USB Hardware Keylogger

Nicolas Hureau

kalenz@lse.epita.fr
`http://lse.epita.fr`

February 12, 2013

Plan

1. Introduction

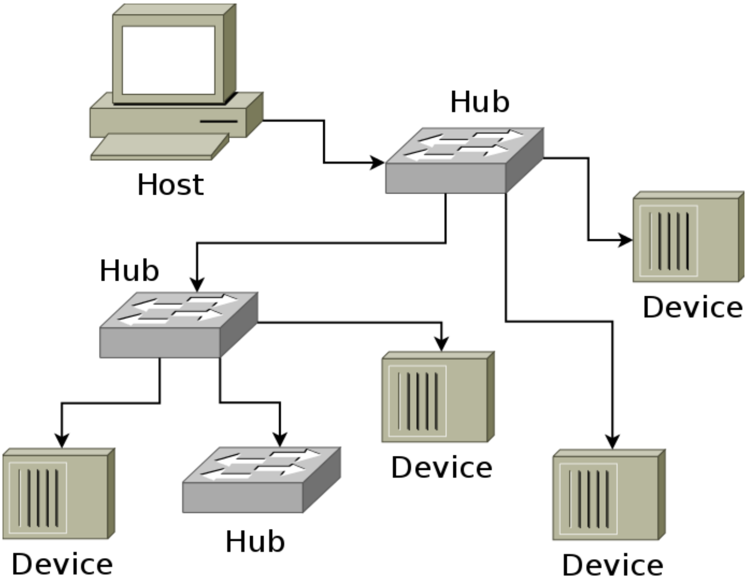
BS intro slide

- **Universal Serial Bus**
- Standard with multiple versions
- Developped mid-90s
- Designed for connection, communication and power supply

Architecture

- Single host controller
- Up to 127 slave devices connected (7-bits address)
- Tiered star topology

Topology



Plan

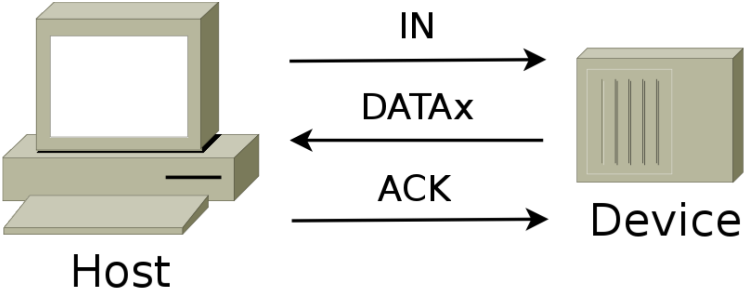
2 USB

- Basics
- Device configuration
- Transfers

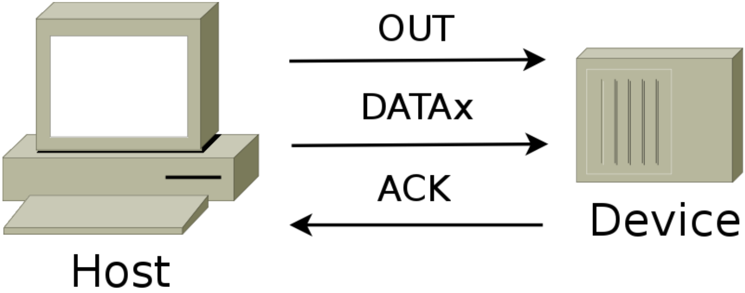
Info

- We will mostly focus on the USB protocol, ignoring lower levels
- All communications on the bus are initiated by the host

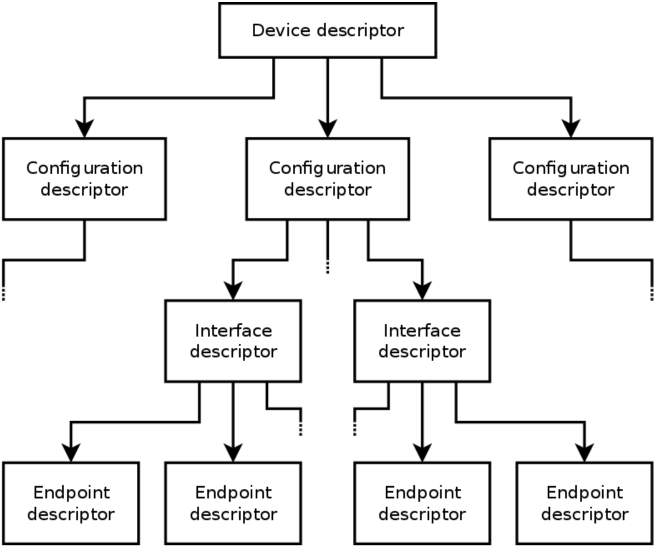
Host requesting data from device



Host pushing data to device



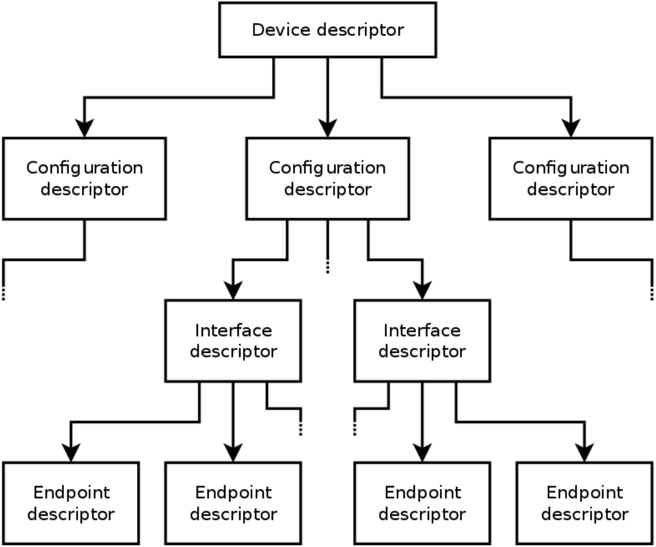
Global configuration



Device descriptor

- idVendor
- idProduct
- bNumConfiguration
- bDeviceClass, bDeviceSubClass, bDeviceProtocol
- iManufacturer, iProduct, iSerialNumber
- . . .

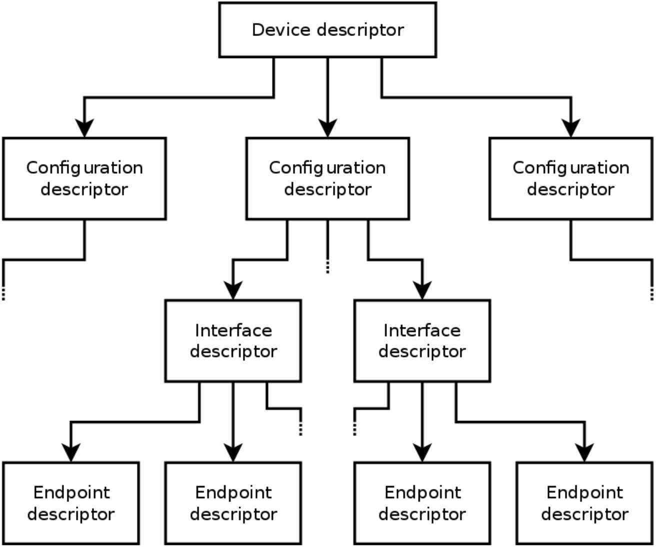
Global configuration



Configuration descriptor

- bNumInterface
- . . .

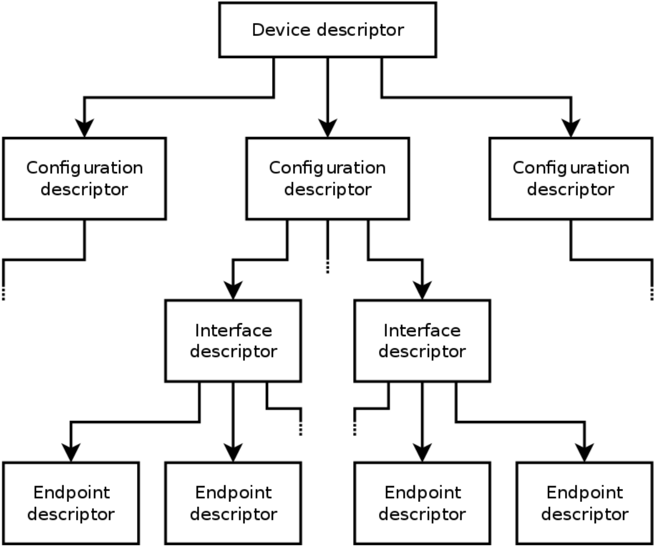
Global configuration



Interface descriptor

- bInterfaceNumber
- bInterfaceClass, bInterfaceSubClass, bInterfaceProtocol
- bNumEndpoints
- bAlternateSetting
- ...

Global configuration



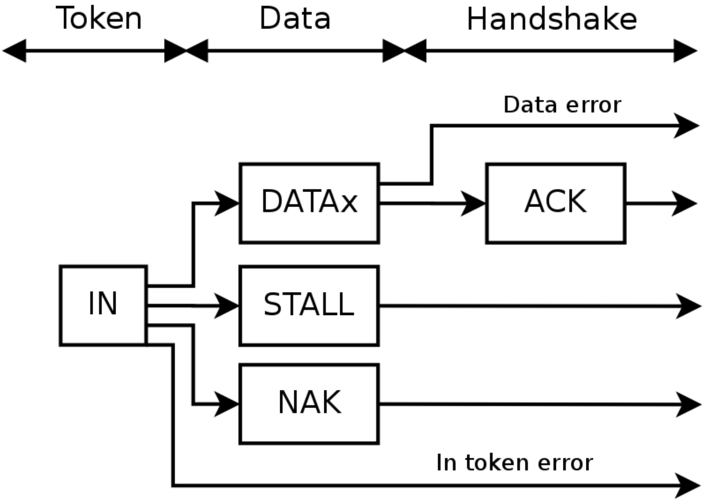
Endpoint descriptor

- bEndpointAdress
- wMaxPacketSize
- bInterval
- . . .

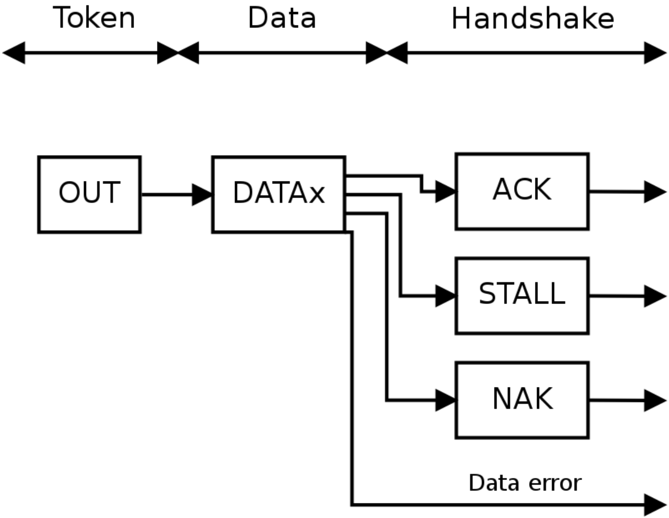
Transfer types

- Control (device setup)
- Interrupt (guaranteed bandwidth, polled by the host)
- Isochronous (guaranteed bandwidth, but no delivery guaranty)
- Bulk (large transfer, no guaranteed bandwidth)

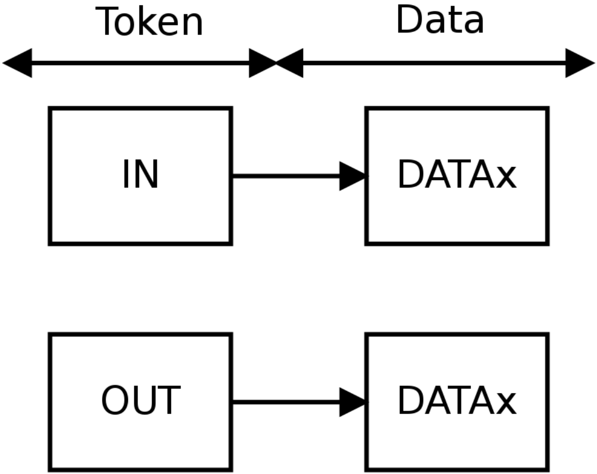
Interrupt IN



Interrupt OUT



Isochronous



Plan

3 HID

- Types
- Keyboard

HID Types

- **Human Interface Device**
- Part of the USB specification dealing with devices such as keyboards, mice and game controllers
- Also mention lots of other devices:
  - Simulation controls
  - Alphanumeric displays
  - Medical instruments
  - . . .

Report Descriptor

- Describe the format of device messages
- Use "Usage Tables" to do so:
  - 150 page documents
  - Standardized controls for devices mentioned earlier

Report Descriptor

```
Usage Page (Generic Desktop),
Usage (Keyboard),
Collection (Application),
  Usage Page (Key Codes);
  Usage Minimum (224),
  Usage Maximum (231),
  Logical Minimum (0),
  Logical Maximum (1),
  Report Size (1),
  Report Count (8),
  Input (Data, Variable, Absolute),  ;Modifier byte
  Report Count (1),
  Report Size (8),
  Input (Constant),    ;Reserved byte
  Report Count (5),
  Report Size (1),
  Usage Page (Page# for LEDs),
  Usage Minimum (1),
  Usage Maximum (5),
  Output (Data, Variable, Absolute), ;LED report
  Report Count (1),
  Report Size (3),
  Output (Constant),   ;LED report padding
  Report Count (6),
  Report Size (8),
  Logical Minimum (0),
  Logical Maximum(101),
  Usage Page (Key Codes),
  Usage Minimum (0),
  Usage Maximum (101),
  Input (Data, Array),    ;Key arrays (6 bytes)
End Collection
```

Report Descriptor

```
05 01
09 06
A1 01
05 07
19 E0
29 E7
15 00
25 01
75 01
95 08
81 02
95 01
75 08
81 01
95 05
75 01
05 08
19 01
29 05
91 02
95 01
75 03
91 01
95 06
75 08
15 00
25 65
05 07
19 00
29 65
81 00
C0
```

Report Descriptor

```
+----------------+-----------+-----------+
| Modifiers (1B) | LEDs (1B) | Keys (6B) |
+----------------+-----------+-----------+
```

Keyboard

- bDescriptorType = DT_DEVICE (0x1)
- bDeviceClass = CLASS_PER_INTERFACE (0x0)
- bInterfaceClass = CLASS_HID (0x3)
- bInterfaceSubClass = CLASS_HID_BOOT_PROTOCOL (0x1)
- bInterfaceProtocol = HID_KEYBOARD (0x1)

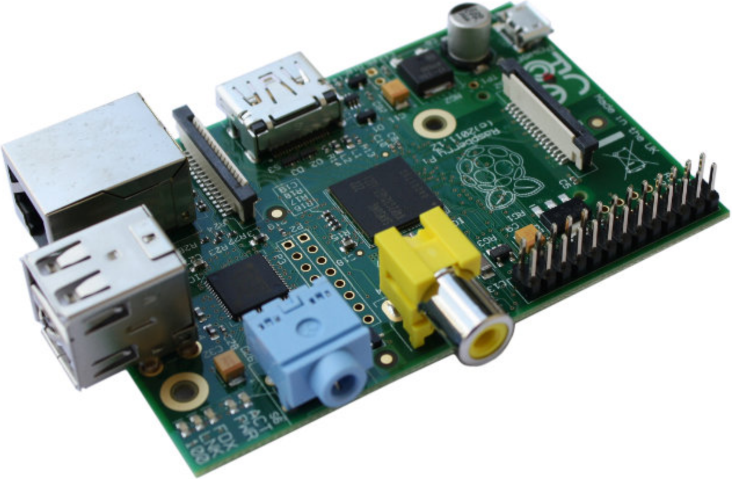
Plan

4 Keylogger
- Sniffer
- Keyboad emulator
- Misc

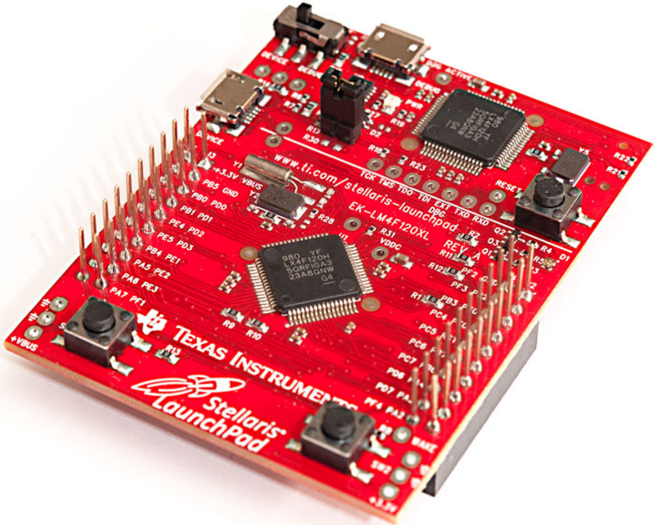
Software

- Using libusb(x) through pyusb:
  - Enumerate keyboards
  - Claim the first one
  - Listen to what is typed

Hardware



Hardware


www.ti.com/stellaris-launchpad
EK-LM4F120XL
REV A
Texas Instruments
Stellaris LaunchPad
RESET
DEVICE
DEBUG
+VBUS
+3.3V

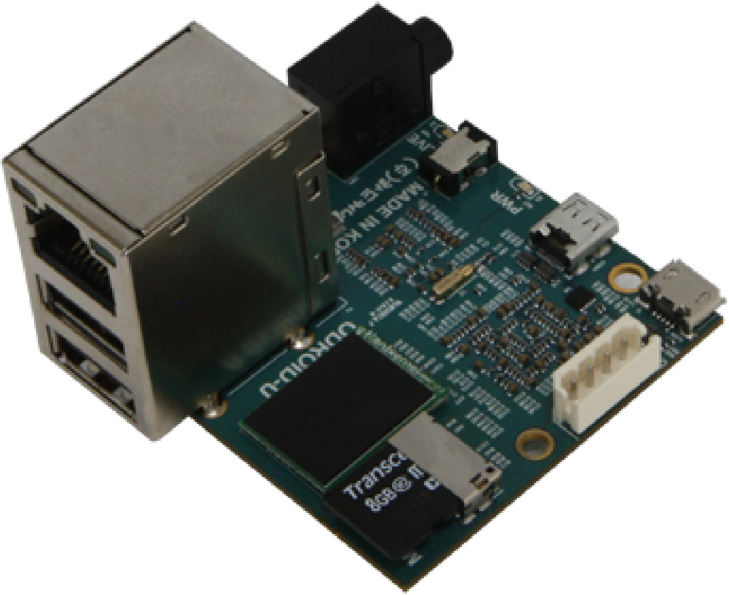
Software

- Using Stellaris SDK
  - Register as a keyboard
  - Print stuff hen pressing a button

Missing feature

- Passing status from Pi to Stellaris obviously
- Given available intefaces, donc through serial
- Should be straightforward, Ti gives helper functions

Other solutions



Plan

5 Conclusion

That’s all folks!

Questions?

@kalenz
http://bitbucket.org/kalenz