USB Controlled Stepper Motor Interface



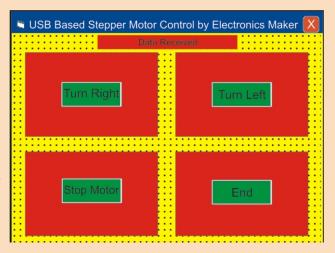
MUDIT AGARWAL

The Universal Serial Bus is one of the most common interfaces used in electronic consumer products today, including PCs, cameras, GPS devices, MP3 players, modems, printers, and scanners etc. The USB was originally developed by Compaq, Microsoft, Intel, and NEC, and later by Hewlett-Packard, Lucent, and Philips as well. These companies eventually formed the nonprofit corporation USB Implementers Forum Inc. to organize the development and publication of USB specifications. In this article we shows how to control a stepper motor with a USB port of a PC with the help of PIC microcontroller.

The USB is a high-speed serial interface that can also provide power to devices connected to it. A USB bus supports up to 127 devices connected through a four-wire serial cable of up to three or even five meters in length. Many USB devices can be connected to the same bus with hubs, which can have 4, 8, or even 16 ports. A device can be plugged into a hub which is plugged into another hub, and so on.

USB Signal

USB signals are bi-phase, and signals are sent from the host computer using the NRZI data encoding technique. In this technique, the signal levels inverted for each change to a logic 0. The signal level for a logic 1 is not changed. A 0 bit is "stuffed" after every six consecutive ones in the data stream to make the data dynamic (this is called bit stuffing because the extra bit lengthens the data stream). A packet of data transmitted by the host is sent to every device connected to the bus, traveling downward through the chain of hubs. All the devices receive the signal, but only one of them, the addressed one, accepts the data. Conversely, only one device at any time can transmit to the host, and the data travels upward through the chain of hubs until it reaches the host. USB devices attached to the bus may be full-custom devices, requiring a full-custom device driver, or they may belong to a device class.



Deviced classes enable the same device driver to be used for several devices having similar functionalities. For example, a printer device has the device class 0x07, and most printers use drivers of this type.

Endpoint: An endpoint is either a source or a sink of data. A single USB device can have a number of endpoints, the limit being sixteen IN and sixteen OUT endpoints.

Transaction: A transaction is a transfer of data on the bus.

Pipe: A pipe is a logical data connection between the host and an endpoint.

USB States

Idle: The bus is in idle state when the pulled-up line is high and the other line is low. This is the state of the lines before and after a packet transmission.

Detached: When no device is connected to the bus, the host sees both lines as low.

Attached: When a device is connected to the bus, the host sees either D+ or D_ go to logic high, which means a device has been plugged in.

J state: The same as idle state. K state: The opposite of J state.

SEO: The single ended zero state, where both lines

on the bus are pulled low.

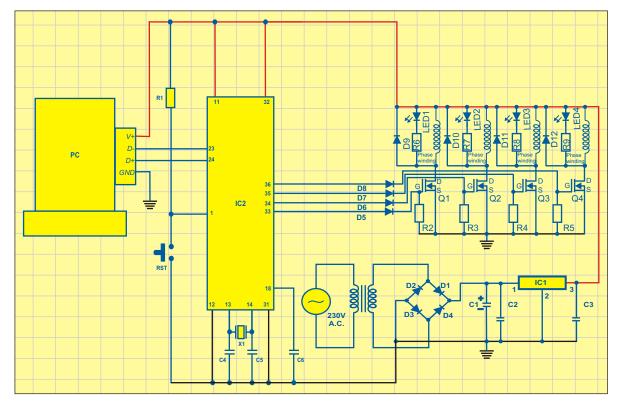


Fig. 1. Circuit Diagram of USB controlled Stepper Motor Interface

SE1: The single ended one state, where both lines on the bus are high. SE1 is an illegal condition on the bus; it must never be in this state.

Reset: When the host wants to communicate with a device on the bus, it first sends a "reset" condition by pulling low both data lines (SEO state) for at least 10ms.

EOP: The end of packet state, which is basically an SEO state for 2 bit times, followed by a J state for 1 bit time.

Keep alive: The state achieved by EOP. Keep alive is sent at least once every millisecond to keep the device from suspending.

Suspend: Used to save power, suspend is implemented by not sending anything to a device for 3ms. A suspended device draws less than 0.5mA from the bus and must recognize reset and

Sync	PID	ADDR	ENDP	CRC	EOP
	8 bits	7 bits	4 bits	5 bits	

resume signals.

Resume: A suspended device is woken up by reversing the polarity of the signal on the data lines for at least 20ms, followed by a low-speed EOP signal.

Data Transmission

Data is transmitted on a USB bus in packets. A packet starts with a sync pattern to allow the receiver clock to synchronize with the data. A packet identifier (PID) byte immediately follows the sync field of every USB packet. A PID itself is 4 bits long, and the 4 bits are repeated in a complemented form. There are seventeen different PID values There are four packet formats, based on

PID Type	PID name	Bits	Description	
Token	OUT	11100001	Host to device transaction	
	IN	01101001	Device to host transaction	
	SOF	10100101	Start of frame	
	SETUP	00101101	Setup command	
Data	DATA0	11000011	Data packet PID even	
	DATA1	01001011	Data packet PID odd	
	DATA2	10000111	Data packet PID high speed	
	MDATA	00001111	Data packet PID high speed	
Handshake	ACK	11010010	Receiver accepts packet	
	NAK	01011010	Receiver does not accept packet	
	STALL	00011110	Stalled	
	NYET	00111100	No response from receiver	
Special	PRE	00111100	Host preample	
	ERR	01111000	Split transaction error	
	SPLIT	01111000	High speed split transaction	
	PING	10110100	High speed flow control	
	Reserved	11110000	Reserved	

which PID is at the start of the packet: token packets, data packets, handshake packets, and special packets.

Enumeration

When a device is plugged into a USB bus, it becomes known to the host through a process

called enumeration. The steps of enumeration are, When a device is plugged in, the host becomes aware of it. The host sends a USB reset signal to the device to place the device in a known state. The reset device responds to address 0. The host sends a request on address 0 to the device to find out its maximum packet size using a Get Descriptor command. The device responds by sending a small portion of the device descriptor. The host sends a USB reset again. The host assigns a unique address to the device and sends a Set Address request to the device. After the request is completed, the device assumes the new address. At this point the host is free to reset any other newly plugged-in devices on the bus. The host sends a Get Device Descriptor request to retrieve the complete device descriptor, gathering information such as manufacturer, type of device, and maximum control packet size. The host sends a Get Configuration Descriptors request to receive the device's configuration data, such as power requirements and the types and number of interfaces supported. The host may request any additional descriptors from the device.

The most common USB descriptors are:

Device descriptors, Configuration descriptors
Interface descriptors, HID descriptors, Endpoint descriptors. Here i cannot discuss all these things because of space limitation.

Fig. 2. PCB Layout of (92% of actual size) and RCS

PIC18F4550 microcontroller is use for USB communication with PC. PORTC pins RC4 (pin 23)

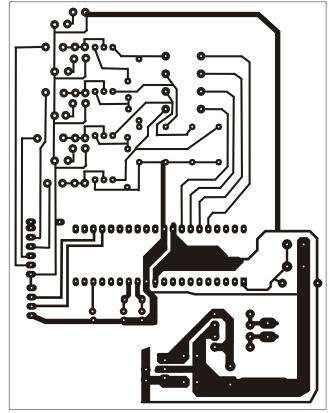


Fig. 2. PCB Layout of USB controlled Stepper Motor Interface. (92% of actual size)

and RC5 (pin 24) are used for USB interface. RC4 is the USB data D_ pin, and RC5 is the USB data D+ pin. Internal pull-up resistors are provided which

can be disabled (setting UPUEN =0) if desired and external pullup resistors can be used instead. For full-speed operation an internal or external resistor should be connected to data pin D+, and for low-speed operation an internal or external resistor should be connected to data pin D-. Operation of the USB module is configured using three control registers, and a total of twenty-two registers are used to manage the actual USB transactions The operation of the USB link requires the microcontroller to keep the connection alive by sending keep-alive messages to the PC every several milliseconds. This is achieved by setting up a timer interrupt service routine using

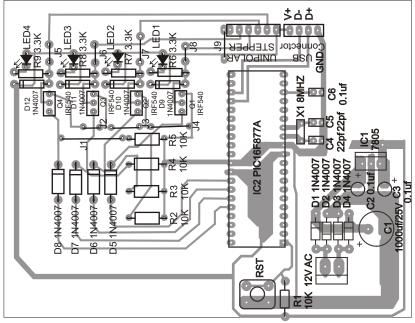


Fig. 3.Component Layout of USB controlled Stepper Motor Interface. (92% of actual size)

TIMER O. Timer TMROL is reloaded and timer interrupts are re-enabled just before returning from the interrupt service routine. Inside the main program PORTB is defined as digital I/O and TRISB is cleared to 0 so all PORTB pins are outputs. All the interrupt registers are then set to their power-onreset values for safety. The timer interrupts are then set up. The timer is operated in 8-bit mode with a prescaler of 256. Although the crystal clock

frequency is 8MHZ, the CPU is operated with a 48MHz clock, by enabling its PLL. Selecting a timer value of TMROL = 100 with a 48MHz clock gives timer interrupt intervals of 3.3ms.

Circuit Diagram of USB controlled Stepper Motor Interface is show in fig. 1. PCB layout of USB controlled Stepper Motor Interface is show in fig. 2. Component layout of USB controlled Stepper Motor Interface is show in fig. 3.

The Software Running On Microcontroller

```
Unsigned char Read buffer[64]:
unsigned char Write buffer[64];
unsigned charnum;
unsigned char const HID_INPUT_REPORT BYTES
unsigned char const HID_OUTPUT_REPORT_BYTES
unsigned char const HID_FEATURE REPORT BYTES
unsigned char const NUM ENDPOINTS
unsigned char const ConfigDescr_wTotalLength =
USB_CONFIG_DESCRIPTOR_LEN +
USB_INTERF_DESCRIPTOR_LEN
USB_HID_DESCRIPTOR_LEN +
(NUM_ENDPOINTS *
USB_ENDP_DESCRIPTOR_LEN);
unsigned char const HID_ReportDesc_len
unsigned char const Low_HID_ReportDesc_len =
HID ReportDesc len;
unsigned char const High_HID_ReportDesc_len
HID ReportDesc len >> 8;
unsigned char const Low_HID_PACKET_SIZE
unsigned char const High_HID_PACKET_SIZE
HID PACKET SIZE >> 8;
unsigned char const
 DeviceDescr[USB_DEVICE_DESCRIPTOR_LEN*2]
USB_DEVICE_DESCRIPTOR_LEN, 0,
USB_DEVICE_DESCRIPTOR_TYPE, 0,
0x00,0,0x02,0,0x00,0,0x00,0,0x00,0,
EPO PACKET SIZE, 0,0x34, 0,0x12, 0,0x01, 0,
0x00.0
0x01, 0,0x00, 0, 0x01, 0, 0x02, 0,0x00, 0,0x01, 0
 ConfigDescr[USB_CONFIG_DESCRIPTOR_LEN*2]
USB CONFIG DESCRIPTOR LEN, O,
USB_CONFIG_DESCRIPTOR_TYPE, 0,
ConfiaDescr wTotalLenath, 0,
0x00, 0, 0\overline{x01}, 0, 0x01, 0, 0x00, 0, 0xA0, 0, 50, 0
```

```
unsigned char const
 InterfaceDescr[USB_INTERF_DESCRIPTOR LEN*2]
USB INTERF DESCRIPTOR LEN, O,
USB INTERFACE DESCRIPTOR TYPE, 0,
0x00, 0, 0x00, 0, NUM_ENDPOINTS, 0,0x03, 0,
0x00, 0, 0x00, 0, 0x00, 0;
unsigned char const
 HID Descriptor[USB HID DESCRIPTOR LEN*2] =
USB HID DESCRIPTOR LEN, O,
USB HID DESCRIPTOR TYPE, O
0x0\overline{1}, 0, \overline{0}x01, 0, 0x00, \overline{0}, 0x01, 0, 0x22, 0,
        HID_ReportDesc_len,
0,High_HID_ReportDesc_len, 0};
unsigned char const
 EP1 RXDescr[USB ENDP DESCRIPTOR LEN*2] =
USB_ENDP_DESCRIPTOR_LEN, 0,
USB_ENDPOINT_DESCRIPTOR_TYPE, 0,
0x81, 0, USB_ENDPOINT_TYPE_INTERRUPT, 0,
Low_HID_PACKET_SIZE, 0, High_HID_PACKET_SIZE, 0, 0 };
unsigned char const
 EP1 TXDescr[USB ENDP DESCRIPTOR LEN*2] =
ÙSB_ENDP_DESCRIPTOR_LEN, 0,
USB_ENDPOINT_DESCRIPTOR_TYPE, 0,
0x01, 0,USB ENDPOINT TYPE INTERRUPT, 0,
Low_HID_PACKET_SIZE, 0,HID_PACKET_SIZE, 0,1,
unsigned char const
HID ReportDesc[HID ReportDesc len*2] = {0x06,
0,0xA0,0,
0xFF, 0,0x09, 0,0x01, 0,0xA1, 0,0x01, 0,0x09,
0,0x03,\ 0,0x15,\ 0,0x00,\ 0,0x26,\ 0,0x00,\ 0,0xFF,
0,0x75,0,0x08,0,0x95,0,
HID INPUT REPORT BYTES, 0,0x81, 0, 0x02,
0.0 \times 09.0
0x04, 0,0x15, 0,0x00, 0,0x26, 0, 0x00, 0,0xFF,
0x08, 0,0x95, 0, HID OUTPUT REPORT BYTES, 0,
0x91, 0, 0x02, 0,0x09, 0, 0x05, 0,0x15, 0,0x00,
0.0x26.0.0x00.0.0xFF.0.0x75.0.0x08.0.0x95.
O,HID_FEATURE_REPORT_BYTES,O,0xB1, O, 0x02,
0.0xC0.0}:
```

```
\label{eq:unsigned} \begin{array}{l} unsigned\ char\ const\ LangIDDescr[8] = \{0x04,0,\\ USB\_STRING\_DESCRIPTOR\_TYPE,\ 0,0x09,\ 0,0x04,\\ \end{array}
unsigned char const ManufacturerDescr[24] = {12,
USB STRING DESCRIPTOR TYPE, 0, 'M', 0, 0, 0,
'u',0,0,0, 'd',0,0,0, 'i',0,0,0, 't',0,0,0};
USB_STRING_DESCRIPTOR_TYPE, 0, 'A', 0, 0, 0,
 'g', 0, 0, 0, 'a', 0, 0, 0, 'r', 0, 0, 0, 'w', 0, 0, 0, 'a', 0, 0,
0, 1, 0, 0, 0;
unsigned char const StrUnknownDescr[4] = {
2,0,USB STRING DESCRIPTOR TYPE, 0};
void interrupt()
{HID_InterruptProc(); TMROL = 100;
INTCON.TMR0IF = 0; }
void InitUSBdsc()
\{Byte\_tmp\_0[0] = NUM\_ENDPOINTS;
Byte_tmp_0[0] = ConfigDescr_wTotalLength;
Byte_tmp_0[0] = HID_ReportDesc_len;
Byte tmp 0[0] = Low HID ReportDesc len;
Byte_tmp_0[0] = High_HID_ReportDesc_len;
Byte_tmp_0[0] = Low_HID_PACKET_SIZE;
Byte_tmp_0[0] = High_HID_PACKET_SIZE;
DeviceDescr;ConfigDescr;InterfaceDescr;HID Desc
EP1_RXDescr;EP1_TXDescr;HID_ReportDesc;
LangIDDescr;ManufacturerDescr;ProductDescr;
StrUnknownDescr:}
void main()
\{ADCON^{\dagger} = 0xFF;TRISB = 0; PORTB = 0;
INTCON=0;
INTCON2=0xF5;RCON.IPEN=0;PIE1=0;PIE2=0;
PIR1=0;
PIR2=0;TOCON = 0x47; TMROL = 100;
INTCON.TMR0IE = 1;
T0CON.TMR0ON = 1; INTCON = 0xE0;
Hid_Enable(&Read_buffer, &Write_buffer);Delay_ms(1000);
Delay_ms(1000);for(;;) {num=0;while(num != 4)
Read_buffer[2];}}
Hid_Disable();}
```

The Software on PC

Option Explicit

Declare Function hidConnect Lib "mcHID.dll" Alias "Connect" (ByVal pHostWin As Long) As Boolean Declare Function hidDisconnect Lib "mcHID.dll" Alias "Disconnect" () As Boolean Declare Function hidGetItem Lib "mcHID.dll" Alias

"GetItem" (BvVal pIndex As Long) As Long

Declare Function hidGetItemCount Lib "mcHID.dll"

Alias "GetltemCount" () As Long

Declare Function hidRead Lib "mcHID.dll" Alias "Read" (ByVal pHandle As Long, ByRef pData As Byte)

Declare Function hidWrite Lib "mcHID.dll" Alias "Write" (ByVal pHandle As Long, ByRef pData As Byte) As Boolean

Declare Function hidReadEx Lib "mcHID.dll" Alias

"ReadEx" (ByVal pVendorID As Long, ByVal pProductID As Long, ByRef pData As Byte) As Boolean

Declare Function hidWriteEx Lib "mcHID.dll" Alias "WriteEx" (ByVal pVendorID As Long, ByVal pProductID As Long, ByRef pData As Byte) As Boolean Declare Function hidGetHandle Lib "mcHID.dll" Alias "GetHandle" (ByVal pVendoID As Long, ByVal pProductID As Long) As Long

Provide Contribution Provide Provide Contribution Provided Contribution Provided Contribution Provide			
Declare Function MidSelfeoducidD No incell Did All Action Control	Declare Function hidGetVendorID Lib "mcHID.dll"	Private Const BufferInSize = 4	DoEvents
Decimal Function in Microefford (1) No. in cell 10.0 His Action (1) No. Index. (1) No. Index. (1) No. Index. (1) No. Index. (2) No. Index.	Alias "GetVendorID" (ByVal pHandle As Long) As Long	Private Const BufferOutSize = 4	Next
Alias Cerifolderiol (Pyrel Information & Long) Activation Activa	Declare Function hidGetProductID Lib "mcHID.dll"		BufferOut(0) = 0 ' first by is always the report ID
Press Declare Function 10 - recitl 10 Press	Alias "GetProductID" (ByVal pHandle As Long) As		
Declare Function Mode-Welstein Mode			
Section Birdin Protein As I completed Protein As I completed Protein Assignment Birdin Could Protein Assignment Birdin Protein Birdin			
Declare Function Mid-Grave More Library Mid-Grave More Mid-Grave Mi			
The company of the			
Actionary Mykrol print As Simony Mykrol print As Long Astronomics Mykrol print Astronomics		BufferOut(1) = Asc("P") ' first data item ("P")	hidWriteEx VendorID, ProductID, BufferOut(0)
Declare Function hidGetProductName (Ptr) introduction (Ptr)	"mcHID.dll" Alias "GetVendorName" (ByVal pHandle	BufferOut(2) = $Asc("=")$ 'second data item ("-")	hidWriteEx VendorID, ProductID, BufferOut(1)
	As Long, ByVal pText As String, ByVal pLen As Long) As	BufferOut(3) = Asc("a")	hidWriteEx VendorID, ProductID, BufferOut(2)
Declare Function hid Gerifford active Mark Mark Function National Nation	Long		
Betaltus = "Time Right" Security Secur			
Declare Function Indicate Prof. Policy			
Declare Function hisGerlssrin Number (1) by discharded by the control of the cont			
Declare Function hidGes far of Number Lb	0		
Mad Mala SeaSenal Number (Byke) planted Surge, Byke planted Surge, Byke planted Surge, Byke planted Surge, Byke Surg		DoEvents	Next
Martin M	Declare Function hidGetSerialNumber Lib	Next	BufferOut(0) = 0 ' first by is always the report ID
As Long, 8/vol plant As String, 8/vol plant As Long, 8/vol plant As Long, 8/vol plant As String (\$\text{Micro}(1) = \text{As}(-7)^2 \text{stable table label}) \text{As}(\text{Micro}(1)) = \text{As}(-7)^2 \text{stable}(1) \text{As}(-1)^2 \text	"mcHID.dll" Alias "GetSerialNumber" (ByVal pHandle	BufferOut(0) = 0	
SufferOut(2) - Acct 1*) second doto lem ("-")	As Long, ByVal pText As String, ByVal pLen As Long) As		
Declare Function Mid-Gelinpulisportings (Mid-Mid-Market) (Mid-Market)	0		
mcHDdll" Alias "Gethopulkeportangth" (BVal phrodie As Long As Long Profession in McGetOupulkeportangth" (BVal Declars Function in McGetOupulkeportangth" (BVal Declars Sub Indicated Profession			
phonde As Long JA stong Declare Function in McGeOutpulleportlength Lib mcHDDII Allan "Sectory June Part and Market Vendorf). Product(D). Buffer Out(1) shidwiset Evendorf). Product(D). Buffer Out(1) shidwiset Evend			
Declare Function IndiGenOutpulsepartenght Byth Individual Extended Product(D) Buffer(Out[2) Product(D) Buff			
medital Allary (SetOrgunReportlength* (pilval phradie As Long) As Long Declare Sub hidseRecolNotify Lith "method II" Allas SetRecolNotify (pilval phradie) As Long, Bylat phradie) As Long, Bylat phradie As Long (pilval phradie) As Long (pilval phr			hidWriteEx VendorID, ProductID, BufferOut(1)
pérdens Sub MaisferaoNneit) ills "mcHID.dil" Allas SerfacoNneit)" (By/or jaronile As Long, By/or Johns As Long, By		hidWriteEx VendorID, ProductID, BufferOut(1)	hidWriteEx VendorID, ProductID, BufferOut(2)
pictoria Sub MaskeanNeshi jub "mcHID.dil" Allas SakeanNeshi jub "mcHID.dil	"mcHID.dll" Alias "GetOutputReportLength" (ByVal	hidWriteEx VendorID, ProductID, BufferOut(2)	hidWriteEx VendorID, ProductID, BufferOut(3)
Declare Function Mish ReadNotiny Finabled Ub morth Julia	pHandle As Long) As Long		
Selfendhotinfy (Bylor) phroadie As Long, Bylor brown As Long, Bylor brow			
Declare Function hidiskandNotifyEnabled Lib "mcHDdli" Alias "IskeadNotifyEnabled Lib "mcHDdli" Alias "IskeadNotifyEnabled Lib "mcHDdli" Alias "IskeadNotifyEnabled Lib "mcHDdli" Alias "IskeadNotifyEnabled" Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "Iskead Notified Lib "mcHDdli" Alias "Call WindowProc A" Elly Wold "motor Canl Wold Motor was alias "Call WindowProc A" Elly "mcHDdli" Alias "Call "mcHDdli" Alias "Call "mcHDdli" Alias "Call "mcHDdli" Alias "Call "mcHDdli" Alias "mcHDdli "motor Canl Wold Motor was alias "mcHDdli" Alias "mcHDdli			
Declare Function hidsReadNotifyEnabled (b) micritiDull* Also "SteadNotifyEnabled" (Byte) pHandle As Isong JAs Boolean pRoduct(I) A color hids/weitable bit micritiDull* Also "Sevariable" (Byte) pMandet D As Long, Byte) pProduct(I) A color hids/weitable bit micritiDull* Also "Sevariable" (Byte) pMandet D As Long, Byte) pProduct(I) A color hids/weitable bit micritiDull* Also "Sevariable" (Byte) pMandet D As Long, Byte) pProduct(I) A color hids/weitable bit micritiDull* Also "Sevariable" (Byte) pMandet D As Long, Byte) product(I) A color hids As Long, Byte) product(I) A color hids As Long, Byte) product(I) A color hids As Long, Byte) product(I) As Long, Byte) product(II) As Long			
mcHID_allY_Aliss "saReadNotifenabled" (ByVal phander) & Florida Paton (and Share Declare Function hids/wailable Lib "in-child" Aliss "sakrollable" (ByVal phander) & Florida Paton (and Share) BufferOut(1) = Asc(1") "sacrollable (ByVal phander) & Florida Paton (Coll Paton (
Delard Put Conton Mish Available Ltb "micht ID.dll" Alas Salvailable Ltb "micht ID.dll" Ala			
Declare Function hids/wailable Lb "mcHild Milat" SufferOv(1) = Asc (1-3) second data item (1-3)		BufferOut(0) = 0 ' first by is always the report ID	End Sub
Declare Function hidsk-validable Lb* "mcHiDLIA" Alsos, psylol phroductID As Long, My North 2014 As Long, Psylon phroductID As Long As Boolean Private Declare Function CallWindowProc Lb* SufferOut(1) = Asc[T]* SufferOut(1) = SufferOut(1)		BufferOut(1) = $Asc("P")$ ' first data item ("P")	Private Sub Command3 Click()
"Indication of the County of t	Declare Function hidlsAvailable Lib "mcHID.dll" Alias		
pProtect Declare Function CallWindowProc Lib "user32" Alias "CallWindowProc A" (ByVal) high-Rest MendorD, ProductID, BufferOut(1) high-Rest MendorD, ProductID, BufferOut(2) high-Rest MendorD, ProductID, BufferOut(2) high-Rest MendorD, ProductID, BufferOut(3) high-Rest MendorD, ProductID, BufferOut(3) high-Rest MendorD, ProductID, BufferOut(3) high-Rest MendorD, ProductID, BufferOut(4) high-Rest MendorD, Produ	"IsAvailable" (ByVal pVendorID As Long, ByVal		
Private Declare Function CallWindowProc. II but viser32" Alias "CallWindowProc. A" (Byt) A strong Mycl hard As Long, Byd Individual Street (Part Vision (Part Vis			
Just 232" Alias "CallWindowFrocA" (ByVal May As Long, ByVal March & Long, ByVal (Brarm As Long), ByVal (Brarm), ByVal (Brarm As Long), ByVal (Brarm), ByVal (Brarm), ByVal (Brarm), ByVal (Brarm), ByVal (Brarm), Brarm As Long), ByVal (Brarm), Brarm As Long), ByVal (Brarm As Long), ByVal (Brarm), Brarm As Long), Brarm As Long, ByVal (Brarm As Long), Brarm (Brarm), Brarm As Long), Brarm (Brarm), Brarm (Brarm			
IndivineEx Mendord D, Product ID, Buffer Court (2) IndivineEx Mendord D, Product ID, Buffer Court (3) IndivineEx Mendord D, Product ID, Buffer Court (4) IndivineEx Mendord D, Product ID, Buffer Court (5) IndivineEx Mendord D, Product ID, Buffer Court (6) IndivineEx Mendord D, Product ID, Buffer Court (6) IndivineEx Mendord D, Product ID, Buffer Court (7) IndivineEx Mendord D, Product			
Mag As Long, ByVal vParam As Long, ByVal vPa			
Indivines Long As Long Private Declare Function SetWindowLong Lib			
Private Declare Function SetWindowLong Lib viser32* Alias "SetWindowLongA" (Bydal hand A Sa Long, Bydal Indiax As Long, Bydal wheelung As Long Private Const WM. APP = 327.68 Buffar-Out(9) = 0 * first by is always the report ID Buffar-Out(9) = 0 * first by is always the report ID Buffar-Out(9) = Asc("P") * first data item ("P") Buffar-Out(9) = Asc("P") * first data item		hidWriteEx VendorID, ProductID, BufferOut(3)	hidWriteEx VendorID, ProductID, BufferOut(0)
Friste Declare Function SetWindowLong Lib visuar(32* Alias SetWindowLong* (ByVal hand As Long, ByVal Indiax As Long, Buffacout(1) = Asc(*P) first data tem(*P) Long Each End Indiax As Long, Buffacout(1) = Asc(*P) first data tem(*P) Long Each End Indiax As Long, Buffacout(1) = Asc(*P) first data tem(*P) Long Each End Indiax As Long, Buffacout(1) = Asc(*P) first data tem(*P) Long End End End Indiax As Long, Buffacout(1) = Asc(*P) first d	Long) As Long	hidWriteEx VendorID, ProductID, BufferOut(4)	hidWriteEx VendorID, ProductID, BufferOut(1)
Juser22 Alias "SetWindowLongA" (ByVal Membrd As Long, ByVal Indiaex As Long, ByVal Indiaex As Long, ByVal Indiaex As Long ByVal devilors As Long As Long (ByVal MindowLong As Long Private Const WML APP = 32768 **Brivate Const WML MID EVENT = WML APP + 200 **Brivate Const WML MID EVENT = WML APP + 200 **Brivate Const NOTIFY PLUIGGED = 1 **BufferOut(3) = Asc("+)" (first data item ("") **BufferOut(4) = Asc("+)" (first data item ("") **BufferOut(5) = Asc("+)" (first data item ("") **BufferOut(6) = Asc("+)" (first data item ("") **BufferOut(6) = Asc("+)"	Private Declare Function SetWindowLong Lib		
Next Longl Ax Long Bydol Indiex Ax Long, Bydol dwNewLong Ax Longl Ax Long BydrerOut(0) = 0 first by is always the report ID BydrerOut(0) = Axc("P") first data item ("P") BydrerOut(0) = Axc("P") first data it	"user32" Alias "SetWindowLonaA" (BvVal hwnd As		
BufferOut() = 0 if rist by is always the report ID			
Private Const GVII. MDPROC = .4 Private Const GWI. MDPROC = .4 Private Const GWI. MDPROC = .4 Private Const WM. HID EVENT = WM. APP + 200 Private Const NOTIFY FUNGESE = .7 Private Const NOTIFY CHANGED = .2 Private Const NOTIFY CHANGED = .2 Private Const NOTIFY CHANGED = .2 Private Const NOTIFY CHANGED = .3 Private Windreck VendorID, ProductID, BufferOut(1) Private PrevWinProc As Long Private PrevWinProc As Long Private PrevWinProc As Long Private PrevWinProc = .5 End Sub Command 4_Click() Form _Unload (0) End			
Private Const QWT_WNDPRCC = -4 Private Const NOTIFY_CHANGED = 1 Private Const NOTIFY_UNFLUGGED = 1 Private Const NOTIFY_UNFLUGGED = 2 Private Const NOTIFY_CHANGED = 3 Private Sub Commond 4_Click() Private Priva			
Private Const MOTIFY PLUGGED = 1 Private Const NOTIFY PLUGGED = 2 Private Const NOTIFY UNPLUGGED = 2 Private Const NOTIFY CHANGED = 3 Private Const NOTIFY CHANGED = 3 Private Const NOTIFY CHANGED = 4 Private Const NOTIFY CHANGED = 3 Private Const NOTIFY CHANGED = 4 Private Const NOTIFY CHANGED = 4 Private Evendroil Productil D. Buffer Out(1) Private PrewVinProc As Long Public Function ConnectToHID(ByVal pHostWin As Long) As Boolean PVinIndraide As Long Private PrewVinProc = SetWindowLong(FWinHandle) PrevWinProc = SetWindowLong(FWinHandle) PrevWinProc = SetWindowLong(FWinHandle) PrevWinProc = SetWindowLong FWinHandle, GWL_WNDPROC, Productil D, Ba Boolean Public Function DisconnectFromHID() As Boolean Public Function DisconnectFromHID() As Boolean Public Function Public Public Sub Command2 Click() For i = 0 To 9999 BufferOut(2) = Asc(*P)* first data item (*P*) BufferOut(2) = Asc(*P)* first data item (*P*) BufferOut(2) = Asc(*P)* first data item (*P*) BufferOut(3) = Asc(*P)* first data item (*P*) BufferOut(4) = Asc(*P)* first data item (*P*) BufferOut(6) = Asc(*P)* first data item (*P*) BufferOut(6) = ProductID, BufferOut(6) BufferOut(7) = Dotorate Sub Form Load(7) Private Sub For			
Private Const NOTIFY _PLUGGED = 1 Private Const NOTIFY _PLUGGED = 2 Private Const NOTIFY _CHANGED = 3 Private Pr		BufferOut(2) = $Asc("=")$ 'second data item ("-")	Private Sub Command4_Click()
Private Const NOTIFY_PLUGGED = 1 Private Const NOTIFY_CHANGED = 2 Private Const NOTIFY_CHANGED = 3 Private Const NoTIFY_CHANGED = 4 Private Const NoTIFY_CHANGED = 3 Private Const NoTIFY_CHANGED = 4 Private Const NoTIFY_CHANGED = 3 Private Const NoTIFY_CHANGED = 4 Private Const NoTIFY_CHANGED = 3 Private Const NoTIFY_CHANGED = 2 Private Const NoTIFY_CHANGED = 3 Private Const NoTIFY_CHANGED = 4 Private Const NoTIFY_CHANGED = 3 Private Const NoTIFY_CHANGED = 4 Private Const NoTIFY_CHANGED = 5		BufferOut(3) = Asc("h")	Form Unload (0)
Private Const NOTIFY_UNPLUGGED = 2 hidWriteEx VendorID, ProductID, BufferOut(0) Private Const NOTIFY_READ = 4 hidWriteEx VendorID, ProductID, BufferOut(2) Private Const NOTIFY_READ = 4 hidWriteEx VendorID, ProductID, BufferOut(2) hidWriteEx VendorID, ProductID, BufferOut(3) hidWriteEx VendorID, ProductID, BufferOut(3) hidWriteEx VendorID, ProductID, BufferOut(3) hidWriteEx VendorID, ProductID, BufferOut(3) Private Sub Form_Load() ConnectToHID (Me.hwnd) Private Sub Form_Load() ConnectToHID (Me.hwnd) Private Sub Form_Load() Privat	Private Const NOTIFY_PLUGGED = 1	BufferOut(4) = $Asc("T")$ fourth data item ("T")	
Private Const NOTIFY_CHANGED = 3 Private Const NOTIFY_CHANGED = 3 Private Const NOTIFY_CHANGED = 3 Private FreeWinProc As Long Private PreeWinProc As Long Private PreeWinProc As Long Private PreeWinProc As Long Private PreeWinProc = SetWindowLong(FWinHandle) PreeWinProc = SetWindowLong(FWinHandle) Private Public Function Disconnect(FWinHandle) Private Public Function Disconnect(FWinHandle) Private Sub Command2_Click() For i = 0 To 100 DoEvents Next Next PreeWinProc = SetWindowLong(FWinHandle) Private Sub Command2_Click() For i = 0 To 100 DoEvents Next Next Private Sub Command2_Click() For i = 0 To 100 DoEvents Next Next Private Sub Command2_Click() For i = 0 To 100 DoEvents Next Next Private Sub Command2_Click() For i = 0 To 100 DoEvents Next Next End Sub Private Sub Command2_Click() For i = 0 To 100 DoEvents Next Next End Sub Private Sub Command2_Click() For i = 0 To 100 DoEvents Next Next End Sub Private Sub Command2_Click() For i = 0 To 79999 End Sub Private Sub Command2_Click() For i = 0 To 79999 End Sub Private Sub Command2_Click() For i = 0 To 79999 End Sub Private Sub Command2_Click() For i = 0 To 79999 End Sub Private Sub Commund2_Click() For i = 0 To 79999 End Sub Private Sub Commund2_Click() For i = 0 To 79999 End Sub Private Sub Commund2_Click() For i = 0 To 79999 End Sub Private Sub Commund2_Click() For i = 0 To 79999 End Sub Private Sub Commund2_Click() For i = 0 To 79999 End Sub Private Sub Commund2_Click() For i = 0 To 79999 End Sub Private Sub Commund2_Click() For i = 0 To 79999 End Sub End Sub It hidGetProductID(pHandle) = VendorID And It hidded It hidd			
Private Const NOTIFY_READ = 4 Private FrewWinProc As Long Private FrewWinProc Biddlish Private Sub Command2_Click()			
Private FWinHrandle As Long Private FWinHrandle As Long Public Function ConnectToHID(ByVal pHostWin As Long) As Boolean FWinHrandle = pHostWin ConnectToHID(ByVal pHostWin As Long) As Boolean FWinHrandle = pHostWin ConnectToHID = hidConnect(FWinHrandle) FPrevWinProc = SetWindowLong(FWinHrandle) FPrevWinProC, AddressOf WinProc) End Function DisconnectFromHID() As Boolean			
Private FWinHandle As Lang Public Function ConnectToHID(ByVal pHostWin As Long) As Boolean FWinHandle = pHostWin ConnectToHID(ID = hidConnect(FWinHandle) FPrevWinProc = SetWindowLong(FWinHandle) FPrevW			
Public Function ConnectToHID(ByVal pHostWin As Long) As Boolean FVinHandle = pHostWin ConnectToHID = hidConnect[FWinHandle) FrewWinProc = SetWindowLong[FWinHandle, GWL_WNDPROC, AddressOf WinProc) End Function Public Function DisconnectFromHID() As Boolean DisconnectFromHID() = hidDisconnect BetWindowLong FWinHandle, GWL_WNDPROC, End Function Private Function WinProc(ByVal pHWnd As Long, Brivate Function WinProc(ByVal pHWnd As Long) It indicetred them ("-") BufferOut(B) = Asc("T)' fourth data item ("-") BufferOut(B) = Asc("D) FroductID, BufferOut(B) BufferOut(B) = Asc("D) Froduct	•		
Long) As Boolean FWinHandle = pHostWin ConnectIoHID = hidConnect(FWinHandle) FPrevWinProc = SetWindowLong(FWinHandle, GML_WNDPROC, AddressOf WinProc) End Function DisconnectFromHID As Boolean DisconnectFromHID hidDisconnect SetWindowLong FWinHandle, GWL_WNDPROC, AddressOf WinProc) End Function DisconnectFromHID hidDisconnect SetWindowLong FWinHandle, GWL_WNDPROC, FPrevWinProc SetWindowLong FWinHandle, GWL_WNDPROC, FPrevWinProc SetWindowLong FWinHandle, GWL_WNDPROC, FPrevWinProc SetWindowLong FWinHandle, GWL_WNDPROC, FPrevWinProc SetWindowLong FWinHandle, SetWindowLong FWinHandl			
FWinHandle = pHostWin ConnectToHID = hidConnect(FWinHandle)		For ii = 0 To 100	
ConnectToHID = hidConnect(FWinthandle) FPrevWinProc = SetWindowLong(FWinthandle) FPrevWinProc = SetWindowLong(FWinthandle) For i = 0 To 9999 Frivate Sub Command2 Click() For i = 0 To 9999 BufferOut(0) = 0 'first by is always the report ID BufferOut(1) = Asc("P") 'first data item ("P") SetWindowLong FWinthandle, GWL_VNDPROC, FrewWinProc End Function Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal wParam As Long, ByVal pHSg = WM-HID_EVENT Then Select Case wParam Case Is = NOTIFY_LUGGED MainForm.OnUnplugged (IParam) Case Is = NOTIFY_LUGGED MainForm.OnChanged MainForm.OnChanged MainForm.OnChanged MainForm.OnChanged MainForm.OnRead (IParam) End Select End If WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, IParam End Sub SufferOut(1) = Asc("P") 'first data item ("P") IndWriteEx VendorID, ProductID, BufferOut(1) BufferOut(2) = Asc("m") 'second data item ("F") HidWriteEx VendorID, ProductID, BufferOut(2) HidWriteEx VendorID, ProductID, BufferOut(2) HidWriteEx VendorID, ProductID, BufferOut(2) HidWriteEx VendorID, ProductID, BufferOut(2) HidWriteEx VendorID, ProductID, BufferOut(4) HidWriteEx VendorID, ProductID, BufferOut(1) HidWriteEx VendorID,	Long) As Boolean	DoEvents	End Sub
ConnectToHID = hidGonnect[FWinHandle] FPrevWinProc = SetWindowLong(FWinHandle), GWL_WNDPROC, AddressOfWinProc) End Function DisconnectFromHID() As Boolean DisconnectFromHID() As CPP) First data item ("") HidWriteEx VendorID, ProductID, BufferOut(0) DisdWriteEx VendorID, ProductID, BufferOut(1) DisdWriteEx VendorID, ProductID, BufferOut(1) DisdWriteEx VendorID, ProductID, BufferOut(1) DisdWriteEx VendorID, ProductID, BufferOut(1) D	FWinHandle = pHostWin	Next	Public Sub OnPlugged (BvVal pHandle As Long)
FPrevWinProc = SetWindowLong(FWinHandle, GWL_WNDPROC, AddressOf WinProc)	ConnectToHID = hidConnect(FWinHandle)		
GWL_WNDPROC, AddressOf WinProc) End Function Private Sub Command2_Click() End Function DisconnectFromHID() As Boolean DisconnectFromHID = hidDisconnect SetWindowLong FWinHandle, GWL_WNDPROC, End Function Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal wParam As Long, ByVal Param As Long) As Long If pMsg = WM. HID EVENT Then Select Case wParam Case Is = NOTIFY_PLUGGED MainForm.OnPlugged (IParam) Case Is = NOTIFY_CHANGED MainForm.OnChanged MainForm.OnChanged MainForm.OnChanged Case Is = NOTIFY_CHANGED MainForm.OnChanged MainForm.OnChanged MainForm.OnChanged Case Is = NOTIFY_CHANGED MainForm.OnChanged MainF			
End Function Public Function DisconnectFromHID() As Boolean DisconnectFromHID() Bidfboolean DistrictFromHID() Bidfbo			u ,
Public Function DisconnectFromHID() As Boolean DisconnectFromHID = hidDisconnect SetWindowLong FWinHandle, GWL_WNDPROC, FPrevWinProc End Function Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal wParam As Long, ByVal IParam As Long) As Long If pMsg = WM_HID_EVENT Then Select Case wParam Case Is = NOTIFY_PLUGGED MainForm.OnPlugged (IParam) Case Is = NOTIFY_PUNPLUGGED MainForm.OnChanged Case Is = NOTIFY_CHANGED MainForm.OnChanged Case Is = NOTIFY_CHANGED MainForm.OnChanged Case Is = NOTIFY_READ MainForm.OnChanged Case Is = NOTIFY_CHANGED MainForm.OnChanged Ca	_ '		
DisconnectFromHID = hidDisconnect SetWindowLong FWinHandle, GWL_WNDPROC, FPrevWinProc End Function Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal wParam As Long, ByVal IParam As Long) ByVal pMsg As Long If pMsg = WM_HID_EVENT Then Case Is = NOTIFY_PLUGGED MainForm.OnPlugged (IParam) Case Is = NOTIFY_UNPLUGGED MainForm.OnUnplugged (IParam) Case Is = NOTIFY_CHANGED MainForm.OnChanged Case Is = NOTIFY_CHANGED MainForm.OnRead(IParam) BufferOut(0) = 0 *first by is always the report ID MainForm.OnRead (IParam) End Select End If End If End If End If End Sub End Sub Notice YendorID, ProductID, BufferOut(0) Dim DeviceHandle As Long NoticeHandle = hidGetHandle (VendorID) DeviceHandle = hidGetHandle (VendorID) Dim DeviceHandle = hidGetHandle (VendorID) Dim DeviceHandle As Long If WinFrec VendorID, ProductID, BufferOut(4) Iblistatus = "Turn Leff" Public Sub OnRead(ByVal pHandle As Long) If hidSetReadNotify DeviceHandle, True End Sub Public Sub OnRead(ByVal pHandle As Long) If hidSetReadNotify DeviceHandle, True End Sub Public Sub OnRead(ByVal pHandle As Long) If hidSetReadNotify DeviceHandle, True End Sub Public Sub OnRead(ByVal pHandle As Long) If hidSetReadNotify DeviceHandle, True End Sub Public Sub OnRead(ByVal pHandle As Long) If hidSetReadNotify DeviceHandle (VendorID) In the firm of the firm			
SetWindowLong FWinHandle, GWL_WNDPROC, FPrevWinProc End Function Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As L			
FPrevWinProc End Function BufferOut(3) = Asc("h") BufferOut(4) = Asc("h") BufferOut(4) = Asc("h") BufferOut(4) = Asc("h") BufferOut(1) = Asc("h") BufferOut(2) = Asc("h) BufferOut(3) = Asc("h) BufferOut(3) = Asc(h) BufferOut(4) = Asc(h) BufferOut(5) = BufferOut(6) BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by is always the report ID BufferOut(6) = O 'first by		BufferOut(1) = Asc("P") ' first data item ("P")	hidGetProductID(pHandle) = ProductID Then
End Function Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal wParam As Long, ByVal IParam As Long) As Long IParam As Long As Long IParoductID, BufferOut(0) IParoductID, BufferOut		BufferOut(2) = $Asc("=")$ 'second data item ("-")	End If
End Function Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal wParam Long, ByVal Long,		BufferOut(3) = Asc("h")	End Sub
Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal pMsg As Long, ByVal wParam As Long, ByVal pMsg As Long, ByVal wParam As Long, ByVal pMsg As Long, ByVal wParam As Long, ByVal pMsg As Long by Val wParam As Long, ByVal PMsg As Long by Val wParam As Long, ByVal PMsg As Long by Val wParam As Long, ByVal PMsg As Long by Val wParam As Long, ByVal PMsg As Long by Val wParam As Long, ByVal PMsg As Long by Val wParam As Long, ByVal PMsg As Long by Val WParam As Long, ByVal PMsg As Long	End Function		Public Sub On Changed()
ByVal pMsg As Long, ByVal wParam As Long, ByVal hidWriteEx VendorID, ProductID, BufferOut(1) ProductID, BufferOut(2) hidWriteEx VendorID, ProductID, BufferOut(2) hidWriteEx VendorID, ProductID, BufferOut(3) hidWriteEx VendorID, ProductID, BufferOut(3) hidWriteEx VendorID, ProductID, BufferOut(3) hidWriteEx VendorID, ProductID, BufferOut(4) End Sub Public Sub OnRead(ByVal pHandle As Long) If hidRead(pHandle, BufferIn(0)) Then If (BufferIn(1) = Asc("P") And BufferIn(2) = Asc("=' And BufferIn(4) = Asc("P") And BufferIn(2) = Asc("=' And BufferIn(4) = Asc("P") And BufferIn(3) End If End If End If End If End Sub Public Sub OnRead(ByVal pHandle As Long) If hidRead(pHandle, BufferIn(0)) Then If (BufferIn(1) = Asc("P") And BufferIn(2) = Asc("=' And BufferIn(1) = Asc("P") And BufferIn(2) = Asc("=' And BufferIn(1) = Asc("P") And BufferIn(2) = Asc("=' And BufferIn(3)) End If End Sub Public Sub WriteSomeData() Public Sub WriteSup In It in IdWriteEx VendorID, ProductID, BufferOut(1) Buffe	Private Function WinProc(BvVal pHWnd As Long,		
Param As Long			
If pMsg = WM_HID_EVENT Then Select Case wParam Lagranger Select Case Select			
Select Case wParam hidWriteEx VendorID, ProductID, BufferOut(4) Case Is = NOTIFY_PLUGGED MainForm.OnPlugged (IParam) Case Is = NOTIFY_UNPLUGGED MainForm.OnUnplugged (IParam) Case Is = NOTIFY_UNPLUGGED MainForm.OnUnplugged (IParam) Case Is = NOTIFY_CHANGED MainForm.OnChanged Case Is = NOTIFY_CHANGED MainForm.OnChanged Case Is = NOTIFY_READ MainForm.OnRead (IParam) End Sub Further(1) = Asc("=") Asc("=") And BufferIn(2) = Asc("=") And BufferIn(4) = Asc("") Then txtreceived = BufferIn(3) End If End Sub Public Sub OnRead (ByVal pHandle As Long) If hidRead (pHandle, BufferIn(2) = Asc("=") And BufferIn(1) = Asc("=") And BufferIn(1) = Asc("=") And BufferIn(1) = Asc("=") And BufferIn(1) = Asc("=") And BufferIn(2) = Asc("=") Extreceived = BufferIn(3)) End If End If End Sub It (BufferIn(1) = Asc("=") It (BufferIn(1) = Asc("=") And BufferIn(2) = Asc("=") Extreceived = BufferIn(3)) End If End If End Sub WriteEx VendorID, ProductID, BufferOut(0) Public Sub WriteSomeData()			
Case Is = NOTIFY_PLUGGED MainForm.OnPlugged ((Param)) Case Is = NOTIFY_UNPLUGGED MainForm.OnDlugged ((Param)) Case Is = NOTIFY_UNPLUGGED MainForm.OnUnplugged ((Param)) Case Is = NOTIFY_CHANGED MainForm.OnChanged Case Is = NOTIFY_CHANGED MainForm.OnChanged Case Is = NOTIFY_READ MainForm.OnChanged End Select End If WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, Param) End Function pMsg, wParam, Param) End Function Dim i, ii, j As Variant Private Const VendorID = 4660 hidWriteEx VendorID, ProductID, BufferOut(2) bidWriteEx VendorID, ProductID, BufferOut(3) BufferOut(4) End Sub Public Sub OnRead((ByVal pHandle As Long) If hidRead(pHandle, BufferIn(3)) Then If (BufferIn(1) = Asc("=") Asc("=") And BufferIn(1) = Asc("=") And BufferIn(2)			
MainForm.OnPlugged (Param) Case Is = NOTIFY_UNPLUGGED DoEvents MainForm.OnUnplugged (Param) Case Is = NOTIFY_CHANGED BufferOut(0) = 0 'first by is always the report ID MainForm.OnChanged Case Is = NOTIFY_CHANGED BufferOut(1) = Asc("P") 'first data item ("P") Case Is = NOTIFY_READ MainForm.OnRead (Param) BufferOut(1) = Asc("P") 'first data item ("-") MainForm.OnRead (Param) BufferOut(2) = Asc("=")' 'second data item ("-") End Select End If WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, Param) End Flenction Dim i, ii, j As Variant Private Const VendorID_ 4660 MainForm.OnPead (Param) BufferOut(2) = Asc("=")' 'second data item ("-") BufferOut(3) = Asc("T") 'fourth data item ("-") End SufferOut(0) = 0 'first by is always the report ID BufferOut(1) BufferOut(1) BufferOut(0) = 0 'first by is always the report ID BufferOut(0) = 0 'first data item, etc etc bidWriteEx VendorID, ProductID, BufferOut(2) BufferOut(1) = 10 'first data item, etc etc bidWriteEx VendorID, ProductID, BufferOut(4) BufferOut(1) End Sub			
Case Is = NOTIFY_UNPLUGGED MainForm.OnUnplugged (IPrarm) Case Is = NOTIFY_CHANGED MinForm.OnChanged MainForm.OnChanged MainForm.OnChanged Case Is = NOTIFY_READ MainForm.OnRead (IPrarm) MainForm.OnRead (IPrarm) MainForm.OnRead (IPrarm) BufferOut(1) = Asc("P")' first data item ("P") BufferOut(2) = Asc("e")' second data item ("-") BufferOut(3) = Asc("d")' o110 0010 BufferOut(4) = Asc("T")' tourh data item ("T") WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, IPrarm) End Function DoEvents If (BufferIn(1) = Asc("P")' And BufferIn(2) = Asc("=")' And BufferIn(4) = Asc("T")' Then txtreceived = BufferIn(3) Label1.Caption = Str\$ (BufferIn(3)) End If End If End If End Sub Public Sub WriteSomeData() bidWriteEx VendorID, ProductID, BufferOut(2) BufferOut(0) = 0 'first by is always the report ID BufferOut(1) = 10 'first data item, etc etc bidWriteEx VendorID, ProductID, BufferOut(4) BufferOut(1) = 10 'first data item, etc etc bidWriteEx VendorID, ProductID, BufferOut(4) End Sub		lblstatus = "Turn Left"	Public Sub OnRead(ByVal pHandle As Long)
Case Is = NOTIFY_UNPLUGGED MainForm.OnUnplugged (IParam) Case Is = NOTIFY_CHANGED BufferOut(0) = 0 'first by is always the report ID MainForm.OnChanged Case Is = NOTIFY_CHANGED BufferOut(1) = Asc("P") 'first data item ("P") Case Is = NOTIFY_READ MainForm.OnRead (IParam) BufferOut(2) = Asc("=")' second data item ("-") BufferOut(3) = Asc("d")' 101100010 BufferOut(4) = Asc("T")' fourth data item ("T") WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, IParam) End Flenction In I, i, As Variant Private Const VendorID = 4660 DoEvents If (BufferIn(1) = Asc("P") And BufferIn(2) = Asc("=")' And BufferIn(1) = Asc("P") And BufferIn(2) = Asc("P") And Buf	MainForm.OnPlugged (IParam)	For ii = 0 To 100	If hidRead(pHandle, BufferIn(0)) Then
MainForm.OnUnplugged (Param) Case Is = NOTIFY_CHANGED BufferOut(0) = 0 'first by is always the report ID MainForm.OnChanged BufferOut(1) = Asc("P") 'first data item ("P") Case Is = NOTIFY_READ BufferOut(2) = Asc("e")' second data item ("-") MainForm.OnRead (Param) BufferOut(3) = Asc("e")' second data item ("-") End If BufferOut(3) = Asc("d")' 10110 0010 End Select End If WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, Param) End Function Dim i, ii, j As Variant Private Const VendorID = 4660 Next And BufferOut(0) = Asc("T") Then txtreceived = BufferIn(3) Label 1. Caption = Str\$ (BufferIn(3)) End If End Sub ("-") End If End Sub WriteSomeData() Public Sub WriteSomeData() BufferOut(0) = 0 'first by is always the report ID BufferOut(1) BufferOut(1) BufferOut(1) = 10 'first data item, etc etc hidWriteEx VendorID, ProductID, BufferOut(3) HidWriteEx VendorID, ProductID, BufferOut(4) End Sub	Case Is = NOTIFY UNPLUGGED		If $(BufferIn(1) = Asc("P") And BufferIn(2) = Asc("=")$
Case Is = NOTIFY_CHANGED MainForm.OnChanged Case Is = NOTIFY_READ MainForm.OnChanged Case Is = NOTIFY_READ MainForm.OnChanged BufferOut(1) = Asc("P") 'first data item ("P") BufferOut(2) = Asc("=") 'second data item ("-") End If BufferOut(4) = Asc("T") 'fourth data item ("-") BufferOut(4) = Asc("T") 'fourth data item ("T") End If End If End Sub WriteSub WriteSomeData() Public Sub WriteSomeData() Public Sub WriteSomeData() BufferOut(0) = 0 'first by is always the report ID BufferOut(1) BufferOut(1) BufferOut(1) BufferOut(1) BufferOut(1) BufferOut(1) BufferOut(0) = 0 'first by is always the report ID BufferOut(1) BufferOut(1) BufferOut(0) = 0 'first by is always the report ID BufferOut(0) = 0 'first by is always the report ID BufferOut(0) = 0 'first by is always the report ID BufferOut(0) = 0 'first by is always the report ID BufferOut(0) = 10			
MainForm.OnChanged Case Is = NOTIFY_READ BufferOut(1) = Asc("P") 'first data item ("P") BufferOut(2) = Asc("=") 'second data item ("-") End If BufferOut(3) = Asc("d") 'fourth data item ("-") End If BufferOut(4) = Asc("T") 'fourth data item ("-") End If BufferOut(4) = Asc("T") 'fourth data item ("T") End Select End If WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, Param) End Function Dim i, ii, j As Variant Private Const VendorID = 4660 BufferOut(1) = Asc("P") 'first data item ("P") BufferOut(2) = End If End If End Sub Public Sub WriteSomeData() BufferOut(0) = 0 'first by is always the report ID BufferOut(1) = BufferOut(0) = 0 'first by is always the report ID BufferOut(1) = 10 'first data item ("P") End If BufferOut(0) = 0 'first by is always the report ID BufferOut(1) = 10 'first data item ("P") BufferOut(2) = 10 'first data item ("P") BufferOut(2) = 10 'first data item ("P") BufferOut(2) = 10 'first data item			
Case Is = NOTIFY_READ MainForm.OnRead (IParam) End Select End If WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, IParam) End Function Dim i, ii, j As Variant Private Const VendorID = 4660 BufferOut(2) = Asc("=") 'second data item ("-") BufferOut(3) = Asc("=") 'income data item ("-") BufferOut(4) = Asc("T") 'fourth data item ("T") BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) = 0 'first by is always the report ID BufferOut(1) Bu			(/
MainForm.OnRead (IParam) End Select End If WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, IParam) End Function Dim i, ii, j As Variant Private Const VendorID = 4660 BufferOut(3) = Asc("d") '1010 0010 End If End Sub ProductID, BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) BufferOut(0) = 0 'first by is always the report ID BufferOut(1) = 10 'first data item, etc etc hidWriteEx VendorID, ProductID, BufferOut(3) HidWriteEx VendorID, ProductID, BufferOut(4) End Sub			
End Select End If WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, IParam) End Function Dim i, ii, j As Variant Private Const VendorID = 4660 End Sub(FFrevWinProc, pHWnd, pidWriteEx VendorID, ProductID, BufferOut(1) End Function BufferOut(2) Dim i, iii, j As Variant Private Const VendorID = 4660 End Sub(FFrevWinProc, pHWnd, pidWriteEx VendorID, ProductID, BufferOut(2) BufferOut(3) BufferOut(1) = 10 'first data item, etc etc hidWriteEx VendorID, ProductID, BufferOut(3) HidWriteEx VendorID, ProductID, BufferOut(4) End Sub End Sub End Sub End Sub First data item, ("T") BufferOut(0) = 10 'first by is always the report ID BufferOut(1) = 10 'first data item, etc etc hidWriteEx VendorID, ProductID, BufferOut(3) BufferOut(1) = 10 'first data item, etc etc hidWriteEx VendorID, ProductID, BufferOut(4) End Sub			
WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, Param)		BufferOut(3) = $Asc("d") \cdot 01100010$	
WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, IParam) hidWriteEx VendorID, ProductID, BufferOut(1) BufferOut(0) = 0 'first by is always the report ID BufferOut(1) BufferOut(1) = 10 'first by is always the report ID BufferOut(2) BufferOut(1) = 10 'first data item, etc etc hidWriteEx VendorID, ProductID, BufferOut(3) HidWriteEx VendorID, ProductID, BufferOut(4) End Sub		BufferOut(4) = $Asc("T")$ 'fourth data item ("T")	End Sub
pMsg, wParam, IParam) hidWriteEx VendorID, ProductID, BufferOut(1) End Function hidWriteEx VendorID, ProductID, BufferOut(2) Dim i, ii, j As Variant Private Const VendorID = 4660 hidWriteEx VendorID, ProductID, BufferOut(3) hidWriteEx VendorID, ProductID, BufferOut(4) hidWriteEx VendorID, ProductID, BufferOut(4) hidWriteEx VendorID, ProductID, BufferOut(4)	WinProc = CallWindowProc(FPrevWinProc, pHWnd,		Public Sub WriteSomeData()
End Function hidWriteEx VendorID, ProductID, BufferOut(1) BufferOut(1) = 10 'first data item, etc etc Dim i, ii, j As Variant hidWriteEx VendorID, ProductID, BufferOut(3) HidWriteEx VendorID, ProductID, BufferOut(0) Private Const VendorID = 4660 hidWriteEx VendorID, ProductID, BufferOut(4) End Sub			
Dim i, ii, j As Variant hidWriteEx VendorID, ProductID, BufferOut(3) HidWriteEx VendorID, ProductID, BufferOut(0) Private Const VendorID = 4660 hidWriteEx VendorID, ProductID, BufferOut(4) End Sub			
Private Const VendorID = 4660 hidWriteEx VendorID, ProductID, BufferOut(4) End Sub			
Bit to the state of the state o			
For ii = 0 to 100			EUG 200
	Trivale Constribution	For ii = 0 Io 100	