

The Design and Development of General Interface Program for Embedded System

Min Qi¹, Gong Cheng¹, Yue-lei Xu² and Ke Li³

School of electronics and information, Northwestern Polytechnical University, Xi'an, P.R. China
drqimin@nwpu.edu.cn

Cheng Gong

School of electronics and information, Northwestern Polytechnical University, Xi'an, P.R. China
931315949@qq.com

Xu Yue-Lei

School of Aeronautics and Astronautics Engineering, Air Force University of Engineering, Xi'an, P.R. China
Li Ke

School of automation, Northwestern Polytechnical University, Xi'an, P.R. China

Abstract—With the rapid development of embedded industry, embedded products emerge in endlessly, which brings great convenience for people's life. But there is no suitable development kit for interface design, the standardization and code reusability are poor. Therefore, based on the message-driven mechanism and combined with ideas of the hierarchical structure and modular design, an interface program development kit for the embedded system is designed and developed. Also, an general interface program implemented with the development kit is realized. It has been verified by an instance that this general interface program has many advantages such as explicit structure, small amount of code, easy to understand and modify, better maintainability and portability.

Keywords—embedded system; general interface program; hierarchical structure; modular design

I. INTRODUCTION

Embedded system is widely used to control or monitor machinery, equipment, factories and other large-scale systems equipment. It is application-centric and based on computer technology, which is one of today's hottest concepts[1]. Embedded products are found in people's daily life, from cell phones, electronic dictionaries, videophones in home to air conditioners, refrigerators, digital imaging machine. Digital age is no longer a symbol of PC to PC, but different patterns of embedded systems[2].

With the rapid development of IT industry, software systems emerge in endlessly. Interface procedures account for a large proportion of the entire software design and development process. Due to the lack of suitable general interface development kit, software developers are forced to do a lot of repetitive work and spend a lot of time and effort in every software development process, but the results are not always satisfactory[3]. It is extremely important and valuable to design and develop a lightweight, small footprint, high performance, high reliability general interface program which is based on the message-driven mechanism and combined with ideas of the hierarchical structure and modular design.

II. ARCHITECTURE DESIGN

As we can see in Figure 1, this general interface program

adopts hierarchical structure.

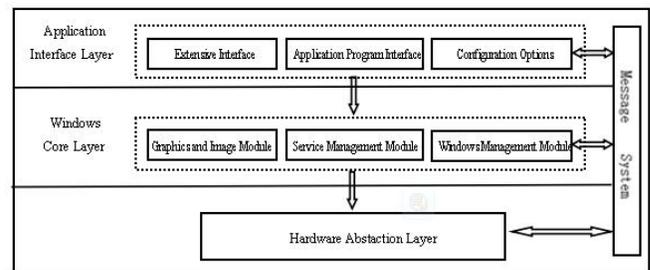


Figure 1. Architecture of the general interface program.

In the general interface program, layer is and relatively independent and interconnected between layer. Each layer finishes one's own functions and services for the upper through interfaces. It isolates changes inside one layer and does not affect the upper calling the lower to complete corresponding functions. Windows core layer provides core graphics operating functions, such as message mechanism, graphics device interface, fonts, windows and other desktop management functions. Application interface layer provides users with application programming interfaces and configuration options so that users can extend or clip this program.

A. Windows Core Layer

Windows core layer is the most important part of the general interface program which occupies a large component of work throughout the process of program design and development. Based on modular design, it includes graphics and image module, service management module, windows management module.

1) *Graphics and Image Module*: Graphics and image module is an independent drawing application interface based on the output abstraction layer. It encapsulates and extends the interface functions of the output abstraction layer further, providing the application programming interface of higher layer with more simple and more sophisticated drawing functions[4]. Graphics and image module includes a variety of functions, draw points, lines, circles, rectangles, fill color, text output, etc. By writing these functions, drawing a variety of graphics, and cooperating with the resources and fonts, it generates

various graphics interface, making programmers' thinking into reality.

2) *Service Management Module*: Service management module includes initialization, timer operation, predefined heap and data management sub-modules, etc.

a) *Initialization sub-module*: Initialization sub-module is the basic and indispensable part of interface program. We have to initialize the port, registers and system clock by using this module before we call the interface program in almost every instance. Program will not run properly without this module.

b) *Timer sub-module*: Timer sub-module is also often used in the interface program. It is commonly used in the control of blinking cursor and progress bar[5].

c) *Predefined heap*: Predefined heap plays an important role in the course of writing program. It will be more efficient to define and more convenient to transplant some functions if we use predefined heap properly. In addition, it avoids repeating the header files effectively.

d) *Data management*: The storage space in single-chip computer is very small. We often set up external memory on the hardware platform in practical applications. Data management module is responsible for storing, deleting and calling data in the external memory, giving interface program the information management capacity.

3) *Windows Management Module*: Windows management module is responsible for creating, deleting and managing various windows to achieve a variety of predefined window logic and the generation and processing of various window logic events.

Windows are the foundation of the graphical user interface and all operations of graphical user interface are done on them. In practical applications, we create a basis form called desktop first, covering the entire LCD screen. Then create another task window and the child window on it. By using multi-threading technology, we can hide and repaint the windows, etc[6].

Controls are the parts of window which can be understood as the child of the main window. In order to facility the users, and generate a consistent man-machine interface, this module provides a common predefined control classes, including buttons, edit boxes, list boxes, combo boxes, and so on.

B. Application Interface Layer

Application interface layer is the application programming interface of the general interface program provided to users. Developers can call the interface function encapsulated in application interface layer to complete interface design, processing of data logic, showing updates and message response. In addition, by using the extensive interface and configuration options, users can extend or clip the general interface program to meet individual needs.

III. DESIGN AND IMPLEMENTATION

This general interface program was programmed by C language. It has been proved that C language has many

advantages such as easy to understand and better portability. It is convenient for program development process, which laid a foundation for the commonality. Adopting modular design and proper packaging, they are independent of each other between modules. Internal structural modifications do not affect the other modules, meeting the needs of extension and clipping.

A. Graphics and Image Module

Graphics and image module is the core module of windows core layer and the basic of the general interface program. It implements basic graphics generation, image display, text output and other basic functions.

1) *Basic graphics generation*: As Figure 2 shows, this general interface program covers all required functions under normal circumstances, such as painting, draw lines, draw arcs, draw a rectangle, circle, painted oval, rectangular face, circle face and oval face painting.

Many points are combined into lines and many lines are combined into planes. Straight lines and arcs are the basis for graphics and images and their generation algorithm directly affect the efficiency of the program. Based on Bresenham, giving full consideration to the position of points users want to draw down and the ideal straight lines, it improves the efficiency and reduces the rendering error at the same time[7].

2) *Image generation*: First, we should transform images into hex code by using the software tools ,such as Image2Lcd. Then we will call this code in the program and generate images displayed on the LCD screen finally. This hex code should be stored in an array whose type is code. Only in this way will the single-chip computer stores it in the storage space of programme (ROM). Otherwise, a lot of hex code will be stored in the storage space of data (RAM). This is not only a waste of RAM space, but also can cause some compile errors.

3) *Text output*: Bitmap font has advantages of high efficiency, less resource-intensive, etc. We define each character in an array, including its height and width, the number of bytes per row and the pointer. All characters compose a assembly to be called by functions and displayed on the screen .

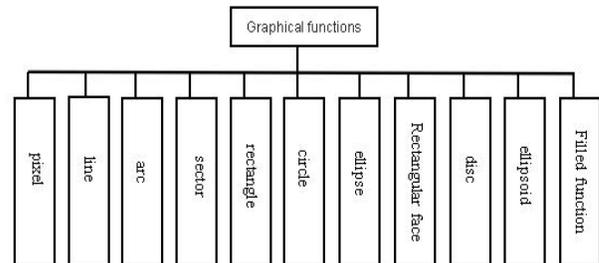


Figure 2. Architecture of graphical functions

As the machine code of Chinese characters distribute in several sections discontinuous, we need divide all the Chinese characters into a number of character sets, and combined them into a linked list. Each set includes indexes

of the first and the last character and pointers pointed to the first character and the next set.

First, we should take advantage of the indexes of the linked list which is the Chinese characters machine code to find the section of the character to be displayed and get the pointer. Then, we can find this character according to the offset relative to the first character in this section.

B. Service Management Module

Service management module includes initialization, timer operation, predefined heap and data management sub-modules, etc. We will introduce how to realize the data management sub-module.

It is necessary to store data outside the chip since the storage space in single-chip computer is very small. Service management module is responsible for reading and writing outside the chip.

There are two main functions in this part. Read_Flash (unsigned char *nContent, unsigned char nAddr, unsigned char nLen) is used for reading chips. This function reads nLen bytes of data continuously and stores it in the storage space beginning with the pointer named nContent. Similar to Read_Flash(), Write_Flash (unsigned char *nContent, unsigned char nAddr, unsigned char nLen) is used for writing chips.

We need to define the data to be saved in the form of structure and linked list. Together with some necessary hardware underlying driver functions, the former reading and writing functions can achieve data storage, output, search and delete, etc.

C. Windows Management Module

Windows management module is designed to complete windows organization, windows drawing and other windows operations. As shown in Figure 3, the structure of windows organization is a double linked list.

We create a node for each new window. Then, these corresponding window controls add into this node after inheriting window features. All nodes form a double linked list. When a node moves, the window appears and disappears corresponding. When a node is removed from the list, the window and its controls will be destroyed recursively.

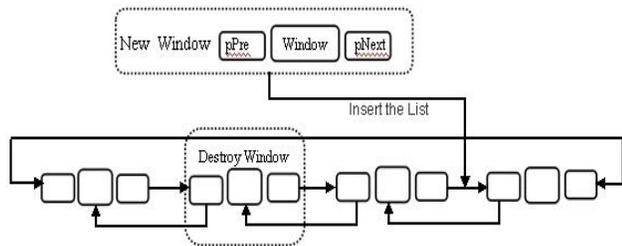


Figure 3. Structure of windows organization.

D. Message System

This general interface program is based on message-driven mechanism. The major difference between event-driven and process-driven is the message system which ensures the entire program running in an orderly manner.

Message is generated by events, such as input for devices, the operation of windows and configuration options, etc. Message loop is the core part of message system and the general interface program. It is a cyclic process that distribute message get from the queue to the target window[8].

Message loop iterates the message queue constantly. It preferred to wait for an external message when the queue is empty. Then, it encapsulates this message and pushes it into the queue. On the other hand, it will get the message and delivered it to the corresponding object when the message queue is not empty.

IV. APPLICATION EXAMPLES

We have designed and developed a software interface of leak-testing apparatus based on 8051F by using this general interface program. As shown in Figure 4 and Figure 5, we have achieved the following functions: image display, character display, graphics generation, matrix keyboard control and data management. It has been proved that this general interface program easy to use and saves a lot of manpower and resources.



Figure 4. Power Interface of the leak-testing apparatus.



Figure 5. Main Interface of the leak-testing apparatus.

V. CONCLUSION

Based on the message-driven mechanism and combined with ideas of the hierarchical structure and modular design, this kit has advantages of fully functional, wide universality, small amount of code, high efficient, easy learning and operation, and so on. Also, users can extend or clip it themselves by using the application interface and configuration options. It can be widely used in embedded

industry, to solve the problem of lacking in suitable interface development kit effectively.

REFERENCES

- [1] Zhen-liang Zhao, "The Research and Design of Graphic User Interface of Embedded System Based on Linux," Shanghai: Tongji University, April 2007.
- [2] Chao Feng, "The Research and Design of Graphic User Interface of Embedded System Based on Linux," Wuhan: Huazhong University of Science & Technology, April 2007.
- [3] Blanchette J, Summerfield M, "C++ GUI Programming with Qt4," Prentice HallPTR, 2006.
- [4] Yong-jun Cai, Jing-dong Wu, "The Research and Design of Graphic User Interface of Embedded System," *Microcomputer Information*, 2007, 23(8-2): pp. 64-66.
- [5] Ke-jin Bao, "The principle and applying of C8051F microcomputer," Beijing: China Electric Power Press, 2006.
- [6] Ji-ping LI, Shu-e BI, Li-qiong YI, "Design of user interface of teaching box based on QT4 and Windows CE," *Modern Computer*, 2010, (5) : pp. 175-177.
- [7] Jia-guang Sun, "Computer graphics," Beijing: tsinghua university press, 1998.
- [8] Jie Xiong, Xin-de Liu, Ren-jian Feng, "The Design and Implementation of Graphic User Interface of Embedded System," *Computer Engineering and Design*, 2012, 33(7): pp. 2602-2606.