BIP in Embedded Systems

USB devices on embedded Systems

Thomas Herpoel

February 12, 2024

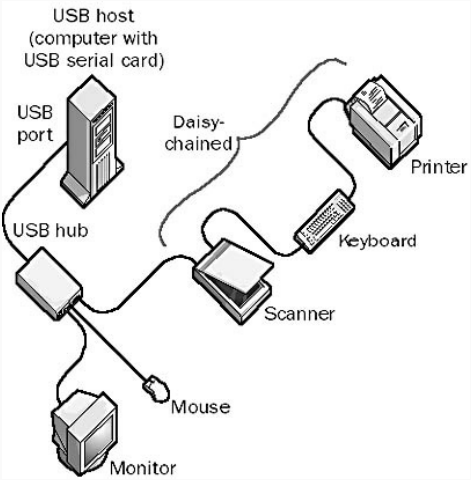
École d'ingénieurs de la HELHa

HELHa
Haute École Louvain en Hainaut

USB on embedded devices

Part 1: Understand USB in a few slides?

Universal Serial Bus



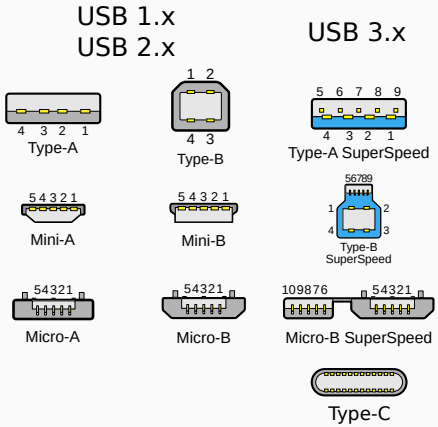
One HOST device connected to **several** USB devices!

Versions

The USB protocol was created in 1996. Since then it evolved quite a bit:

- USB 1.1 Low-Speed (LS): 1.5 Mbps
- USB 1.1 Full-Speed (FS): 12 Mbps
- USB 2.0 High-Speed (HS): 480 Mbps
- USB 3.0, 3.1, 3.2 SuperSpeed (SS): 5 Gbps, 10 Gbps, 20 Gbps
- USB4: 20Gbps, up to 120 Gbps (!)

USB Type-A Type-B or Type-C? Only describes the physical connector!



- USB 1.x: one differential pair
- USB 2.x: one differential pair
- USB 3.x and USB4: 2 or 4 differential pairs!

Type-C connector

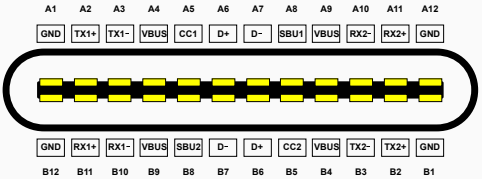


Fig. 1: https://upload.wikimedia.org/wikipedia/commons/0/07/USB_Type-C_Receptacle_Pinout.svg

Slowly replacing all USB connectors. It has the advantage of being compatible with older USB versions (thanks to a legacy differential pair) while still allowing more modern USB 3.x and USB4 to operate.

Data Link layer

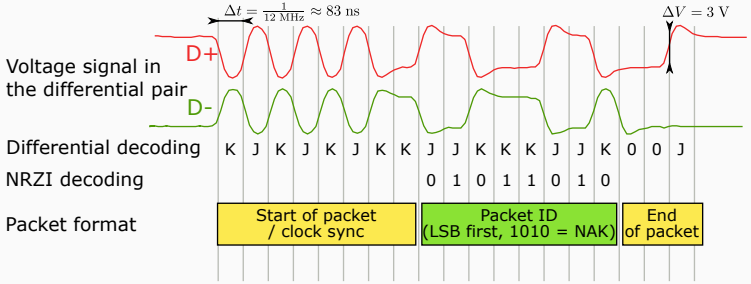


Fig. 2:
https://upload.wikimedia.org/wikipedia/commons/e/e0/USB_signal_example.svg

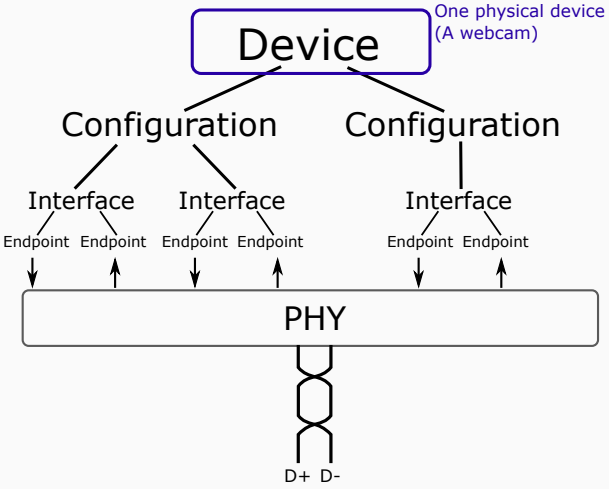
Universal?

The goal of the Universal Serial Bus is to be **universal**!

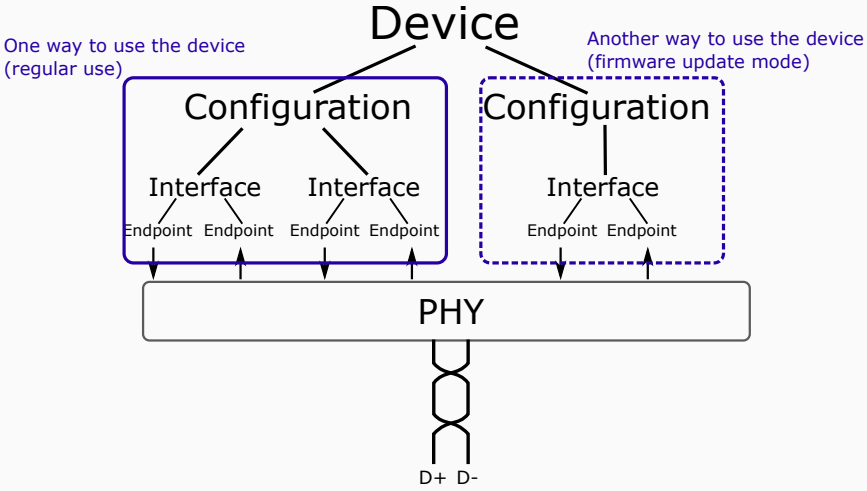
A USB device could be **anything**, so the USB protocol uses a **hierarchical** device structure and **descriptors** to help the host use the devices.

At startup, the host asks the device for information about it. They are given in the form of **descriptors**.

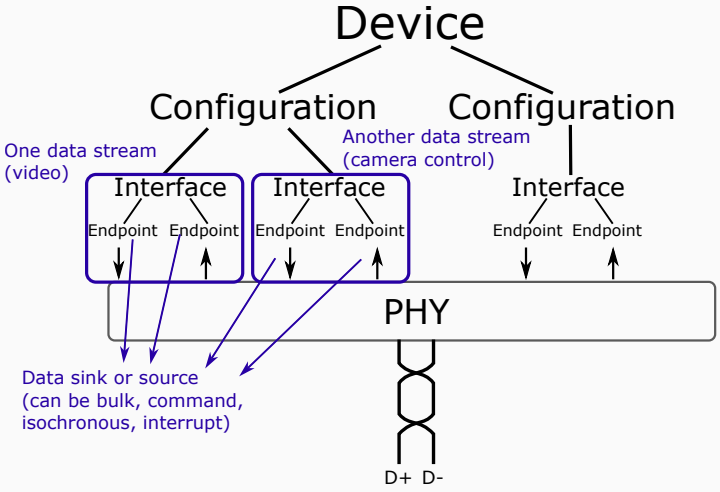
USB device description



USB device description



USB device description



USB Enumeration

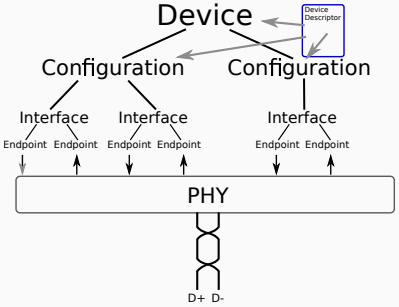
Enumeration is the process where the host detects a device and gather information about it. It helps the host loading a proper driver in order to use the device correctly. Enumeration steps are:

- Device presence and speed detection
- Get Device descriptor
- Get Configuration descriptor
- Get Interface descriptors

USB Enumeration

Enumeration steps are:

- Device presence and speed detection
- **Get Device descriptor**
- Get Configuration descriptor
- Get Interface descriptors

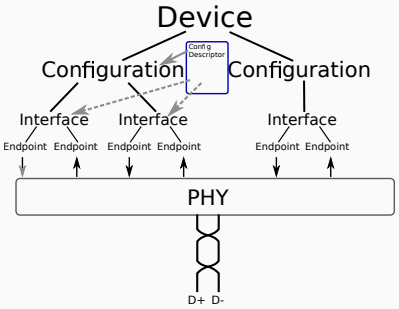


Device descriptor: includes information about the device, and the number of configurations for this device.

USB Enumeration

Enumeration steps are:

- Device presence and speed detection
- Get Device descriptor
- **Get Configuration descriptor**
- Get Interface descriptors

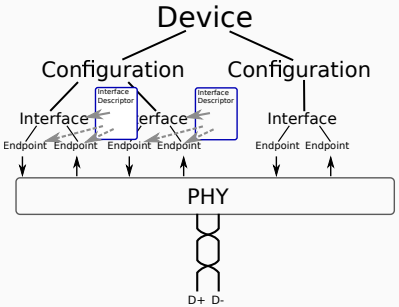


Configuration descriptor: A device can be in only one configuration at any time. It also gives the number of interfaces for this configuration.

USB Enumeration

Enumeration steps are:

- Device presence and speed detection
- Get Device descriptor
- Get Configuration descriptor
- **Get Interface descriptors**



Interface descriptor: all the interfaces for a configuration are active at the same time.
Endpoint descriptor: they are the data sinks or sources used between the host and the device.

USB Enumeration Example


```
Device Descriptor:
0x12    bLength
0x01    bDescriptorType
0x0200  bcdUSB
0x00    bDeviceClass
0x00    bDeviceSubClass
0x00    bDeviceProtocol
0x08    bMaxPacketSize0   (8 bytes)
0x046D  idVendor
0xC051  idProduct
0x3000  bcdDevice
0x01    iManufacturer     "Logitech"
0x02    iProduct          "USB-PS/2 Optical Mouse"
0x00    iSerialNumber
0x01    bNumConfigurations
```

USB Enumeration Example


```
Configuration Descriptor:
0x09    bLength
0x02    bDescriptorType
0x0022  wTotalLength    (34 bytes)
0x01    bNumInterfaces
0x01    bConfigurationValue
0x00    iConfiguration
0xA0    bmAttributes
(Bus-powered Device, Remote-Wakeup)
0x31    bMaxPower       (98 mA)
```

USB Enumeration Example


```
Interface Descriptor:
-----------------------------------------
0x09    bLength
0x04    bDescriptorType
0x00    bInterfaceNumber
0x00    bAlternateSetting
0x01    bNumEndPoints
0x03    bInterfaceClass
(Human Interface Device Class)
0x01    bInterfaceSubClass
0x02    bInterfaceProtocol
0x00    iInterface
```

USB Enumeration Example


```
Endpoint Descriptor:
-------------------------------------------------
0x07     bLength
0x05     bDescriptorType
0x81     bEndpointAddress  (IN endpoint 1)
0x03     bmAttributes
(Transfer: Interrupt / Synch: None / Usage: Data)
0x0008   wMaxPacketSize    (1 x 8 bytes)
0x0A     bInterval         (10 frames)
```

USB Enumeration

Once enumeration is done, the host has enough information to configure the device and load the correct driver...

But what about the end utility of the device? How to use a specific type of device?

USB Enumeration Example

Interface Descriptor:

...

0x03 bInterfaceClass (Human Interface Device Class)

...

Classes

Multiple classes of device possible, including:

- **CDC**: Communication Device Class: Serial to USB device, Ethernet adapter, ...
- **HID**: Human Interface Device: mouse, keyboard, joystick, ...
- **PTP/MTP**: picture transfer, webcam, ...
- **MSC/UMS**: mass storage devices (USB storage key, ...)

And many more...

They regroup devices that offers the same function, to offer the possibility of a **generic driver** handling them in the host!

Class-specific descriptors

Once again, descriptors are used to give information about the device functionality to the host:

- **HID class descriptor**: its added to the regular interface descriptor
- **HID Report Descriptor**: it gives information of the report format given by the device to the host

The report contains the actual data sent to the host! For instance, the button clicked on a mouse, the relative movement of the mouse or a keystroke on a keyboard!

USB Enumeration Example - HID Report Descriptor


```
05 01 09 02 a1 01 09 01 a1 00 05 09 19 01 29 08
15 00 25 01 95 08 75 01 81 02 95 00 81 03 06 00
ff 09 40 95 02 75 08 15 81 25 7f 81 02 05 01 09
38 15 81 25 7f 75 08 95 01 81 06 09 30 09 31 16
01 80 26 ff 7f 75 10 95 02 81 06 c0 c0
```

USB Enumeration Example - HID Report Descriptor


```
0x05, 0x01,        // Usage Page (Generic Desktop Ctrls)
0x09, 0x02,        // Usage (Mouse)
0xA1, 0x01,        // Collection (Application)
0x09, 0x01,        //   Usage (Pointer)
0xA1, 0x00,        //   Collection (Physical)
0x05, 0x09,        //     Usage Page (Button)
0x19, 0x01,        //     Usage Minimum (0x01)
0x29, 0x08,        //     Usage Maximum (0x08)
0x15, 0x00,        //     Logical Minimum (0)
0x25, 0x01,        //     Logical Maximum (1)
0x95, 0x08,        //     Report Count (8)
0x75, 0x01,        //     Report Size (1)
0x81, 0x02,        //     Input (Data,Var,Abs,No Wrap,Linear,Preferred State,N
0x95, 0x00,        //     Report Count (0)
0x81, 0x03,        //     Input (Const,Var,Abs,No Wrap,Linear,Preferred State,
```

USB Enumeration Example - HID Report Descriptor


```
0x06, 0x00, 0xFF,  //     Usage Page (Vendor Defined 0xFF00)
0x09, 0x40,        //     Usage (0x40)
0x95, 0x02,        //     Report Count (2)
0x75, 0x08,        //     Report Size (8)
0x15, 0x81,        //     Logical Minimum (-127)
0x25, 0x7F,        //     Logical Maximum (127)
0x81, 0x02,        //     Input (Data,Var,Abs,No Wrap,Linear,Preferred State,N
```

USB Enumeration Example - HID Report Descriptor


```
0x05, 0x01,         //     Usage Page (Generic Desktop Ctrls)
0x09, 0x38,         //     Usage (Wheel)
0x15, 0x81,         //     Logical Minimum (-127)
0x25, 0x7F,         //     Logical Maximum (127)
0x75, 0x08,         //     Report Size (8)
0x95, 0x01,         //     Report Count (1)
0x81, 0x06,         //     Input (Data,Var,Rel,No Wrap,Linear,Preferred State,N
0x09, 0x30,         //     Usage (X)
0x09, 0x31,         //     Usage (Y)
0x16, 0x01, 0x80,   //     Logical Minimum (-32767)
0x26, 0xFF, 0x7F,   //     Logical Maximum (32767)
0x75, 0x10,         //     Report Size (16)
0x95, 0x02,         //     Report Count (2)
0x81, 0x06,         //     Input (Data,Var,Rel,No Wrap,Linear,Preferred State,N
0xC0,               //   End Collection
0xC0,               // End Collection
// 77 bytes
```

USB Enumeration Example - HID Report

Buttons X move

0x 01 FC 02 00 FC FF 02 00

Wheel Y move

A few USB Host options

Fig. 3: Nucleo F446RE

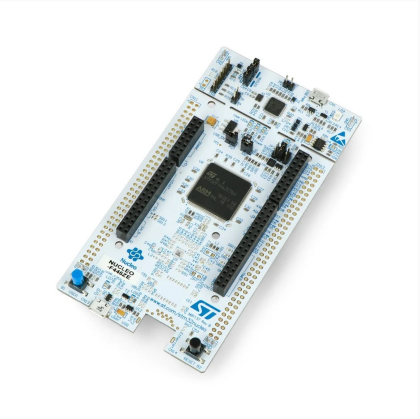


Fig. 4: Nucleo F446ZE

A few USB Host options

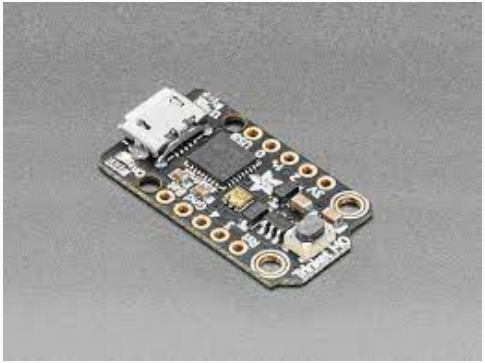


Fig. 5: Adafruit Trinket m0

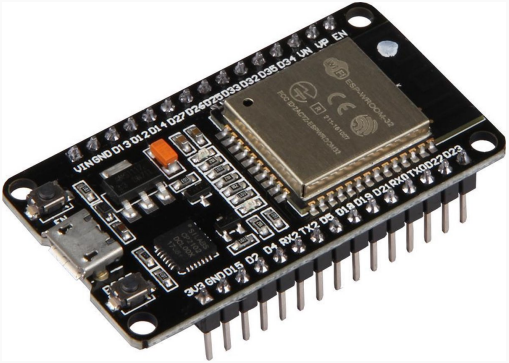
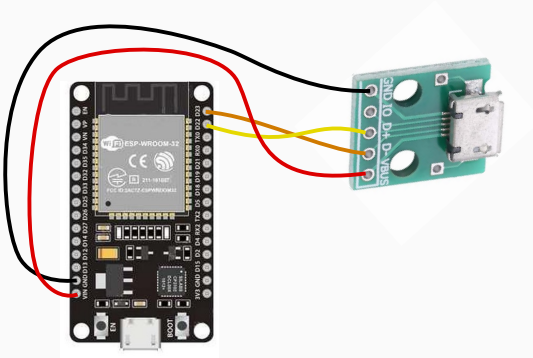


Fig. 6: ESP32 module - **Emulated** USB Host

A few USB Host options

Fig. 7: Raspberry Pi pico - Native and emulated (with PIOs)

Demo time



Resources

The (online) bible on the USB protocol:
https://www.beyondlogic.org/usbnutshell/usb1.shtml

French translation:
http://www.abcelectronique.com/acquier/usb1_fr.htm

More french information:
http://www.rennes.supelec.fr/ren/fi/elec/docs/usb/hid.html

STM32 USB Training:
https://www.st.com/content/st_com/en/support/learning/stm32-education/stm32-moocs/STM32-USB-training.html

Resources

HID official documentation:
https://www.usb.org/sites/default/files/documents/hid1_11.pdf

To create HID descriptors:
https://www.usb.org/document-library/hid-descriptor-tool

To parse USB descriptors:
https://eleccelerator.com/usbdescreqparser/

Tool to investigate descriptors of a USB device:
https://www.thesycon.de/eng/usb_descriptordumper.shtml