

The default branch for this project has been changed. Please update your bookmarks.
main

- [bluepad32](#)
- [docs](#)
- **[plat_nina.md](#)**

[Find file](#)[Blame](#)[History](#)[Permalink](#)



fix backticks shell

[Ricardo Quesada](#) authored 1 month ago

7bc2790f

plat_nina.md 5.93 KiB

[Blame](#)

[Edit](#)

Bluepad32 firmware for NINA

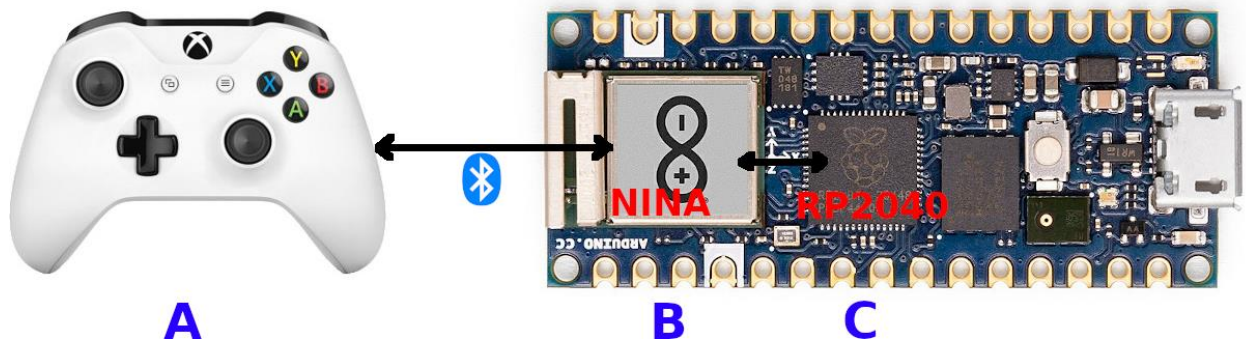
What is NINA

NINA is a family of [ESP32 modules](#). These modules are present on some Arduino boards like:

- [Arduino Nano RP2040 Connect](#)
- [Arduino Nano 33 IoT](#)
- [Arduino MKR WiFi 1010](#)
- [Arduino UNO WiFi Rev.2](#)
- [Arduino Arduino MKR Vidor 4000](#)

NINA modules are co-processors, usually used only to bring WiFi or BLE to the main processor.

In order to have gamepad support, the original NINA firmware must be replaced with Bluepad32 firmware. This is a simple step that needs to be done just once, and can be "undone" at any time.



This is how it works:

- Gamepad (A) talks to NINA module (B)
- NINA module (B) talks to main processor (C)

Bluepad32 firmware is "compatible-enough" with the original firmware:

- Uses SPI, and the same GPIOs to talk to the main processor
- Uses the same protocol that runs on top of SPI
- But not all messages are implemented. Only the ones that are needed to have gamepad support working.

Flashing pre-compiled Bluepad32 firmware

To flash Bluepad32 firmware, you have to:

Download latest pre-compiled Bluepad32 firmware for NINA

Download latest binary from here: <https://gitlab.com/ricardoquesada/bluepad32/-/releases>

- Download the file with "nina" in its name. E.g: `bluepad32-nina-x.y.z.tar.gz`
- Uncompress it using [7-zip](#), or from command line (`cmd.exe`):

```
tar -xf bluepad32-nina-x.y.z.tar.gz
```

```
cd bluepad32-nina-x.y.z
```

```
dir
```

- And you should see a file named `bluepad32-nina-x.y.z.bin` (or similar). You will use it later. Keep reading.

```
ESP-IDF 4.4 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-7a538bcf490d7cccefd2fcb09abae4f
C:\Users\riq\Downloads>
C:\Users\riq\Downloads>
C:\Users\riq\Downloads>
C:\Users\riq\Downloads>
C:\Users\riq\Downloads>tar -xf bluepad32-nina-v3.9.1.tar.gz
C:\Users\riq\Downloads>cd bluepad32-nina-v3.9.1
C:\Users\riq\Downloads\bluepad32-nina-v3.9.1>dir
Volume in drive C has no label.
Volume Serial Number is A007-3B54

Directory of C:\Users\riq\Downloads\bluepad32-nina-v3.9.1

10/13/2023  12:28 PM  <DIR>          .
10/13/2023  12:28 PM  <DIR>          ..
10/13/2023  12:28 PM                735,232 bluepad32-nina-full-v3.9.1.bin
10/13/2023  12:28 PM                2,417  README.md
                2 File(s)      737,649 bytes
                2 Dir(s)  630,723,833,856 bytes free

C:\Users\riq\Downloads\bluepad32-nina-v3.9.1>
```

Download arduino-fwuploader

Download latest binary from here: <https://github.com/arduino/arduino-fwuploader/releases>

Select correct board name

- `arduino:samd:mkrwifi1010` for Arduino MKR WiFi 1010
- `arduino:samd:nano_33_iot` for Arduino NANO 33 IoT
- `arduino:samd:mkrvidor4000` for Arduino MKR Vidor 4000
- `arduino:megaavr:uno2018` for Arduino Uno WiFi Rev2
- `arduino:mbed_nano:nanorp2040connect` for Arduino Nano RP2040 Connect

You can see all boards names by doing:

```
$ arduino-fwuploader firmware list
```

Flash it

Windows

You have to know:

- COM port: If you don't know which one it is, open Arduino IDE, and go to **Tools** -> **Port**: It should be something like **COM3**.
 - **VERY IMPORTANT**: Close Arduino IDE after that. The COM port must be "free". Nobody should be using it to flash the firmware.
- The board name: Choose the correct one from the list above

```
arduino-fwuploader firmware flash -b arduino:mbed_nano:nanorp2040connect -a COM3 -i P
ATH\T0\bluepad32-nina-full.bin
```

```
ESP-IDF 4.4 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-7a538bcf490d7cccefd2fcb09abae4f
C:\Espressif\frameworks\esp-idf-v4.4.6>cd \users\rig\Downloads\arduino-fwuploader_2.4.1_Windows_64bit
C:\Users\rig\Downloads\arduino-fwuploader_2.4.1_Windows_64bit>arduino-fwuploader firmware flash -b arduino:mbed_nano:nanorp2040connect -a COM3 -i bluepad32-nina-full-v3.9.1.bin
rp2040load 1.0.6 - compiled with go1.16.2
Loading into Flash: [=====] 100%
Flashing progress: 99%
Upload completed!
C:\Users\rig\Downloads\arduino-fwuploader_2.4.1_Windows_64bit>
```

Linux & macOS

Replace name and address with the correct ones

```
export BOARD=arduino:samd:nano_33_iot
```

```
export ADDRESS=/dev/ttyACM0
```

```
$ arduino-fwuploader firmware flash -b $BOARD -a $ADDRESS -i bluepad32-nina-full.bin
```

Verify

To verify that the flash was successful, do:

```
$ arduino-fwuploader firmware get-version -b $BOARD -a $ADDRESS
```

And you should see:

```
...
```

Firmware version installed: Bluepad32 for NINA v3.6.0-rc0

Flashing self-compiled Bluepad32 firmware

To flash a self-compiled firmware, you should do:

1. Put the Arduino board in "pass-through" mode
2. Compile it yourself and flash it.

1. Put Arduino board in "passthrough" mode

Before flash Bluepad32 firmware, you have to put the Arduino board in "pass-through" mode:

1. Open Arduino IDE
2. Install the WiFiNINA library (just do it once)
3. And finally open the `SerialNINAPassthrough` sketch:
 - File -> Examples -> WiFiNINA -> Tools -> SerialNINAPassthrough

Compile it and flash it to the Arduino board.

2. Compile it yourself and flash it

Install the requirements described here: [README.md](#).

Chose `nina` as the target platform:

```
cd ${BLUEPAD32}/src

# Select Nina platform:

# Components config -> Bluepad32 -> Target Platform -> Nina

idf.py menuconfig
```

```
# And then compile it!
```

```
idf.py build
```

On Nano 32 IoT / MKR WIFI 1010, doing `idf.py flash` will just work.

```
# Only valid for:
```

```
# * Nano 33 IoT
```

```
# * MKR WIFI 1010
```

```
# Port might be different
```

```
export ESPPORT=/dev/ttyACM0
```

```
idf.py flash
```

But on NANO RP2040 Connect and UNO WiFi Rev.2, you have to flash it using the `--before no_reset` option, and **NOT** `--before default_reset`. E.g:

```
# Only valid for:
```

```
# * Nano RP2040 Connect
```

```
# * UNO WiFi Rev.2
```

```
# Port might be different
```

```
export ESPPORT=/dev/ttyACM0
```

```
esptool.py --port ${ESPPORT} --baud 115200 --before no_reset write_flash 0x1000 ./build/bootloader/bootloader.bin 0x10000 ./build/bluepad32-airlift.bin 0x8000 ./build/partitions_singleapp.bin
```

Example

The Bluepad32 library for Arduino with examples is available here:

- <http://gitlab.com/ricardoquesada/bluepad32-arduino>