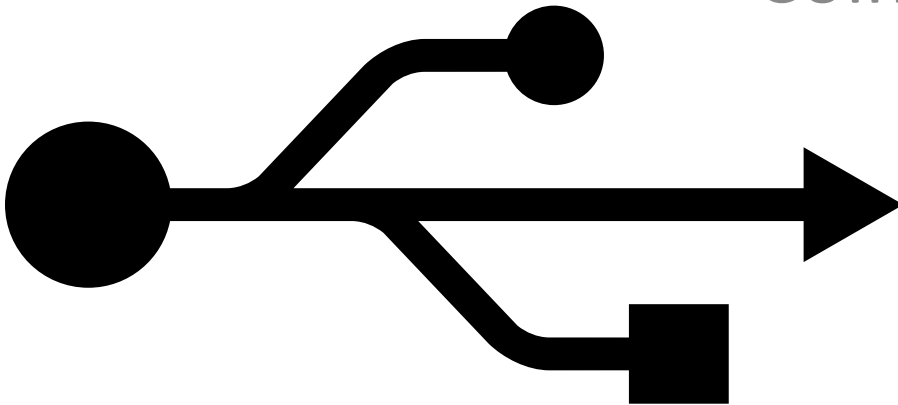


USB device on embedded systems

Coimbra BIP Workshop 2024

Thomas Herpoel - herpoelt@helha.be

Laurence Baclin – blaclinl@helha.be



Workshop organization

Digipad: interactive ressources

<https://tinyurl.com/3rzfhwzv>

Code: BIP2024



Workshop organization

Github repository

https://github.com/ThomHPL/USB_Workshop_BIP2024



Workshop organization

- Session Goal
- Toolchain setup
 - Software configuration
 - Hardware setup
- Proof Of Concept

Session Goal

Adapt a USB Low Speed HID device to a Bluetooth HID device

- Get the USB HID reports on the ESP32
- Decode the HID reports into structured data
- Emulate a BLE HID device with the ESP32

Software configuration

- Wireshark
 - <https://www.wireshark.org/download.html>
- Install USBPcap when asked by the setup



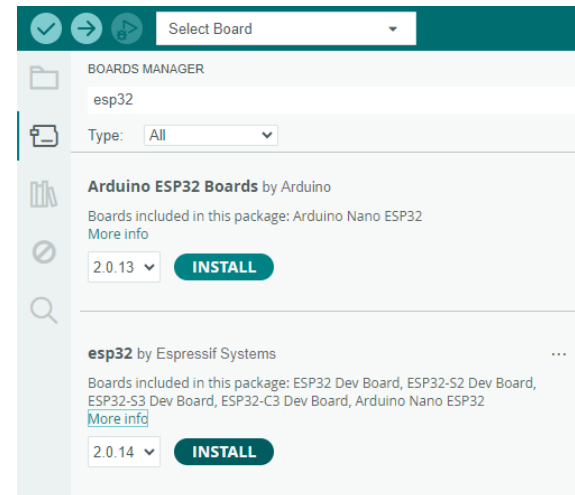
Software configuration

- Arduino IDE
 - <https://www.arduino.cc/en/software>



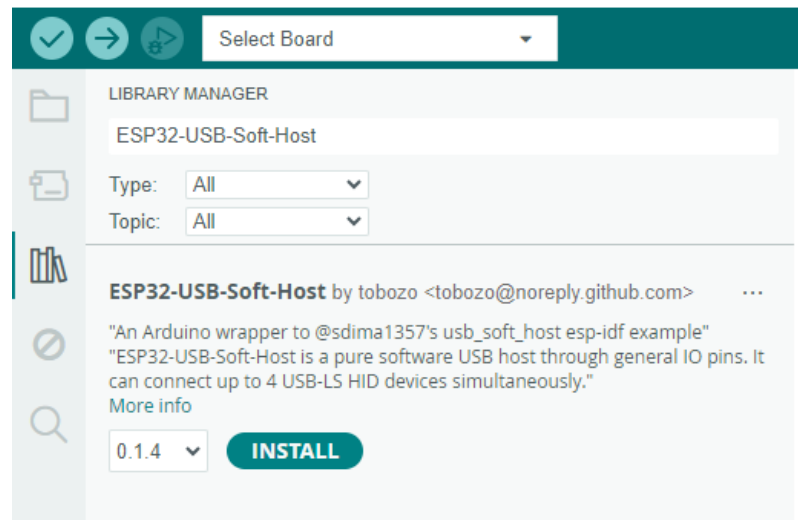
Software configuration

- ESP32 support in Arduino IDE
 - File → Preferences
 - Add to “Additional boards manager”:
 - https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
 - Boards
 - Search for “esp32”
 - Install “esp32 by Espressif”



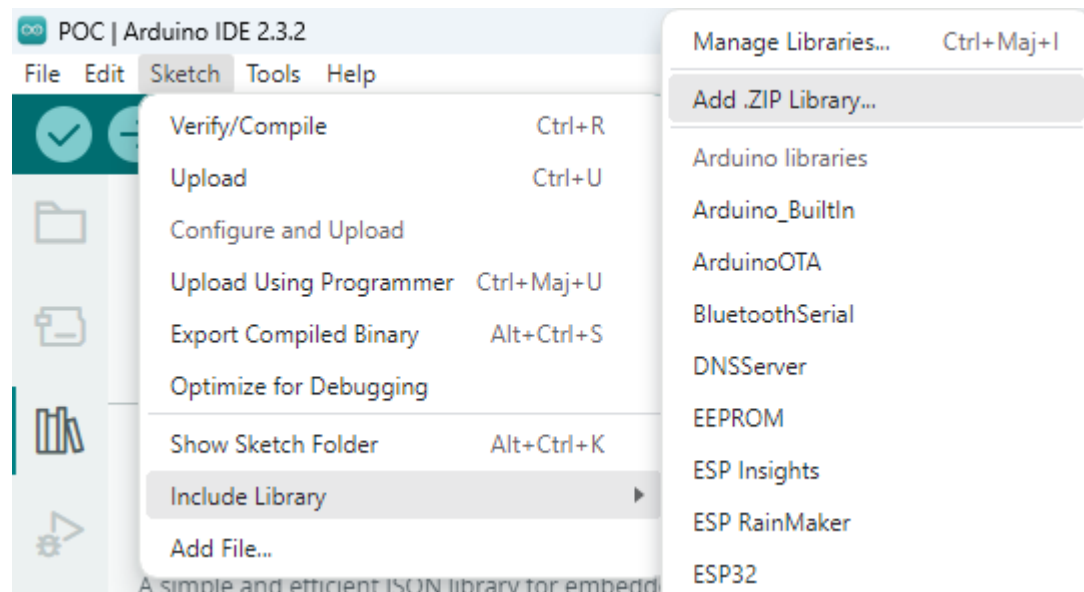
Software configuration

- ESP32 USB Host Library
 - Library Manager
 - Search for “ESP32-USB-Soft-Host”
 - Install “ESP32-USB-Soft-Host by tobozo”



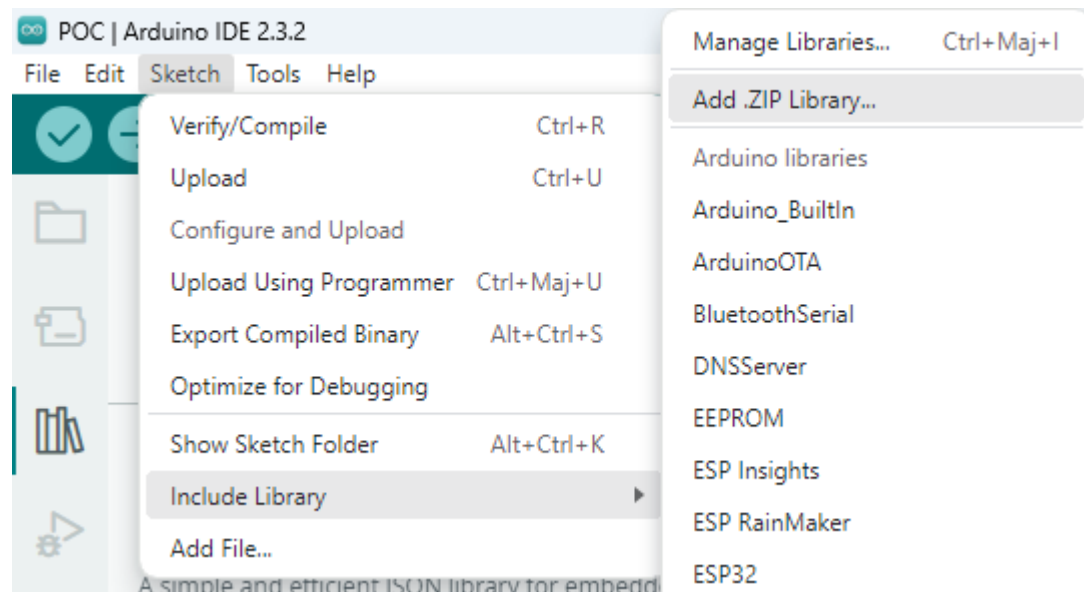
Software configuration

- ESP BLE Keyboard library
 - <https://github.com/T-vK/ESP32-BLE-Keyboard>
 - Download library as a ZIP archive
 - Install it through the Arduino IDE

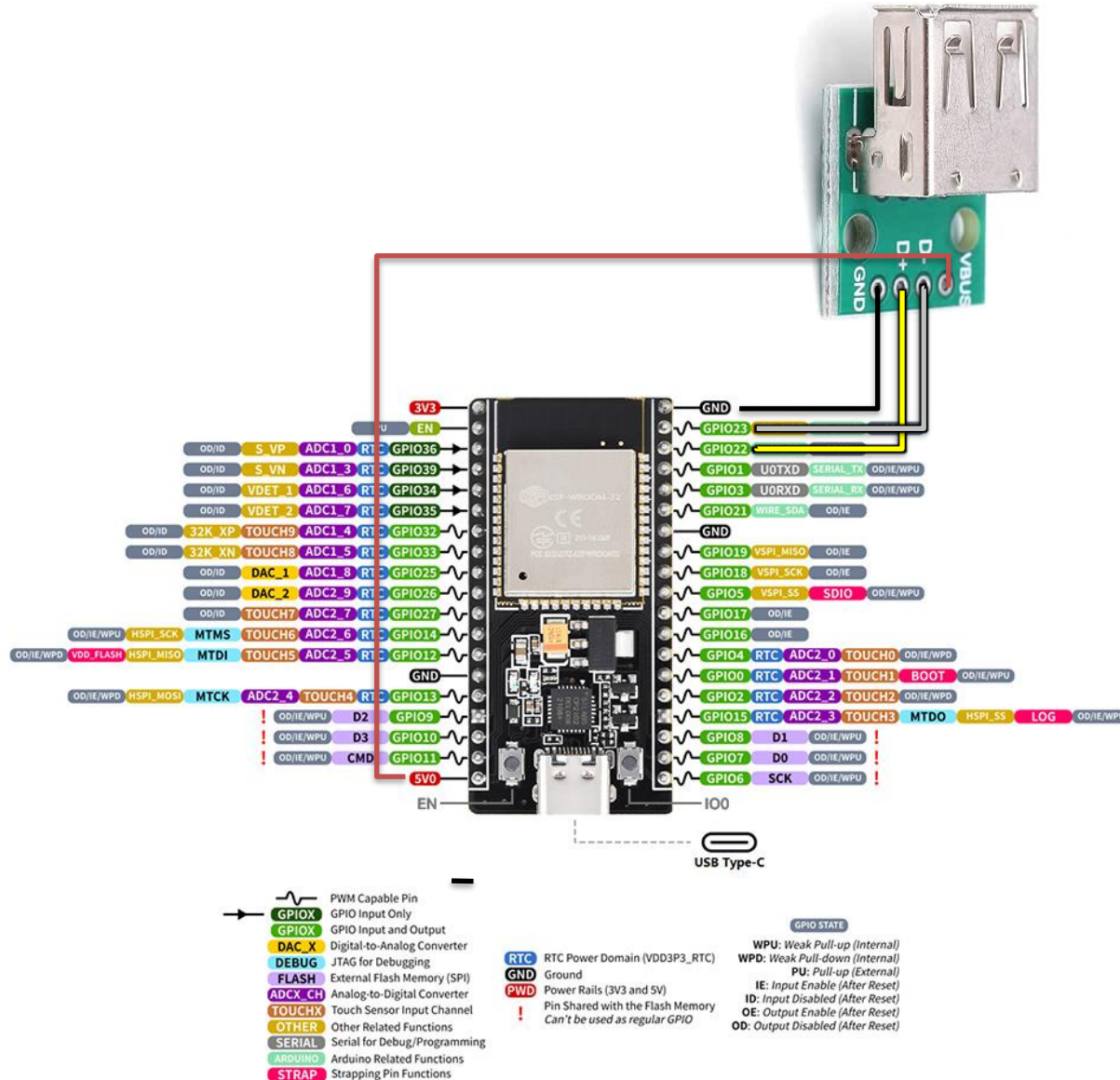


Software configuration

- ESP BLE Mouse library
 - <https://github.com/T-vK/ESP32-BLE-Mouse>
 - Download library as a ZIP archive
 - Install it through the Arduino IDE



Hardware setup



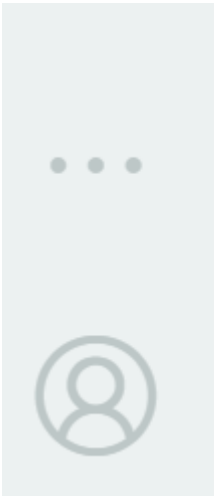
Getting reports on the ESP32

- Use the USB_Test example from ESP32-USB-Soft-Host library to get descriptors and raw data
- Notice **my_USB_PrintCB**: a callback function that is called on each received HID report.

```
static void my_USB_PrintCB(uint8_t usbNum, uint8_t byte_depth, uint8_t* data, uint8_t data_len)
{
    // if( myListenUSBPort != usbNum ) return;
    printf("in: ");
    for(int k=0;k<data_len;k++) {
        printf("0x%02x ", data[k] );
    }
    printf("\n");
}
```

Getting reports on the ESP32

- After compiling and uploading, you should then receive the descriptors and the raw data from the device on the serial terminal.



```

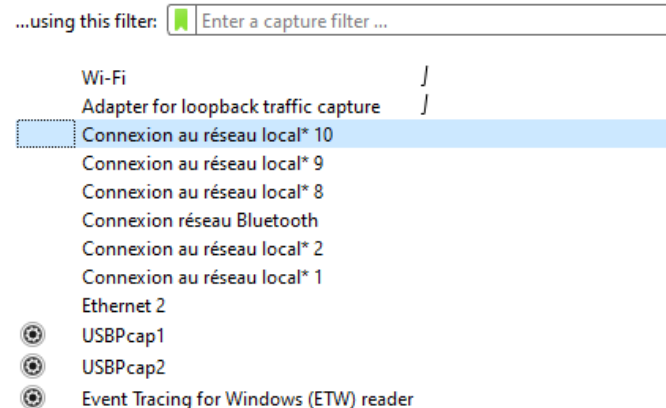
...
in:  0x01  0x02  0x00  0x00
in:  0x01  0x01  0x00  0x00
in:  0x01  0x01  0xff  0x00
in:  0x01  0x01  0xff  0x00
in:  0x01  0x01  0x00  0x00
  
```

➤ **How to interpret the data?**

Decoding HID reports

- **We need the report descriptor!**
 - Use Wireshark to sniff on the USB enumeration
 - Select USBPcap interface

Capture



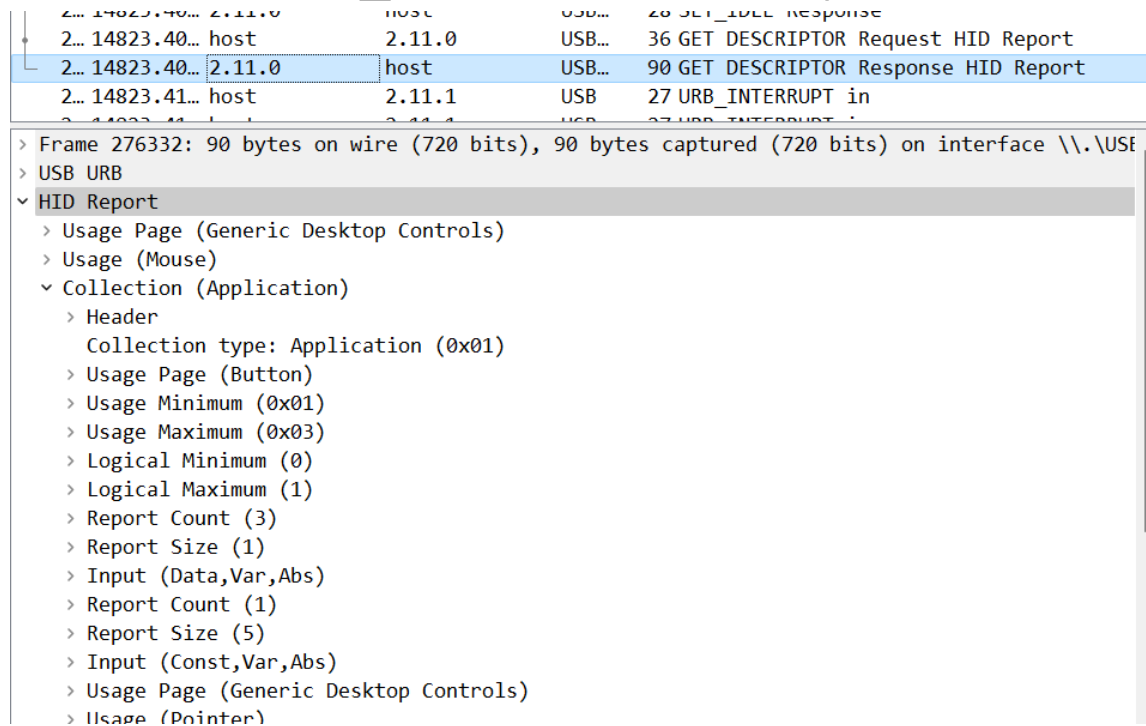
Learn

[User's Guide](#) · [Wiki](#) · [Questions and Answers](#) · [Mailing Lists](#) · [SharkF](#)

You are running Wireshark 4.2.3 (v4.2.3-0-ga15d7331476c). You receive automatic

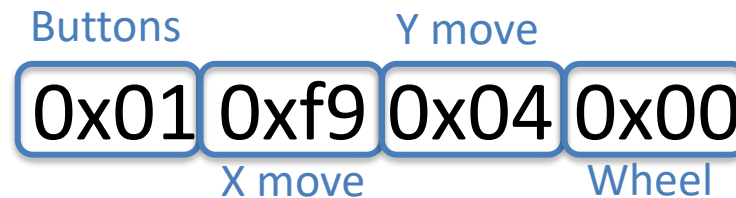
Decoding HID reports

- How to get the report descriptor?
 - Plug in your USB device into the Extract HID Descriptor
 - Check for the GET_DESCRIPTOR Response



Decoding HID reports

- Compare the reports given by the esp32 with the report descriptor
 - Check <https://docs.kernel.org/hid/hidintro.html> for help understanding the report



- Write a structure typedef that matches this report format

Emulating BLE HID device

- We'll use the following libraries:
 - <https://github.com/T-vK/ESP32-BLE-Mouse>
 - <https://github.com/T-vK/ESP32-BLE-Keyboard>
- Once installed, checkout the example codes
 - MouseButtons
 - SendKeyStrokes
- Integrate everything into a single code that
 - Uses the USB host library to get reports from your HID device
 - Formats the data into a structure
 - Uses the structure to send BLE HID data

Emulating BLE HID device

- **Important!**

- When integrating USB Host and BLE into the same code, this should be added in the code **before** the USB Host initialization!

```
// necessary for USB to work with BLE  
USH.setISRAllocFlag(NULL);  
USH.init(USB_Pins_Config);
```

- BLE library has to begin **after** the USB initialization!

```
// has to begin AFTER USB host  
bleMouse.begin();
```

Emulating BLE HID device

- Example for BleMouse, use the class functions to translate the HID report structure into BLE actions

```
void click(uint8_t b = MOUSE_LEFT);
void move(signed char x, signed char y, signed char wheel = 0,
signed char hWheel = 0);
void press(uint8_t b = MOUSE_LEFT);    // press LEFT by default
void release(uint8_t b = MOUSE_LEFT);  // release LEFT by default
bool isPressed(uint8_t b = MOUSE_LEFT); // check LEFT by default
bool isConnected(void);
void setBatteryLevel(uint8_t level);
```