Realtime and Embedded System Testing for Biomedical Applications

Jinhe Wang, Jing Zhao, Bing Yan
School of Computer Engineering, Qingdao Technological University, Qingdao, 266033, China
jinhewang@qtech.edu.cn

***Abstract*—With realtime and embedded systems we propose a software testing approach to build a testing architecture for biomedical applications, it can check the reliability based on the failure data observed during software testing and can be applied to make the use of test task more flexible. The reliability of data of the system is computed through the test panel and simulation of the testing system by testing the reliabilities of the individual modules in the embedded system. One of the fundamental requirements for embedded system is the ability to obtain the testing data from command line of the program for process control in the testing system. Hence, the testing instructions can be formally described as a sequential decision commands in terms of their temperature, action time, incremental quantity and gradient. The testing approach has been applied in the system and achieved the testing requirements of the embedded system.**

***Index Terms*—Software testing, reliability, embedded system**

I. INTRODUCTION

Embedded System have been developed in high speed, and used for many electronic devices in the field of routine diagnostics of biomedical applications. It performs a variety of functions for the users, and the complexity of embedded system is increased dramatically. Because of mass-production and fast time-to-market, the quality of product and overall cost has become important factors. The multi-level testing technique allows using partially defective logic and routing resources for normal operation, it spans over several engineering disciplines: mechanical engineering, electrical engineering and control engineering. There are many types and design methods for software testing in the Biomedical Embedded system, especially for temperature control testing [1-12]. The application of this method is focused on polymerase chain reaction embedded system, which has cycling reactions such as Denaturation, Annealing and extension [13-15]. Several different models have been designed and applied for testing in the field of basic research. The following sections will provide a description for the software testing on the realtime& embedded system. In Section II, we describe the architecture of the system testing. Finally, a brief conclusion is given in Section III.

II. THE ARCHITECTURE OF THE SYSTEM TESTING

The embedded system is for a gene amplification system consisting of data sampling, heat execution, man-machine interface and peripheral interface, and control module. Figure 1 shows the overall diagram of the embedded system.

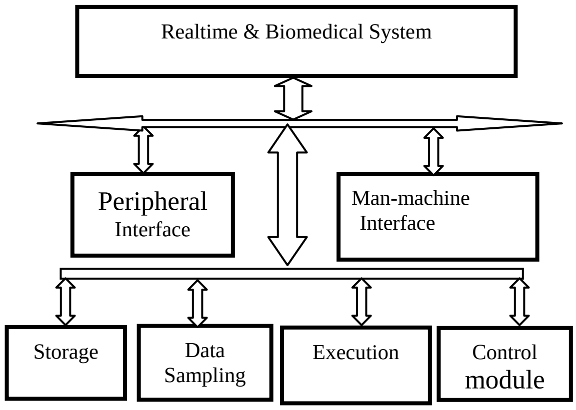

Figure 1. The overall diagram of the system.

Testing system includes several modules for the biomedical embedded system, such as digital signal test man-machine interface test, analog-to-digital module test, control unit test and heat execution module test.

A. Man-machine Interface Test

Man-machine interface is a human-machine interaction platform, which receive commands from users and translate them into operating steps, and then transfer the steps into execution module, whereas, Sensor information, parameters of output or other output informaiton will be sent to human-machine interface, and then as results, transfer them to the user via LCD display.

With the Human-machine interface module, users edit a program (existing program or new one) with program editor that developed, the program can be loaded into the processing level via user command. This is applicable for programs from the internal device memory. After programming, the program data can be saved and remains stored even when the apparatus is switched off. Only programs can be started for execution. The program that displayed in the processing level can be started directly by user command via human-machine interface, and at the same time, the human-machine interface will translate the program into control instructions for execution line by line.

Here we will introduce the functions of keys briefly, the key board consists of 24 keys in the embedded

doi:10.4304/jsw.5.10.1060-1067

system, they are programming keys, cursor/confirm keys, control keys, file key, setting key, reverse key and 10 digital keys.

The programming key consists of Ins Key, Del Key and Exit Key; cursor/confirm key consists of Left key, Right key, Up key, Down key and Confirm key; Control key includes Start key, Stop key and Pause key.

Let us take the programming key as an example: During the creation of a program, program lines can be inserted by pressing Ins Key, or they would be deleted by pressing Del Key.

In order to test the program executed in the Man-machine interface module, a test system is designed, the panel of the system is show as Figure 2, There are 40 lights on the panel, the first light-line includes 20 lights, which named LED1, LED2, LED3, …, LED20 respectively, the second light-line includes 10 lights, the names of these lights are LED21, LED22, LED23, …, LED30, and the third includes LED31, LED32, LED33, …, LED40.

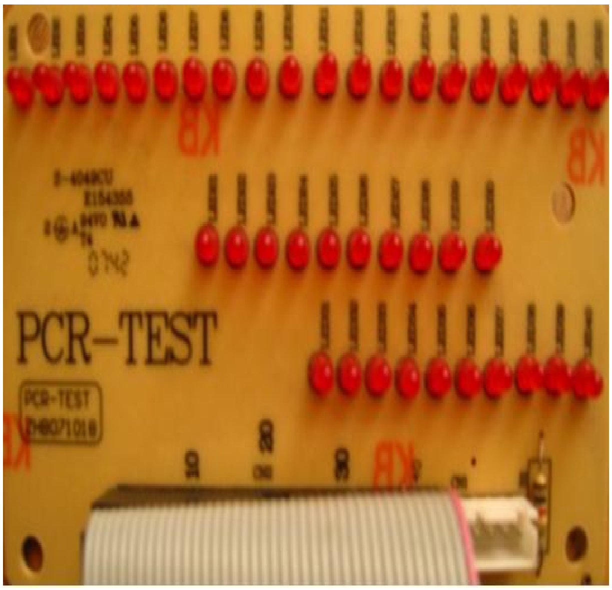

Figure 2. The test panel of the system

There three software testing tasks for Man-machine interface module.

1 Key-board Test

Keys on the key-board are pointed to their very LEDs on the test panel , the relationship between keys and LEDs is shown in Table I.

The steps for testing task are as following:

Step 1. connect the 40-pin cable to the hardware of the embedded system and the test panel

Step 2. connect the 4-pin cable to the hardware of the embedded system and the test panel

Step 3. start testing the program of man-machine interface

Step 4. For each testing key, it corresponds to one of the LEDs shown in Table I, press the key, if the LED fails to light, the key, functions with the key, or the program codes edited for the key should be checked properly.

Step 5. End.

TABLE I.
THE RELATIONSHIP BETWEEN KEYS AND LEDS

Type	Relationship	
	Description	LED
Programming keys	Ins Key	LED1
	Del Key	LED2
	Exit Key	LED3
Cursor/confirm key	Left key	LED4
	Right key	LED5
	Up key	LED6
	Down key	LED7
	Confirm key	LED8
Control key	Start key	LED9
	Stop key	LED10
	Pause key	LED11
Digital Key	0-key	LED12
	1-key	LED13
	2-key	LED14
	3-key	LED15
	4-key	LED16
	5-key	LED17
	6-key	LED18
	7-key	LED19
	8-key	LED20
	9-key	LED21
Other key	File key	LED22
	Setting key	LED23
	Reverse key	LED24

Special test for keys:

In order to avoid the twice times or more between a given interval when a key is pressed, the testing strategy let the corresponding LED to be triggered many times when the phenomenon occurs.

2 Command-line Test

A program edited for the embedded system contains two basic commands for controlling the temperature of the thermal block and of the heated lid as well as seven different command lines for programming. A program can contain up to 200 program lines. The command-lines (CML) may be repeated as desired. Digits or letters may be pressed via key board. Selections should be made for all the command lines in programming. The command lines are divided into seven categories, such as Temperature Command Lines (TCL) , Hold Command Lines (HCL), Repeat Command lines (RCL), Goto Command Lines (GCL), Pause Command Lines (PCL) , Voice Command Lines (VCL) , Link Command Lines (LCL) and End Command Lines (ECL). All the digits inputted by the user should be within the given specifications. Figure 3 shows a program, for example, it contains 8 command-lines, they are TCL(Index 1), HCL(Index 2) , TCL(Index 3), TCL(Index 4), GCL(Index 5), RCL(Index 5), HCL(Index 6), HCL(Index 7) and ECL(Index 8)

Edit User Name:A001 Program Name: B09001

Edit Menu
Control Mode: Block
Lid Temperature: 100
Wait Fix
Body of Program

Index	T	Time	δT	δTime	S	δS	G
1	92.0°C	0:12:08	+0.0	+0.00	0.0	+0.0	0.0
2	84.0°C	0:01:00					
3	44.0°C	0:01:00	+0.0	+0.00	0.0	+0.0	0.0
4	70.0°C	0:01:00	+0.0	+0.00	0.0	+0.0	0.0
5	Goto 02	Rep 10					
6	70.0°C	0:01:00					
7	4°C	1:00:00					
8	End						

Figure 3. An example of program editing.

Each command line is pointed to its corresponding LED on the first line on the test panel, the Block-mode is pointed to LED17, the Wait-mode is pointed to LED18, the Fix-mode is pointed to LED19, and the Gradient-mode is pointed to LED20. In the biomedical system, Temperature is the data which is to be controlled, permitted values is 4 to 99.0, see Figure3; Temperature increment(δT) means that with the times of circles, the temperature is increased or decreased by the value for each cycle; Time is the cycle/running time of a step; speed(S),to describe the ramp speed in one circle; with the times of circles, the speed is raised or lowered by speed increment(δS) in one circle; Gradient(G),the value with gradient mode which lowest temperature is on one side and the highest one on the other, permitted maximum value is 20; the index number of the code. The relationship between command lines, Mode and LEDs is shown in Table II.

1) TCL/HCL Test

The first two digits for temperature in TCL/HCL is pointed to their LEDs on the second line and third line on the test panel , the left digit is for LED on the second line, and the second digit from left is for LED on the third line; the relationship between temperatures and LEDs is shown in Table III.

TABLE II.
THE RELATIONSHIP BETWEEN COMMAND LINE AND LEDS

Type	Relationship	
	Description	***LED***
Command Line	TCL	LED1
	HCL	LED2
	GCL	LED3
	PCL	LED4
	VCL	LED5
	LCL	LED6
	ECL	LED7
	RCL	LED8
Mode	Block	LED11
	Wait	LED12
	Fix	LED13
	Gradient	LED14

The five digits for time in TCL/HCL is pointed to their corresponding LEDs on the first line on the test panel , from left, the first digit is for hours, it is pointed to LED16; the second and third digits are for minutes, they are pointed to LED17 and LED18; the last two digits are for seconds, they are pointed to LED19 and LED20. The relationship between time and LEDs is shown in Table IV.

After Step 4 above, the program code for control mode is executed, the black number means the ID of LEDs, and the red number means digit that the LED is to be indicated. If the LED11,LED12 and LED13 turn on, it succeeds in Step 4, if the result fails, the program code about the control mode and related parameters should be checked carefully, in this case, the testing software will show the message about the area that potential original codes take in the program of the embedded system.

Take an example shown in Figure 3, the steps for testing task for TCL/HCL are as following:

Step 1. connect the 40-pin cable to the hardware of the embedded system and the test panel
Step 2. connect the 4-pin cable to the hardware of the embedded system and the test panel
Step 3. start testing the program of TCL/HCL
Step 4. load the program code from storage of the system, the program is shown in Figure3, as an example, the control mode is Block, wait and fix, so the LED11,LED12 and LED13 turn on (see Figure 6).
Step 5. execute code line by line, until it is not TCL/HCL
Step 6. end.

For Step 5 above, first command line TCL is to be executed, there are three procedures to execute the TCL.

First of all, the LED1 turns on according to Table II, as shown in Figure 5.

TABLE III.
THE RELATIONSHIP BETWEEN TEMPERATURES AND LEDS

Type	Relationship	
	Description	***LED***
Left digit of first two digits of temperature on TCL/HCL	1	LED21
	2	LED22
	3	LED23
	4	LED24
	5	LED25
	6	LED26
	7	LED27
	8	LED28
	9	LED29
	0	LED30
Second digit of first two digits of temperature on TCL/HCL	1	LED31
	2	LED32
	3	LED33
	4	LED34
	5	LED35
	6	LED36
	7	LED37
	8	LED38
	9	LED39
	0	LED40

TABLE IV.
THE RELATIONSHIP BETWEEN TIME AND LEDS

Type	Relationship	
	Description	***LED***
The position of Digits of time on TCL/HCL	1	LED16
	2	LED17
	3	LED18
	4	LED19
	5	LED20

If LED1 turns on, it succeeds in the first procedure in Step 5, if the result fails, the program codes about the order of the execution code perhaps are not normal, the codes for loading program, their execution and related parameters should be checked carefully, at the moment, the testing software shows the message about the area that potential original codes take in the program of the embedded system.

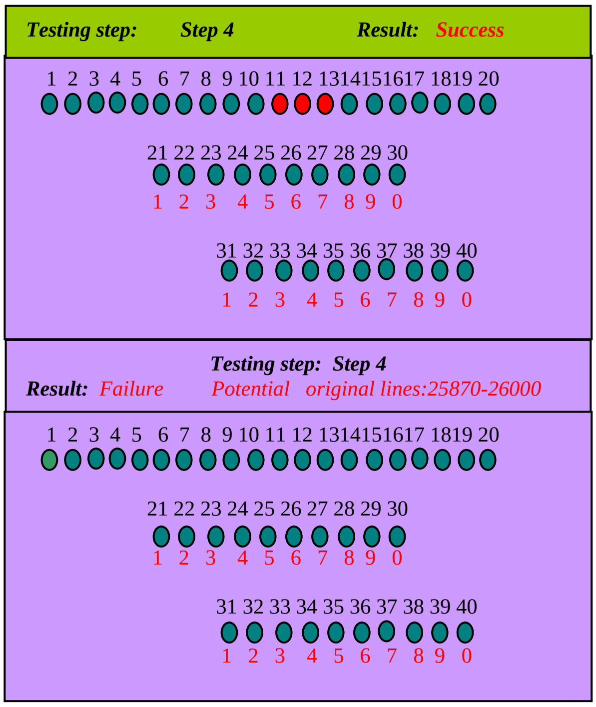

Figure 4. Diagram of the testing step 4 of TCL/HCL testing

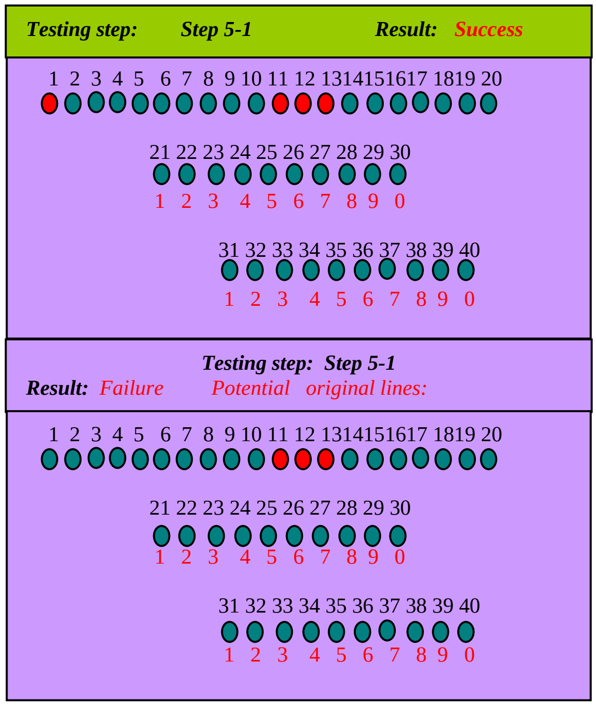

Figure 5. Diagram of the first procedure the testing step 5 of TCL testing

In the second procedure in Step 5, load temperature 92°C, LED29 and LED32 turn on and flash if the temperature on the heating block do not reach to 92°C, otherwise, if the temperature on the heating block catches 92°C, LED29 and LED32 light and do not flash, as shown in Figure 6.

It succeeds in the second procedure in Step 5 (above in Figure 6), if the result fails, the LED29 turns off or LED32 off, the program codes about the loading of the temperature are not normal, as shown in Figure 6, the codes for loading program, the codes for execution and

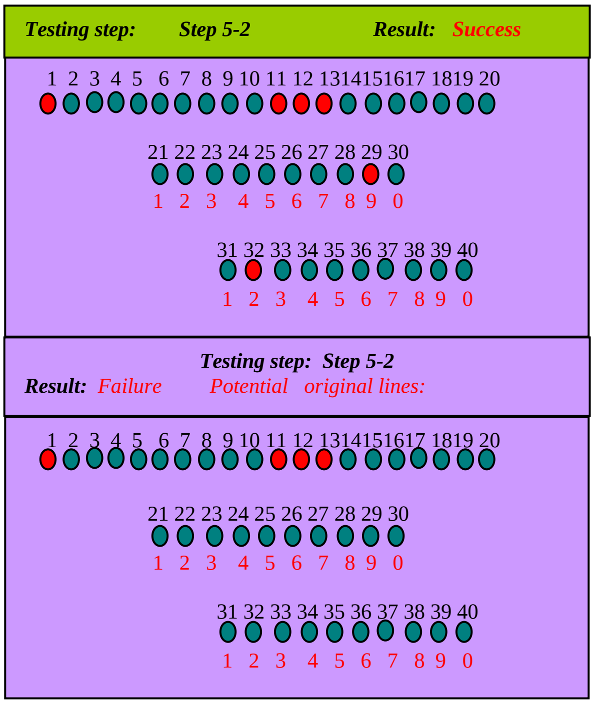

Figure 6. Diagram of the second procedure in the testing step 5 of TCL testing

related parameters should be checked carefully, at the moment, the testing software will show the message about the area that potential original codes take in the program of the embedded system.

In the third procedure in Step 5, load time, 0:12:08, LED17, LED18 and LED20 turn on and flash, the data of the time decreases with the running of the procedure, LED17, LED18 and LED20 indicate the position of the digit of the time (hour, minute, or second). If not all of the LED17, LED18 and LED20 turn on at the first stage of the third procedure in Step 5 ,as shown in Figure 7, the result fails, the program codes about the load of the data code are not correct, the codes for loading program, their execution and related parameters should be checked carefully.

The second command line is HCL in Step 5, it is similar to the first command line, it is to be executed by three procedures, too.

Figure 8 shows the third procedure to execute the HCL in Step 5, No error occurs in this procedure.

2) GCL/RCL Test

The steps for testing task for GCL/RCL are as following:

Step 1. start testing the program of GCL/RCL

Step 2. load the program code from storage of the system

Step 3. execute GCL/RCL code

Step 4. end.

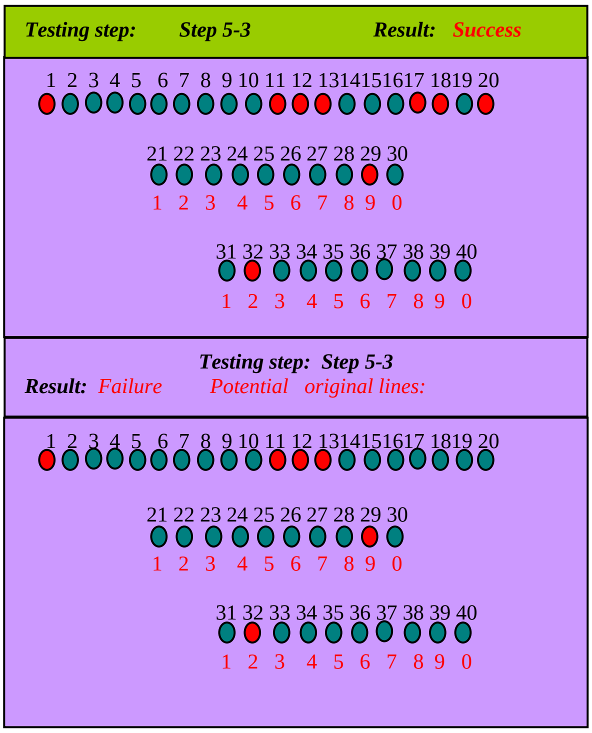

Figure 7. Diagram of the third procedure in the testing step 5 of TCL testing

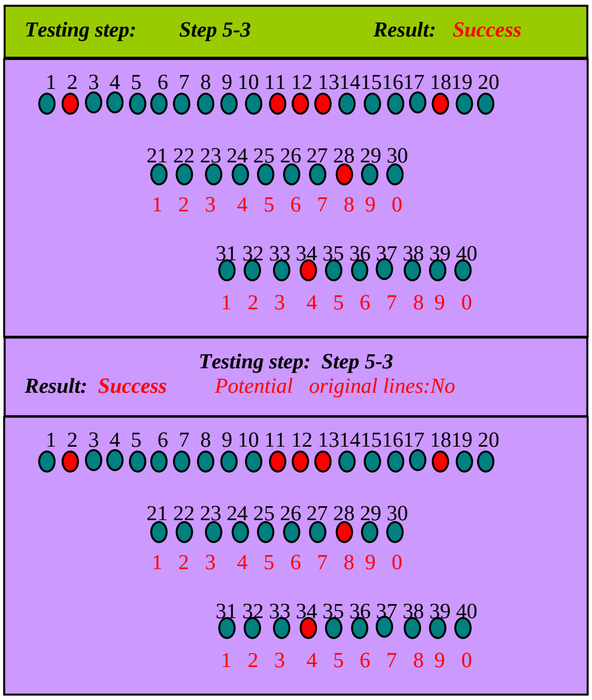

Figure 8. Diagram of the third procedure to execute HCL in the testing step 5

For Step 3, it is similar to the TCL/HCL testing, it is to be executed by two procedures.

Figure 9 shows the second procedure to execute the GCL in Step 3, No error occurs in this procedure.

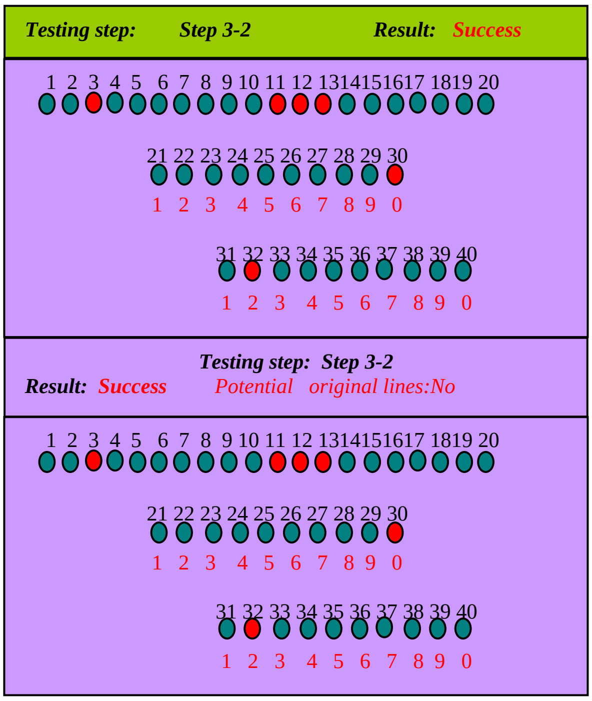

Figure 9. Diagram of the second procedure to execute GCL in the testing step 3

3 Original Code Test

The program code is edited by user through Man-machine interface, and then it saved to flash storage in the embedded system in order to load and to execute it sometimes. When a program is loaded for running from flash memory or USB via human-machine interface, the program code should be checked before the execution of the program. All lines and other commands in the program code will be translated into testing system, in this case, the functions/meanings of LEDs on the test panel will be redefined, if error occurs, the testing system will send message to point out the location of the mistake code in the program. Only the checked program is able to run.

4 User command Test

The embedded system tries to achieve an application goal in an efficient way; it should be executed through man-machine interface by user. The user sends several commands from interface by handling the keyboard in the embedded system. For example, Start command, Load command, Run command, Stop command, Pause command, Print command. Load command is for calling up the existing programs. The selected program is loaded for execution or other purpose. Run command can cause the loaded program running. Pause command is for the interruption of the program. If you want to abort a running program, the Stop command should be sent. All the commands mentioned above should be translated by man-machine interface, and should be transferred to Control Unit through human-machine interface for executing/display.

TABLE V.
THE VOLTAGE OBTAINED BY THE TEST SYSTEM

Date	Channel-1	Channel-2	Channel-3
06-07-2009 13:30:22	7.640055	8.1245757	9.3034378
06-07-2009 13:30:32	7.1392351	8.4115767	9.3197592
06-07-2009 13:30:42	7.4844733	8.0956444	9.2550023
06-07-2009 13:30:52	7.2269653	8.3901012	9.3823906
06-07-2009 13:31:02	7.3999809	8.1200163	9.2049369
06-07-2009 13:31:12	7.0618131	7.9101841	9.1928115
06-07-2009 13:31:22	7.3483185	8.4501555	9.2712796
06-07-2009 13:31:32	7.5631287	7.8135554	8.9273633
06-07-2009 13:31:42	7.0132893	7.9537298	9.2428879
06-07-2009 13:31:52	6.7928735	7.8749313	9.4858366
06-07-2009 13:32:02	7.4457733	8.4258937	9.243813
06-07-2009 13:32:12	7.1511072	8.3054987	8.8745556
06-07-2009 13:32:22	7.1579463	7.9696879	9.4832926
06-07-2009 13:32:32	7.480883	7.6636893	8.9514159
06-07-2009 13:32:42	7.0767138	7.6740746	9.1335281
06-07-2009 13:32:52	6.9660762	8.3725353	9.0386064
06-07-2009 13:33:02	7.5585142	7.5289881	9.1696069
06-07-2009 13:33:12	7.1296316	7.9263733	9.120015
06-07-2009 13:33:22	2.9115743	8.2689463	10.177556
06-07-2009 13:33:32	5.9876958	8.2285504	9.4551762
06-07-2009 13:33:42	5.8926307	8.7853826	9.2717752
06-07-2009 13:33:52	5.7151989	8.831109	9.6085113
06-07-2009 13:34:02	5.6787786	8.0032227	9.0578793
06-07-2009 13:34:12	5.7350885	8.2026586	9.142735
06-07-2009 13:34:22	5.6130414	8.2888249	9.5062439
06-07-2009 13:34:32	5.8781045	8.0117578	9.0460402
06-07-2009 13:34:42	5.9179828	8.4251007	9.0143115
06-07-2009 13:34:52	5.5670949	8.2356098	9.2010823
06-07-2009 13:35:02	5.566368	7.8208681	9.365882
06-07-2009 13:35:12	6.0495893	8.321699	9.2524473
06-07-2009 13:35:22	5.2272527	8.5359365	8.9409975
06-07-2009 13:35:32	5.885307	8.2993204	8.995314
06-07-2009 13:35:42	5.6490211	8.0709751	9.4890525
06-07-2009 13:35:52	5.3964469	8.6015084	8.652598

The task of testing procedure is to find out the mistakes or other potential errors before executing such commands, for each command in the system, there are special definitions with test panel, and it performs such interactions with the user efficiently.

B. Data sampling module Test

Data sampling module is very important in this system, Signals from this module are amplified, filtered, and the analogy signals should be converted into digital signals before imported into the Control Unit.

In this paper, the test task of the data sampling module includes the multi-channels data testing, and filtration test. The test panel should be redefined when applied in this module, the first 20 LEDs (from LED1 to LED 20) is for signals imported through 20-pin cable from the module, and other LEDs indicate the states of the module.In this data sampling system, interleaving multiple analog-to-digital converters is performed with the intent to increase a converters effective sample rate, however, time-interleaving data converters is not an easy task, because the offset mismatches and timing errors can cause undesired spurs in the spectrum of the system. Therefore, you wouldn't gain a good performance for the temperature control without excellent data sampling system. The test system can automatically set the sampling parameter such as sampling channel and sampling frequency, and the results show that the test system has good performance in the testing procedure.

C. Other module Test

After a program is loaded for running from flash memory, the test task is to check the user commands, the testing system asks the corresponding question first, and then activates serial instructions according to the user's answer, If the answer is for execution, the next test task is to check the program mentioned above. If no error occurs in the testing procedure, then the test program activates process control module according to the user's answer and provides the related parameters/states of the process to the human-machine interface. When the program begin some parameters should be got by testing system, for example, if the command that the operator sent is Run command, the biomedical system will execute a user program. It chooses the current action from its set of possible actions according to the program conditions, the parameters for Waiting, Block, Gradient will be got by the system. When the next CML executes, the parameters such as action time, the ramp rate, and the possible remaining time will be transferred from the embedded system and then displayed on the test panel of the test system and on the LCD of the embedded system. After the program stop, the test data of the all of the running parameters will be obtained by the testing system. Table V and Table VI show the data of the output voltage and the data of the temperature of the three channels in the biomedical system respectively.

TABLE VI.
THE TEMPERATURE OBTAINED BY THE TEST SYSTEM

Date	Channel-1	Channel-2	Channel-3
06-07-2009 13:30:22	23.967	23.916	23.967
06-07-2009 13:30:32	23.888	23.862	23.888
06-07-2009 13:30:42	23.91	23.775	23.91
06-07-2009 13:30:52	23.782	23.709	23.782
06-07-2009 13:31:02	23.885	23.678	23.885
06-07-2009 13:31:12	23.672	23.604	23.672
06-07-2009 13:31:22	23.664	23.563	23.664
06-07-2009 13:31:32	23.571	23.533	23.571
06-07-2009 13:31:42	23.56	23.52	23.56
06-07-2009 13:31:52	23.492	23.457	23.492
06-07-2009 13:32:02	23.392	23.364	23.392
06-07-2009 13:32:12	23.398	23.375	23.398
06-07-2009 13:32:22	23.324	23.376	23.324
06-07-2009 13:32:32	23.261	23.272	23.261
06-07-2009 13:32:42	23.269	23.255	23.269
06-07-2009 13:32:52	23.231	23.208	23.231
06-07-2009 13:33:02	23.212	23.166	23.212
06-07-2009 13:33:12	23.168	23.13	23.168
06-07-2009 13:33:22	23.147	23.119	23.147
06-07-2009 13:33:32	23.092	23.098	23.092
06-07-2009 13:33:42	23.103	23.057	23.103
06-07-2009 13:33:52	23.067	23.065	23.067
06-07-2009 13:34:02	23.022	23.007	23.022
06-07-2009 13:34:12	22.979	22.999	22.979
06-07-2009 13:34:22	22.981	23.002	22.981
06-07-2009 13:34:32	23.171	22.921	23.171
06-07-2009 13:34:42	22.861	22.872	22.861
06-07-2009 13:34:52	22.869	22.866	22.869
06-07-2009 13:35:02	22.886	22.843	22.886
06-07-2009 13:35:12	22.891	22.807	22.891
06-07-2009 13:35:22	22.839	22.801	22.839

III. CONCLUSION

In this paper, we have proposed a software testing technique for biomedical system, the test instructions have been described in the sequential test process, and some parameters, such as temperature, incremental quantity for next cycle, action time, gradients, and other parameters, have been discussed in detail.

The designed software testing system has been implemented and achieved the requirements of human-machine interface in the embedded system.

REFERENCES

[1] Morari, M., and E. Zafiriou, E. Robust Process Control. Prentice Hall, NJ. 1989.

[2] Desborough, L., and T.J. Harris. Performance Assessment Meas ures for Univariate Feedback Control, Can. J. Chem. Eng., vol.70, pp. 1186-1197, 1992.

[3] Desborough, L., and T.J. Harris. Performance Assessment Measures for Univariate Feedforward/Feedback Control, Can. J. Chem. Eng.,vol. 71, pp. 605-616, 1993.

[4] Harris, T.J. Assessment of Control Loop Performance, Can. J. Chem. Eng., vol.67, pp. 856-861, 1989.

[5] Harris, T.J., F. Boudreau and J.F. MacGregor, Performance Assessment of Multivariable feedback Controllers, Automatica,vol. 32,pp. 1505-1518, 1996.

[6] Harris, T.J., C.T. Seppala and L.D. Desborough. A Review of Performance Monitoring and Assessment Techniques for Univariate and Multivariate Control Systems, J. Proc. Cont., vol.9, pp. 1-17, 1999.

[7] Huang, B., S.L. Shah and K.E. Kwok. Good, Bad or Optimal? Performance Assessment of Multivariable Processes, Automatica, vol.33, pp. 1175-1183, 1997.

[8] Ko, B.-S. and T.F. Edgar. Performance Assessment of Cascade Control Loops, AIChE J., 46, pp. 281-291, 2000.

[9] Krishnaswamy, P.R., G.P. Rangaiah, R.K. Jha and P.B. Deshpande. When to Use Cascade Control, Ind. Eng. Chem. Res., vol.29, pp. 2163-2166, 1990.

[10] Krishnaswamy, P.R. and G.P. Rangaiah. Role of Secondary Integral Action in Cascade Control, Trans. IChemE., vol.70, pp. 149-152, 1992.

[11] Kary Mullis. The Unusual Origin of the Polymerase Chain Reaction,Scientific American, pp. 56-65,1990.

[12] Robert F. Service. The Incredible Shrinking Laboratory, Science, vol.268, pp.26-27,1995.

[13] A.F. Markham. A Tool for Molecular Medicine, British Medical Journal, vol.306, pp.441-447, 1993.

[14] Stephen.P.Naber. Molecular Pathology: Diagnosis of Infectious Disease.vol.331,pp.1212-1215,1994.

[15] Hansen J, Nussbaum M. Application of Bismuth-telluride thermoelectrics in driving DNA amplification and sequencing react ions. Proceedings of 15th International conference on Thermoelectrics, vol.1, 256-258, 1996.

[16] T.M. Teo1, S. Lakshminarayanan1 and G.P. Rangaiah1. Performance Assessment of Cascade Control Systems, Journal of The Institution of Engineers, Singapore, vol.45, pp27-38, 2005.

Jin-He Wang was born in Henan Province, China, in 1963. He received the M.S. degree from the Jilin University, Changchun, Jilin, in 1993 and the Ph.D. degree from the Dalian University of Technology, Dalian, Liaoning, in 1999. He is currently a professor with School of Computer Engineering at Qingdao Technological University. His research interests include image understanding, signal processing, and control theory.

Jing Zhao was born in Shandong province,China, in 1984. She received the diploma from the Qingdao Agricultural University,Qingdao,Shandong in 2008, She is currently a Graduate student with School of Computer Engineering at Qingdao Technological University. Her research interests include computer application and control theory.

Bing Yan was born in Shandong Province, China, in 1982. She received the diploma from the Qingdao Technological University, Qingdao, Shandong in 2008, She is currently a Graduate student with School of Computer Engineering at Qingdao Technological University. Her research interests include image understanding and computer application.