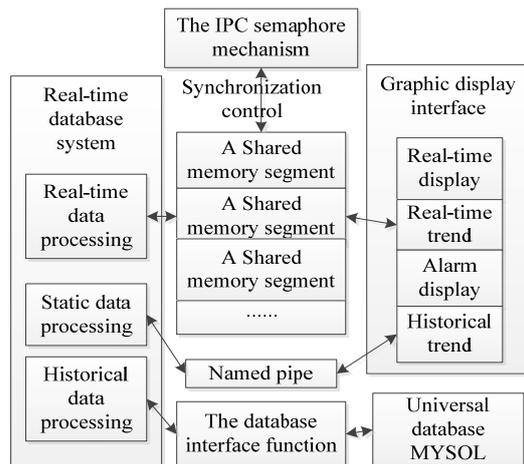


Figure 5. Three Floors Storage Structure of Real-time Database

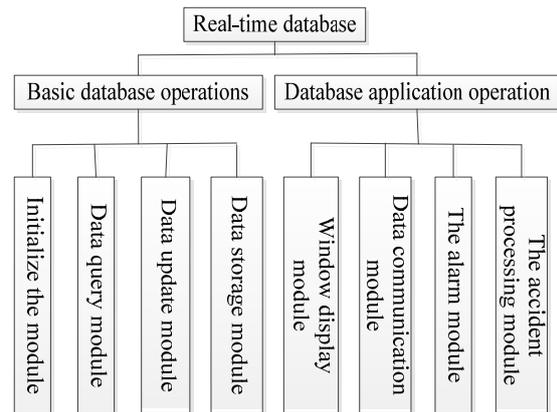


Figure 6. Real-time Database Structure Designs

Real-time database designs into modular to meet the requirements of embedded configuration system. The whole real-time database is divided into relatively independent modules, it is convenient to develop and test system fast. Its structure as shown in Figure 6.

(1) Initialization module: It is mainly used to constructing and initializing data in the memory database, and the establishment work of historical database.

(2) Data query module: According to the operation of customer's choice or the demands of system, to retrieval the data which meet the conditions in real-time database, and return query results.

(3) Data update module: To update the data which needs to be updated in the system, and to refresh the history database tasks according to actual conditions, etc.

(4) Data storage module: To save the data which satisfy the trigger condition or time condition to the history database.

(5) The window display module: According to the current display window ID, to query the data that corresponds to the query window's pixel in real-time database, and according to the return value to adjust the window's pixel.

(6) Data communication modules: To communicate with the field I/O devices according to the agreement, read the current production process data from device, issue the control instruction according to the device ID, achieve the scene device's control function.

(7) Alarm module: To test whether the data beyond the alarm limit, give the alarm information and save if it beyond.

(8) Accident processing module: To save the state of the system, the field data and operation records of the operators when the accident happened to the system.

### 3.2. The Implementation of Real-time Database System

Industrial field data including real-time data of on-site acquisition, system data, calculate data, attribute data, control and management data. All data can be represented by three data types that are analog quantity and switch quantity and strings. Real-time data implemented by

structure type, the different process type to distinguish by the real-time data type field in the structure. Implementation of real-time data structure types is as follows:

```

/* enumeration type tag real-time data process type* /
typedef enum {
double_t = 1,
bool_t
} pv_type_set;
/* joint type realize real-time data process value */
typedef union {
double dPV;
bool swHPV;
} pv_data_set;
/* the data type of the real-time data* /
#define NAME_LEN 20
#define DESC_LEN 50
typedef struct {
char name[NAME_LEN +1]; // Name of the data points
pv_type_set type; // Data point type
char desc[DESC_LEN+1]; // Data points describing information
pv_data_set pv; // Data point process
char domain[3]; // The domain of the data points
char eu[DESC_LEN+1]; // Data point engineering unit description
double euLow; // Data point engineering unit lower limit
double euHigh; // Data point engineering unit upper limit
double pvRaw; // Field measurement data
bool IsRanCon; // Whether the scale transform
double pvRawLow; // Data range lower limit
double pvRawHigh; // Upper limit of the data range
bool static; // Static data, historical data store to the file system
int storecyc; // The backup cycle
bool Is Alarm; // Whether the alarm
int Alarm Priority; // The alarm priority
double LowLowValue; // Alarm lower limit
double LowVaule; // Alarm Low limit
double HighHighvalue; // Alarm higher limit
double Highvalue; // Alarm high limit
double LowDevvalue; // Alarm low deviation values
double HighDevvalue; // Alarm high deviation values
} tag_node;

```

Using the support for real-time multitasking operating provide by Linux system, to perform the data acquisition and processing tasks of real-time database in the form of multi-process concurrent. Using the shared memory technology of IPC, can distribute the required memory space discretely in memory according to the application, then all discrete shared memory address form an index table, reached on managing all shared memory [13]. In practice, often combine the different processes or different workshop production data points into a separate data domain, built into indexing table structure of the domain table and the point table's two levels structure address, the structure as shown in Figure 7:

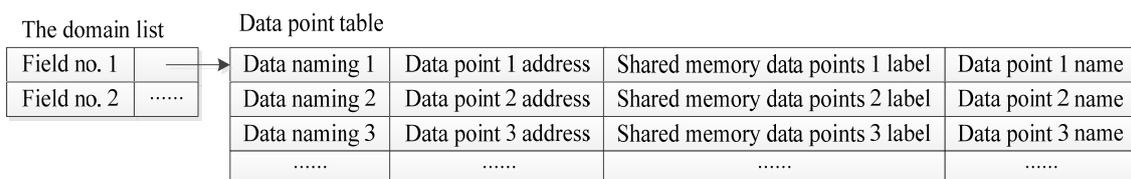


Figure 7. Domain Table and Data Point Table Two Levels Index Structure Diagram

The relevant data structure of domain table and point table as follows:

```

/* describe domain table the data structure of data item */
typedef struct {
char domIndex[3]; // The domain,
tbTag_item *tbTag_ptr; // Data point table address of the domain
} tbDom_item;
/* describe data point table the data structure of data item */
typedef struct {
char tagIndex[3]; //data point number
tag_node *tag_ptr; // Pointer to the data points
int shmid; // Store the data shared memory label
char name[NAME_LEN + 1]; // data point name
} tbTag_item;

```

Using the database interface can access and manipulate the database directly. It is convenient for user to develop I/O interface driver and exchange the data with other devices, make the real-time database has good versatility and openness. Some common database interface function as shown in Table 2:

Table 2. The Part Interface Function of Real-time Database

Function name	The return value type	Function description
GetTagNum();	int	Query the number of data points
GetNameByID(char * tagID);	Char*	Obtain the data point named by data ID
GetIDByName(char*tagName);	Char*	Obtain the data ID by data point named
GetPVType(char *tagName);	pv_type_set	Obtain data point process value type by data point named
GetPVByName(char*tagName, pv_data_set *pv);	int	Obtain data point process value by data point named
SetPVByName(char*tagName, pv_data_set *pv);	int	According to the data point named written to the data point calling process value

Real-time database development environment provides a real-time database configuration interface, and generate the database configuration file. In the configuration interface it is convenient for user to define all kinds of memory variables, I/O variables, set a variable process mode, etc., and generate a data dictionary; configuration files provide the basis for running environment generated data.

(1) The definition of data dictionary

Data dictionary refers to define the need variables in industrial control object and the device parameters need configure, part of the variable will be used as real-time database management object in the kernel.

The definition of data dictionary in the development environment has the following features:

(a) Specifies the data variable types: there are usually a variety of data types in the configuration system, such as analog quantity and switch quantity and character variables, etc. Specified data types in the configuration interface can be convenient the running environment to allocate reasonable storage space in memory for the corresponding data.

(b) Specifies the field devices which I/O variable should associate with: after specifies the associated equipment for the variable, can be collected the corresponding data in the field devices by the I/O drivers under the operating conditions;

(c) Set up the method of data processing: the original data acquire from field devices cannot be directly used for interface, should take the corresponding conversion operation or process, so set up the corresponding processing way for the data on the configuration interface;

(d) Specify the data sampling time: different data variables in the system have different requirements for sampling and need set up different sampling time for data;

(e) Set up the data preservation attributes: there's some data often needs to be saved in the system, convenient the system to statistical analysis for production status and alarm failure, so need set up different ways of data storage according to the actual needs, such as set-time storage, event trigger storage, and so on;

(f) Set up data alarm attributes: setting alarm limit and priority of variables data.

(2) The historical database

For the data need to long-term preservation of system, such as alarm information, trends information, and the data for realizes the compensation mechanism need save on regular or according the agreed conditions setting the corresponding storage table in configuration interface, such as system information table, the fault information table, etc.

(3) The configuration files storage

To save the XML format database configuration file generated from data dictionary and historical database information configured by develop environmental, for the use of system running environment.

Running environment is the ultimate use model of real-time database; this part of the design is good or bad directly affecting the efficiency of program.

(1) Configuration files parsing

Configuration file parsing is deploying file according to the database generate in the development environment, generate the tables which needed for the corresponding memory database and general database in memory space.

(2) Real-time database runtime environment

When the configuration software at run, the system to generate memory data file and historical database according to the configuration file. Memory database to store raw data collect from the field I/O devices, and make corresponding processing and save later; history database storage the data needed long-term preservation.

#### 4. Conclusion

Real-time database is divided into the development environment and running environment. Development environment provides a database design interface, the interface can set the data variable name, type and so on some regular option, and also can set the device node which the data is associated with and data processing methods, such as data sampling period, refresh time, etc. At the same time to design table in the history database for historical data in the system need long-term preservation and the table for recovery mechanism, such as alarm table, system information table, etc.; After configuration completed will to generate real-time database configuration file for the use of running environment analysis. In the running environment, the application program generate real-time database according to the configuration file of the configuration at first. If the history database does not exist, generate a new historical database; generate memory database at the same time and refresh data variable according to the configuration variable configuration of the sampling time and trigger event; to save the data need to save according to the event or at regular time. This component also provides historical data query, processing alarm etc. functions.

#### References

- [1] Tianliang Hu, Chengrui Zhang, Lin Yang, Junzhe Tan. *Engine Based Embedded Control System Design and Implementation*. Mechatronic and Embedded Systems and Application. IEEE/ASME International Conference. 2008: 269-275.
- [2] Wei Lin. Introduction to the development trend of configuration software. *Automation expo*. 2003; (1): 45-47.
- [3] Xiang Zhang, Jian Xiao. Embedded system and embedded operating system. *The instrument and automation*. 2002; (5): 12-16.
- [4] Nan Sang. The principle and application of embedded system development technology. Beijing: Beijing University of aeronautics and astronautics press. 2002: 4-6.
- [5] Jing Zhang, Xianyun Zeng. Summary of the embedded system. *Electric measurement and instrument*. 2002; 39(436): 42-45.
- [6] Yao Li, Wilson. *P.PARTOS-11: an efficient real-time operating system for low-cost microcontrollers*. Electronic Design, Test and Applications, Proceedings. The First IEEE International Workshop. 2002; 235-239.
- [7] Jianwei Li, Hong Guo. The design and development of monitoring configuration software. Beijing: metallurgical industry press. 2007; 105-118.
- [8] Qingbo Zeng, Hua Sun, Weihong Zhou. Monitoring configuration software and its application technology. Harbin: Harbin industrial university press. 2005; 12-21.

- 
- [9] Peng Duan, Min Jin. The design and implementation of Embedded Linux configuration software graphics interface. *Microcomputer information*. 2008; 24(2-2): 47-49.
  - [10] Xuechen Li. The research and implementation of industrial control configuration software platform, Thesis of master degree. Dalian: Dalian University of Technology. 2005.
  - [11] Yun Shang, Hailong Wang, Minglu Gao. The research of real-time database in industrial control system. *Gansu Science and Technology*. 2006; 22: 70-72.
  - [12] Neil Matthew, Richard Stones. *Linux programming*. Beijing: posts and telecom press, 2007.111-126.
  - [13] Shengjian Chen, Yuan Wang, Shuai Li. The design of real-time database in Embedded Linux configuration software. *Micro computer information*. 2009; 25(6-2): 166-168.