

# Player!

## Hardware

### Bill of Materials

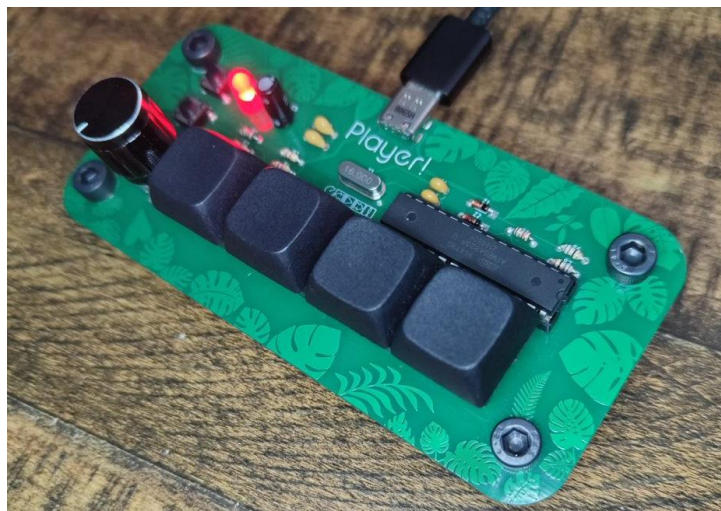
The bill of materials is hosted on Octopart, available [here](#). Almost all the components are generic and as a result can be sourced from any good electronic supplier. The only real exceptions are the USB-C connector and the ATmega328 for which there are no alternatives (as far as I know).

### Base/Case

Player! is currently housed in a 3D printed black PLA base, which holds the Player! PCB facing forwards at 10°. The base uses M5x10mm brass threaded inserts to accept M5x10mm hex head screws. The self-tapping brass inserts used to prevent damage caused by screwing/unscrewing the hex head screws directly into the base.



Whilst testing the sizing for the brass threaded inserts, I found small PLA standoffs to be quite an elegant solution, leaving the PCB “floating” ... however, these can be tricky to align at an angle.



The STL files to print your own base or angled standoffs can be found [here](#).

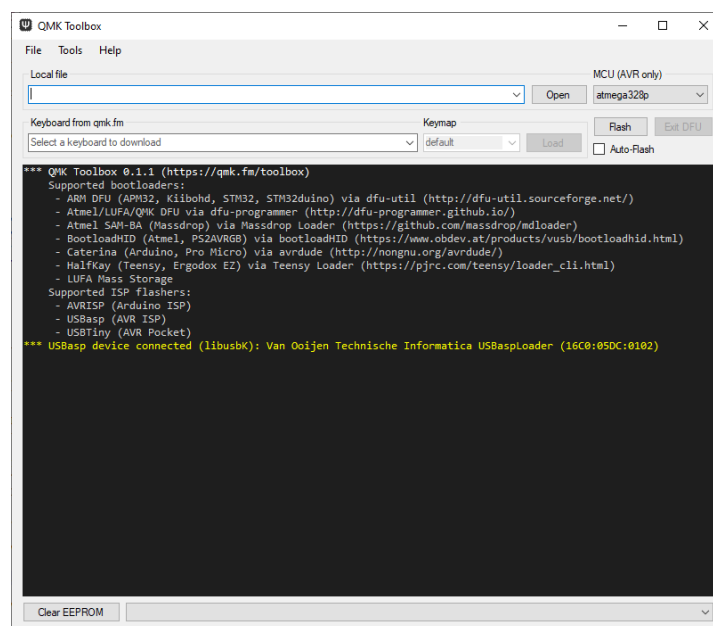
## Firmware - QMK

### Entering Bootloader Mode

Player! needs to be put into bootloader mode to allow QMK to be flashed via a USB connection:

1. Press and hold “Boot” button
2. Tap “Reset” button
3. Release “Boot” button

USBaspLoader will show up in device manager or as a connected device in QMK toolbox when the device has successfully entered bootloader mode, like so:



### Flashing Firmware

Once Player! has been successfully put into bootloader mode, the firmware can be flashed. Download the [player\\_vial.hex](#), and point QMK toolbox to its location, and then click flash. Upon completion, press the “Reset” button on the Player! and it will be ready to use with the default keybinds.

## Firmware – Bootloader Information

The Player! Uses a USB bootloader that allows the ATmega328 to program itself with QMK, namely [USBaspLoader](#).

- My flavour of the bootloader is available [here](#).
- The instructions for setting up the build environment can be found in the “readme”.
- Some changes will be required, such as changing the port or programmer definitions for your setup. These can be found on lines 9 and 44 of the makefile.inc respectively. **Never** edit the makefile directly.
- Flashing the ATmega328 once the build environment is setup:

“make flash” (flashes the makefile)

“make fuse” (sets fuses for the ATmega328)

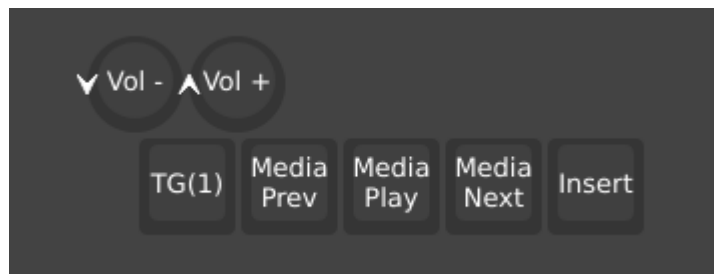
## Rebinding Keys

Player! Supports the rebinding of keys using Vial. This can be done using the downloaded vial client, or simply navigating to [vial.rocks](https://vial.rocks) in your web browser. When using the browser version, click “Start Vial” and select “johnboysmooth Player” from the pop up:

vial.rocks wants to connect to a HID device

johnboysmooth Player!

Vial should start and present the buttons and rotary knob of the Player!, which can then be rebound as required.



The two circles represent the rotary encoder, the directional arrows indicating the binds for clockwise/anti-clockwise rotation. TG(1) is for the rotary encoder switch, which by default toggles between the first and second layer. The remaining four buttons correspond to the four keyswitch buttons on the Player!.

## Future Work/Issues

Feedback is welcomed, including any issues which can be logged in the GitHub [issues](https://github.com/johnboysmooth/Player!/issues) page. Player! continues to be developed, here are some future ideas:

- RGB backlighting.
- A case with transparent/translucent edges for LED shine through.
- Choc/Cherry switch compatibility, using footprints that support either switch.
- A “true” wireless device, supporting Bluetooth connectivity as well as Qi wireless charging to completely remove the need to ever connect via USB. This would have to include a small battery.