doi:10.3772/j.issn.1006-6748.2021.02.012

Research on rapid development platform of PLC control system[①]

Wang Xing(王 兴)[*][**][***], Tang Xianwei[②][***], Dong Zengshou[****], Zhen Liaomo[*]

([*] School of Electronic Information Engineering, Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, P. R. China)

([**] School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, P. R. China)

([***] School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, P. R. China)

([****] School of Electronic Information Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, P. R. China)

Abstract

In the field of industrial process control, a fast-development platform for programmable logic controller (PLC) systems is designed in order to solve two main problems of rapid development of PLC control system and programmability of controlling software. In the aspect of design, the platform is composed of hardware controlling and software monitoring and is taking industrial computer as the core. Under the Windows environment, the platform establishes the control instruction set, develops the configuration function and visual programming function of the monitoring software and it integrates PLC controller based on Visual Basic software. In order to achieve the function of data monitoring, it has realized the serial communication between computer and PLC by using RS-485 and RS-232 serial ports line communication. The platform designs the intelligent instruction scheduling strategy by studying the encoding and decoding rules of the communication instruction set. It proposes a method for rapidly developing control programs by adopting the expert control mode, which enables clients to develop and modify programs conveniently by importing instructions in a non-coded manner. After experimental testing, the platform is proved successful achieving both the rapid development of PLC control system and the rapid modification of monitoring software.

Key words: programmable logic controller (PLC), rapid development, instruction set, instruction scheduling, expert control

0 Introduction

At present, most of the monitoring software that are used in the industrial field can only be applied to a fixed control system and they do not have the function of program refactoring. Once the original process changes, professionals of these software will need to modify the source code, which is very complicated and time-consuming. In some cases, companies that provide monitoring software do not provide updated services because they are beyond expiration time. Based on the above situations, users cannot modify the program based on a changed original process, which brings great inconvenience to the users. Thus on one hand, people can take advantage of this system proposed in this paper by ignoring a shifty fundamental process or a demanded new version of service. On the other hand, the proposal of intelligent manufacturing meets today's requirements for intellectualization. However, traditional control systems have a low level of intellectualization and they are lack of self-learning function. The defects of traditional control systems are causing unnecessary waste of human resources and they would make the control system more rigid. As a result, it is necessary to design a versatile intelligent measuement and controlling platform system. The system could be programmed by users and refactored based on different processes[1], and it has great flexibility and low barriers to use. Furthermore, it could develop within a certain time frame. In this research, the proposed platform is elaborated.

1 The design of the system

1.1 Composition of the system

The system is mainly composed of 6 parts, the up-

[①] Supported by the International S&T Cooperation Program of China (No. 2014DFR70280), Key Research and Development Project of Shanxi Province (No. 201903D321012), Key Research and Development Project of Shanxi Province (No. 201903D121023) and Patent Promotion and Implementation Funding Research Project of Shanxi Province (No. 20210521).

[②] To whom correspondence should be addressed. E-mail: S20180562@stu.tyust.edu.cn

Received on Apr. 17, 2020

per computer configuration software, the lower computer, the cloud database, the personal computer (PC) or mobile device, the measurement and control object and other external communication equipment. The overall architecture of the system is shown in Fig. 1.

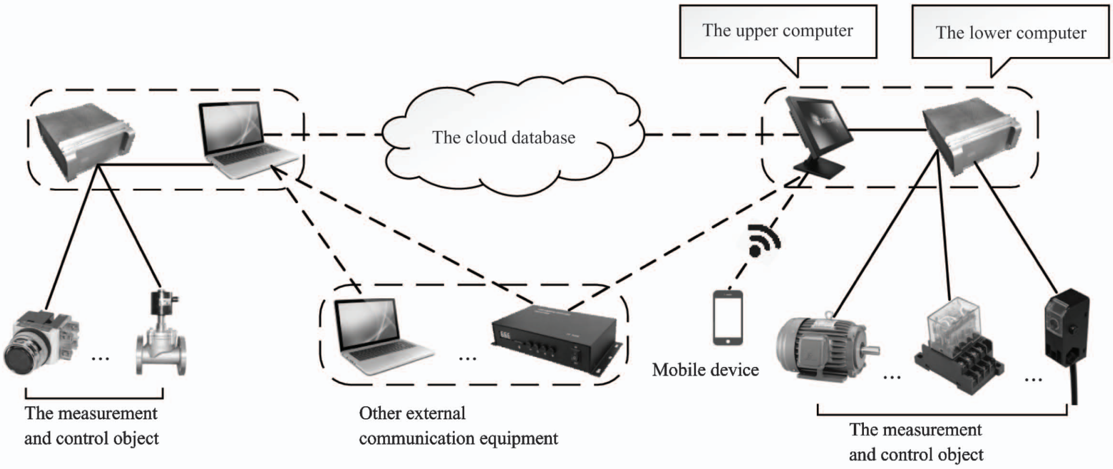

The upper computer
The lower computer
The cloud database
Mobile device
The measurement and control object
Other external communication equipment
The measurement and control object
...

Fig. 1 System architecture

1.2 Working principle of the system

The first step is to formulate an I/O table based on the determination of the composition of the system. Then, design a background picture and import the picture into the configuration software, and add monitoring points based on the I/O table to formulate a complete monitoring picture. Enter programming mode after the monitoring picture is designed. Users need to convert the process into steps and each step corresponds to an instruction. By clicking the corresponding instruction button, each instruction is added to the program running window in the order of the process leading to the programming is completed. The program is saved as . txt format, which can be called in part or in full. In addition, an advanced programming method is also available. According to the established standard of instruction set, the program files can be edited from the PC or mobile devices directly and the programming results can be sent to the upper computer by the Internet. By identifying the instruction statements in the program file the writing of the program is completed.

All measurements and control objects in the control system have access to the input and output interfaces that are provided by the lower computer from the I/O table, and then, connect the upper computer and the lower computer can be connected serial communication. When users operate the control system by clicking the run button, the upper computer will send the program that is written by users to the lower computer under the instructions, at the same time, the lower computer will make the corresponding action and feed the current state back to the upper computer in real time[2]. Finally, the state information will be presented in the monitoring image that is designed by users[3]. The control system could connect to other external communication devices by one-to-many or many-to-many. The upper computer configuration software could control other systems or equipments based on the rules from the instruction sets of both sides or vice versa.

Finally, the program can be uploaded to cloud database and it can be directly searched by other users. Users can directly embed a part or all of the functions to the program in their own control systems by downloading them. The growing quantity of cloud database keeps the system growing[4]. In addition, the expert control adopted by the system would perform operation based on the information provided by the knowledge base for achieving smart control and intelligent processing. The system operation flow is shown in Fig. 2.

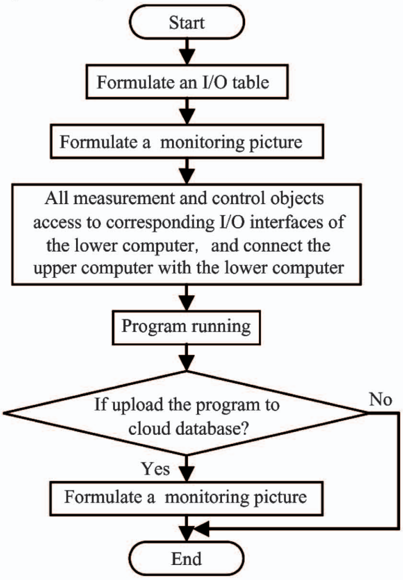

Start
Formulate an I/O table
Formulate a monitoring picture
All measurement and control objects access to corresponding I/O interfaces of the lower computer, and connect the upper computer with the lower computer
Program running
If upload the program to cloud database?
No
Yes
Formulate a monitoring picture
End

Fig. 2 System operation flow chart

2 The design of communication instruction set

2.1 Instruction design

Instructions act as messengers. On the one hand, when users use the upper computer configuration software, a series of operations are converted into instructions and the configuration software makes corresponding actions by identifying different instructions. On the other hand, the interaction between the upper computer and lower computer is also achieved by identifying instructions. Thus, making a set of instruction rules is a premise of the communication between people and machines, machines and machines. A collection of all instructions is called an instruction set. The types of instructions included in this program's instruction set are system operating instructions, alarm instructions, program call instructions, device action instructions, logic judgment instructions, parameter setting instructions and function instructions. The design of instructions needs to take into account the readability and quick recognition[5]. Some of the instruction sets of the program are presented in Table 1.

The following contents describe the structure of all types of instructions, and the first three digits of each instruction are instruction serial numbers.

The functions of the system operation instructions are Run, Pause, Execute Next Instruction and Interrupt Program. Every instruction is composed of instruction serial number and four characters. For example, 003NEXT, it means to execute the next instruction operation.

The alarm instruction is composed of instruction serial number, Wam and the three alarm numbers. For example, 000Wam100, it represents programmble logic controller (PLC) communication failure.

The program call instruction is composed of instruction serial number, Load _ and ten characters. For example, 004Load _ TrcLig, it means to call the local program named TrcLig.

The device action instruction is also a digital output instruction. The structure of the instruction includes instruction serial number and QX. Y-Z in which X. Y represents the address of PLC's digital quantity output and Z represents 0 or 1. For example, 026Q1. 5 = 1 means to set PLC's digital quantity output Q1. 5 to high level.

The parameter setting instruction is composed of the instruction serial number, AOX = and twelve setting values. For example, 004AO5 = 100 means to set the value of analog quantity output 5 to 100.

Table 1 System partial instruction set

Instruction	Explanation of the instructions
000? I0. 0 = 1	Whether I0. 0 is a high level or not
001? I0. 0 = 0	Whether I0. 0 is a low level or not
…	…
000Q0. 0 = 1	Set Q0. 0 as a high level
001Q0. 0 = 0	Set Q0. 0 as a low level
…	…
000? Q0. 0 = 1	Whether Q0. 0 is a high level or not
001? Q0. 0 = 0	Whether Q0. 0 is a low level or not
…	…
000? AI1 >> Value	Whether AI1 is greater than the set value or not
001? AI1 << Value	Whether AI1 is less than the set value or not
…	…
000AO1 = Value	Set AO1 is equal to the set value
001AO2 = Value	Set AO2 is equal to the set value
…	…
000RUN *	Program running
001PAUS	Program halt
…	…
004Load _ Name	Call the program named Name
005Time = Value	Set the time
006Loop = Value	Set the loops number
007EndLP	Loop end
008Goto = Value	Set the liner table of jump
009ElGoto = Value	Line number to jump to when setting condition
010or	The logical or instruction
011//	It is available to add comments after //
000Wam100	Error code 100
…	…

The logic judgment instruction includes the digital quantity input logic judgment, the digital quantity output logic judgment and the analog quantity input logic judgment. The structure of the three instructions are instruction serial number, ? IX. Y = Z and a twelve bit setting values, instruction serial number, ? QX. Y = Z and a twelve bit setting values, and instruction serial number, ? AIXYY and a twelve bit setting values. The structure of the first two listed obove is similar to the action instructions' structure. YY of the analog quantity input logic judgment represents logical symbols that includes == , >> , << , >= , <= as five different categories. The instruction 011? Q0. 5 = = 0 and 015? AI4 > > 200 separately represents a judgment on whether the digital output Q0. 5 is low level and the value of analog quantity input 4 is greater than 200.

The function instruction is composed of the timer instruction, the cycle instruction, the jump instruction, the logical or instruction and comment instruction.

2.2 Dynamic adjustment of instruction sets

The instruction set contains both public and custom parts, and the public part is the same for all users but the custom part is different. The former can not be operated, thus, users can only operate on the latter. The adjustment of instruction set needs to be synchronized with the control systems to achieve the right correspondence[6]. The configuration software will dynamically modify the instruction set in the following cases.

(1) When it comes to complex control systems, the I/O port may not meet the needs of users, therefore the system provides an extended function that can install the input and output module based on users' needs.

(2) Remove excess I/O ports.

(3) The system provides the setting function of custom alarm condition, which allows users to add or remove alarm instructions by themselves.

(4) When the platform system adjusts its functionalities.

(5) When the instruction set used by the program that is downloaded from cloud databases is different from users' own instruction set.

In order to reduce the number of modifications for the instruction set after adding or removing instructions, the instructions that can be adjusted are given a set of independent instruction serial numbers. However, system operation instructions, program call instructions and function instructions can not be changed, leading to a combination of those above with a separate instruction serial numbers. The part with a separate serial numbers is called an instruction block and each particular part is stored in a separate array.

The adjustment of instruction set is divided into two cases including adding or deleting instructions. In the case of adding instructions, according to the types of instruction to find the instruction block to which it belongs, add the instruction at the end of the instruction block, at the same time, it is given an instruction serial number sequentially. As for the deleting of instructions, in the instruction block, all instructions that are located after the instruction which has been deleted need to be moved forward by one position and the instruction serial number needs to be modified to be continuous.

For cloud databases, whenever the functionality of the entire platform system is adjusted, some instructions may be added or deleted, thus, the common part of the instruction set needs to be modified. The result of the modification will be sent to all users by the network to remind them to timely update the instruction set. When users download someone else's program from cloud databases, the corresponding instruction set will be downloaded synchronously. The custom part of the instruction set which is downloaded from the cloud databases will be compared with the existing instruction set, and instructions which are not included in the existing instruction set will be added in. This is the way to run someone else's program correctly.

2.3 Execution of program

The selection and application of data structure have attracted more attention in the execution of a program, because the right data structure can greatly increase the processing speed of instructions. Normally, the control process in the industrial environment is carried out step by step in accordance with the production process, and each step of the production process corresponds to an instruction. This has followed the principle of first-come-first-served in order to avoid the phenomenon of false overflow. A circular queue in the linear table is selected as the data structure. However, the loop execution for partial operation often occurs in the industrial control, and even multiple layers of loops can be nested, causing rapid enqueue and dequeue of the phenomenon in the loop and wasting a lot of time on read and write instructions. Thus, a data structure of nested loop team is proposed to nest and add linear tables on the basis of a loop queue.

While the program is being executed, instructions will sequentially enqueue. When receiving a loop start instruction, all instructions between the end and start loop pair will be stored in a separate linear table. Store the address pointer of the first instruction *Do in the linear table at the position that the previous layer of loop-start instruction has located. Then record the address pointer *Upt of the instruction next to the previous layer loop-start instruction and the number of loops of the current layer Num. For linear tables, after the last instruction is executed, the address pointer can point to the first instruction by subtracting the length of the linear table. At this point, a loop is completed. To subtract Num by 1 every time after the loop is executed, when Num = 0, to call the *Upt back to the previous layer and free the memory of the linear table. The principle of nesting loop team is shown in Fig. 3.

2.4 Instruction query

By instruction transmission, users can operate the upper computer and the lower computer can feed the

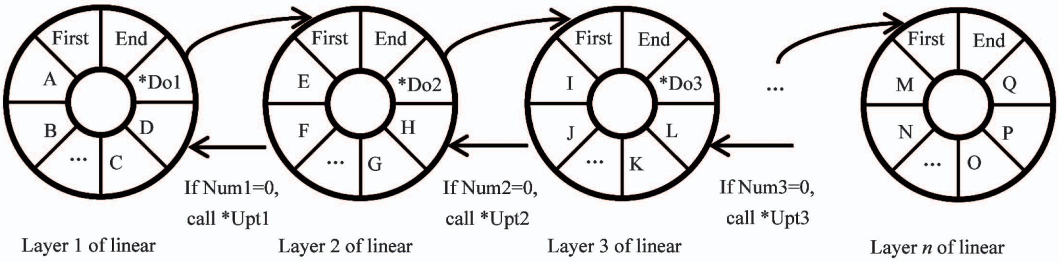


Fig. 3 Nested circular queue

information back to upper computer. Once the upper computer receives the instruction, it will look up the corresponding meaning in the instruction set and perform the corresponding action. For complex systems with many instructions comparing to small systems, the method of instruction query can affect the running speed of the entire system[7]. Based on the design of instructions that is mentioned in this research, the index look-up method is proposed[8]. The main idea of this method is to store different instruction blocks into separate arrays and store the instructions into the array in ascending order, so that the instruction serial number of each instruction is the index in the corresponding array. There are two steps for the execution of instruction query. The first step needs to extract the fourth and fifth bits of the instruction, and find the array to which the instruction belongs according to the index table. The second step is to find the target instruction by confirming that the array index is the instruction number. The index table is shown in Table 2.

Table 2 Index table

Number	The fourth bit of the instruction	The fifth bit of the instruction	Array name
1	Q		Com1
2	?	Q	Com2
3	?	I	Com3
4	?	A	Com4
5	A		Com5
6	W		Com6
7	Others		Com7

Therefore, the average search length (ASL) of the index search method is the sum of the average search lengths of the two parts, which is shown in Eq. (1).

$$ASL = ASL_1 + ASL_2 \tag{1}$$

Each instruction has the same search probability $P_i = 1/n$ for the instruction set $C[1:n]$. The second step is the direct addressing method. The number of comparisons of each element is 1. Assume that the number of logical judgment instructions is L, which means that the sum of the instructions in the arrays Com2, Com3, and Com4 is L. Then the average search length is shown in Eq. (2).

$$\begin{aligned} ASL &= ASL_1 + ASL_2 \\ &= \frac{1}{n}\sum_{i=1}^{n} C_i + \frac{1}{n}\sum_{i=1}^{n} 1 \\ &= \frac{1}{n}[L \times 2 + (n-1) \times 1] + \frac{1}{n}\sum_{i=1}^{n} 1 \\ &= 2 + \frac{L}{n} \end{aligned} \tag{2}$$

The result shows that $0 < L/n < 1$, so the average search length should be $2 < ASL < 3$. The time complexity is $O(1)$. It is not difficult to see that the design of the instruction set, while taking into account readability and scalability, greatly shortens the instruction search time and improves the overall operating efficiency of the system.

2.5 Instruction optimization scheduling

System instructions are divided into internal and external instructions. The instructions stored in the upper computer instruction library are internal instructions, which are used to provide programming with judgments, loops, functions, logic, and other statements. And the external instructions cause an external communication device to operate. Each instruction in the system has a priority, some are built-in to the system, and some need to be defined by users. The external communication equipment in the system is mainly based on the lower computer PLC. It can also be connected to other communication equipment or systems help them control the rest other systems or equipment. It can also be controlled by other systems. Because there are many types of external communication devices, the instruction set of each device is different, and the communication interfaces are not uniform. Such a good instruction scheduling strategy will make the system operate in an orderly manner[9]. The system instruction schedule is shown in Fig. 4.

The external instructions defined in the system are set for the PLC, because it is very difficult to record all the instruction sets of all external communication devices. In order to provide the function of connecting other external communication devices, the system provides an external instruction input function. The user enters control instructions according to the instruction set rules of the external communication device, and then adds them to the program running window. When the upper computer is programmed, the system will run the program linearly, and then execute the next one after receiving feedback from the instruction. When running to different external instructions, the system will automatically open the corresponding communication channel. But when the upper computer receives control instructions from other external communication devices, it will conflict with the running program, and there may be multiple external control instructions at the same time[10]. It needs to be executed in the order of priority. The highest priority will be compared with the priority of the instruction that is running in the upper computer[11]. The higher priority will get the control of the upper computer. Follow this strategy until the program end.

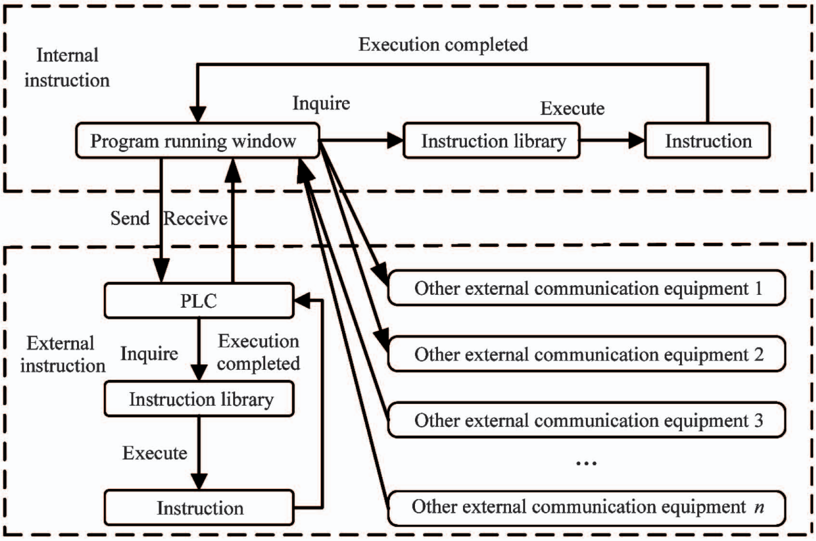


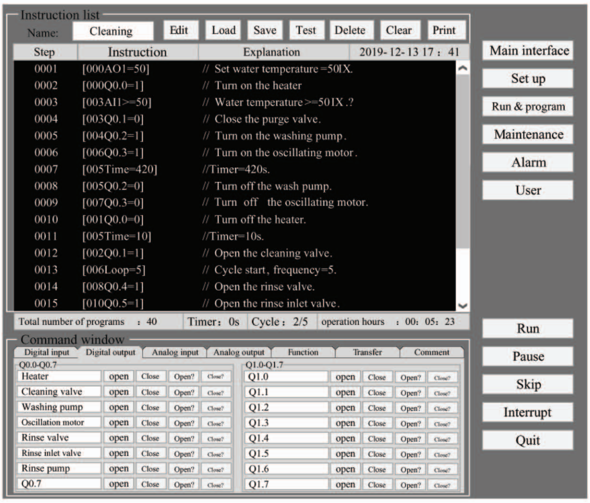
Fig. 4 Instruction scheduling

3 The design of software

In terms of the process control in the industrial environment, although the control flow is diverse, it can be split into the fundamental operation of ' input' and ' output'. Considering the upper computer configuration software of the platform, give sufficient consideration to the feature, programming mode and program modification of the process control in the industrial environment. A new design philosophy is proposed in this research in which user can devise monitoring menu based on different features of control system and select or define operation upon needs within the offered operations of ' input' and ' output'[12]. The software encapsulates each control instruction into a button, so that users can do programming by themselves. The control flow needs to be split into many basic operations referring to a single control instruction. To complete the programming simply and efficiently, users need to click the corresponding instruction button or input the external instruction; then, the instruction needs to be added to the program running window orderly incorporating with some other operations, such as loading, program invocation, adding, deleting and clearing. Thus, it is obviously that the modification of the program can be accomplished simply, which makes the system have the feature of refactoring and can check the syntax[13], logic and format fault of a program[14]. This software set up fault alarm library to store the failures that would happen in the system. Adopting expert control structural, this system can intelligently process failure and provide user with extended function. Thus, users can define warning condition and solution[15]. Besides, it has the function of self-define report printing, curve chart painting and printing. Fig. 5 shows the programming interface of the software testing the water cleaning machine.

4 Expert control method

Expert system (ES) is an intelligent computer program and it records one or more experts' experience and knowledge of a certain field. When it comes to problems, ES can use the stored knowledge to imitate the expert in doing reasoning and judgement, in order



Name: Cleaning

Step	Instruction	Explanation	2019-12-13 17：41
0001	[000AO1=50]	// Set water temperature=50IX.	
0002	[000Q0.0=1]	// Turn on the heater	
0003	[003AI1>=50]	// Water temperature>=50IX.?	
0004	[003Q0.1=0]	// Close the purge valve.	
0005	[004Q0.2=1]	// Turn on the washing pump.	
0006	[006Q0.3=1]	// Turn on the oscillating motor.	
0007	[005Time=420]	//Timer=420s.	
0008	[005Q0.2=0]	// Turn off the wash pump.	
0009	[007Q0.3=0]	// Turn off the oscillating motor.	
0010	[001Q0.0=0]	// Turn off the heater.	
0011	[005Time=10]	//Timer=10s.	
0012	[002Q0.1=1]	// Open the cleaning valve.	
0013	[006Loop=5]	// Cycle start, frequency=5.	
0014	[008Q0.4=1]	// Open the rinse valve.	
0015	[010Q0.5=1]	// Open the rinse inlet valve.	

Total number of programs ：40 | Timer：0s | Cycle：2/5 | operation hours ：00：05：23

Q0.0-Q0.7					Q1.0-Q1.7				
Heater	open	Close	Open?	Close?	Q1.0	open	Close	Open?	Close?
Cleaning valve	open	Close	Open?	Close?	Q1.1	open	Close	Open?	Close?
Washing pump	open	Close	Open?	Close?	Q1.2	open	Close	Open?	Close?
Oscillation motor	open	Close	Open?	Close?	Q1.3	open	Close	Open?	Close?
Rinse valve	open	Close	Open?	Close?	Q1.4	open	Close	Open?	Close?
Rinse inlet valve	open	Close	Open?	Close?	Q1.5	open	Close	Open?	Close?
Rinse pump	open	Close	Open?	Close?	Q1.6	open	Close	Open?	Close?
Q0.7	open	Close	Open?	Close?	Q1.7	open	Close	Open?	Close?

Fig. 5 Programming interface

to solve those complex problems[16]. The expert control in this system is used for fault detection and alarming, syntax checking, instruction scheduling, instruction searching and instruction dynamic adjustment.

4.1 The design of knowledge base

First, the syntax checking rule, instruction scheduling strategy, instruction lookup method, instruction set dynamic adjustment strategy, fault information and solution are put into knowledge base. Then, the custom interface is provided and it is used for the opening of the knowledge base to users. The system can transform the user-defined fault alarm messages into the ES's specialized form and store them into knowledge base. The problems that user's device cannot solve will be uploaded to cloud databases, and they will be handled by professionals. Then the professionals would update the knowledge base. For all users, as long as the devices access the Internet, the software will automatically upload the content of the knowledge base to cloud shared databases. Then the cloud shared database will integrate the collected knowledge base, and it will delete the meaningless rules by using the screening and discriminating mechanism. Combining the upgrading of the platform system which includes many modifications and expansions of every rule, strategy and method, a new knowledge base will be constructed, at the same time, it will be sent to all users. The system is improved by the expanded knowledge base which has been achieved by constant interactions between cloud platforms and different users.

4.2 The design of inference machine

The inference machine is the main part of the expert system. For this system, a problem can be caused by a variety of reasons and then be assigned a priority level to every situation. The value of every reason is initialized as 0, the case holding the higher value of priority will be judged at first, and increase the frequency of this situation by 1 after a successful diagnosis. After a round of judgement, the priority of every reason will be adjusted according to the value of the frequency. The flowchart of the judgement and reasoning is shown in Fig. 6.

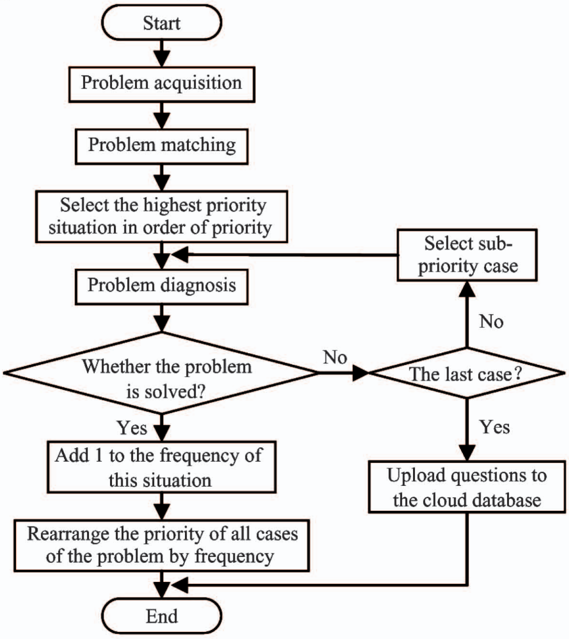

Start
Problem acquisition
Problem matching
Select the highest priority situation in order of priority
Problem diagnosis
Whether the problem is solved?
No
The last case?
No
Select sub-priority case
Yes
Yes
Add 1 to the frequency of this situation
Upload questions to the cloud database
Rearrange the priority of all cases of the problem by frequency
End

Fig. 6 Diagnostic flowchart

4.3 The constitute of rules

The expert system utilizes production rules which are constructed by if-else for the representing of knowledge, and the general representation of the rules is as follows.

(1) If(term 1)Then (action or conclusion 1)

(2) If(term 1)And (term 2)…(term n) Then (action or conclusion 1)
Else(action or conclusion 2)

(3) If(term 1)Then (action or conclusion 1)
ElseIf(term 2)Then(action or conclusion 2)
…

(4) If(term 1)And (term 2)…(term n) Then (action or conclusion 1)
ElseIf(term 1)And (term 2)…(term n)
Then(action or conclusion 2)

5 Conclusions

In view of the monitoring software in the present industrial environment, it can only be used in a fixed control system and it does not have the function of program refactoring. This research has built the instruction sets and turns the instruction sets into buttons to make users program and amend the codes by themselves[17]. According to a number of tests on different control systems, the platform system has been proved achieving strong versatility and better readability, and it has simple operation logic and can complete programming really fast. Also, it works well in fast processing failures and detecting the validity of programs. In addition, users can alter the program by themselves once again with the change of technologies. All in all, this system works on programming not just for professionals, but also for the improvement of industrial process control, because it provides new ideas for the development and modification efficiency.

References

[1] Al Dallal J, Abdin A. Empirical evaluation of the impact of object-oriented code refactoring on quality attributes: a systematic literature review[J]. *IEEE Transactions on Software Engineering*, 2018, 44(1): 44-69

[2] Zhu X Q, Zhu W H, Zhong C P, et al. Design of multi-functional microcontroller and programmable logic device experimental box[J]. *Experimental Technology and Management*, 2019, 36(9): 77-81

[3] Luo J L, Shao H, Wu W M, et al. Synthesis of logical controllers and discrete-event systems supervisory control theory[J]. *Control Theory and Applications*, 2018, 35(1): 86-91

[4] Tan D P, Li L, Zhu Y L, et al. An embe-ddedcloud database service method for di-stributedindustry monitoring[J]. *IEEE Transactions on Industrial Informatics*, 2017, 14(7):2881-2893

[5] Zhu D X. UNI-SPEC: an instruction set description language[J]. *High Technology Letters*, 2003, 9(4): 33-38

[6] Beak B, Zamanipour M, Head K L, et al. Peer-to-peer priority signal control strategy in a connected vehicle environment[J]. *Transportation Research Record*, 2018, 2672(18): 15-26

[7] Tran K A, Carlson T E, Koukos K, et al. Static instruction scheduling for high performance on limited hardware[J]. *IEEE Transactions on Computers*, 2018, 67(4): 513-527

[8] Ren J J. Digital predistorter with improving index accuracy of lookup table based on FPGA[J]. *Integration the VlSI Journal*, 2020, 71: 70-75

[9] Nguyen V A, Hardy D, Puaut I. Cache-conscious off-line real-time scheduling for multi-core platforms: algorithms and implementation[J]. *Real-Time Systems*, 2019, 55(4):810-849

[10] Feng Y J, Ou Y, Ye X C, et al. Study of network loading based instruction scheduling mechanism in dataflow architecture[J]. *Chinese High Technology Letters*, 2018, 28(2): 885-898

[11] He Q, Head K L, Ding J, et al. Multi-modal traffic signal control with priority, signal actuation and coordination[J]. *Transportation Research Part Cemerging Technologies*, 2014, 46: 65-82

[12] Al Dallal J. Identifying refactoring opportunities in object-oriented code: a systematic literature review[J]. *Information and Software Technology*, 2015, 58:231-249

[13] Alkhalid A, Alshayeb M, Mahmoud A. Software refactoring at the package level using clustering techniques[J]. *IET Software*, 2011, 5(3): 274-286

[14] Maldonado E D, Shihab E, Tsantalis N. Using natural language processing to automatically detect self-admitted technical debt[J]. *IEEE Transactions on Software Engineering*, 2017, 43(11): 1044-1062

[15] Koo S, Song Y, Lim S H, et al. Development of a remote supervisory control and data acquisition system for offshore waste final disposal facility[J]. *Journal of Coastal Research*, 2019, 90: 205-213

[16] Zhang W Y, Huang D X, Wang Y D, et al. Adaptive state feedback predictive control and expert control for a delayed coking furnace[J]. *Chinese Journal of Chemical Engineering*, 2008, 4:100-108

[17] Bavota G, De Lucia A, Oliveto R. Identifying extract class refactoring opportunities using structural and semantic cohesion measures[J]. *Journal of Systems and Software*, 2011, 84(3): 397-414

Wang Xing, born in 1981. He is currently an associate professor. He obtained his B. S. and M. S. degrees in electrical engineering from Taiyuan University of Science and Technology in 2004 and 2007 respectively. He is currently pursuing a Ph. D degree in computer application technology at University of Chinese Academy of Sciences. His current research interests include computer application technology and intelligent control.