Future Technology Devices International Ltd.

Application Note

AN_163

Vinculum-II USB Slave Disconnect Detection

Document Reference No.: FT_000365

Version 1.0

Issue Date: 2010-11-30

This application note provides an example of how to detect when an FTDI Vinculum-II (VNC2) USB Slave device has been disconnected from USB. Sample source code is included for reference.

Future Technology Devices International Limited (FTDI)

Unit 1, 2 Seaward Place, Glasgow G41 1HH, United Kingdom
Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758
E-Mail (Support): **support1@ftdichip.com** Web: http://www.ftdichip.com

Table of Contents

1 Introduction

This application note describes how an application can detect that a Vinculum-II USB slave device has been disconnected from USB.

The sample source code for the application is provided as an example and is neither guaranteed nor supported by FTDI.

1.1 Overview

The method involves connecting the 5V bus power from the USB Slave to a GPIO input, and implementing an application thread to monitor the GPIO input.

The remainder of this application note describes a specific implementation of this method.

1.2 Hardware Requirements

- V2EVAL board
- 64-pin Vinculum II daughter board

See Appendix A for references to the datasheets.

2 Implementation

This section describes a specific implementation of disconnect detection. The reader should have a sound understanding of firmware application structure and the fundamental concepts of how the Vinculum Operating System (VOS) works. For an introduction to VinIDE, VOS and an overview of the general application structure please refer to *Vinculum-II_Tool_Chain_Getting_Started_Guide* [1]. The Vinculum-II tool-chain consists of a compiler, assembler, linker and debugger encapsulated in a graphical interface (the IDE) and is used to develop customised firmware for VNC2 [2].

2.1 Hardware Configuration

For the purpose of this application note, disconnect is detected on GPIO Port B pin 0 which is routed to IOBUS8 on the V2Eval Board [7]. A jumper wire is connected from VBUS Enable on the VII Slave JP4 to physically detect the VBUS from the USB Host.

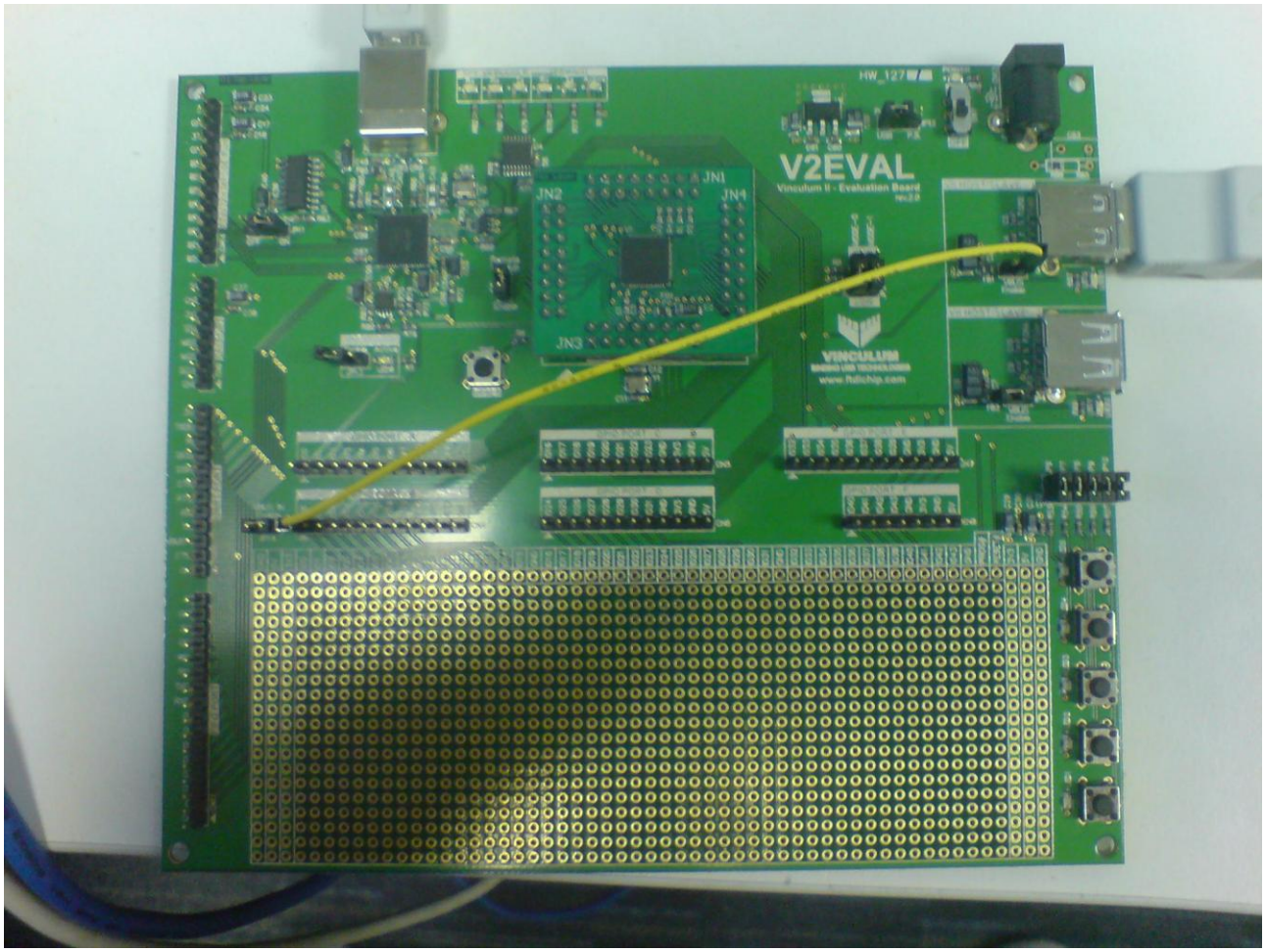

Figure 1: Connecting VBUS Enable to IOBUS8

2.2 Firmware Configuration

2.2.1 IOMux Configuration

Signals on VNC2 must be configured using the IOMux. In addition, FTDI provides an IOMux configuration utility, as part of the IDE, to aid with configuring IOMux signals. For more information on IOMux and the IOMux configuration utility, see [3] and [4].

For this implementation using a 64-pin VNC2, Pin 19 is defined as an input and is controlled by GPIO Port B 0. The following source code performs this configuration, and sets the IO cell characteristics for the pin. See [5] for more details.

```
// Route GPIO Port B 0 to pin 19
// this is used to detect a disconnect
vos_iomux_define_input(19, IOMUX_IN_GPIO_PORT_B_0);

// Also configure GPIO Port B 0 with a pull-down
vos_iocell_set_config(19,
                      VOS_IOCELL_DRIVE_CURRENT_4MA,
                      VOS_IOCELL_TRIGGER_NORMAL,
                      VOS_IOCELL_SLEW_RATE_FAST,
                      VOS_IOCELL_PULL_DOWN_75K);
```

2.2.2 GPIO Configuration

The VNC2 GPIO driver supports configurable interrupts. GPIO Port B is configured to interrupt when a state change occurs on pin 0. The following source code shows how to setup interrupt 0 on GPIO Port B to interrupt on the negative edge of the signal on Pin 0:

```
#define GPIOB    2    // user-defined device number

VOS_HANDLE hGpioB;

// Open GPIO port B
hGpioB = vos_dev_open(GPIOB);

// Set GPIO mask
gpio_iocb.ioctl_code = VOS_IOCTL_GPIO_SET_MASK;
gpio_iocb.value = 0x00; // all input
vos_dev_ioctl(hGpioB, &gpio_iocb);

// configure interrupt for GPIO B 0 for falling edge
gpio_iocb.ioctl_code = VOS_IOCTL_GPIO_SET_PROG_INT0_PIN;
gpio_iocb.value = GPIO_PIN_0;
vos_dev_ioctl(hGpioB, &gpio_iocb);

gpio_iocb.ioctl_code = VOS_IOCTL_GPIO_SET_PROG_INT0_MODE;
gpio_iocb.value = GPIO_INT_ON_NEG_EDGE;
vos_dev_ioctl(hGpioB, &gpio_iocb);

vos_enable_interrupts(VOS_GPIO_INT_IEN);
```

2.2.3 Monitoring the GPIO Input

An application thread is dedicated to monitoring the GPIO Port B 0 input signal. The thread blocks on a call to VOS_IOCTL_GPIO_WAIT_ON_INT0. When the USB Slave device is disconnected from the bus, the change in state of VBUS Enable causes an interrupt on GPIO Port B 0, and the thread is unblocked as the IOCTL completes.

The thread sends a VOS_IOCTL_USBSLAVE_DISCONNECT to the USB Slave driver to reset the USB Slave port. In this state, it is ready to be reconnected.

Finally, the thread calls VOS_IOCTL_GPIO_SET_PROG_INT0_MODE to re-enable the interrupt.

Note that the thread must be of sufficiently high priority that it will get to run when the GPIO interrupt has fired, and not be starved of CPU time by other higher priority threads. Since this thread will normally be blocked, it is safe to make this thread the highest priority in the application.

An implementation of a monitor thread is shown in the following source code:

```
void monitor_usb_disconnect(void) {

    gpio_ioctl_cb_t gpio_iocb;
    usbslave_ioctl_cb_t iocb;

    while (1) {
        // GPIO interrupt is configured - wait on detecting a disconnect
        gpio_iocb.ioctl_code = VOS_IOCTL_GPIO_WAIT_ON_INT0;
        vos_dev_ioctl(hGpioB, &gpio_iocb);

        // if we get here, we've been disconnected from the host!
        // put our slave back in a known state ready to be re-connected

        iocb.ioctl_code = VOS_IOCTL_USBSLAVE_DISCONNECT;
        iocb.set = (void *) 0;
        vos_dev_ioctl(hA,&iocb);

        // re-enable GPIO interrupt
        gpio_iocb.ioctl_code = VOS_IOCTL_GPIO_SET_PROG_INT0_MODE;
        gpio_iocb.value = GPIO_INT_ON_NEG_EDGE;
        vos_dev_ioctl(hGpioB, &gpio_iocb);
    }
}
```

3 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com
Web Site URL http://www.ftdichip.com
Web Shop URL http://www.ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan , R.O.C.
Tel: +886 (0) 2 8791 3570
Fax: +886 (0) 2 8791 3576

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com
Web Site URL http://www.ftdichip.com

Branch Office – Hillsboro, Oregon, USA

Future Technology Devices International Limited (USA)
7235 NW Evergreen Parkway, Suite 600
Hillsboro, OR 97123-5803
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) us.admin@ftdichip.com
Web Site URL http://www.ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited (China)
Room 408, 317 Xianxia Road,
Shanghai, 200051
China
Tel: +86 21 62351596
Fax: +86 21 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com
Web Site URL http://www.ftdichip.com

Distributor and Sales Representatives

Please visit the Sales Network page of the FTDI Web site for the contact details of our distributor(s) and sales representative(s) in your country.

4 Appendix A – References

Document References

[1] FTDI Application Note 142, *Vinculum-II_Tool_Chain_Getting_Started_Guide*, FTDI, 2010.

[2] FTDI Application Note 151, *Vinculum-II User Guide*, FTDI, 2010.

[3] FTDI Application Note 139, *IO_MUX Explained*, FTDI, 2010.

[4] FTDI Application Note 144, *Vinculum-II_IO_Mux_Config_Utility_User_Guide*, FTDI, 2010.

[5] FTDI Application Note 137, *Vinculum-II IO Cell Description*, FTDI, 2010.

[6] FTDI Datasheet, *Vinculum II Embedded Dual USB Host Controller IC*, Version 1.2, FTDI, 2010.

[7] FTDI Datasheet, *V2-EVAL Vinculum II Evaluation Board*, FTDI, 2010.

Acronyms and Abbreviations

Terms	Description
IOMux	Input Output Multiplexer – Used to configure pin selection on different package types of the VNC2.
V2EVAL	Vinculum II Evaluation Board- Customer evaluation board for the VNC2 allowing prototype development.
VOS	Vinculum Operating System
IDE	Integrated Development Environment
VNC2	Vinculum II

5 Appendix B – Revision History

Revision	Changes	Date
1.0	Initial Release	2010-11-30