

HID oder „Die Sendung mit der Maus“

Eine HID (*Human Interface Device class*) soll programmiert werden, um auch ohne libusb Daten austauschen zu können. Bei der Auslieferung des AT90USBKEY ist von ATMEL eine Maus-Imitation implementiert. Dies soll hier imitiert und implementiert werden ;-).

Die ursprüngliche Bibliothek verwendet Bulk-Transfers (anwenderspezifische Klasse) um über Endpunkte Daten mit dem PC mit Hilfe der libusb auszutauschen.

Bei der HID-Klasse können dazu Interrupt-Transfers verwendet werden. Der Unterschied besteht darin, dass bei Interrupt-Transfers die Daten in einem bestimmten Zeitraum übertragen werden (Bei Bulk-Transfers hängt die Geschwindigkeit der Bearbeitung von der Busbelegung ab; es wird keine Bandbreite reserviert.) Der PC (Host) fragt in festen Zeitabständen (im Deskriptor festgelegt) das Gerät ab.

In der HID-Klasse können auch über Reports (GET_REPORT) Daten über Endpunkt 0 gesendet und empfangen werden (keine reservierte Bandbreite). Dieser Datenaustausch wird hier nicht verwendet.

Die HID-Klasse muss die Interrupt-IN-Übertragung über einen Endpunkt zum Senden von Daten an den Host unterstützen. Die Klasse wird auf Interface-Level definiert.

Zu den bekannten Deskriptoren muss ein HID-Deskriptor und ein Report-Deskriptor hinzugefügt werden.

Änderungen an den Deskriptoren:

Geräte-Deskriptor

Klasse, Unterklasse und Protokoll sind nicht mehr anbieterspezifisch (0xFF) sondern werden mit Null (0x00) initialisiert. Null bedeutet, dass diese Größen erst im Schnittstellen (Interface)-Deskriptor definiert werden.

Feldbezeichnung	Byte	Wert	Beschreibung
bDeviceClass	1	0x00	Klassencode (auf Interface-Level definiert! = 0x00)
bDeviceSubClass	1	0x00	Unterklassencode (auf Interface-Level definiert! = 0x00)
bDeviceProtocoll	1	0x00	Protokollcode (auf Interface-Level definiert! = 0x00)

Konfigurations-Deskriptor

Die Gesamtlänge (Feld **wTotalLength**) wird um die Größe des HID-Deskriptor (9 Byte) erhöht (9 Byte (1 Konfigurations-Deskriptor) + 9 Byte (1 Schnittstellen-Deskriptor) + 9 Byte (1 HID-Deskriptor) + 2*7 Byte (2 Endpunkte (ohne Endpunkt 0!)) = 41 Byte.

Feldbezeichnung	Byte	Wert	Beschreibung
-----------------	------	------	--------------

wTotalLength	2	41 (0x29)	Länge des Konfigurations-Deskriptors und aller untergeordneter Deskriptoren
--------------	---	-----------	---

Schnittstellen (Interface)-Deskriptor

Als Klasse wird im Interface-Deskriptor HID definiert und das Protokoll für eine Maus angegeben.

Feldbezeichnung	Byte	Wert	Beschreibung
bInterfaceClass	1	0x03	Klassencode (HID = 0x03)
bInterfaceSubClass	1	0x00	Unterklassencode (keine Unterklasse = 0x00)
bInterfaceProtocol	1	0x02	Protokollcode (Maus = 0x02)

Neu! Der HID-Deskriptor

Als neuer Deskriptor kommt der HID Deskriptor dazu. Er besitzt zwingend als untergeordneten Deskriptor einen Bericht (Report)-Deskriptor. Dieser wird allerdings nicht gemeinsam mit dem Konfigurations-Deskriptor übertragen sondern gesondert bei einer Report-Anfrage des Host.

Feldbezeichnung	Byte	Wert	Beschreibung
bLength	1	9	Größe des Deskriptors in Byte
bDescriptorType	1	33 (0x21)	die HID-Klasse hat die Nummer 33 (0x21)
bcdHID	2	0x0110	nach HID Spezifikation 1.1
bCountryCode	1	0	wird hier nicht unterstützt
bNumDescriptors	1	1	Anzahl der untergeordneten Deskriptoren = 1 Der Bericht (Report)-Deskriptor ist obligatorisch.
bDescriptorType	1	34 (0x22)	Report-Deskriptor 0 34 (0x22) Dieser Deskriptor ist obligatorisch!
wDescriptorLength	2	0x0034	Länge des Report-Deskriptor (hier 52 Byte)

Endpunkt-Deskriptor

Zwingend ist ein IN-Interrupt-Endpunkt erforderlich.

Feldbezeichnung	Byte	Wert	Beschreibung
bmAttributes	1	3	Transfertyp = Interrupt (contr. = 0, iso. = 1, bulk = 2)
bInterval	1	1	Polling Interval = 1 (1-255) Bei Interrupt: Latenzzeit in Millisekunden (Host fragt jede Millisekunde nach)

Neu! Der Bericht (Report)-Deskriptor

Bei der GET_DESCRIPTOR-Anforderung des PC ein muss in der Software ein neuer Fall hinzugefügt werden um den Report Deskriptor senden zu können. Sendet der PC im SETUP-Paket 34 (0x22) in wValue_h so fordert er damit den Bericht-Deskriptor an.

Im Bericht-Deskriptor wird festgelegt, wie die Informationen im Endpunkt-FIFO abgelegt werden. Der Endpunkt bildet in der Maus einen gemeinsam vom PC und der Maus einsehbaren Speicherbereich, der innerhalb eines garantierten Zeitabstandes vom Computer ausgelesen wird.

In der Firmware wird ein Standard-HID-Bericht für eine Maus mit drei Tasten und Scroll-Rad verwendet. Um solche Berichte zu erstellen bietet der USB Implementers Forum (USB-IF, www.usb.org) eine Windows Software als Hilfe an. (**HID Descriptor Tool**, www.usb.org/developers/hidpage). Diese wird mit einigen Standard-Berichten als Beispiel geliefert.

Im Bericht gibt das erste Byte die Größe an und das zweite Byte den Wert. Der benutzte Bericht definiert wie die 4 Byte die im Fifo des IN-Endpunkts abgelegt werden zu interpretieren sind:

1 Byte	3 Bit für die drei Tasten	0b00000xxx (2 ⁰ = Linke Taste)
2 Byte	signed (-127 bis 127) X	relativer Wert
3 Byte	signed (-127 bis 127) Y	relativer Wert
4 Byte	signed (-127 bis 127) Wheel	relativer Wert
USAGE_PAGE (Generic Desktop)	05 01	//allgemeine Funktion des Gerätes //hier Generic Desktop Controls
USAGE (Mouse)	09 02	//Funktion eines einzelnen Berichtes //in der Usage Page (hier Maus)
COLLECTION (Application)	A1 01	//Beginn einer Gruppe von Einträgen //die die Funktion bilden
USAGE (Pointer)	09 01	//
COLLECTION (Physical)	A1 00	//Gruppe von Einträgen
USAGE_PAGE (Buttons)	05 09	//Maustasten
USAGE_MINIMUM (Button 1)	19 01	//mindestens eine
USAGE_MAXIMUM (Button 3)	29 03	//maximal 3
LOGICAL_MINIMUM (0)	15 00	//Legen Bereich für die Werte fest die
LOGICAL_MAXIMUM (1)	25 01	//im Bericht enthalten sind (binär)
REPORT_SIZE (1)	75 01	//Anzahl der Bits pro Datenelement 1Bit
REPORT_COUNT (3)	95 03	//Anzahl der Datenelemente im Bericht 3
INPUT (Data,Var,Abs)	81 02	//INPUT Bericht, Werte absolut
REPORT_SIZE (5)	75 05	//5 Bits (füllt Byte?)
REPORT_COUNT (1)	95 01	//1*
INPUT (Cnst,Var,Abs)	81 01	//Konstantes Feld
USAGE_PAGE (Generic Desktop)	05 01	//Mauszeiger
USAGE (X)	09 30	//48
USAGE (Y)	09 31	//49
USAGE (Wheel)	09 38	//56
LOGICAL_MINIMUM (-127)	15 81	//-127 bis 127
LOGICAL_MAXIMUM (127)	25 7F	//
REPORT_SIZE (8)	75 08	//8 Bits
REPORT_COUNT (3)	95 03	//3*8 Bits (X,Y,Wheel)
INPUT (Data,Var,Rel)	81 06	//Werte relativ!

```
END_COLLECTION          C0
END_COLLECTION          C0
```

Unter Linux kann man sich mit **lsusb** den den Bericht-Deskriptor eines Gerätes anzeigen lassen.

Vorgehensweise:

Gerät einstecken, und dann mit **dmesg** Gerätenummer rsausfinden

```
[ 2000.028357] input: ATMEL AVR USB KEY COMPOSITE DEVICE as
/devices/pci0000:00/0000:00:04.1/usb2/2-1/2-1.2/2-1.2:1.0/input/input4
```

Die Gerätenummer besteht aus:

Bus-Portnummer (string mit Punkt als Trenner):Konfiguration.Interface

Hier: **2-1.2:1.0**

Gerät muss als root losgebunden werden, damit der Deskriptor angezeigt wird:

```
sudo bash -c "echo -n 2-1.2:1.0 > /sys/bus/usb/drivers/usbhid/unbind"
```

Dann **lsusb -v -s [[bus]:][devnum]** oder **lsusb -d vendor:[product]**

Mit bind statt unbind dann wieder einbinden.

Weitere Änderungen an der Firmware

Im Hauptprogramm wurden die Verweise auf den Namen der Bibliothek durch den neuen Namen ersetzt (in C Anpassung der Header-Datei und der Namen im Makefile). Die fünf Eingangspins des Joystick (PB5-7, und PE4-5) wurden als Eingang mit Pull-Up initialisiert.

Die Namen der Endpunktunterprogramme wurden verändert.

Alle oben angegebenen Änderungen der Deskriptoren wurden vorgenommen. Die GET_DESCRIPTOR-Abarbeitung wurde um einen Eintrag erweitert (0x22 (34) in wValue_h).

Der Endpunkttyp der Endpunkte 1 und 2 wurden auf 3 (Interrupt) gesetzt.

Da nur der IN-Endpunkt (EP2) genutzt wird wurde dieser verändert. Es werden 4 Byte ins FIFO geladen. Das erste Byte ist null, wenn der Taster (Linke Maustaste) nicht gedrückt wurde (sonst 0x01). Wird der Joystick nach Links bewegt, so wird wird 1 von X abgezogen (Zeierkomplement: 0xFF = -1). Nach Rechts wird 1 addiert (0x01). In Mittelstellung wird 0x00 in die FIFO-Speicherzelle geschrieben. Analog wird bei Rauf abgezogen und bei Runter addiert und in Mittelstellung 0x00 eingetragen. Das vierte Byte ist Null, da kein Scrollrad vorhanden ist (und kein weiterer Taster zur Verfügung steht). Alle Taster sind nullaktiv (negative Logik).