

## A Multi-thread UI Framework Without Graphics Driver Of Mouse On VxWorks System

---

**Congliang Hu**<sup>1</sup>

*719 Research Institute*

*Wuhan, 430000, China*

*E-mail: hucongliang0015@163.com*

**Wenpei Rao**

*719 Research Institute*

*Wuhan, 430000, China*

*E-mail: raowenpei@vip.qq.com*

**Mingjie Fang**

*719 Research Institute*

*Wuhan, 430000, China*

*E-mail: fangmingjie18@126.com*

In order to solve the problem of low efficiency to draw UI interface without hardware graphics driver of mouse, this paper proposes a multi-thread user interface framework by utilizing WindML. The framework is divided into three parts, specifically, main thread, peripherals response thread and display thread. Main thread implements some significant functions including receiving data, processing data and sending data. Peripherals response thread monitors the input messages of peripheral equipment, and transmits them to main thread and display thread. Display thread receives the displaying messages from main thread and that of coordinates of cursor from peripherals response thread, then integrate and paint the interface. The results of the experiment show that the multi-thread user interface framework proposed in this paper significantly decreases the time of drawing per frame and increases the refresh rate of the interface resulting in a smoother user interface and more comfortable operation.

*CENet2017*

*22-23 July, 2017*

*Shanghai, China*

---

<sup>1</sup>Hu Congliang

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0).

<http://pos.sissa.it/>

## 1.Introduction

Vxworks is an embedded real-time operating system developed by the Wind River System company. In order to work more efficiently on VxWorks, WRS also provides users with Tornado, which integrates user-friendly development environment, simulation environment and many other development tools. As an embedded real-time operating system, VxWorks is highly reliable and has outstanding real-time performance[1]. VxWorks is designed to use in the embedded systems requiring real-time, deterministic performance, especially for industries such as aerospace and defense, medical devices, industrial equipment, robotics, energy, transportation, network infrastructure, automotive, and consumer electronics. Besides Tornado, VxWorks is provided with WindML, a media development library that can be utilized to develop UI programs and process graphics, audio, and video on the embedded operating systems. Moreover, WindML supports input devices and can process events.

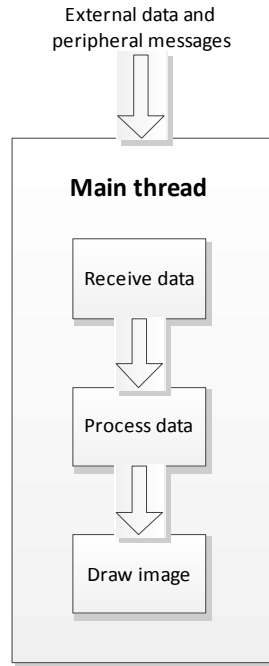
However VxWorks system also has its inherent flaws, for example, its poor scalability and incapability to support some drivers and chips. The official does not provide graphics driver, which needs to be developed by third-party professional companies[2]. The general equipment and chip manufactures do not provide driver for VxWorks.

In the given example of WindML, the cursor is drawn with the drawing function in the UGL library. Furthermore, both peripheral response functions and drawing functions run in the main thread. The main thread usually handles complex tasks, receives and responds to the input of external devices. It is time-consuming and inefficient to draw every frame if the drawing function operates in the main thread. In order to solve these problems, there are many methods to improve the frame rate in graphical interface [3][4][5]. In this paper, we proposed a multi-thread user interface framework on VxWorks system by utilizing WindML.

## 2.The Theory of Framework

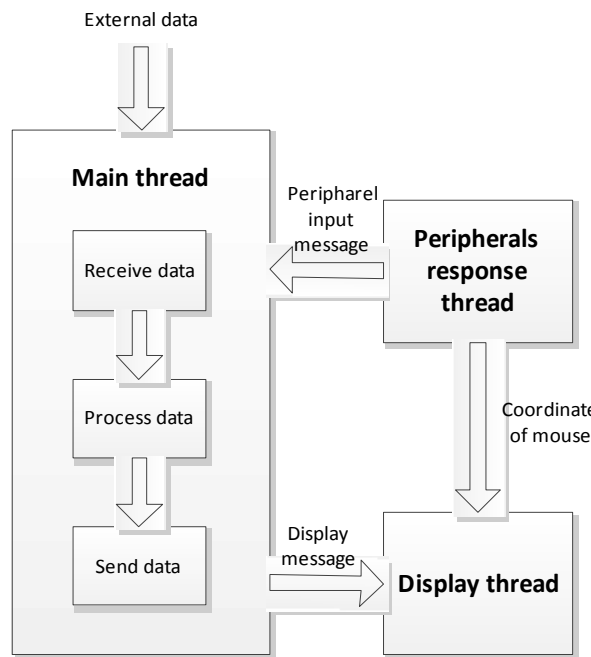
This paper is to overcome the shortcomings mentioned above. Without the specialized graphics driver for the mouse, a multi-thread user interface framework on VxWorks system is proposed by utilizing WindML. This framework can handle complex tasks, respond to input of external devices and draw interface efficiently.

WindML-based interactive UI program on VxWorks system usually runs as shown below. The main thread receives external data and peripheral messages, processes data, and draws the interface graphics. The time of updating frame,  $T_0 = T_r + T_p + T_d$ .  $T_r$  is the time of receiving data.  $T_p$  is the time of processing data and  $T_d$  is the time of drawing the graphics. When the processing function is complex and the amount of computation is large and  $T_p$  becomes large, resulting in larger  $T_0$ . When the refresh cycle of graphical interface becomes longer, the refresh rate becomes lower. When the frame rate is less than 25 fps, we can visualize its discontinuity. Furthermore, in each update period,  $T_p$  changes with the amount of calculation.  $T_0$  is not fixed and the refresh rate of display changes. All these factors lead to visual discomfort.



**Figure 1:** The flow chart of traditional method

In order to improve the display efficiency, a multi-thread user interface framework without graphics driver of the mouse on VxWorks system is proposed. The framework design flow chart is shown as follows.



**Figure 2:** The flow chart of multi-thread framework

### 3.The Multi-thread Framework

#### 3.1Main Thread

Main thread is responsible for the main functions of the program, including data reception, data processing, and data transmission. Firstly, the data reception receives the input data from the external interface and the messages from peripheral response thread. Then, the data

processing responds to the external signal and processes the received data. Finally, the data transmission sends the display data the display thread needs. These three steps are executed in a loop.

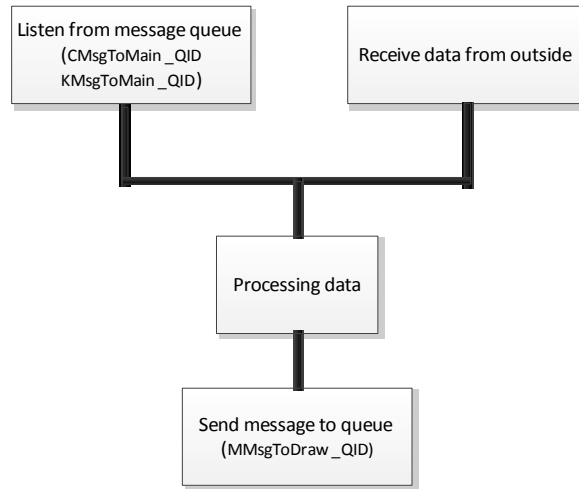


Figure 3: The flow chart of main thread

### 3.2 Peripherals Response Thread

Peripherals response thread is responsible for monitoring the peripheral input messages. On the one hand, the mouse or keyboard operation is sent to the main thread from the message queue to complete the data processing operations; on the other hand, the coordinates of the cursor are sent to the display thread from message queue to update the real-time position of the cursor.

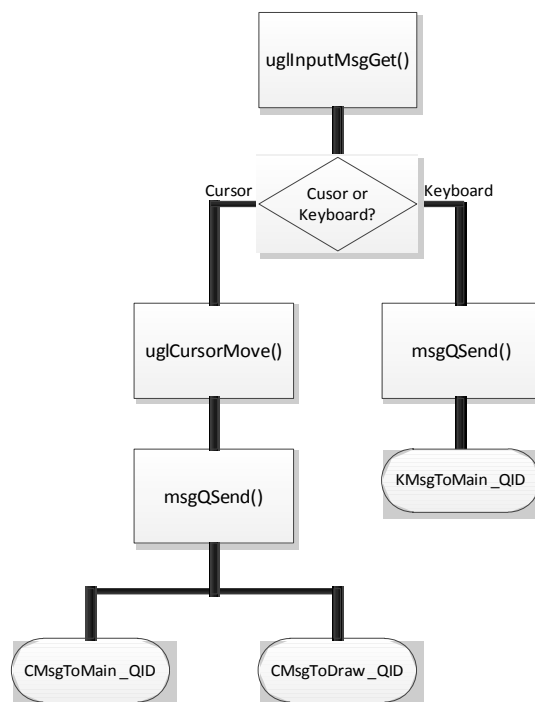


Figure 4: The flow chart of peripherals response thread

1. Obtain the external device input messages by *uglInputMsgGet()* function, and determine the mouse input or the keyboard input.

POS(CENet2017)054

2. If it is the mouse input, the cursor's coordinates are updated by *uglCursorMove()* function.

3. The message of cursor is sent to the main thread from *CMsgToMain<sub>Q</sub>ID* message queue by *msgQSend()* function, and the message of cursor is also sent to the display thread from *CMsgToDraw<sub>Q</sub>ID* message queue.

4. If it is the keyboard input, the message of keyboard is sent to the main thread from the *KMsgToMain<sub>Q</sub>ID* message queue by *msgQSend()* function.

5. Step 1, 2, 3, 4 are executed in loop.

### 3.3 Display Thread

Display thread is responsible for monitoring the messages needed to display from the main thread and that of the coordinates of cursor from the peripheral response thread. As the cursor often moves, the refresh frequency needs to be more than 25 fps so that the operations displaying on the screen can look smooth. Therefore, the frequency of message sent from the peripherals response thread needs to be at least 25Hz. Meanwhile the cycle of sending updated data from the main thread should be within 500ms. The display thread integrates the messages from the other two threads and draws the graphics with WindML function.

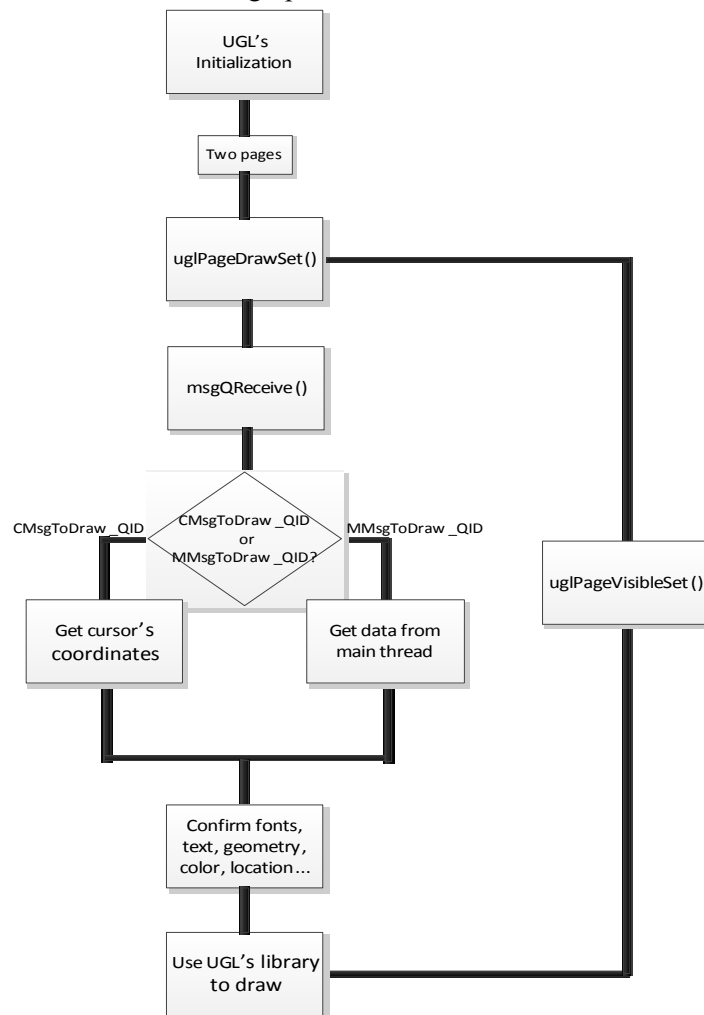
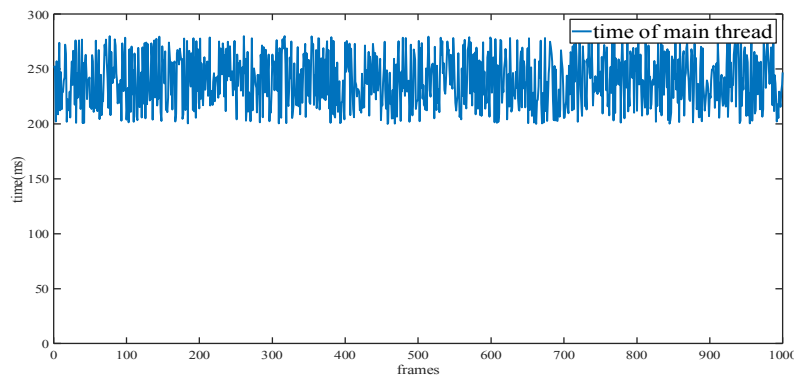


Figure 5: The flow chart of display thread

1. Initialize UGL members.
2. Define `page[0]` and `page[1]` for double-buffering, and set the page to display by `uglPageDrawSet()` function.
3. Receive the messages of cursor's coordinates from `CMsgToDrawQID` message queue by `msgQReceive()` function. And receive the messages of display from `MMsgToDrawQID` message queue.
4. Determine the color and position of the font, text, geometry etc. and use the UGL library function to draw the image.
5. Set the `page[0]` or `page[1]` to display by `uglPageVisibleSet()` function.
6. Step 1, 2, 3, 4 and 5 are executed in loop.

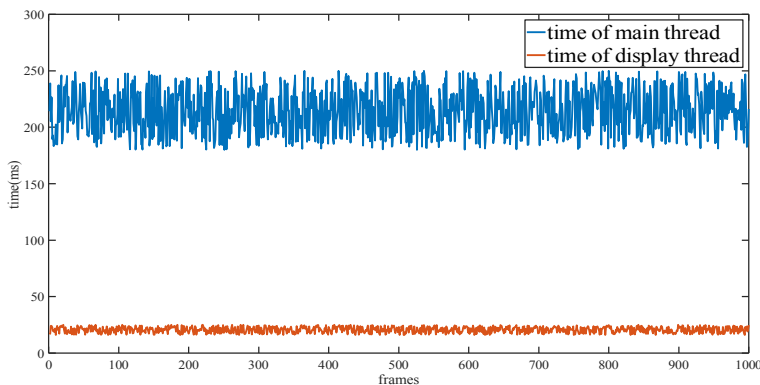
#### 4.Results of the Experiment

In the experiment, we compared the efficiency of two UI frameworks. In the test program, the complex calculation takes at least 200ms. At first, the UI interface program needs to receive data from the external, and then process and calculate, finally display the results. We can choose the method of calculating by using the mouse.



**Figure 6:** The results of traditional method

In the traditional framework, the time of main thread equals the time of updating frame. In equation  $T_0 = T_r + T_p + T_d$ ,  $T_0$  is about 250ms, and  $T_p$  takes the most time of one frame, which significantly decreases the frame rate. As the frame rate is about 4 fps, it's very difficult for human to operate in the UI program by using the mouse.



**Figure 7:** The results of our method

In the multi-thread user interface framework proposed in this paper, firstly, the main thread assumes the computation part is complex and time-consuming, and the display thread is responsible for drawing graphics, which takes much less time. Secondly, the display thread can update the position of cursor very quickly, because it only depends on the frequency of the messages sent by peripherals response thread. Thirdly, the peripherals response thread is able to send peripherals messages, including the coordinate messages of the cursor in real time, in a word, the multi-thread user interface framework proposed in this paper greatly increases the frame rate of the UI program on VxWork system. In the experiment, the cycle time of display thread is about 25ms, which means the frame rate is at least 30 fps. It will be much smoother and more convenient for human to operate.

## 5.Conclusion

Compared with the traditional WindML interface framework based on the mouse operation, the multi-thread interactive UI design framework in the VxWorks system proposed in this paper has solved the problems of low efficiency to draw UI interface without hardware graphics driver of mouse. The framework improves the frame rate when there are complex computation or tasks, and makes it more comfortable for human to operate in the VxWorks UI program.

## References

- [1]Zha Z Y, Xiong H G. *Development platform for real-time control system Chinese UI based on VxWorks*[J]. Computer Engineering & Design, 2009, 30(8):1875-1868.
- [2]Chen Y. *Research on Implementation of Dynamic UI Based on MiniGUI on VxWorks System*[J]. Ship Electronic Engineering, 2009.
- [3]Cai H. *Method of Design Graphical Interfaces with WindML Based on VxWorks*[J]. Industrial Control Computer, 2005.
- [4]Li H. *Analysis of WindML Graphical Interface Program Frame on VxWorks*[J]. Industrial Control Computer, 2007.
- [5]Guo T Y, Liu P, Wang H C, et al. *Design and Implementation on PC Control Interface of Robot Based on VxWorks Operating System*[J]. Applied Mechanics & Materials, 2014, 602-605:1109-1112.