論文
4-4-3

GUI Environmental Controller Design with USB IF (USB IF를 이용한 GUI 환경의 제어기 개발)

Sun-Hag Hong(홍 선 학)[1]

ABSTRACT

In this paper, GUI environmental controller is implemented with USB(Universal serial bus) IF(interface). Especially, the performance examination of ultrasound diagnostic device which connects external USB port with front panel is automatically inspected by VC++ examination program. The test program minimizes the operation of inspector with making the examined result audible or visible, and maximizes the productivity of custom device. The GUI environmental controller controls a standard keyboard, a track ball with dedicated buttons to emulate a standard two-button mouse, switches, rotary encoders, an analog potentiometer group, two alpha-numeric displays, and a touch screen; all of which generate events for the host.

요 약

본 논문에서는 USB 인터페이스를 이용한 GUI 환경의 제어기를 구현해 보였다. 특히, 프론트 판넬에 취부된 외부 USB 포트의 성능 검사는 비쥬얼 C++ 검사 프로그램을 사용하여 자동적으로 실행되도록 하였다. 성능검사 프로그램은 실험 결과를 가청적 또는 가시적으로 표시함으로서 작업자의 조작을 최소화하고, 생산성을 최대화할 수 있도록 하였다. GUI 환경의 제어기는 검사 프로그램을 사용하여 표준방식 키보드, 트랙볼, 엔코더, 아날로그 퍼텐시오미터, 2개의 디스플레이장치 및 터치스크린을 제어 할 수 있다.

1. Introduction

The ultrasound diagnostic device, JUPITER, has the USB hub which is connected to the A/N key compound device, track ball for low speed, special function key for full speed and touch panel. USB transfers will be implemented using the USB bulk transfer mode, which limits all packets to a maximum of 64 bytes.

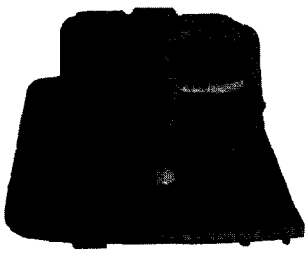
Examination program with GUI environment identifies the panel LED status with light and back light of LEDs, test the status of A/N keys and foot switches, examines the encoder by rotating between clockwise and count clockwise, displays the TGC bars in the VC++ window programs, and sequentially verifies the touch panel with dividing panel cell range into arithmetic values.

1) 정회원 : 서일대학 컴퓨터응용시스템 조교수

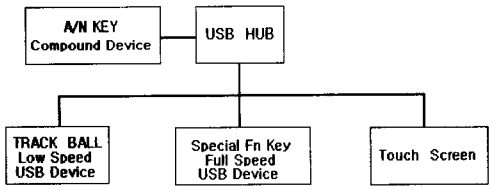
논문심사 : 2003. 4 .15
심사완료 : 2003. 5 .16

2. System configuration

The ultrasound system has a specified purpose keyboard, and has 3 USB devices, that is, USB Hub with A/N key, track ball and special function I/O devices which are switches, LEDs, encoders and variable re-sisters as like [Fig. 1]



[Fig. 1] Front panel View



[Fig. 2] Block diagram of system

Special key and others are Input/Output devices like USB full speed devices. They report switch status, volume position, encoder pulses and direction to the host, and they receive commands from host and indicating LEDs and operating mode of keys for example, repeat enable/disable key, simulation key or single key assignments. [Fig. 2]shows the block diagram of Ultra sound diagnostic device.

2.1 Software structure for diagnostic device

The diagnostic device connects via a single USB cable to the mother board. Host software is Win2000. The device contains switches, rotary encoders, and analog poten-tiometer group, two alpha-numeric displays and a touch screen : all of which generate events for the host. The panel-initiated events of diagnostic device are the switch events which contain depress, repeat, and release events, the encoder events which contain change and rest events, the analog potentiometer group which contains change and sending events with unsigned value scaled between 0 - 255, the touch screen which will send the current absolute XY location whenever a touch is detected, the XY position changes, and when a touch is released.

Above all events shall be queued in temporal order and sent to the host via USB, the host will accept the events and process them sequentially buffering when needed. When the host is sending commands to the front panel of diagnostic device, it is usually a burst of 20 or more. The button depression paradigm should not add more than five milliseconds delay no average.

2.2 USB device driver

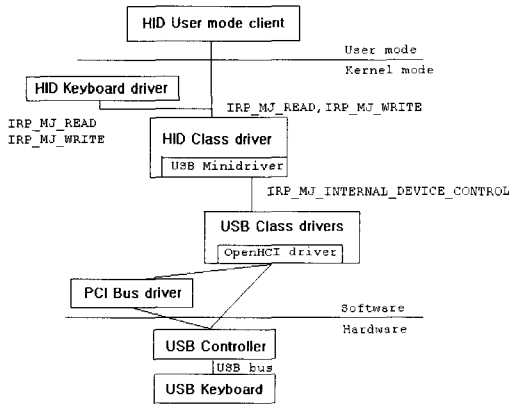
The USB driver shall queue any writes to the front panel to avoid waiting in app-lication code for any USB latencies. The USB driver shall initiate, execute, and release without hanging the operating system. The driver will fully support the CreateFile, ReadFile, WriteFile, and CancelIo calls as

documented by Microsoft. In addition, it will provide two new calls to support the management of the receive queue.

For the receiving side, on power up the application will call CreateFile to get a receive handle. It will then call panel-start-read-monitor to initiate the receive queue to 0. When the application wishes to terminate the read operation, it will call panel-stop-read-monitor, them CancelIo, and finally Close-Handle, calling panel-stop-read-monitor prior to calling panel-start-read-monitor will not be considered an error. The receive queue is of fixed size specified by two parameters defined in the **.ini file, readbuffer size. On the transmit side, the first request to send data to the device will initiate a call to CreateFile to get a write handle. Again, this handle will remain open throughout the life of the application. After insuring a valid handle, WriteFile will be called, and upon return the application may immediately reuse, destroy, or invalidate the contents of the input buffer to the WriteFile call.

The driver will insure the data integrity of the transfer without forcing the application to wait for the completion of the data transfer across the USB bus. There may be as many as 2000 calls in the output queue, so the transmit queue will not be statically sized. This queue shall be managed in such a manner as to avoid long-term fragmentation. CancelIo has no meaning an transmit side, but CloseHandle will terminate all pending transfers and return any memory to the system. If an error occurs during the transfer of a data packet, that data packet will be retried three times, at which time it will be thrown away and this error status set. This error status will be returned by the next call to WriteFile.

USB host control device confirms the status of keyboard, and therefore the status information is delivered to the USB root hub driver. Root hub driver informs the addition of new device to the PnP manager. HID keyboard driver requests the keyboard data by Read IRP. [Fig 3] shows the flow of infor- mation transfer procedure.



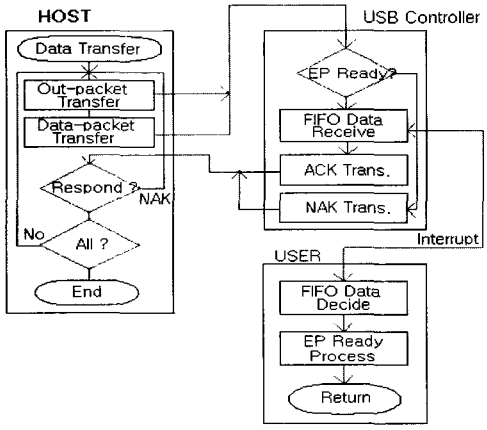
[Fig. 3] Flow of information transfer

3. System requirements

3.1 Software requirements

For switch events, the time between the declaration of the DepressEvent and the RepeatEvent is programmable as an individual value for each switch. For encoder events, the time between the first detected change and the first EncoderChangeEvent is programmable as a single value for all events. The time between the first Encoder-ChangeEvent and subsequent Encoder-ChangeEvents is individually programmable

for each encoder. The time from the last detected change in an encoder and when the EncoderRestEvent is sent is individually programmable for each encoder. The time between the first detected change and the reporting of the analog potentiometer group is programmable. The touch screen initial delay and repeat delay are programmable. The blink rate is programmable. The overall intensity of the indicators is programmable. The overall intensity of the keyboard backlighting is programmable. The programmable delays mentioned above shall be in integer units of 5 millisecond times. [Fig. 4] shows the data transfer mode between host and usb controller. The pipe is the way of data transfer between host and target.



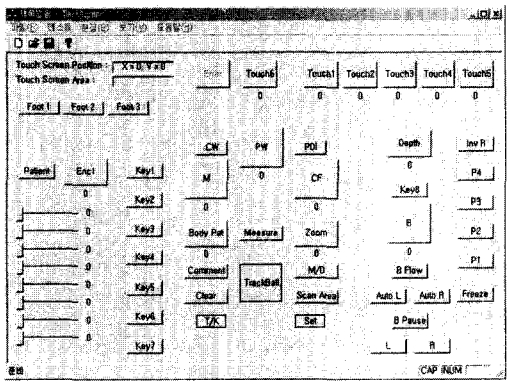
[Fig. 4] Data Transfer mode

3.2 Driver requirements

The USB driver shall be implemented such that no USB transfers from the front panel to the host are dropped. The generic driver shall block whenever the front panel attempts to send more imput than has been handled by the application. It will be the responsibility of the application to remove the inputs from the receiver in a timely manner. The generic driver shall send commands synchronously to the front panel. The generic driver will support obtaining synchronous read handles and write handles with the Windows CreatFile call. The generic driver needs to support three functional calls, read, write, and cancel. It would be preferable if these were implemented via IoCntrl calls.

4. Experiments

Ultrasound diagnostic device, JUPITER, is fully enable to examine the performance test, and decide whether it is normal operation or not. Test program of the device contain the examination of input, output, and communication units. Input examination verifies the actions of key, freeze key, foot switch, encoder, brightness, TGC, touch screen.



[Fig. 5] VC++ test program

Output examination verifies the actions of light and back light functions. Communication examination verifies the specification if USB1.1 under the local test condition. And therefore, software programs give the inspectors visible and audible test results. [Fig. 5] is the result of VC++ program.

5. Conclusions

In these works, USB interfacing is implemented for ultrasound diagnostic device which examines the performance by VC++ program like as [Fig. 5] The GUI program of USB IF is enable to control status of front panel for diagnostic device, to read a number of operation mode for switch groups, and to display the status of lights and back lights. Through these works, GUI environmental control designs are accomplished by the USB IF and VC++ tools. Afterwards, enhanced USB IF specification 2.0 is applied to the diagnostic device system.

Acknowledgement

This work is granted by 2002 year Seoil college research award.

※ References

[1] Visual C++ 6.0 Davis Chapman, SAMs, 1998.

[2] 알기 쉬운 Visual C++, 김태훈외, 사이버출판사, 2002.

[3] 오리지널 USB 어뎁터 제작, 월간전자기술, 2001.1 http://www.chomdan.co.kr

[4] Universal Serial Bus Specification Revision 1.1, 1998, http://www.usb.org

[5] Universal Serial Bus Device Class Definition for Human Interface Devices version 1.1 1999, http://www.usb.org

[6] Universal Serial Bus HID Usage Tables Version 1.1, 1999, http://www.usb.org

[7] Writing Windows WDM Device Drivers, 에이콘 출판사, 2001, Chris Cant.

[8] NEW 비쥬얼 C ++6 입문, 영진출판사, 2002, Haruhiko Hayashi

[9] Microsoft Taeyo's ASP, 삼영출판사, 2001. 8, 김태영

홍 선 학

1986. 2 광운대학교 전기과
졸업
1986. 3 한국전력공사 입사
1994. 3 서일대학 교수
1995. 2 광운대학교 박사학위
취득
1996. 3 중소기업청
기술지도 및 지원 사업
연구분야 : 디지털 제어계측 및
서보 제어분야