UNIVERSITY OF Southampton
Electronics and Computer Science

Design Exercise D1

HIDden secrets of USB

http://www.ucas.com/documents/annualdatasets/2012/insts12.exe

January 2013

Some information from

– USB Made Simple

History

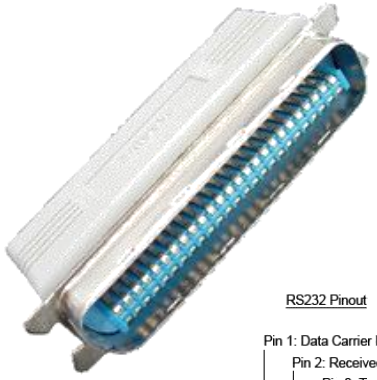
- USB: Universal Serial Bus.
- Invented and standardized by a group of computer and peripherals manufactures in 1995.
- Competes with IEEE1394 *FireWire*.

Evolution

Revision	Issue Date	Comments
0.7	November 11, 1994	Supersedes 0.6e.
0.8	December 30, 1994	Revisions to Chapters 3-8, 10, and 11. Added appendixes.
0.9	April 13, 1995	Revisions to all the chapters.
0.99	August 25, 1995	Revisions to all the chapters.
1.0 FDR	November 13, 1995	Revisions to Chapters 1, 2, 5-11.
1.0	January 15, 1996	Edits to Chapters 5, 6, 7, 8, 9, 10, and 11 for consistency.
1.1	September 23, 1998	Updates to all chapters to fix problems identified.
2.0 (draft 0.79)	October 5, 1999	Revisions to chapters 5, 7, 8, 9, 11 to add high speed.
2.0 (draft 0.9)	December 21, 1999	Revisions to all chapters to add high speed.
2.0	April 27, 2000	Revisions for high-speed mode.

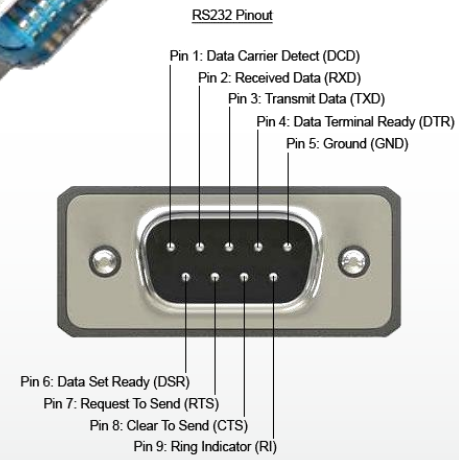
PC interfaces were a mess

- Big Centronics printer port.



- PS/2 Mouse, Keyboard

- RS232 serial: + - 12V
  and all sorts of flow control.



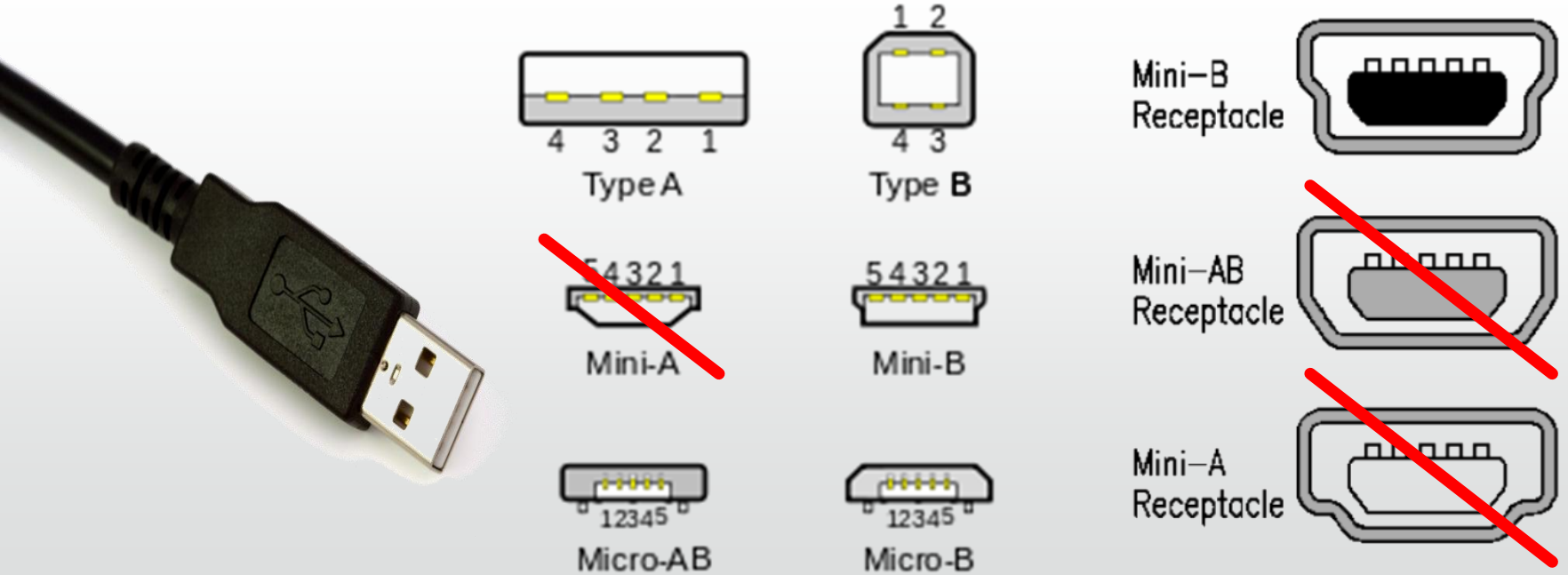
- Data peripherals over
  bidirectional Centronics port.

Basic characteristics

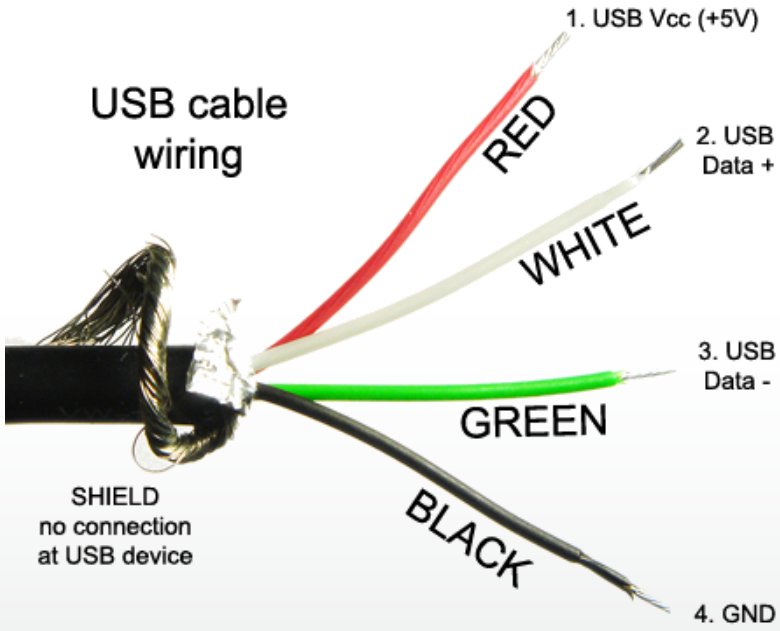
- Low voltage (at the time) 5V interface supplies power to small peripherals.
- Expandable via hubs to 127 devices, but *NOT* a bus.
- Two-wire differential signaling.
- Serial protocol => no skew problems.
- Largely self-configuring:
  - No need for user to allocate interrupts, IO ports
  - Many devices (e.g. keyboard, mouse) completely self-describing.
- Devices can be attached and detached on a running host.

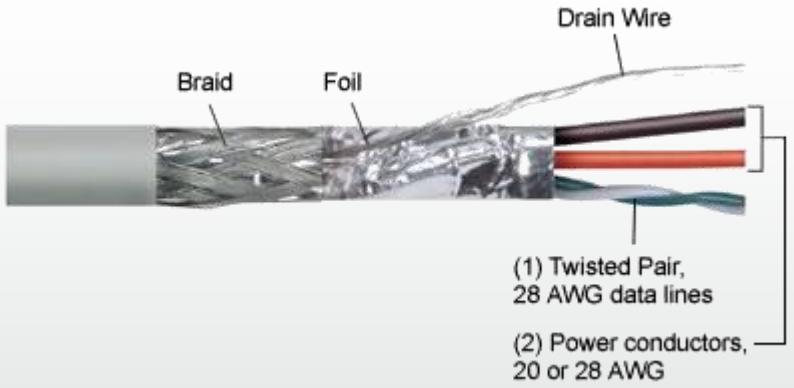
Network and Connectors

- The logical topology of the USB is a tree structure.
- The (single) host polls the peripherals, possibly via hubs; you cannot easily directly connect two hosts or two slaves (but see USB On-the-go). The A connector is the host; B is the slave.
- Three generations of ever-smaller connectors.



Common cable colour assignments





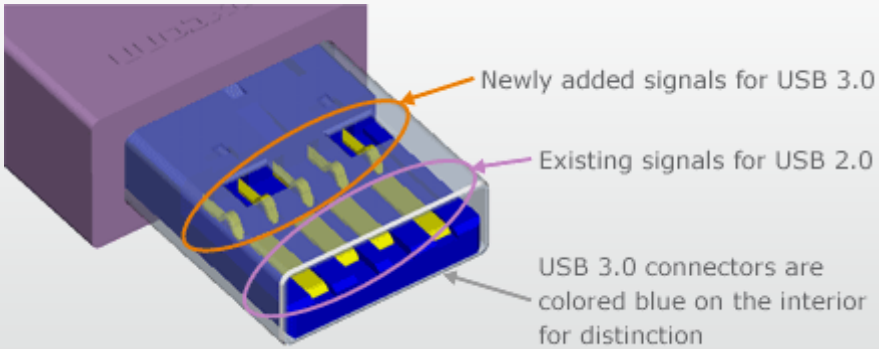
Connector evolution

- Mini A and mini AB connectors were deprecated by *USB Implementers Forum* on 23rd May 2007.
- The micro connectors support more insertions (have a longer life) and the wear is concentrated in the cable rather than the device.
- Micro USB B connectors are also used in standard EN 62684:2010 for a common mobile charger.

Speed

The latest invocation is Super Speed USB 3.0 with a theoretical maximum transfer rate of 5Gbit/s, similar to PCIe Gen2, and five additional pins (including signal ground) in a backward compatible connector:

- USB 1: Low speed: 1.5Mbit/s signalling, full speed: 12Mbit/s
- USB2: + High Speed: 480Mbit/s
- USB3: + Super Speed: 5Gbit/s signalling.



A modern standard?

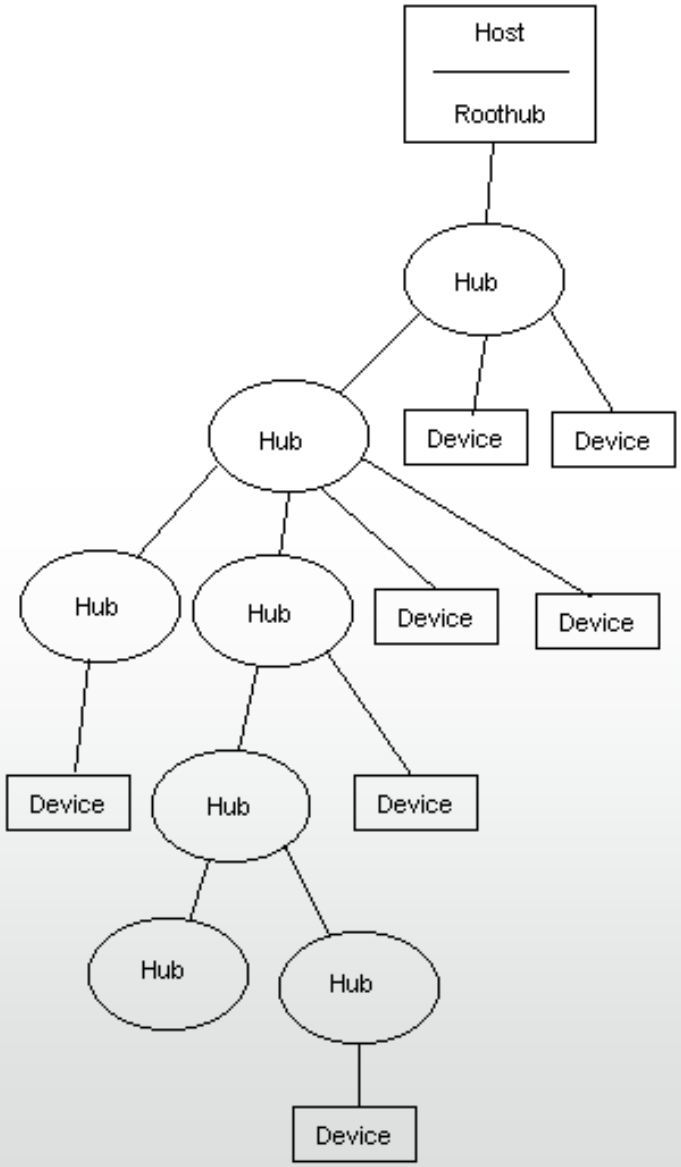
- Most modern interface standards use serial communication and support automatic configuration by interrogating self-describing devices.
- In modern high-speed systems, the data communication is typically *unidirectional* to avoid the delays involved in *turning around* the bus signals.
- It is also common to use separate clocks in the fastest systems as *clock recovery* can be problematic at the highest speeds.

Software environment

- The *heavy lifting* is done by the host software; slaves can be simple. The host software has to recognize various device *class*es and install appropriate, sometimes vendor-supplied drivers. It also has to manage the various hubs in the network.
- Every device has to be registered with a *vendor ID* and a *product* ID, 16 bit each.
- The University of Southampton has a Vendor ID; it is administered by Denis Nicole [dan@ecs.soton.ac.uk]; he will issue Product IDs on request.

A USB network is a tree

UNIVERSITY OF Southampton
Electronics and Computer Science



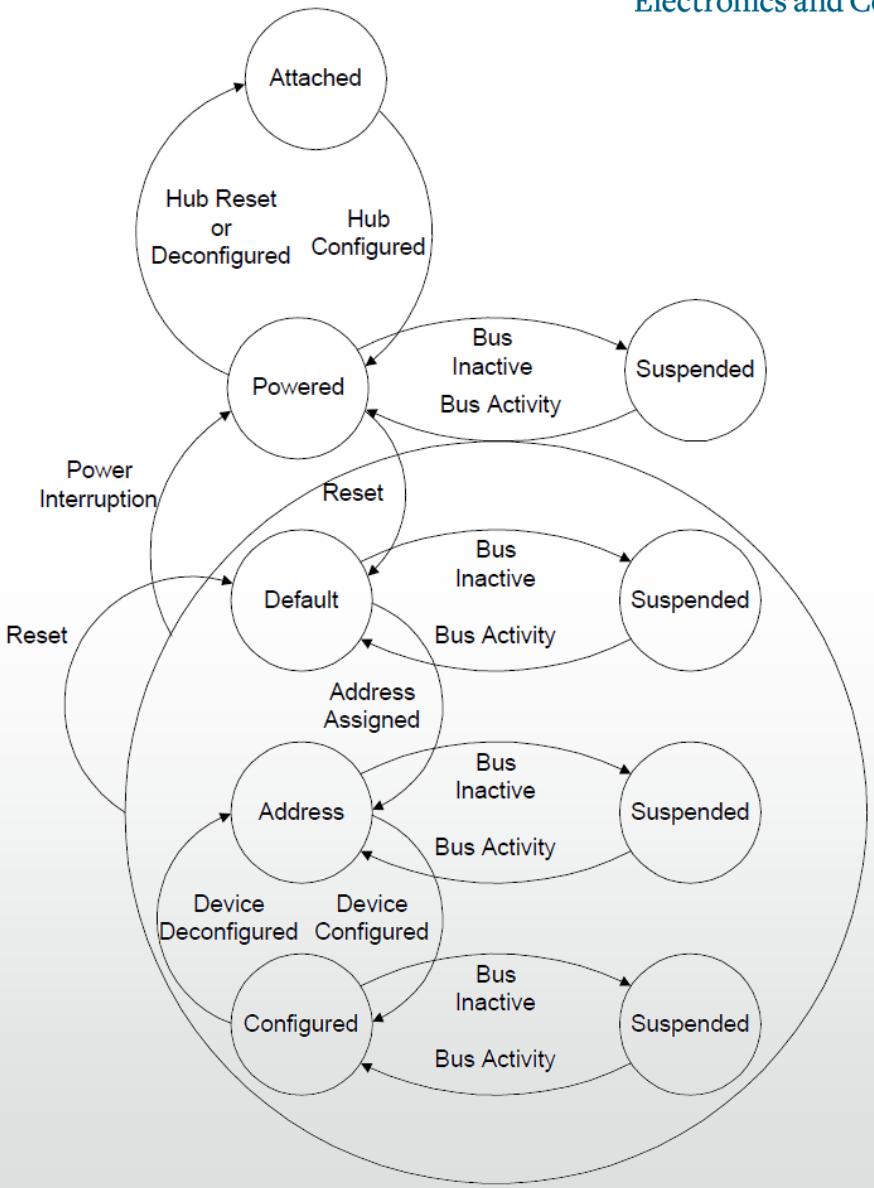
Hubs are managed by the host. A device only need worry about its Device ID.

Power

- USB supplies V_{BUS} at about 5V. Do not connect a decoupling capacitor of more than 10µF to this line; it might crash the hub or other devices with its inrush current. There *must* be at least 1µF present (at all V_{BUS} voltages) to support on-the-go device recognition in the absence of power on V_{BUS}.
- A low-speed device announces its presence by pulling the D- pin up via a 1k5Ω resistor. This pull-up should be towards 3.3V derived from V_{BUS}, not from self power; pulling up D+ in the absence of V_{BUS} can confuse on-the-go devices.
- A device can draw up to 100mA in operation, but must be able to reduce this to 2.5mA if *suspend*ed. It can request up to 500mA, but this may not be granted. Laptops have been known to agree a high current load, then fail to deliver it. If you want more than 100mA, you will probably need to use a *powered hub*. Or your device can be *self powered*. A device is suspended if the host sends it nothing for 3ms. Wake-up is by reversing the polarity of D+ and D-, either from host or device.

Device States

UNIVERSITY OF Southampton
Electronics and Computer Science



On-the-go

- There is support for systems to act as both host and device, e.g. a camera might act as a device when connected to a PC but as a host when connected to a printer. Such systems are called *on-the-go (OTG)*. Such units must nowadays have a Micro AB connector.
- An OTG cable *must* have an A connector at one end and a B connector at the other end. The cable grounds the ID pin (pin 5) at the A end to indicate that that system should act as host.
- It can get much more complicated. There are now various resistances for the ID pin, and there is a protocol to swap host and device, e.g. when a printer needs to supply power to a camera.

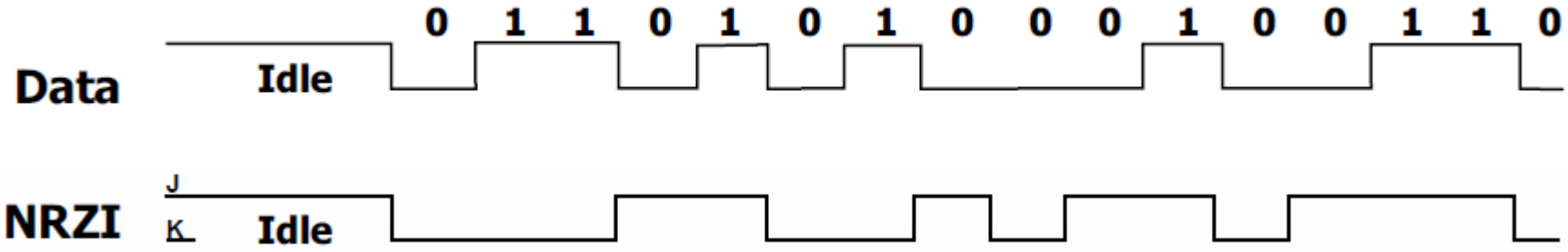
Layered protocols: Low Speed Electrical

- All modern communications protocols are *layered*. We describe the relationship between corresponding layers at each end.
- Electrical layer: on D+, D-
  LOW is -1V to 0.3V in presence of 1.5kΩ pull-up
  HIGH is 2.8V to 4.6V in presence of 1.5k Ω pull-down
  Rise/fall times 75ns to 300ns with 200pF to 600pF load
- Low speed signalling is *differential* at 1.5Mbit/s
- Reset is signalled by holding D+, D- both low for 10ms
- End of Packet (EOP) is signalled by taking D+, D- both low for 2 bit times, then taking D- high (the idle state).
- D+ and D- are never both high at the same time.

See: http://www.usbmadesimple.co.uk/ums_3.htm

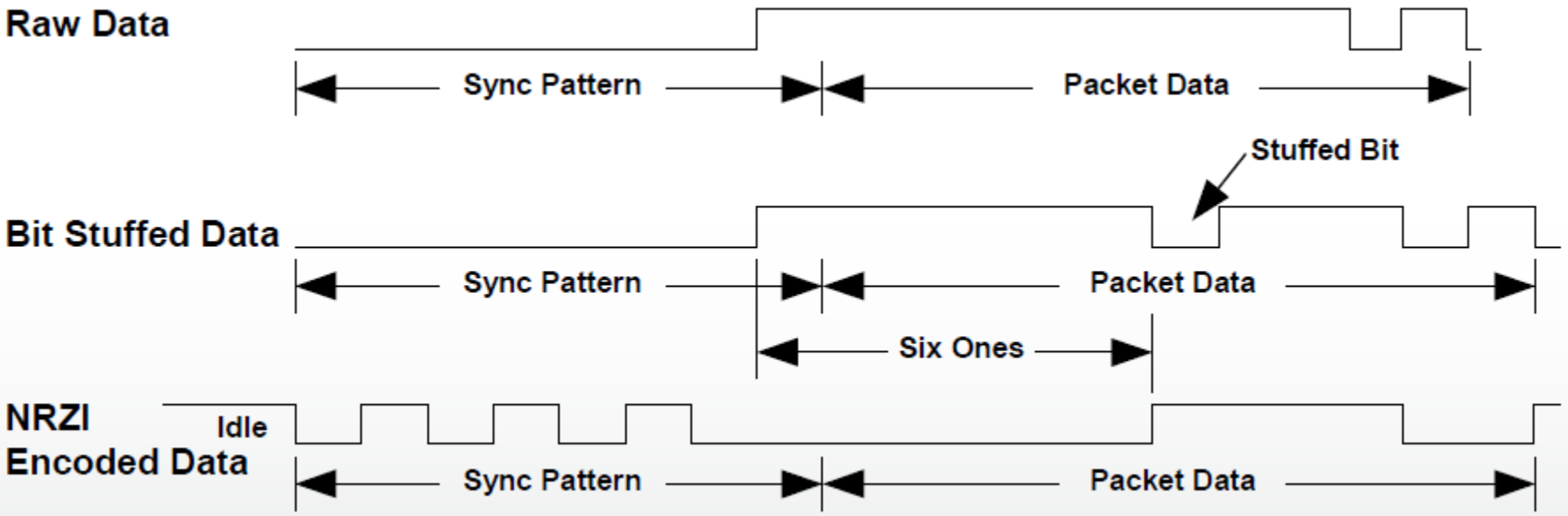
Bit layer

- Data encoding is NRZI with *bit-stuffing*



- There will never be seven consecutive ones.

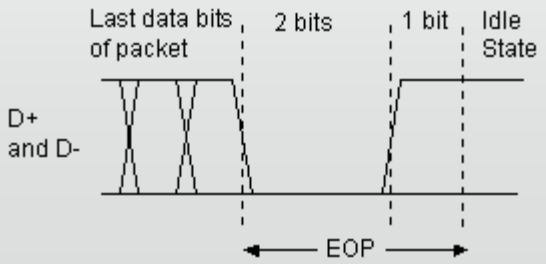
NRZI encoding of data bits



Protocol layer

All data is assembled into *packets:*

- SYNC synchronisation bits: NRZI-encoded 0s.
- Packet ID packet type
- Address 0…127 USB addresses (sender or receiver)
- Endpoint 0…15 Endpoints within device
- Frame number You wouldn't want to use the same data twice
- Data at most 8 bytes for low-speed USB
- CRC error detecting code
- End of packet



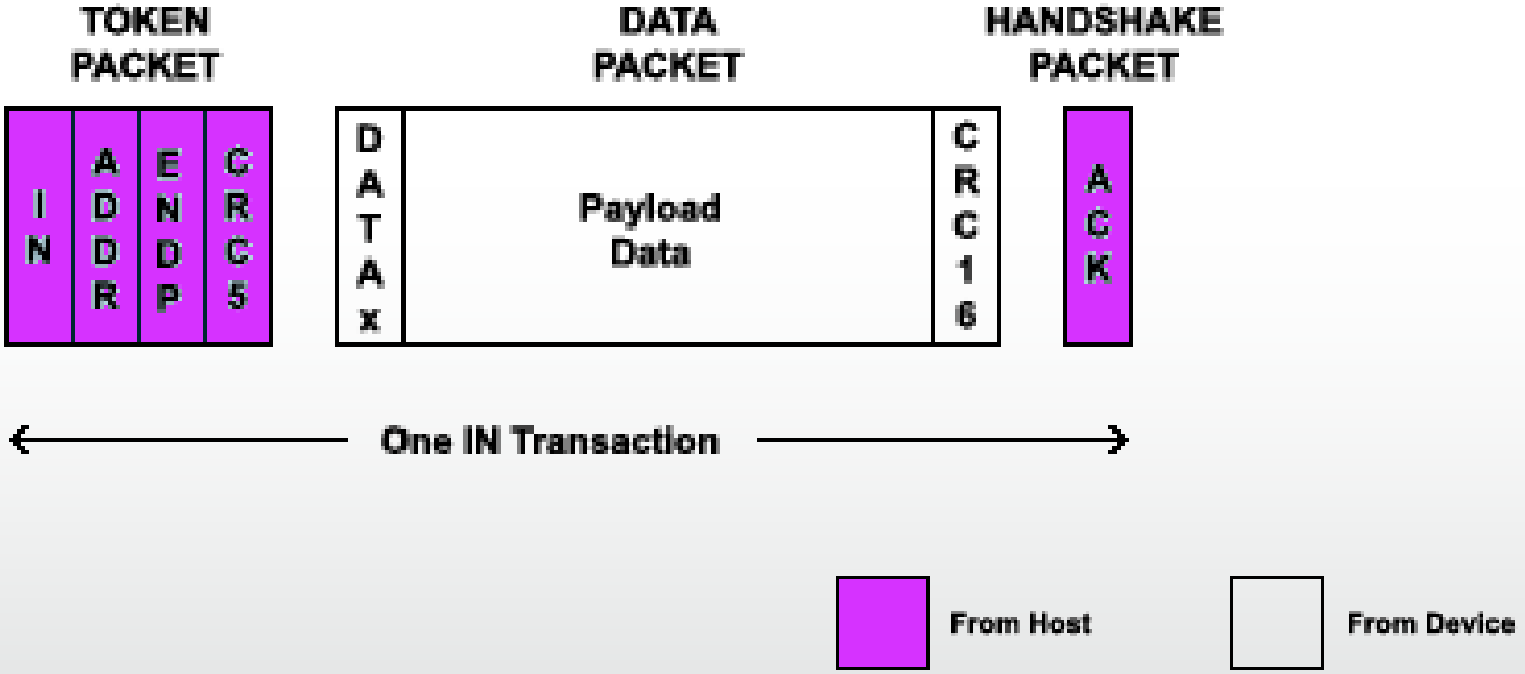
Four classes of packets

UNIVERSITY OF Southampton
Electronics and Computer Science

PID Type	PID Name	PID<3:0>
Token	OUT	0001b
	IN	1001b
	SOF	0101b
	SETUP	1101b
Data	DATA0	0011b
	DATA1	1011b
	DATA2	0111b
	MDATA	1111b
Handshake	ACK	0010b
	NAK	1010b
	STALL	1110b
	NYET	0110b
Special	PRE	1100b
	ERR	1100b
	SPLIT	1000b
	PING	0100b
	Reserved	0000b

Transactions

A transaction is made up of a sequence of packets



Interrupt transfers

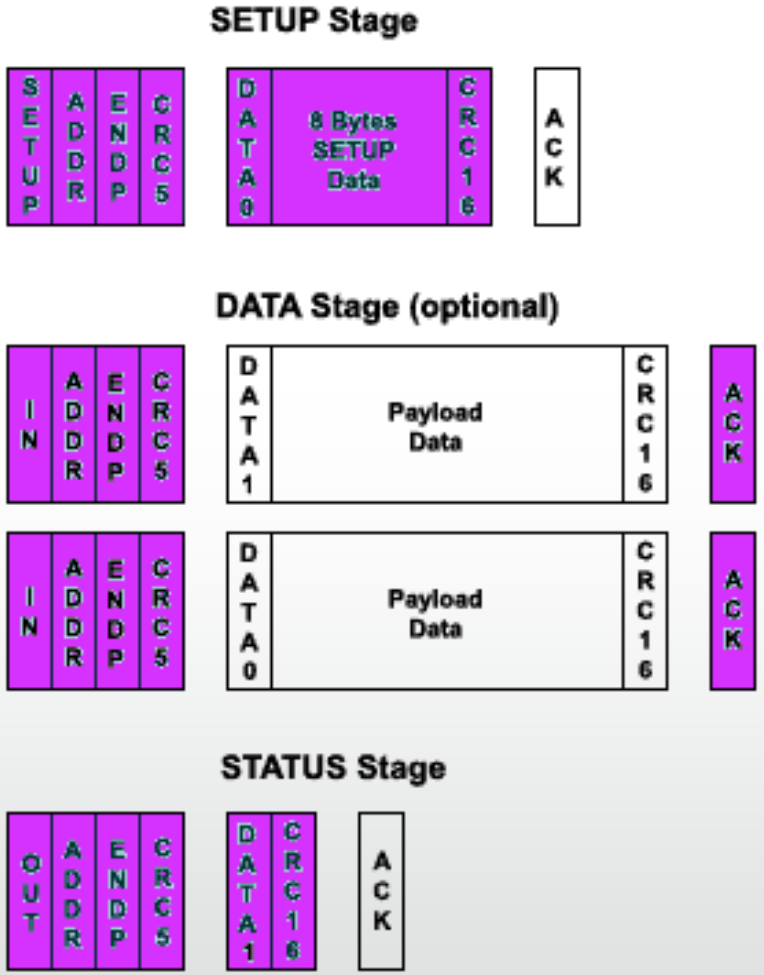
- Have nothing to do with processor interrupts
- They are polled by the host
- A single IN or OUT transaction makes up an interrupt transfer

Control Transfer

A control transfer is made up of several transactions, moving data in both directions

It is divided into three stages.

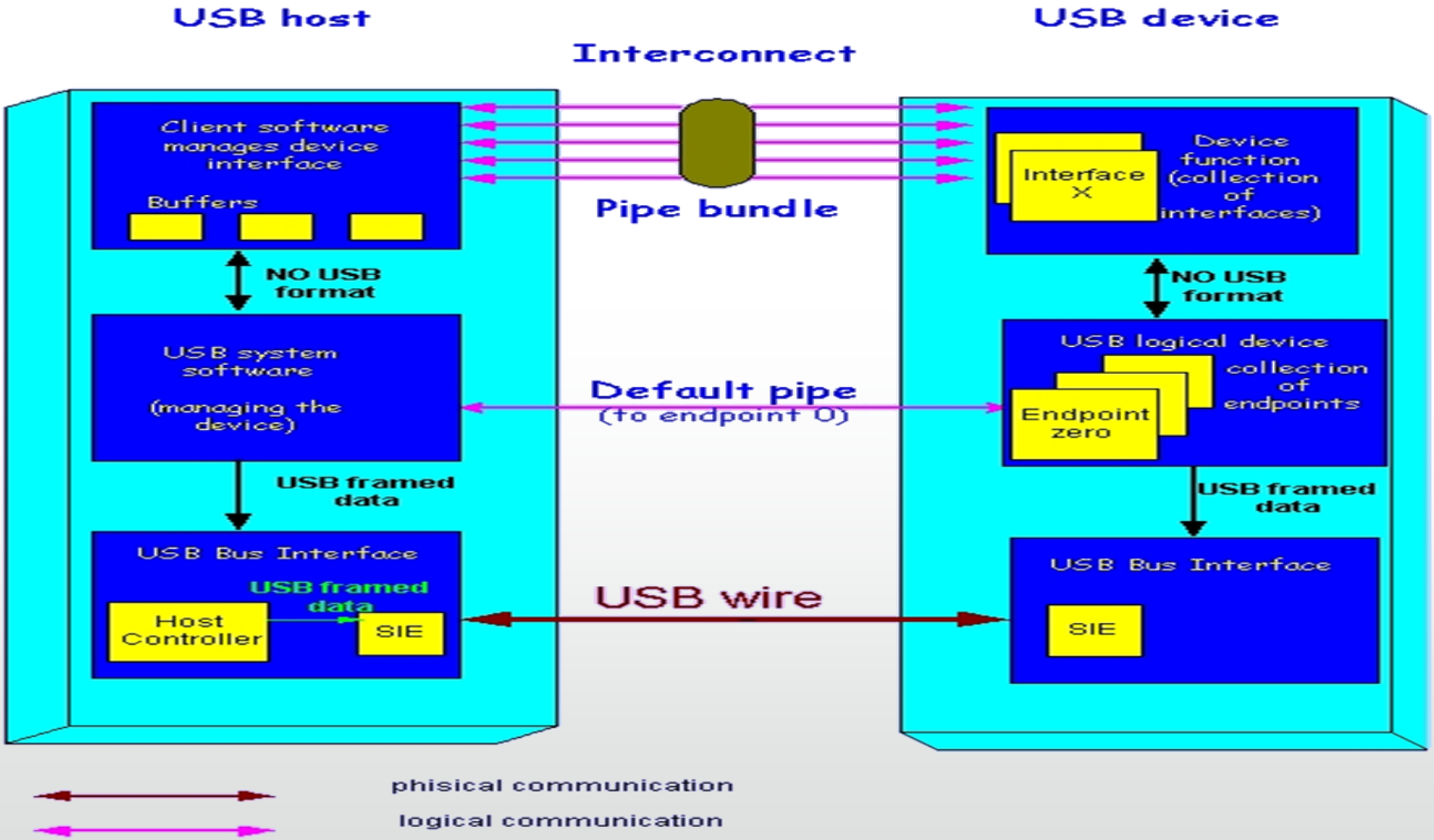
This is how the device describes itself to the host.



That's a lot of layers

- Transfers
- Transactions
- Packets
- On-the wire NRZI
- Bit-stuffing
- Raw bits

Communication Flow



Transfer Types

- Control Transfer
- Isochronous Transfer
- Bulk Transfer
- Interrupt Transfer

[Not for low-speed USB]

Endpoints

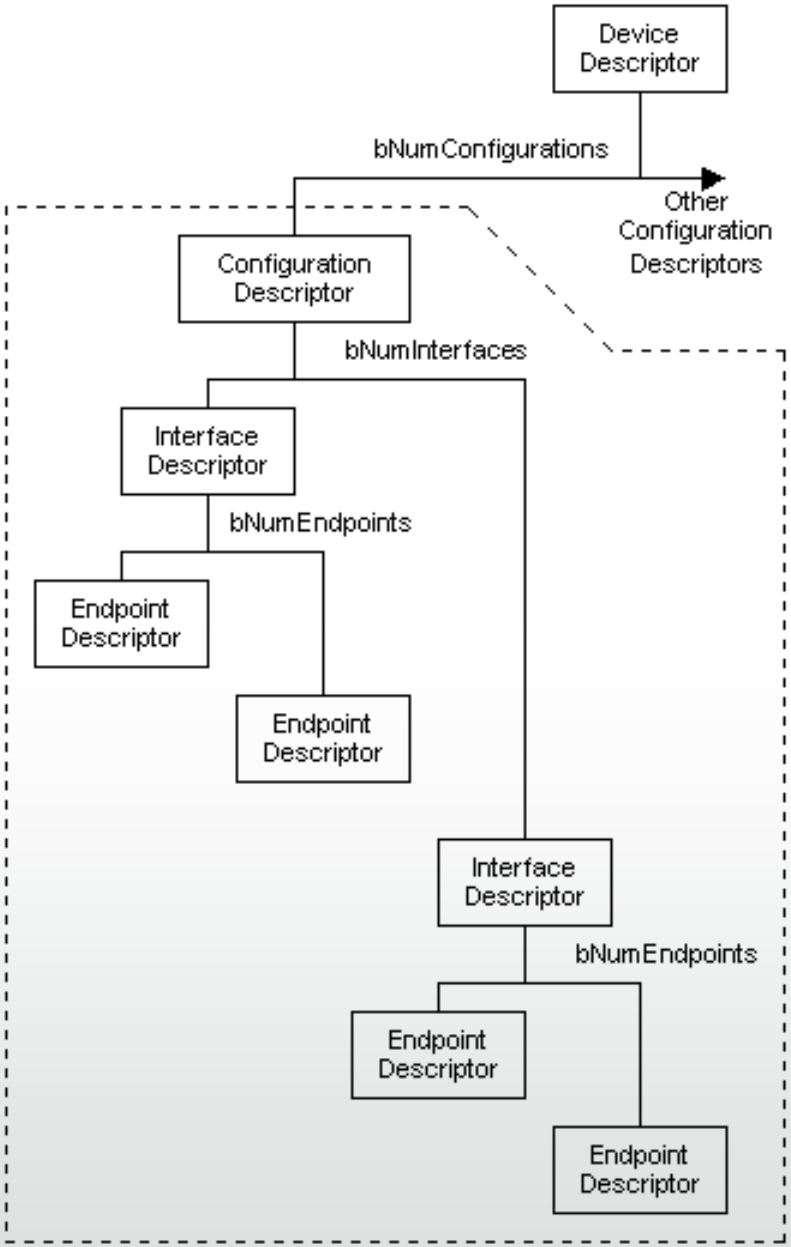
- Every device is required to provide endpoint 0 for control and enumeration: the *control transfers* are bidirectional.
- Other endpoints can transfer data using *interrupt transfers*. These endpoints are unidirectional.
- A typical keyboard would have one *device->host* endpoint for keyboard data and one *host->device* endpoint for keyboard lights.
- Low-speed USB cannot do isochronous or bulk transfers, so no COM: ports or memory sticks.

Device startup

- The host uses control transfers to endpoint 0 to learn the capabilities of the device. The first important information is the maximum packet size.
- After that, there are a range of descriptors. They lead to the setting up of logical devices and additional endpoints.
- For the details, see USB Made Simple
- You can compare these mouse descriptors with the ones on the following pages from

Descriptors

UNIVERSITY OF Southampton
Electronics and Computer Science



Enumeration

```
#define  USB_CFG_VENDOR_ID       0xda, 0x1b
#define  USB_CFG_DEVICE_ID       0x31, 0xe1
#define USB_CFG_DEVICE_VERSION  0x00, 0x01
#define USB_CFG_VENDOR_NAME     'U', 'n', 'i', 'v', 'e', 'r', 's', 'i', \
                                't', 'y', ' ', 'o', 'f', ' ', 'S', 'o', \
                                'u', 't', 'h', 'a', 'm', 'p', 't', 'o', 'n'
#define USB_CFG_VENDOR_NAME_LEN 25
#define USB_CFG_DEVICE_NAME     'P', 'a', 's', 's', 'g', 'e', 'n'
#define USB_CFG_DEVICE_NAME_LEN 7
#define USB_CFG_INTERFACE_CLASS     0x03 // HID
#define USB_CFG_INTERFACE_SUBCLASS  0x01 // Boot
#define USB_CFG_INTERFACE_PROTOCOL  0x01 // Keyboard
```

Device Information

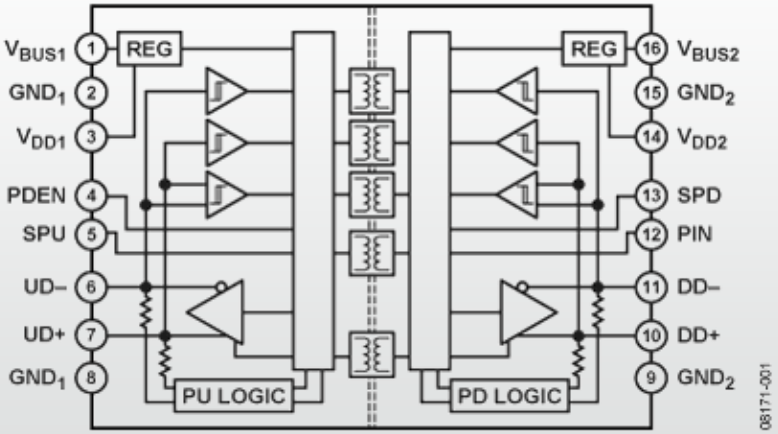
University of Southampton
Electronics and Computer Science

```
PROGMEM char usbHidReportDescriptor[USB_CFG_HID_REPORT_DESCRIPTOR_LENGTH] = {
    0x05, 0x01,                    // USAGE_PAGE (Generic Desktop)
    0x09, 0x06,                    // USAGE (Keyboard)
    0xa1, 0x01,                    // COLLECTION (Application)
    0x75, 0x01,                    //   REPORT_SIZE (1)
    0x95, 0x08,                    //   REPORT_COUNT (8)
    0x05, 0x07,                    //   USAGE_PAGE (Keyboard)(Key Codes)
    0x19, 0xe0,                    //   USAGE_MINIMUM (Keyboard LeftControl)(224)
    0x29, 0xe7,                    //   USAGE_MAXIMUM (Keyboard Right GUI)(231)
    0x15, 0x00,                    //   LOGICAL_MINIMUM (0)
    0x25, 0x01,                    //   LOGICAL_MAXIMUM (1)
    0x81, 0x02,                    //   INPUT (Data,Var,Abs) ; Modifier byte
    0x95, 0x01,                    //   REPORT_COUNT (1)
    0x75, 0x08,                    //   REPORT_SIZE (8)
    0x81, 0x03,                    //   INPUT (Cnst,Var,Abs) ; Reserved byte
    0x95, 0x05,                    //   REPORT_COUNT (5)
    0x75, 0x01,                    //   REPORT_SIZE (1)
    0x05, 0x08,                    //   USAGE_PAGE (LEDs)
    0x19, 0x01,                    //   USAGE_MINIMUM (Num Lock)
    0x29, 0x05,                    //   USAGE_MAXIMUM (Kana)
    0x91, 0x02,                    //   OUTPUT (Data,Var,Abs) ; LED report
    0x95, 0x01,                    //   REPORT_COUNT (1)
    0x75, 0x03,                    //   REPORT_SIZE (3)
    0x91, 0x03,                    //   OUTPUT (Cnst,Var,Abs) ; LED report padding
    0x95, 0x06,                    //   REPORT_COUNT (6)
    0x75, 0x08,                    //   REPORT_SIZE (8)
    0x15, 0x00,                    //   LOGICAL_MINIMUM (0)
    0x25, 0x65,                    //   LOGICAL_MAXIMUM (101)
    0x05, 0x07,                    //   USAGE_PAGE (Keyboard)(Key Codes)
    0x19, 0x00,                    //   USAGE_MINIMUM (Reserved (no event indicated))(0)
    0x29, 0x65,                    //   USAGE_MAXIMUM (Keyboard Application)(101)
    0x81, 0x00,                    //   INPUT (Data,Ary,Abs)
    0xc0                           // END_COLLECTION
};
```

Useful USB gizmo

- Full sped USB DC isolator, good to 1000V. Supplies power across the isolation: Olimex USB-ISO. Costs around £30.

- Uses the Analog Devices ADuM4160 with magnetic coupling



The Motto
Strenuis Ardua Cedunt
may be translated:
The Heights yield to Endeavour.