FLipMouse/Pad Arduino Initialisation

Fabrication Note v1.0, AsTeRICS Foundation

Scope

FLipMouse and FLipPad use from version 3 onward an **Arduino Nano RP2040 Connect** platform.

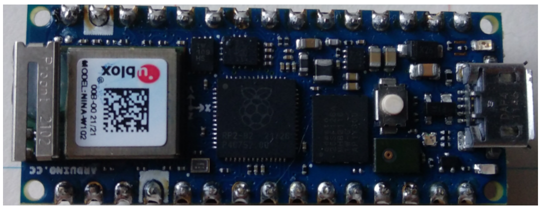
The big advantage over the TeensyLC from version 2, this controller has much more RAM & ROM and already contains the ESP32, which is used for the Bluetooth connectivity.

To fully use the update procedures from the WebGUI, the Arduino needs to be programmed initially:

- **esp32_addon_bootloader:** Bootloader Code to update the ESP32 BLE Mouse/Keyboard firmware without esptool.py
- **esp32_mouse_keyboard:** Program the BLE HID over GATT firmware
- **FLipMouse/FLipPad firmware:** Program the RP2040 with the current FLipMouse / FLipPad Firmware

Preparation

Material

Nr.	Description	Source	Image
1	Arduino Nano RP2040 Connect	https://store.arduino.cc/products/arduino-nano-rp2040-connect	

Note: Material will be referenced in square brackets: []

Tools / Requirements

Nr.	Description	Source
1	Terminal + Python3	Install python according to your OS
2	esptool	Install via pip: `pip install esptool` (if it cannot be executed from a terminal, run this command as root)
2	Permanent marker (red/blue/green)	DYI store, paper store (possibly any store)

Note: Tools will be referenced in curly brackets: {}

Procedure

1. Attach the Arduino Nano RP2040 Connect *[1]* to the computer
2. Open a terminal
3. Call the script: `rp2040_prepare.py -t <FM/FP> -p <serial port>`

-t Select the firmware to be flashed, either FP for FLipPad, FM for FLipMouse of FB for FABI (FP & FB are not finished yet)

-p Select a serial port which should be flashed (normally COMxx on Windows, /dev/ttyxxx on Linux)

Testing

Not available, if procedure is followed, the software is flashed correctly:

- **Blue blinking** LED for a flashed Bluetooth firmware

Documentation

For each produced batch, fill out one document **template_arduino_init_production.ots** and save it as: `arduino_init_<date>.ods`(e.g.: arduino_init_20221118.ods)

Updating the firmware builds to be flashed

1. Replace `bootloader.bin`,`esp32_addon_bootloader.bin`, `ota_initial_data.bin` and `partition-table.bin`with a current build from: *esp32_addon_bootloader/build/esp32_addon_bootloader.bin*
2. Replace `mousekeyboard.bin` with a current build from: *esp32_mouse_keyboard/build/esp32_mouse_keyboard.bin*
3. Replace `FM.uf2`, `FP.uf2`, `FB.uf2` with current builds from the FLipWare/FabiWare repositories (build with Arduino and select *Sketch->Export compiled binary*)
4. Replace `serialflasher1.uf2` with a new build from the serialflasher1.ino sketch from this directory (build with Arduino and select *Sketch->Export compiled binary*)
5. Replace `serialflasher2.uf2` with a new build from the serialflasher2.ino sketch from this directory (build with Arduino and select *Sketch->Export compiled binary*)
6. **Write down the current GIT tags (releases) or commit numbers for the builds to *VERSIONS.md***

Insights

This tool performs following steps:

1. Flashing the serialflasher1.uf2 firmware calling `uf2conv.py`tool
2. Flashing the esp32_addon_bootloader firmware with `esptool.py`
3. Flashing the serialflasher2.uf2 firmware calling `uf2conv.py` tool
4. Flash the esp32_mouse_keyboard_firmware (included in rp2040_prepare.py)

5. Reset the Arduino to UF2 download mode by opening the given serial port with 1200Baud and closing it
6. Flashing the FM/FB/FP.uf2 firmware calling `uf2conv.py`tool
7. Verifying that everything worked by sending "AT BC $ID" to the serial port, which returns the version of the BLE module. This is printed on the command line.
8. Start over (wait until serial port is removed and a new one is detected)