Sensors & Transducers, Vol. 179, Issue 9, September 2014, pp. 55-59

Sensors & Transducers

http://www.sensorsportal.com

The Research about Embedded Soft PLC Running System Based on ARM

*** Fang Ding, Suzhuo Wu**
College of Aeronautical Automation, Civil Aviation University of China, CAUC,
Tianjin, 300300, China
* Tel.: 18322731573
* E-mail: wusuzhuo@163.com

Received: 13 June 2014 /Accepted: 29 August 2014 /Published: 30 September 2014

Abstract: The paper discusses the overall construction and operational principle of soft PLC. Considering the real time request of the system, we use Linux+RTAI dual core system as software platform. In this platform, the implementation method of soft PLC operational system is introduced. Especially, the design of instruction analysis module is emphasized. On the basis of energy flow concept, the logic algorithm is established, aligning to the left bus. Finally, the system is tested to evaluate the desired control behavior. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Soft PLC, Embedded, Linux, Running system.

1. Introduction

Compared with traditional hard PLC, soft PLC technology has more open construction, faster network communication capability and more powerful data processing ability. Thus the application of soft PLC grows quickly in modern automation industry field. With the fast growth of embedded system, embedded soft PLC came into being. The embedded soft PLC can not only realize soft PLC functions on the basis of PC platform, but also has great advantage in real time communication, system clipping, resource distribution and cost control.

Embedded soft PLC system has developing system and running system. The upper developing system runs in PC platform. Windows style interface is adapted, which makes convenience for user to do programming. The running system runs in embedded platform, and has good performance in portability. Among these, the running system is the core of the whole embedded soft PLC system.

2. Operating Principle of Embedded Soft PLC System

Overall structure of embedded soft PLC is shown in Fig. 1 below.

Soft PLC is similar to traditional PLC in working process. Users first develop PLC program in PC environment. This program is compiled to generate the appropriate configuration file, which is downloaded to the embedded soft PLC runtime system after the simulation correctly. When the running system receives the configuration file, interprets it and then run the process in accordance with the traditional PLC.

Like traditional PLC, embedded soft PLC works also taking the manner of scanning cyclically. In a scan cycle, embedded soft PLC first sampled signal of input interface and dump the input image area, then executes the user's program, finally produces corresponding output signal and control an external device via the drive circuit.

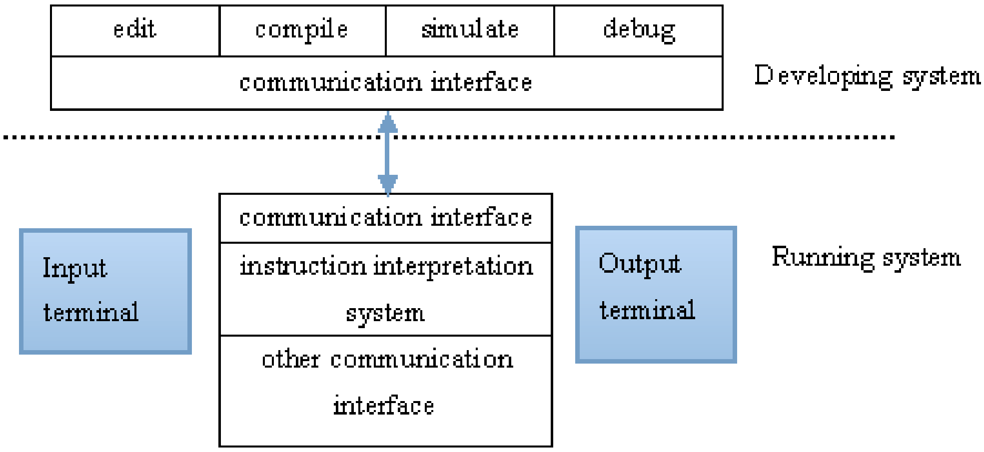

Fig. 1. Overall structure of embedded soft PLC.

3. Hardware Implementation of System

Use the development board FS2410 ARM as the hardware platform. This board uses processor s3c2410 as the core control chip, and it includes the following functional blocks mainly:

Central processor: CPU, s3c2410, frequency, 203 MHz;

External Memory: memory, 64 M byte; Nor Flash, 2 M byte; Nand Flash, 64 M byte;

Serial ports: two five-wire asynchronous serial port;

Network Interface: one 10 M Ethernet port;

USB interface: 2 USB1.1 HOST interface, 1 USB1.1 Device interface;

Storage Interface: one SD interface;

Clock source: internal real-time clock;

Reset circuit: a reset button, and using a dedicated reset chip, stable and reliable;

Debug, download interface: a 20-pin Multi-ICE standard jtag interface supporting ads1.2 and other debugging tools;

Power interface: 12 V power supply, with power switch and indicator light;

Others: 16 small buttons, 4 highlight LED; LCD and touch screen interface; a standard connector is used for the expansion port, eliciting the address lines, data lines, read and write, chip select, interruption, I/O ports, ADC, 5 V and 3.3 V power supply and other signals users may use.

4. Software Implementation of System

4.1. Software Platform

The implementation of the embedded soft PLC running system need the support of real-time operating system, so embedded Linux operating system, which is modified by real-time RTAI kernel, is used in this article. Every function module of the embedded soft PLC running system is built based on the software platform.

4.2. The Implementation of Embedded Soft PLC Running System

The embedded soft PLC running system mainly includes the following function modules: the configuration file loading and parsing module, command parsing module, internal variable accessing module, device driving module and real-time task designing module based on RTAI kernel. Each module is implemented by a separate file.

Command parsing module is also called the logical operation module, which is the core of the embedded soft PLC running system, mainly completes resolution of ladder diagram logic relationship, and refreshes the output of the soft PLC by parsing results, finally realizes the control of external equipment.

In this implementation, a ladder diagram of embedded soft PLC is represented with a data structure:

```
Typedef struct StrRung {
      …
      StrElement
Element[RUNG_WIDTH][RUNG_HEIGHT];
   }StrRung;
```

After the configuration file is loaded and parsed, each variable of the structure is initialized. The two-dimensional structure array Element [][] stores the description information of all components in the ladder diagram. StrElement is an abstract data structure which describes the various properties of soft component. After the configuration file is loaded, these data information will be stored in memory area of the user's program and logic operation module reads the information, then the logical relationship is interpreted.

The embedded soft PLC running system defines corresponding functions for every soft element, which realized the function of the soft components in traditional PLC. If the input signal of the soft components is effective, corresponding functions of the soft components is called to achieve its corresponding function.

A ladder diagram logic relation is interpreted in the order from top to bottom and from left to right. The function of each element and the logical connection relationship between components in the ladder are interpreted in turn. When scanning the ladder diagram, from the first column Element [0] [0] ~ Element[HEIGHT] [0], the type of current soft component is judged one by one, the predefined functions of the components are called depending on the type of corresponding components. Ladder diagram's scanning process is implemented by function RefreshRung(), some of its implementation code is as follows:

```
int RefreshRung (StrRung * Rung)
{ …
for (x=0;x<RUNG_WIDTH;x++)//line: from top to bottom
    {
      for(y=0;y<RUNG_HEIGHT;y++)//column: from left to right
        {
          switch(Rung->Element[x][y].Type) //judge the type of the component
            {
              case ELE_INPUT:   //is input element
CalcTypeInput(x,y,Rung,FALSE,FALSE); //call the function of input element
                break;
              case ELE_CONNECTION: // is connected component
CalcTypeConnection(x,y,Rung);//call the function of connected component
                break;
              case ELE_TIMER: //  is the timer element
                CalcTypeTimer(x,y,Rung); //call the function of timer
                break;
              case ELE_OUTPUT://is output element
CalcTypeOutput(x,y,Rung,FALSE);//call the function of output element
                break;
              …
            }
        }
}
…
}
```

In the scanning process, each element in the ladder diagram need to be scanned, its connection relation is determined and the corresponding input and output state of the element is calculated. The calculation of element's input state is implemented by function StateOnLeft ().

For the convenience of accounting for ladder diagram logic scanning principle, the concept of energy flow is introduced. The logical connection relationship between each element is regarded as a kind of energy flow path. This energy flow starts from the left bus, through the various soft components, and ends at the right bus.

The function of soft component of normally open contacts or normally closed contact is realized by judging the input state and conducting state of the component itself, and then calculating its output. The part of the implementation code is as follows:

```
char CalcTypeInput (int x,int y,StrRung * UpdateRung,char IsNot,char OnlyFronts)
{  …
StateElement= ReadVar(UpdateRung->Element[x][y].VarType,
          UpdateRung->Element[x][y].VarNum);
//read the current soft component's state
    if (IsNot) //  if it is a normally closed contact
      StateElement = !StateElement;  //take the reverse state
     …
    UpdateRung->Element[x][y].DynamicState = StateElement; //get component's state
    if (x==0) { //If connected to the left bus directly-flow circulates or not only related to the component's conducting state
     State = StateElement;
    }
else {//If it is not attached directly to the left bus, need to call function StateOnLeft () to calculate the element's input state
    UpdateRung->Element[x][y].DynamicInput = StateOnLeft(x,y,UpdateRung);
    State = StateElement && UpdateRung->Element[x][y].DynamicInput;
  // energy flows into the current components and it is in a conducting state at the same time, energy flow passes though, or cut-off
   }
    UpdateRung->Element[x][y].DynamicOutput = State;//energy output
    UpdateRung->Element[x][y].DynamicVarBak = StateVar;
   …
    return State;
}
```

There are a multiple logical connections between components and components in LD, such as series connection and parallel connection. To understand the various connections between components, according to the concept of the energy flow in the ladder diagram, a logic operation algorithm which is centered around the left bus is established. This algorithm realizes the interpretation of the logic relations between each element in the ladder diagram in the form of calculating input state of each soft component. The flow chart of this algorithm is shown in Fig. 2.

5. Running Instance

Now an LED control model is used to test the running effect of embedded soft PLC running system. The corresponding relation of LED and KEY in this model and programming components of soft PLC is as shown in Table 1 and Table 2. Experiment: press the normally open switch KEY0, LED0 lights up and LED1 lights up after 4 seconds delay. Press the normally open switch KEY1, LED2 lights up and LED0 put out at the same time.

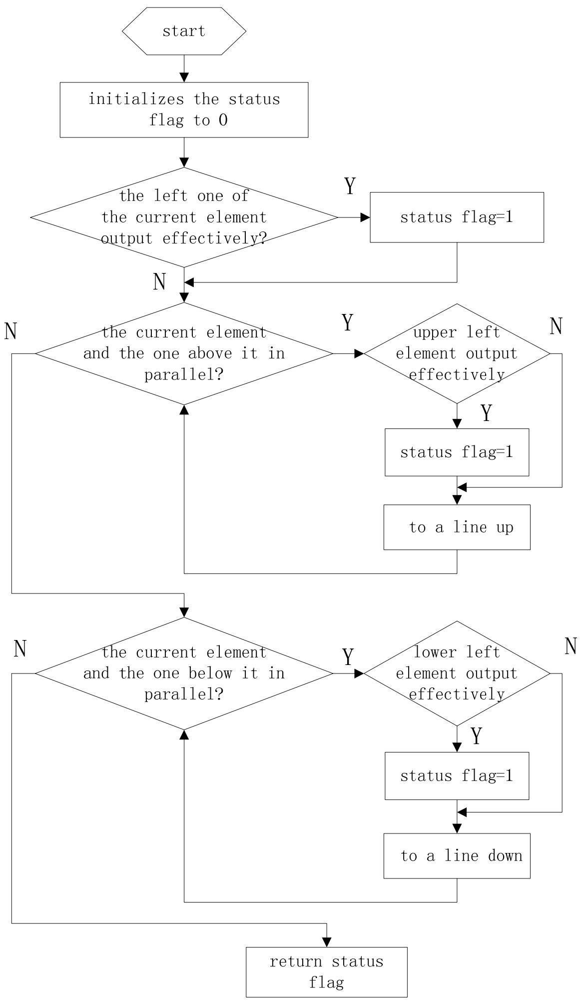

Fig. 2. Flow chart of left-scan algorithm in LD.

Table 1. Key and SoftPLC input elements mapping relations.

KEY0	KEY1	KEY2
I0.0	I0.1	I0.2

Table 2. LED and SoftPLC output elements mapping relations.

LED0	LED1	LED2
Q0.0	Q0.1	Q0.2

First write the corresponding PLC ladder diagram in upper developing system of the embedded soft PLC, and then load the compiled configuration file into the embedded soft PLC running system. The corresponding ladder diagram program is shown in Fig. 3.

Experiment results: press KEY0, LED0 lights up and LED1 lights up after 4 seconds delay. Press KEY1, LED2 lights up and LED0 puts out at the same time. Therefore, the embedded soft PLC running system has reached the expected control effect.

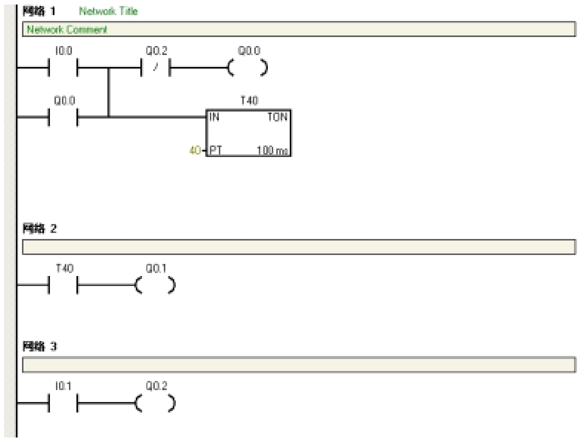

Fig. 3. Ladder diagram of LED control model.

6. Conclusions

In this paper, we realized the function of soft PLC operational system, by simulating the working process of traditional PLC, on the basis of Linux+RTAI dual core system real time environment. Embedded soft PLC technology has great development potential, especially in mid-small control applications. It has big advantage in both control behavior and control cost. The exploitation of soft PLC should have a brilliant future in automation field.

Acknowledgements

This work was financially supported by National High Technology Research and Development Program 863 (2011AA11A102), The Natural Science Foundation of Tianjin (Grant 13JCYBJC39000), The Natural Science Foundation of Civil Aviation University of China (2012QD21x).

References

[1]. Zhibin Ren, The research and implementation of the embedded soft PLC system, PhD Thesis, *Harbin Engineering University*, Harbin, 2005.

[2]. Ben Kao, K. Y. Lam, Brad Adelberg, et al, Updates and view maintenance in soft real-time database systems, in *Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM'99)*, Kansas City, Missouri, November 2-6, 1999, pp. 300-307.

[3]. Yukang Tian, PLC Programming language interpretation research and system implementation, Ph.D. Thesis, *Wuhan University of Technology*, Wuhan, 2004.
[4]. Yisi Liu, Compiler design theory, *Science Press*, 2001.
[5]. Huayun You, The research and development of the soft PLC based on RTlinux, *Computer Engineering and Application*, Vol. 22, 2002, pp. 134-136.
[6]. Chenggyong Chen, Soft PLC technology research and development, Ph. D. Thesis, *Xihua University*, 2008.