Future Technology Devices International Ltd.

Application Note

AN_168

Vinculum-II USB Slave

Customizing an FT232 Device

Document Reference No.: FT_000395

Version 1.0

Issue Date: 2011-02-04

This application note provides an example of how to customize the descriptors of an FTDI Vinculum-II (VNC2) USB Slave FT232 device. Sample source code is included.

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 2
2 Descriptors 3
2.1 Device Descriptor 3
2.2 Configuration Descriptor 3
2.3 Interface Descriptor 4
2.4 Endpoint Descriptors 4
2.5 String Descriptors 5
2.5.1 Zero String 5
2.5.2 Manufacturer String 6
2.5.3 Product String 6
2.5.4 Serial Number String 7
2.5.5 Specifying String Descriptors 7
3 Modifying Descriptors 8
3.1 VOS_IOCTL_USBSLAVEFT232 _SET_DESCRIPTORS 8
4 Contact Information 10
5 Appendix A – References 12
Document References 12
Acronyms and Abbreviations 12
6 Appendix B – Revision History 13

1 Introduction

The function of a USB device is defined by its set of standard USB descriptors. By default, the USB Slave FT232 (FT232) driver uses standard descriptors for an FT232 device. However, the FT232 driver allows some features to be changed in order to customize a device, while still maintaining the functionality of an FT232 device.

This application note contains a description of the FT232 standard descriptors, and discusses the device features that can be customized.

The FT232 driver provides the request *VOS_IOCTL_USBSLAVEFT232_SET_DESCRIPTORS* to allow modification of the standard FT232 descriptors, and this request is defined fully in this note. This request will be supported in the FT232 driver from Vinculum II Development Tools v1.2.4 onwards.

The sample source code contained in this application note is provided as an example and is neither guaranteed nor supported by FTDI.

2 Descriptors

It is clear that, although descriptors can be modified, the device must retain its FT232 functionality. Therefore, only certain fields in the descriptors can be modified. In fact, the interface and endpoint descriptors cannot be changed at all. The remainder of this section details the descriptor fields that are configurable.

2.1 Device Descriptor

For an FT232 device, the default device descriptor is:

```
usb_deviceDescriptor_t
FT232_device_descriptor = {
       18,            // bLength
       1,             // bDescriptorType
       0x110,         // bcdUSB
       0,             // bDeviceClass
       0,             // bDeviceSubClass
       0,             // bDeviceProtocol
       8,             // bMaxPacketSize0
       0x0403,        // idVendor
       0x6001,        // idProduct
       0x0400,        // bcdDevice
       1,             // iManufacturer
       2,             // iProduct
       3,             // iSerialNumber
       1              // bNumConfigurations
};
```

For further details, see Section 9.6.1 in [1].

The fields in the device descriptor that can be configured are highlighted above: *idVendor*, *idProduct*, *iManufacturer*, *iProduct* and *iSerialNumber*.

The combination *idVendor*/*idProduct* represents the device identification; OEMs who have obtained reseller rights to FTDI drivers can safely change these fields to match their unique device identification. *bcdDevice* is the device release number; it is used by the FT232 driver on the host to determine the device type. *bcdDevice* is not configurable, and its default value 0x400 represents an FT232B device.

In principle, the string descriptor indexes, *iManufacturer*, *iProduct* and *iSerialNumber*, can be modified. But, for the implications of this, see section 2.5.5 below.

2.2 Configuration Descriptor

For an FT232 device, the default configuration descriptor is:

```
usb_deviceConfigurationDescriptor_t
FT232_configuration_descriptor = {
       9,             // bLength
       2,             // bDescriptorType
       0x0020,        // wTotalLength
       1,             // bNumInterfaces
       1,             // bConfigurationValue
       0,             // iConfiguration
       0x80,          // bmAttributes
       45             // bMaxPower
};
```

For further details, see Section 9.6.3 in [1].

The fields in the configuration descriptor that can be configured are highlighted above: *bmAttributes* and *bMaxPower*. *bmAttributes* is a bit map that contains the device configuration characteristics as follows:

Bit	**Characteristic**
D7	Reserved (set to one)
D6	Self-powered
D5	Remote Wakeup
D4..0	Reserved (reset to zero)

2.3 Interface Descriptor

For an FT232 device, the default interface descriptor is:

```
usb_deviceInterfaceDescriptor_t
FT232_interface_descriptor = {
       9,              // bLength
       4,              // bDescriptorType
       0,              // bInterfaceNumber
       0,              // bAlternateSetting
       2,              // bNumEndpoints
       0xff,           // bInterfaceClass
       0xff,           // bInterfaceSubClassClass
       0xff,           // bInterfaceProtocol
       2               // iInterface
};
```

For further details, see Section 9.6.5 in [1].

The interface descriptor is not configurable.

2.4 Endpoint Descriptors

An FT232 device has two endpoints: a BULK IN endpoint and a BULK OUT endpoint.

The default endpoint descriptor for the BULK IN endpoint is:

```
usb_deviceEndpointDescriptor_t
FT232_in_endpoint_descriptor = {
       7,              // bLength
       5,              // bDescriptorType
       0x81,           // bEndpointAddress
       2,              // bmAttributes
       0x0040,         // wMaxPacketSize
       0               // bInterval
};
```

The default endpoint descriptor for the BULK OUT endpoint is:

```
usb_deviceEndpointDescriptor_t
FT232_out_endpoint_descriptor = {
       7,             // bLength
       5,             // bDescriptorType
       0x02,          // bEndpointAddress
       2,             // bmAttributes
       0x0040,        // wMaxPacketSize
       0              // bInterval
};
```

For further details, see Section 9.6.6 in [1].

The endpoint descriptors are not configurable.

2.5 String Descriptors

An FT232 device has descriptors for zero, manufacturer, product and serial number strings. All string descriptors are configurable.

The FT232 driver defines a unique index for each string descriptor as follows:

String	Index	Encoding
Zero	0	FT232_STRING_INDEX_NONE
Manufacturer	1	FT232_STRING_INDEX_MANUFACTURER
Product	2	FT232_STRING_INDEX_PRODUCT
Serial Number	3	FT232_STRING_INDEX_SERIAL_NUMBER

The string index is encoded in the U*SBSLAVEFT232_STRING_DESCRIPTOR_INDEX* enumeration type in the header file USBSlaveFT232.h. All references to a string descriptor use the corresponding index; see for example the default device and interface descriptors above. So, although the default FT232 string descriptors can be replaced, the string indexes remain the same.

2.5.1 Zero String

By definition, this string has index zero and contains an array of language ID codes supported by the device. For further details, see Section 9.6.7 in [1]. The default zero string descriptor for an FT232 device is:

```
usb_deviceStringDescriptorZero_t
FT232_zero_string_descriptor = {
       4,             // bLength
       3,             // bDescriptorType
       0x0409         // LANGID code zero
};
```

In principle, the highlighted fields are configurable; in practice, however, it is unlikely that this descriptor would be modified.

2.5.2 Manufacturer String

The manufacturer string descriptor is a UNICODE string descriptor as defined in Section 9.6.7 in [1]. The default manufacturer string descriptor for an FT232 device is:

```
usb_deviceStringDescriptor_t
FT232_manufacturer_string_descriptor = {
        10,             // bLength
        3,              // bDescriptorType
        0x0046,         // bString
        0x0054,
        0x004d,
        0x0049
};
```

bString contains the UNICODE representation of the manufacturer name, "FTDI". The highlighted fields are configurable; *bLength* is a byte field whose value is 2 + the length in bytes of *bString*. All UNICODE strings must use UTF-16LE encoding.

2.5.3 Product String

The product string descriptor is a UNICODE string descriptor as defined in Section 9.6.7 in [1]. The default product string descriptor for an FT232 device is:

```
usb_deviceStringDescriptor_t
FT232_product_string_descriptor = {
        40,             // bLength
        3,              // bDescriptorType
        0x0056,         // bString
        0x004e,
        0x0043,
        0x0032,
        0x0020,
        0x0055,
        0x0053,
        0x0042,
        0x0020,
        0x003c,
        0x002d,
        0x003e,
        0x0020,
        0x0053,
        0x0065,
        0x0072,
        0x0069,
        0x0061,
        0x006c
};
```

bString contains the UNICODE representation of the product name, "VNC2 USB <-> Serial". The highlighted fields are configurable; *bLength* is a byte field whose value is 2 + the length in bytes of *bString*. All UNICODE strings must use UTF-16LE encoding.

2.5.4 Serial Number String

The serial number string descriptor is a UNICODE string descriptor as defined in Section 9.6.7 in [1]. The default serial number string descriptor for an FT232 device is:

```
usb_deviceStringDescriptor_t
FT232_serial_number_string_descriptor = {
        18,             // bLength
        3,              // bDescriptorType
        0x0031,         // bString
        0x0032,
        0x0033,
        0x0034,
        0x0035,
        0x0036,
        0x0037,
        0x0038
};
```

bString contains the UNICODE representation of the serial number, "12345678". The highlighted fields are configurable; *bLength* is a byte field whose value is 2 + the length in bytes of *bString*. All UNICODE strings must use UTF-16LE encoding.

2.5.5 Specifying String Descriptors

As stated above, string descriptors are referenced by an index that is encoded in the *USBSLAVEFT232_STRING_DESCRIPTOR_INDEX* enumeration type in USBSlaveFT232.h. This means that string indexes cannot be changed, although the strings that the indexes refer to can be changed.

This has implications for the configurability of the string descriptor indexes in the device descriptor, described in section 2.1. It is possible to have a device that does not support string descriptors; in this case all references to string descriptors must be set to zero. While this extreme case is an unlikely scenario for an FT232 device, it may be that individual string descriptors are not supported (for example, a device may not have a serial number). So, in practice, the string descriptor index fields in the device descriptor can take on only certain values as follows:

String Descriptor Index	Value
iManufacturer	1 or 0
iProduct	2
iSerialNumber	3 or 0

This shows that a device does not have to support manufacturer and serial number string descriptors. However, the product string descriptor must be supported, since it is referenced by the interface descriptor (section 2.3) which itself is not configurable.

3 Modifying Descriptors

An application modifies the standard descriptors of anFT232 device by calling the IOCTL request *VOS_IOCTL_USBSLAVEFT232_SET_DESCRIPTORS* before the device has been enumerated by a host. If this request is not called, then the device defaults to the standard descriptors described in section 2.

3.1 VOS_IOCTL_USBSLAVEFT232 _SET_DESCRIPTORS

Description

This function modifies the descriptors of an FT232 device.

Parameters

A *usbslaveft232_ioctl_cb_descriptors_t* structure must be initialised and passed in the *set* member of the *common_ioctl_cb_t* structure.

The *usbslaveft232_ioctl_cb_descriptors_t* structure is defined as:

```
typedef struct _usbslaveft232_ioctl_cb_descriptors_t {
        struct {
               unsigned char use;
               unsigned short idVendor;
               unsigned short idProduct;
               unsigned char iManufacturer;
               unsigned char iProduct;
               unsigned char iSerialNumber;
        } device_descriptor;
        struct {
               unsigned char use;
               unsigned char bmAttributes;
               unsigned char bMaxPower;
        } config_descriptor;
        usb_deviceStringDescriptorZero_t *zero_string;
        usb_deviceStringDescriptor_t *manufacturer_string;
        usb_deviceStringDescriptor_t *product_string;
        usb_deviceStringDescriptor_t *serial_number_string;
} usbslaveft232_ioctl_cb_descriptors_t;
```

where,

> *device_descriptor* is a structure containing values for the *idVendor*, *idProduct*, *iManufacturer*, *iProduct* and *iSerialNumber* fields in the device descriptor, as defined in section 2.1. If the *use* field is non-zero, these values modify the standard device descriptor; otherwise the standard device descriptor remains unchanged. Note that the *use* field enables all values in this structure, so they must all be initialised with the required values.
>
> *config_descriptor* is a structure containing values for the *bmAttributes* and *bMaxPower* fields in the configuration descriptor, as defined in section 2.2. If the *use* field is non-zero, these values modify the standard configuration descriptor; otherwise the standard configuration descriptor remains unchanged. Note that the *use* field enables all values in this structure, so they must all be initialised with the required values.
>
> *zero_string* is a pointer to a zero string descriptor, as defined in section 2.5.1.
>
> *manufacturer_string* is a pointer to a manufacturer string descriptor, as defined in section 2.5.2.
>
> *product_string* is a pointer to a product string descriptor, as defined in section 2.5.3.
>
> *serial_number_string* is a pointer to a serial number string descriptor, as defined in section 2.5.4.

Returns

There is no data returned by this call, although the return code indicates the success or otherwise of the operation.

Example

The following code fragment demonstrates how to call *VOS_IOCTL_USBSLAVEFT232_SET_DESCRIPTORS* to modify the device and configuration descriptors, and the manufacturer string descriptor:

```
VOS_HANDLE hFT232;
common_ioctl_cb_t common_cb;
usbslaveft232_ioctl_cb_descriptors_t descriptors_cb;
unsigned char manufacturer_acme[10] = {
        10,
        USB_DESCRIPTOR_TYPE_STRING,
        0x41, 0x00,
        0x43, 0x00,
        0x4d, 0x00,
        0x45, 0x00
};

// open FT232BM
hFT232 = vos_dev_open(USBSFT232);

// initialize request control block
vos_memset(&descriptors_cb,0,sizeof(usbslaveft232_ioctl_cb_descriptors_t));

// set device descriptor in request control block
descriptors_cb.device_descriptor.idVendor = USB_VID_FTDI;
descriptors_cb.device_descriptor.idProduct = 0x9876;
descriptors_cb.device_descriptor.iManufacturer = FT232_STRING_INDEX_MANUFACTURER;
descriptors_cb.device_descriptor.iProduct = FT232_STRING_INDEX_PRODUCT;
descriptors_cb.device_descriptor.iSerialNumber = 0;
descriptors_cb.device_descriptor.use = 1;

// set configuration descriptor in request control block
descriptors_cb.config_descriptor.bmAttributes = 0xa0;
descriptors_cb.config_descriptor.bMaxPower = 0x2d;
descriptors_cb.config_descriptor.use = 1;

// set manufacturer string descriptor in request control block
descriptors_cb.manufacturer_string =
        (usb_deviceStringDescriptor_t *) manufacturer_acme;

common_cb.ioctl_code = VOS_IOCTL_USBSLAVEFT232_SET_DESCRIPTORS;
common_cb.set.data = &descriptors_cb;
vos_dev_ioctl(hFT232,&common_cb);
```

4 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.

System and equipment manufacturers and designers are responsible to ensure that their systems, and any Future Technology Devices International Ltd (FTDI) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested FTDI devices and other materials) is provided for reference only. While FTDI has taken care to assure it is accurate, this information is subject to customer confirmation, and FTDI disclaims all liability for system designs and for any applications assistance provided by FTDI. Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, United Kingdom. Scotland Registered Company Number: SC136640

5 Appendix A – References

Document References

[1] *Universal Serial Bus Specification Revision 2.0*, USB Implementers Forum, 2000. Available from http://www.usb.org/developers/docs/

Acronyms and Abbreviations

Terms	Description
FT232	VNC2 USB Slave FT232 function driver.
VNC2	Vinculum II
VOS	Vinculum Operating System

6 Appendix B – Revision History

Revision	Changes	Date
1.0	Initial Release	2011-02-04