Multi-Mode Virtual Instrument Control Research and Implementation

Yaojie Chen[1, a] and MingQi[2, b*]

[1]College of Computer Science and Technology, Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, P. R. China

[2]College of Computer Science and Technology, Hubei Province Key Laboratory of Intelligent Information Processing and Real- time Industrial System, Wuhan University of Science and Technology, P. R. China

[a]378831830@qq.com, [b]sjms1989@163.com

*The correaponsing author

Keywords: Virtual instrument; Multi-mode; Activex; Customization

Abstract. Aiming at the problems that the current virtual instrument product mode is relatively single, that the code reuse is relatively low, that the development cycle is relatively longer, and that the drawing modules are relatively more, a multi-mode virtual instrument control which is based on GDI +（Graphics Device Interface） of ActiveX control is presented. The control uses the mode as the basic unit, switching among the modes to achieve multi-mode and making the corresponding customized operation to the control according to different environment needs. This paper describes the control principle, multi-mode switching method, design and implementation process, simulation test and practical application in detail. The test and application result shows that the control appears diversity and the mode switching operation is simple, that the control solves the existing problems and it can meet the needs of different environment.

Introduction

With the rapid development of computer technology, virtual simulation technology plays a more and more important role in the production and life. In engineering, education, art and other fields, the virtual simulation technology has been widely used. As a important part of the application of virtual simulation technology, the virtual instrument is the organic combination of computer technology and instrument technology. It overcomes the shortcomings that traditional realistic instrument human-computer interaction interface is poor, that high maintenance cost and poor portability, and that function extension is inconvenient [1,2].

At present, most of the virtual instrument products are developed on the software platform [3]. Its basic guiding ideology follows that National Instruments (NI) proposed "Software is instrument" in the 1980s. Different software platforms have developed their own virtual instrument products in succession. ChenHuai-min develops flight simulation virtual instrument based on GL Studio development platform, which has good portability and reusability[4];CaiLi develops vehicle the virtual instrument based on the Android platform, which has efficient reliability[5];GaoYing develops the virtual instrument ActiveX controls based on COM technology, which can enable the separation of interface and implementation [6];ZhangXu develops the vehicle liquid crystal virtual instrument panel based on HTML5, which has strong extendibility [7];ShenYang develops the virtual aviation instrument based on OpenGL, which has good generality [8] . To sum up, these virtual instrument products show advantages that the easy cross platform porting, strong reusability[9], the separation of interface and implementation, good extendibility, strong generality and so on. But its mode is relatively single, which is shown in that the source code must be changed when modify and set the attribute values to adjust the use of function. This shows problems that the code reuse rate is low, that the development cycle is long, and that the drawing modules are too many.

In view of these problems, this paper designs a kind of multi-mode virtual instrument control which has all the advantages of the above virtual instrument products to solve. The multi-mode virtual instrument control is developed and implemented based on the GDI + of ActiveX control. When the virtual instrument control is completed development, it no longer needs to change any source code to change attribute values. It directly makes click operation on the control interface to modify and set attribute values to complete the switching among multiple modes. Multi-mode virtual instrument control realizes modularization[10], universalization, integration[11] , standardization and makes that the one-time code reuse rate greatly improved, the development cycle significantly shorten, the drawing module number substantially reduced, the high efficiency development. Its significance lies in that it can according to the different needs of the environment to make corresponding customized operations on the interface to modify the property values, which integrates with the same type of virtual instrument products, display functions in one and achieves effect that other modules cannot.

GDI+ and ActiveX Control

GDI+. GDI+ is a subsystem of the Windows system, which is used as a graphical device interface and the upgrade version of previous GDI. Its main function is to be responsible for the exchange of information between the system and the drawing program. GDI+ can make the graphics hardware and applications to separate, so as to achieve the device independent, which greatly reduces the difficulty of the development.

Compared with GDI, GDI+ adds the gradient brushes, the matrix objects, the independent path objects and eliminating pixel saw tooth function. It can complete the drawing of complex graphics transform and make drawing graphics more beautiful, realistic [12].

Activex Control. ActiveX control is an object-oriented program development technology and tool. Its core is COM (Object Model Component) technology and mainly as a object or a component of software to use in a large number [13] .

Many ActiveX controls exist in OCX file format which can be used in a variety of development environment. It has a strong portability and cross platform functionality and possesses a strong reusability which can make one-time code development rate significantly improved.

ActiveX control cannot run independently and must be embedded into the container application to run with it. The ActiveX control should be put into test container before taking the functional test.

Multi-mode Virtual Instrument Control Principle and Design

Control Principle. Multi-mode virtual instrument control adopted the mode as the basic use unit. The mode is composed of a variety of attributes. Each attribute has its own value and a group of different attribute values constitutes a mode. The attribute value can be modified through the edit text on the control or using the written external interface to obtain the external transmission value in the way of communication.

When the new attribute value is input, the written drawing function is called to get the new attribute value and executes it, and then the program redraws the new graph to display the new mode. This is the switching way of the mode. Because the mode switching is achieved by acquiring the new attribute value and calling the drawing function to redraw, thus the double buffer method is used to avoid flicker [14]. All redrawing parts recombine the new whole which is the new mode. The control realizes the changing from one mode to another mode; it can meet the needs in different environment.

Control principle follows the basic idea that it uses the attribute as the unit module to recombine. The schematic diagram of multi-mode virtual instrument control principle is shown in Fig. 1.

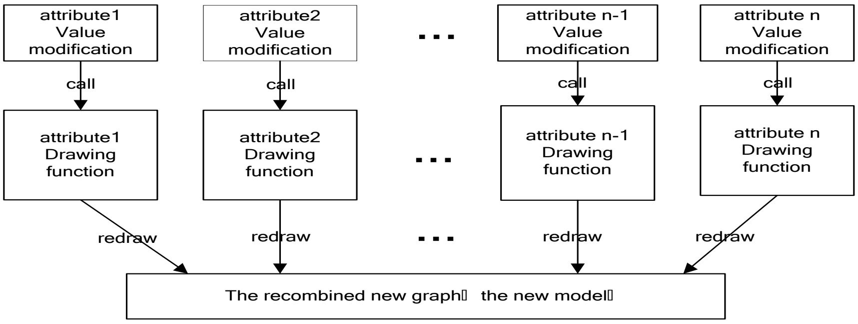

Figure 1. The schematic diagram of multi-mode virtual instrument control principle

Control Design. The designed multi-mode virtual instrument control in this paper is dial type. It can be widely used in industrial control, automotive, shipping, aircraft driving and other fields. It is mainly composed of four parts: the name of instrument dial, the range, the background color, the appearance decoration. The design structure diagram of multi-mode virtual instrument control is shown in Fig. 2.

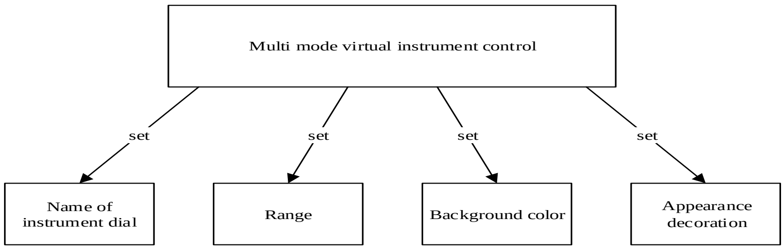

Figure 2. The design structure diagram of multi-mode virtual instrument control

These four attributes are regarded as a group and a group of attribute values is a mode of virtual instrument control. Setting different attribute values will show different modes. In the process of the use, it achieves the multi-mode to switch through modifying property values of the four attributes. The four attributes have a group of default attribute values, which are recognized as the initial mode, and then the attribute values are modified by the edit text on the control or communication input.

Name of Instrument Dial Attribute Design. The main function of this attribute is to display the name of the virtual instrument dial and indicate the use of the dial. The default attribute value before using is "RATE", and then the attribute values can be modified by the edit text on the control or communication input according to needs. The design flow chart of name of instrument dial attribute is shown in Fig. 3.

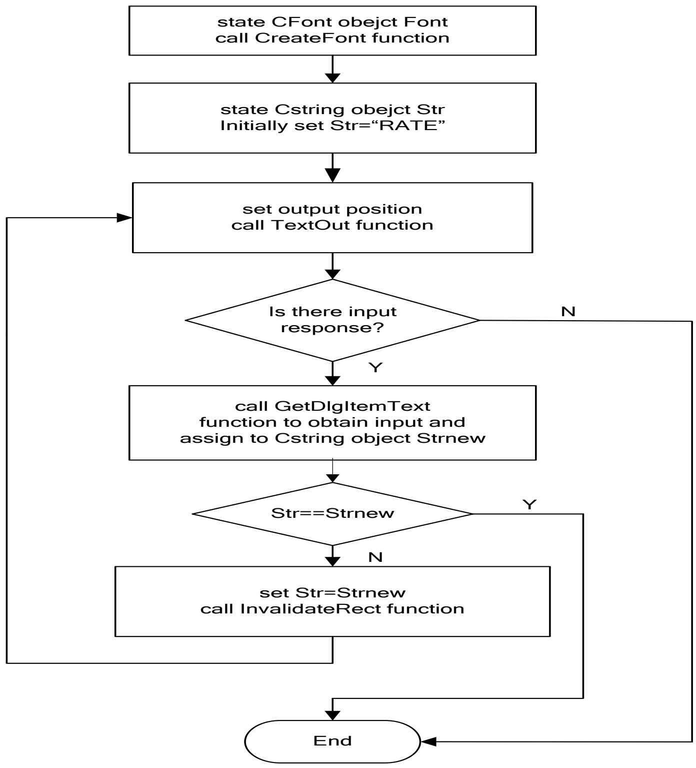

Figure 3. The design flow chart of name of instrument dial attribute

Range Attribute Design. The main function of this attribute is to display the measuring range of the virtual instrument and the current measurement value. The range and value is displayed on the scale dial. To design a realistic scale dial, the drawing of the scale line and the scale dial pointer is the key step in the design of this part.

Scale Line Drawing. The scale line of the scale dial is evenly distributed on the base of circle. Because its scale line is a line segment, so it can not directly by calling the function to draw a circle to implement. But it can reference the thought of circle to complete the drawing by the parameter equation of circle. The parameter equation is shown in Eq. 1.

$$X = a + r\cos\theta$$
$$Y = b + r\sin\theta \qquad (1)$$

X , Y is the horizontal and vertical coordinate of the point on the circle, a , b is the horizontal and vertical coordinate of the centre of a circle, r is the radius, θ is a parameter and $\theta \in [0, 2\pi]$.

What needs to explain is that the coordinate system of the display screen is different from the usual mathematical Descartes coordinate system. Its Y axis down to positive, but the mathematical Descartes coordinate system, Y axis up to positive. Therefore, the parameter of the equation makes a little change to meet the use of the display screen coordinate system. The modified parameter equation is shown in Eq. 2.

$$X = a + r\cos\theta$$
$$Y = b - r\sin\theta \qquad (2)$$

The basic thought of drawing principle is to use the parametric equation to draw several points of the circle evenly, and then the lines connect the center of the circle and the points of the circle to obtain several distributed radiuses of the circle. Then drawing a circle with a smaller radius and filling it with white color, the scale line is realized.

The centre of a circle $o(a,b)$ and radius r (set the size and position of the scale dial), The drawing flow chart of the scale line is shown in Fig. 4.

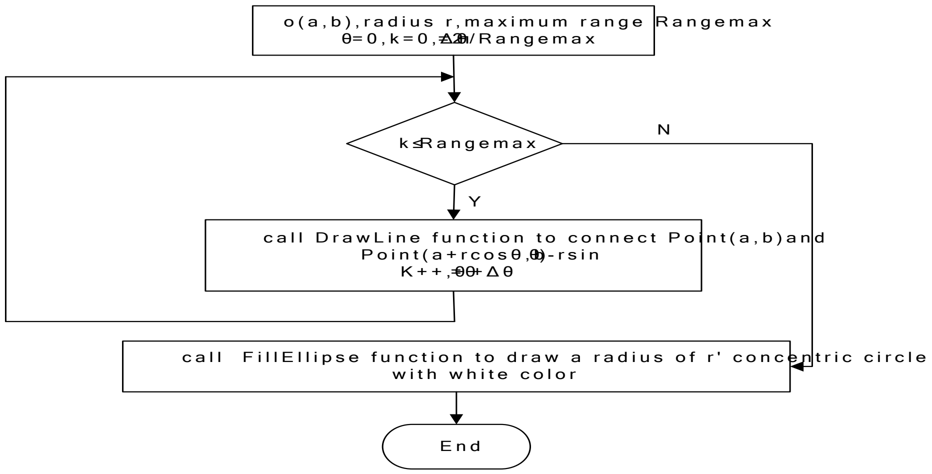

Figure 4. The drawing flow chart of the scale line

In addition, it is also necessary to draw the numbers of the scale value and dye for the scale line to make it more realistic and more eye-catching.

Scale Dial Pointer Drawing. When the scale dial pointer displays the current measurement value, it is necessary to point to the corresponding reading of scale line. If it only relies on a radius to draw a circle to use as a pointer, the visual effect is very poor and does not have the sense of reality. In order to improve the morphological fidelity [15] of the pointer, it uses the method of multiple graphics overlaying to draw the pointer. The method makes the pointer realistic, eye-catching, and achieves a better visual effect. The basic idea is to draw a small circle as the fixed point of the pointer to display, and then superimposing a triangle (its height is the radius) to make it become a common pointer style. The schematic diagram of scale dial pointer is shown in Fig. 5.

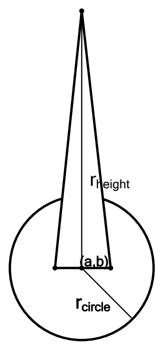

Figure 5. The schematic diagram of scale dial pointer

The centre of a circle in the scale dial is (a,b), r_{circle} is the radius of the small concentric circle, r_{height} is slightly smaller than the radius.

Coordinates of points in the graphics are as follows.

Point1

$(a + r_{height}\sin(m\_runangle)$, $b - r_{height}\cos(m\_runangle))$;

Point2

$(a + r_{height}/5 * \cos(m\_runangle + \pi/2)$, $b - r_{height}/5 * \sin(m\_runangle + \pi/2))$;

Point3

$(a + r_{height}/5 * \cos(m\_runangle - \pi/2)$, $b - r_{height}/5 * \sin(m\_runangle - \pi/2))$;

What needs to explain is that now the pointer points to the North (the position that scale value is 0). If it needs to display the current measured value, the external input data is obtained through the interface function Get Value and transferred to the value $m\_runangle$ that has been correspondingly processed. Then it calls the redrawing function Invalidate Rect to make the original pointer area invalid. After calculating the new results, it draws a new pointer interface and the pointer points to the current measured scale value. The drawing flow chart of scale dial pointer is shown in Fig. 6.

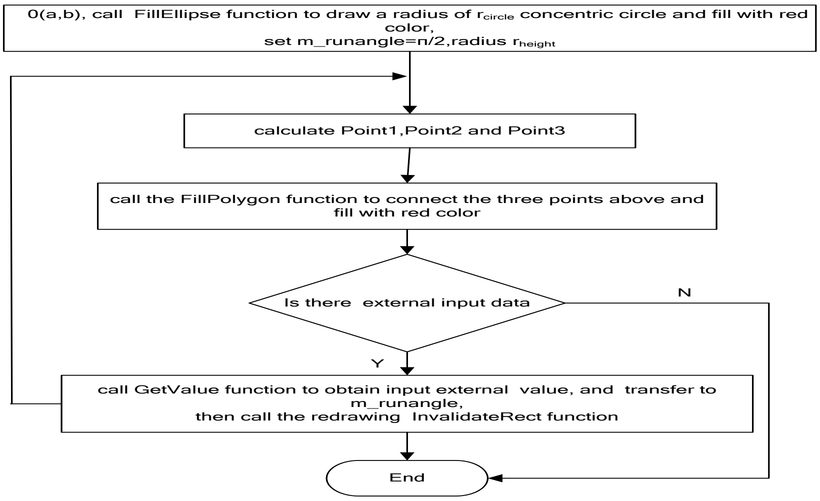

Figure 6. The drawing flow chart of scale dial pointer

After elaborating the drawing principle of the scale dial line and the scale dial pointer, the design idea of the range should be described. Before this part is used, the default property value $Rangemax = 90$ （The value is the maximum range that can be displayed on the entire scale dial）. Then the attribute value that is less than this value can be modified by the edit text on the control or communication input according to needs. The drawing flow chart of range is shown in Fig. 7.

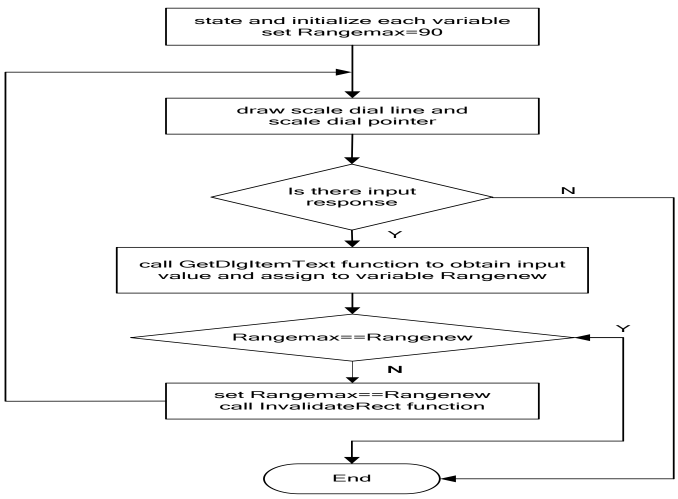

Figure 7. The design flow chart of range attribute

Background Color Attribute Design. The main function of this attribute is to modify the background color of the virtual instrument dial in order to meet the needs of the various environments. Before this attribute is used, the default attribute value is "White". Then the attribute value can be modified by the edit text on the control or communication input according to needs. The drawing flow chart of background color is shown in Fig. 8.

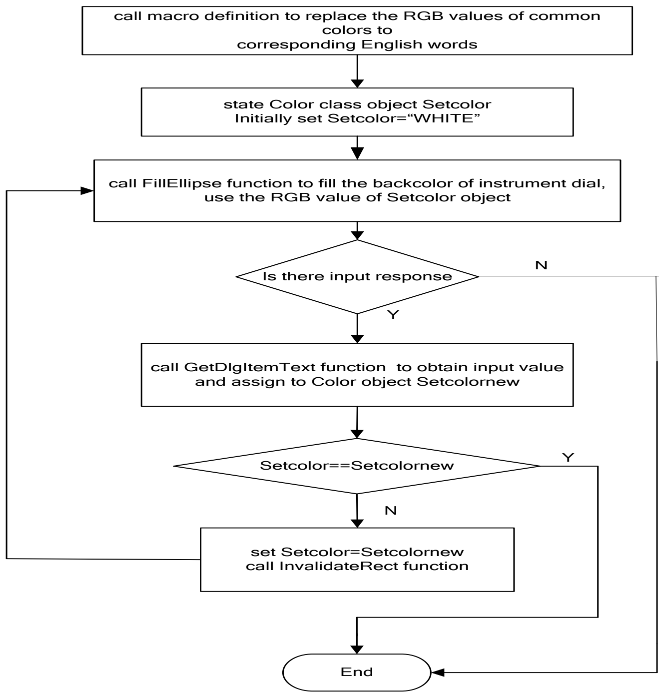

Figure 8. The design flow chart of background color attribute

Appearance Decoration Attribute Design. The main function of this attribute is to decorate the appearance of the virtual instrument dial, making it more beautiful and realistic. Two appearance decoration methods ways are designed. One is to increase the four fixed thumbtacks for the dial, making the appearance more close to the real instrument and enhancing the sense of reality; the other is to add the glass lens which has a color of gradient form, making the dial more gorgeous.

Thumbtack Drawing Principle. In order to make the virtual fixed thumbtacks close to the real ones and maximize their realistic effect, the LinearGradientBrush class in GDI+ is adopted to state and use the linear gradient encapsulation object Brush. Its main function is to make the color linearly gradient along a certain direction, so as to achieve drawing the linear gradient region. The black color and white color are selected as the basic tone of thumbtacks. The schematic diagram of thumbtack is shown in Fig. 9.

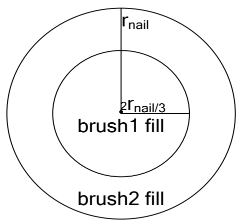

Figure 9. The schematic diagram of thumbtack

Drawing a circle with the radius of r_{nail} and filling with Brush1, its linear change is set along the upper left to lower right and from the shallower to the deeper. All the time drawing a concentric circle with the radius of $2r_{nail}/3$ and filling with Brush2， its linear change is set along top to bottom and from the deeper to the shallower.

Glass Lens Drawing Principle. In order to make the glass lens can play a good decorative effect and do not affect the normal color displaying of the virtual instrument dial at the same time, the PathGradientBrush class in GDI+ is adopted to state and use the path gradient encapsulation object Brush. Its main function is to make the color begin to change gradient from a specified path center to the each border of path, so as to achieve drawing the path gradient region. The main tone of the glass lens is chosen the royal blue color （RGB value is（65,105,255））.

Before this attribute is used, the default property attribute value is "BLANK" （Without any decoration）. Then the attribute value can be modified by the edit text on the control or communication input according to needs. The drawing flow chart of appearance decoration is shown in Fig. 10.

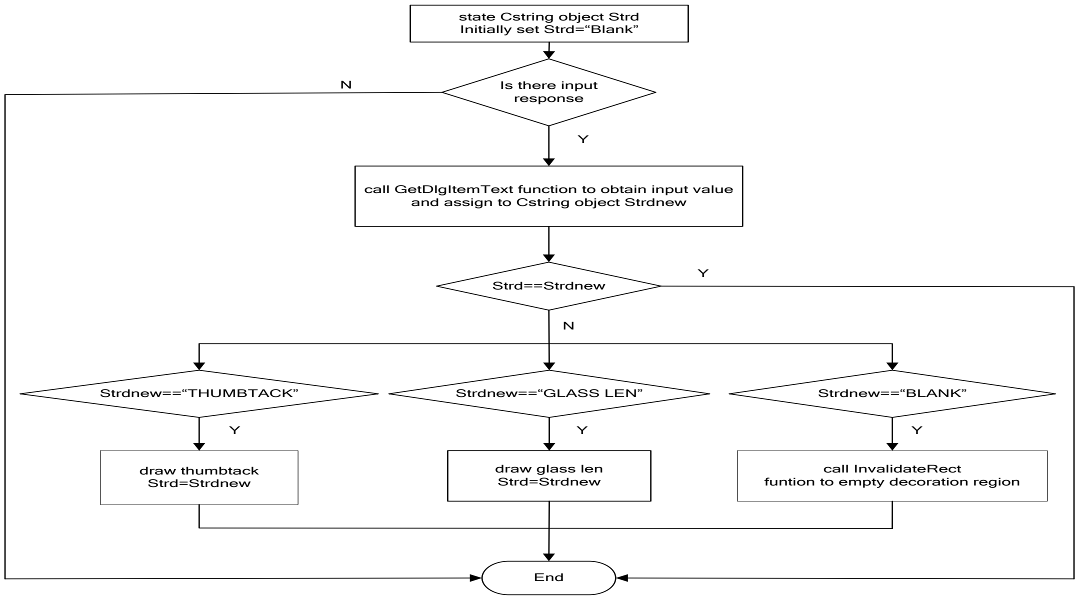

Figure 10. The design flow chart of appearance decoration attribute

Multi-Mode Virtual Instrument Control Simulation Test and Application

Multi-mode virtual instrument control uses Visual C++6.0 as the development tool. It establishes the project and imports GDI+ function library package to achieve programming the control. The thought of programming implementation follows the design idea of the previous section. The control is divided into name of the instrument dial, range, background color, appearance decoration four parts to complete.

The virtual instrument control that has been developed and implemented is the OCX format. It has good portability which can be used across the platform and as a standardized universal module to meet the use of different development environment.

The control is embedded into the VC++6.0 test container which is ActiveX Control Test Container to make the test. Simulation test chooses three modes to test and their attribute value are set as table1. Except the initial mode, the attribute values of the other two models are modified through the edit text on the control.

Table 1 Attribute values of three test modes

Mode	Name of instrument dial	Range	Back group color	Appearance decoration
1(initial mode)	RATE	90	WHITE	BLANK
2	SPEED	60	YELLOW	THUMBTACK
3	VOLUME	40	WHITE	GLASS LEN

Test effect charts of modes are shown in a, b, c of Fig. 11.

(a) **(b)** **(c)**

Figure 11. Test effect charts of modes

Comparing with the effect charts of simulation test, the virtual instrument control can implement the function of multi-mode switching and modify the value through the edit text. Each mode is different obviously in different attribute values, and the control appears diversity.

According to the development principle of the control, button type control, transistor type control and compound control with the same function are designed. When these controls are applied, the ships driving data platform for virtual exhibition which has a large scale display function is developed in a short time. The ships driving data platform for virtual exhibition is shown in Fig. 12.

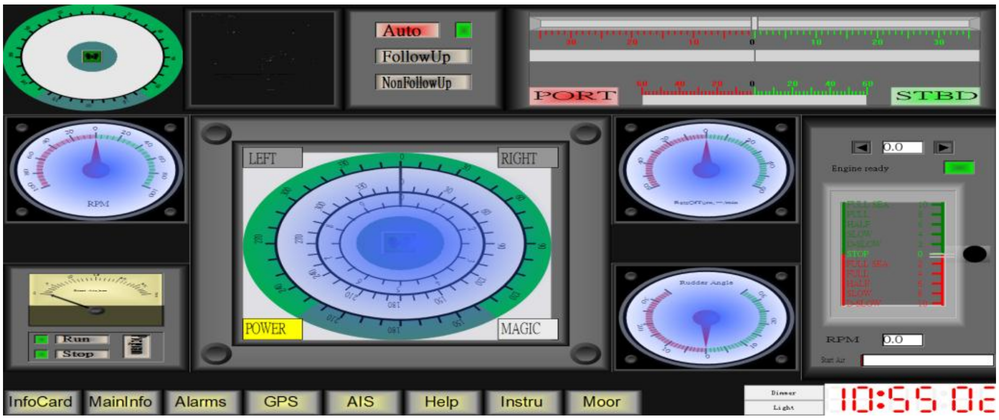

Figure 12. The ships driving data platform for virtual exhibition

Comparing with the virtual instrument products mentioned in the introduction, the virtual instrument control in this paper only uses a simple mode switching to meet the needs in different environment. It avoids the secondary development of the control, which means no longer need to modify any of the code. It only needs to develop one control, which can meet the needs of the same type of products. It reduces the original development cycle and can be used in more environments.

Conclusions

Aiming at the status that the current virtual instrument product model is relatively single, a multi-mode virtual instrument control which is based on GDI + of ActiveX control is given in this paper. This paper describes that the control principle, multi-mode switching method, design and implementation process. What is more, this paper also does simulation test and applies the controls to the platform development. The test result shows that the virtual instrument control has the advantages of simple operation, high one-time code utilization, strong practicability, short development cycle, and diversified modes. The virtual instrument control overcomes the problems in the previous products. It can accord the different needs of the environment to make corresponding customized operations on the interface to modify the property values, which is the effect that other products cannot achieve.

Acknowledgements

Modeling and Simulation of Ship Waves (2016znss09B);

References

[1] Lou Rui: The Design and Implementation of Automotive electronics virtual instrumentation simulation test platform (MS., Zhejiang University, China, 2012).

[2] Tao Lin, Li Zhang, Zi-bo Guo, Lin-cheng Li: Computer Simulation, Vol.29 (2012) No.3, pp.277-280.

[3] J Liu: Computer & Information Technology, 2013, pp.779-787.

[4] Huai-min Chen, Jin-wen Wu, Xiao-bo Huang: Measurement & Control Technology, Vol.32(2013)No.5, pp.89-91.

[5] Li Cai, Ni-na Dai, Ming Deng: Application of Electronic Technique, Vol.37 (2011) No.12, pp.83-86.

[6] Yin Gao, Fei Ge, Nin Liu, Shu-xia Guo: Microprocessors, Vol.34(2013)No.6,pp.42-45.

[7] Xu Zhang: The Design about Virtual LCD Instrument Panel on Vehicles on the Basis of HTML5 (MS., Southwest Jiaotong University, China, 2014).

[8] Yang Shen: Research and Realization of Virtual Aviation Instrument Based on OpenGL (MS., Hua zhong University of Science and Technology, China, 2014).

[9] Jun-ming Xu, Hong-chao Zhao, Guo-lin LI, Shao-li.Fan: Aerospace Control, Vol.28 (2010) No.6, pp.72-76.

[10] Yong-sheng Shi, Xiu-jing Li: Compute Engineering and Design, Vol.36 (2015) No.8, pp. 2287-2290.

[11] Sheng Yang, Neng-ling Tai, Cheng Yuan, Peng-cheng Cui: Computer Engineering, Vol.38 (2012) No.20, pp. 228-231.

[12] Jian-chang Yang: GDI+ advanced programming(Tsinghua University Press, China 2010).

[13] Ri-rong Yang, Cheng-zhi Yang, Xiao-hong Yang: Journal of Kunming University of Science and Technology（Natural Science Edition）, Vol.30 (2005) No.1, pp. 45-47.

[14] Zhi-gang Ma, Wen-yi Liu: Journal of University of Chinese Academy of Sciences, Vol.32 (2015) No.1, pp.116-120.

[15] Wei-ning Fang, Peng Wang: Journal of Beijing Jiaotong University, Vol.38 (2014) No.4, pp.7-13.