

A Design of Embedded Serial Communication Program Based on MiniGUI

Wu Xin-yi⁺, Peng Xiao-hong, Hou Li-gang, He Ming, Wu Wu-chen

VLSI & System Laboratory, Beijing University of Technology, Beijing, China

Abstract. A MiniGUI transplantation from Linux PC to SPARC platform is introduced in this paper, which runs the embedded serial communication program on SPARC platform. The hardware controlled by MiniGUI program works successfully. Through serial communication between SPARC platform and Linux PC, this paper provides a basic design idea of PC monitor test program and PLC application in the graphical interface.

Keywords: MiniGUI; SPARC platform; serial communication; embedded linux

1. Introduction

With the development and popularization of embedded systems, GUI in embedded systems becomes increasingly prominent. When the smart phone, PDA and other systems appear, not only the professional operators can use the embedded systems, but also ordinary people. In the traditional area of embedded control, the content of human-computer interaction is not very complicated, but users also need a friendly interface. Therefore, the popularity of GUI in embedded systems is a trend in the future.

Currently, there are three widely used embedded GUI such as QT / Embedded, Micro windows and MiniGUI. MiniGUI is an excellent China-made software tool launched by Feynman Software Company. It is a graphical user interface support system for embedded systems. In the software system, it is between the kernel and applications, so it can be referred to as a "middleware" product. Because MiniGUI is efficient, reliable, customizable, compact and flexible, it is especially suitable for small and medium real-time embedded application system.

This paper attempts to combine MiniGUI interface design and serial communication design so as to have a good interactive interface between the users and the target board. The operation and control for the target board will be much easier.

2. Design plan

The system mainly consists of two parts, Linux PC and SPARC platform. The version of Linux PC is Ubuntu 9.10. In this paper, we choose leon3-gr-xc3s-1500 development board[1][7] as the SPARC platform(Hereinafter referred to as LEON development board). LEON development board has two PS2 interfaces for keyboard and mouse. It also has a VGA interface, which is used to connect with monitor, so that the information in the LEON development board can be printed on the monitor. Besides, LEON development board has a RS232 interface which is used to complete serial communication with Linux PC. The version of kernel is linux-2.6.21.1 and the version of MiniGUI is MiniGUI1.6.10. Overall structure is as follows:

⁺ Wu Xin-yi
wuxinyig1@hotmail.com

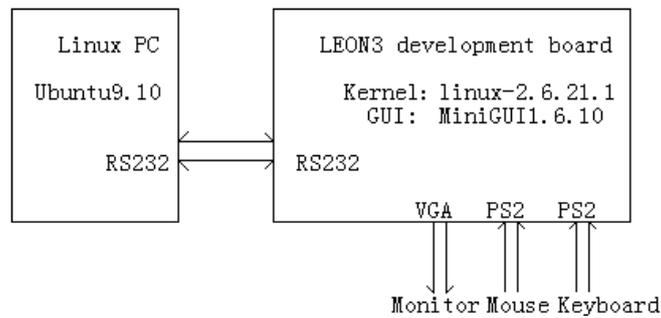


Figure 1. System architecture diagram

3. The architecture and message mechanism of MiniGUI

3.1. The architecture of MiniGUI

MiniGUI is a complete embedded GUI system, and it has a clear hierarchical design structure, the architecture is as follows:

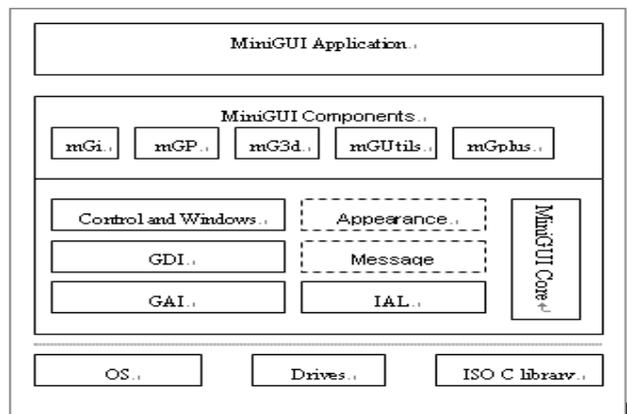


Figure 2. The architecture of MiniGUI

From above figure, we can see that MiniGUI is divided into three parts, which is respectively MiniGUI core, MiniGUI application, and MiniGUI components on the whole [3] [6].

According to the macro-level structure of embedded graphics system, the implementation of MiniGUI core is based on transplant layer (GAL and IAL). GAL is used for the graphics engine, which provides the interface of bottom graphics. IAL is used for the input engine such as keyboard, mouse, and touch screen and so on. The middle layer provides certain components for MiniGUI application. These components provide some special features for upper layer. For example, mGi is the input method component of MiniGUI, mGp is the print component of MiniGUI. The top layer is MiniGUI application which achieves its function by calling MiniGUI API.

3.2. The message mechanism of MiniGUI

MiniGUI is a GUI system driven by messages. All operations of it are implemented around messages. MiniGUI application interacts with the outside world by receiving messages. Messages are generated by the system or application. The system generates messages in response to input events or applications which can complete a task by generating messages and communicate with a window of other applications.

A message is composed of window (hwnd), message number (message), WPARAM parameter (wParam), and LPARAM parameter (lParam). Parameters of message contain important content. For example, lParam generally includes the mouse position in terms of the mouse message, while wParam parameter contains state information of the corresponding SHIFT key.

Message loop is a loop body in which the program uses GetMessage function to get messages from the message queue and then use the Dispatch Message function to send messages to the specified window, and pass the messages and their parameters. A typical message loop is as follows:

```

While(GetMessag (&Msg, hMainWnd) )
{
    TranslateMessage(&Msg);
    DispatchMessage(&Msg);
}

```

4. Design of MiniGUI serial communication program

4.1. Serial Communication Program

Implementation of serial communication program in Linux includes setting, opening, reading, writing, closing and other operations of serial port. The setting of the serial port, which includes setting of baud rate, parity bit and stop bit, is mainly the setting of struct termios members. struct termios is as follows:

```

#include <termios.h>
Struct termio
{
    unsigned short c_iflag;
    unsigned short c_oflag;
    unsigned short c_cflag;
    unsigned short c_lflag;
    unsigned char c_line;

    unsigned char c_cc[NCC];
};

```

In the struct termios, c_cflag, c_iflag and c_cc are very common. Among them, the most important member is c_cflag, because it is related to the setting of baud rate, character size, data bits, stop bits, parity and hardware flow control and so on. Besides, c_iflag control the processing of input character at the receive port. At last, c_cc contains the definition of timeout parameter and control characters.

After the proper configuration of serial port number, baud rate, data bits, parity and so on, we can open, read and write serial port by file I/O functions such as open, read, write, close and so on. The functions are the same as the functions of reading and writing of normal file. The only difference between them is that a serial port is a terminal device. We can often find serial devices into the directory “/dev/ttyS0” or “/dev/ttyS1” in Linux.

4.2. Design of MiniGUI interface

In order to run MiniGUI serial communication program on Linux PC, we have to install MiniGUI library, MiniGUI resources and QVFB and so on[5].

First, download the following resources from the MiniGUI official website: libminigui-1.6.10(MiniGUI library source code), minigui-res-str-1.6.10(MiniGUI resources); and zlib-1.2.3, jpeg-6b, libpng-1.0.10, giflib-4.1.4, freetype-1.3.1, popt-1.7 dependent libraries; And qvfb-1.1(Qt virtual FrameBuffer).

Secondly, configure, compile and install libminigui-1.6.10 and minigui-res-str-1.6.10 on Linux PC. The command is as follows:

```

#./configure CC=/opt/sparc-linux/bin/sparc-linux-gcc --build=i686-pc-linux --target=sparc-linux --
host=sparc-linux prefix=/usr/local/
#make
#make install

```

Finally, install QVFB on Linux PC. QVFB is a Qt Virtual Frame-Buffer tool that can simulate the real device for MiniGUI application. Because this application in the paper is designed to use the same interface between PC and the target board, we can directly code and debug on Linux PC. Then we can verify the correctness of MiniGUI application before transplantation.

After the software mentioned above has been installed, we can verify the function of serial communication program on Linux PC. Connect com1 to com2 with a serial cable on Linux PC, and open two QVFB at the same time. Then run separately serial send program and serial receive program on each QVFB. The result is as follows:

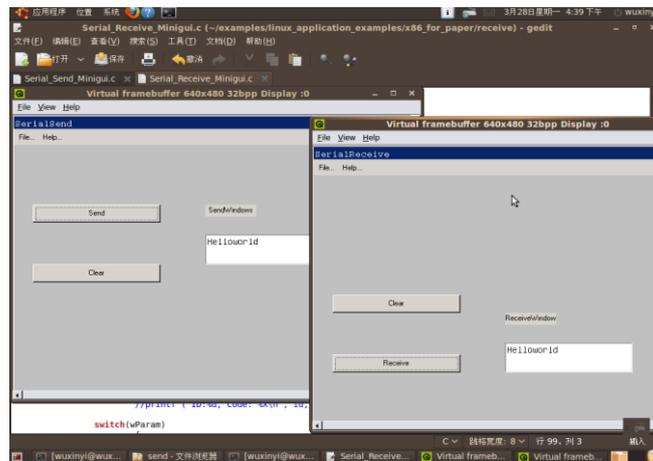


Figure 3. Serial communication program on Linux PC

5. Transplant MiniGUI to SPARC platform

We choose Snapgear Linux operating system for LEON development board, which is published by the Gaisler Research. It's necessary to conduct a software configuration before transplantation. The friendliest configuration way is "make xconfig". We can configure bootloader and kernel in order to meet our needs.

After the software configuration[2][7], We will compile libminigui-1.6.10, minigui-res-str-1.6.10[4], dependent libraries by crossing compilation sparc-linux-gcc, and then move libraries and resources about MiniGUI into /snapgear-2.6-p36/romfs/usr/local/lib, MiniGUI.cfg into /snapgear-2.6-p36/romfs/usr/local/etc, the programs into /snapgear-2.6-p36/romfs/usr/local/src. After all done, we can execute the command "make". Finally, image.flashbz will be generated [7].

After downloading image.flashbz to LEON development board, the little penguin logo will appear on the terminal interface.

Modify manually MiniGUI.cfg file as follows [6]:

```
[system]
#GAL engine and default options
gal_engine=fbcon
defaultmode=640*480-8bpp
#IAL engine
lal_engine=console
mdev=/dev/input/mice
mtype=IMPS2
[fbcon]
defaultmode=640*480-8bpp
```

Execute the following command:

```
#export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
#cd /usr/local/src
#./helloworld
```

The result is as follows:



Figure 4. Transplant MiniGUI to SPARC platform

6. The test of the embedded serial communication program

Before the test, we must build the MiniGUI and NFS environment. After that, connect com1 (LEON development board) to com2 (Linux PC) with a serial cable. Then run serial send program (compiled by `sparc-linux-gcc`) on the LEON development board and serial receive program (compiled by `gcc`) on Linux PC at the same time. The result is as follows:

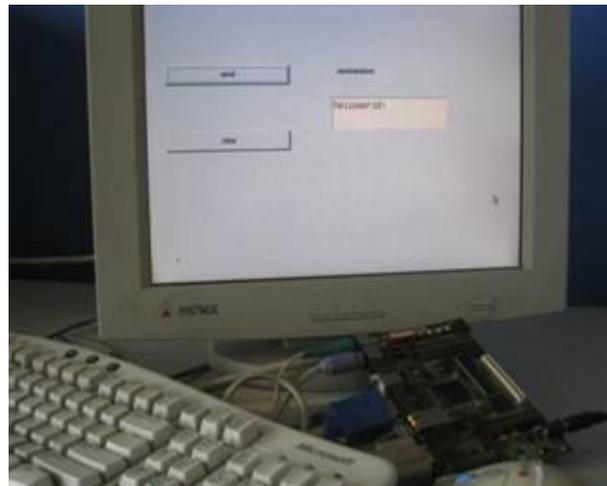


Figure 5. Embedded serial communication program on SPARC platform

7. Conclusion

This paper proves that MiniGUI can run smoothly on SPARC platform, and MiniGUI application can directly operate the hardware to complete certain functions. Besides, LEON development board (SPARC platform) and Linux PC can achieve serial communication based on graphical interface. So PC can easily monitor the status of PLC (Single chip, FPGA, ARM and so on). What's more, it's more convenient for ordinary people to operate PLC in the graphical interface. In short, MiniGUI is applicable and feasible in low-end embedded devices [5] [6].

8. Acknowledgment

This work was supported by the National Natural Science Foundation of China (60976028) and the Startup Foundation for Doctors of BJUT (X0002014201101, X0002012200802, and X0002013201103).

9. References

- [1] Zhou Zhonghua, Wu Wuchen, He Ming, Hou Ligang, "A SoPC design based on LEON3 SoC platform", *Microelectronics & Electronics*, 2009, pp.400-403.

- [2] Munoz, A.; Ostua, E.; Bellido, M.J.; Millan, A.; Juan, J.; Guerrero, D.; Building a SoC for industrial applications based on LEON microprocessor and a GNU/Linux distribution, IEEE international symposium on Industrial Electronics, Cambridge, 2008, pp.1727-1732.
- [3] Liu Junliang; Pan Gang; Li Ping; Embedded Linux Graphic Terminal Design Based On ARM, Control Conference, 2006, pp.1921-1925.
- [4] Liu Chang-sheng; Guo Yong; Xie Xi-hua, Investigation and transplantation of MiniGUI based on embedded Linux, Microcomputer Information, Changsha, 2008, pp.103-105.
- [5] MINIGUI-PROG-GUIDE-V2.0-4C, Version 2.0.4/1.6.10, Aug, 2008.
- [6] MINIGUI-USER-MANUAL-V2.0-4C, Version 2.0.4/1.6.10, Aug, 2008.
- [7] GRLIB IP Core Manual, Version 1.0.19, Sep, 2008