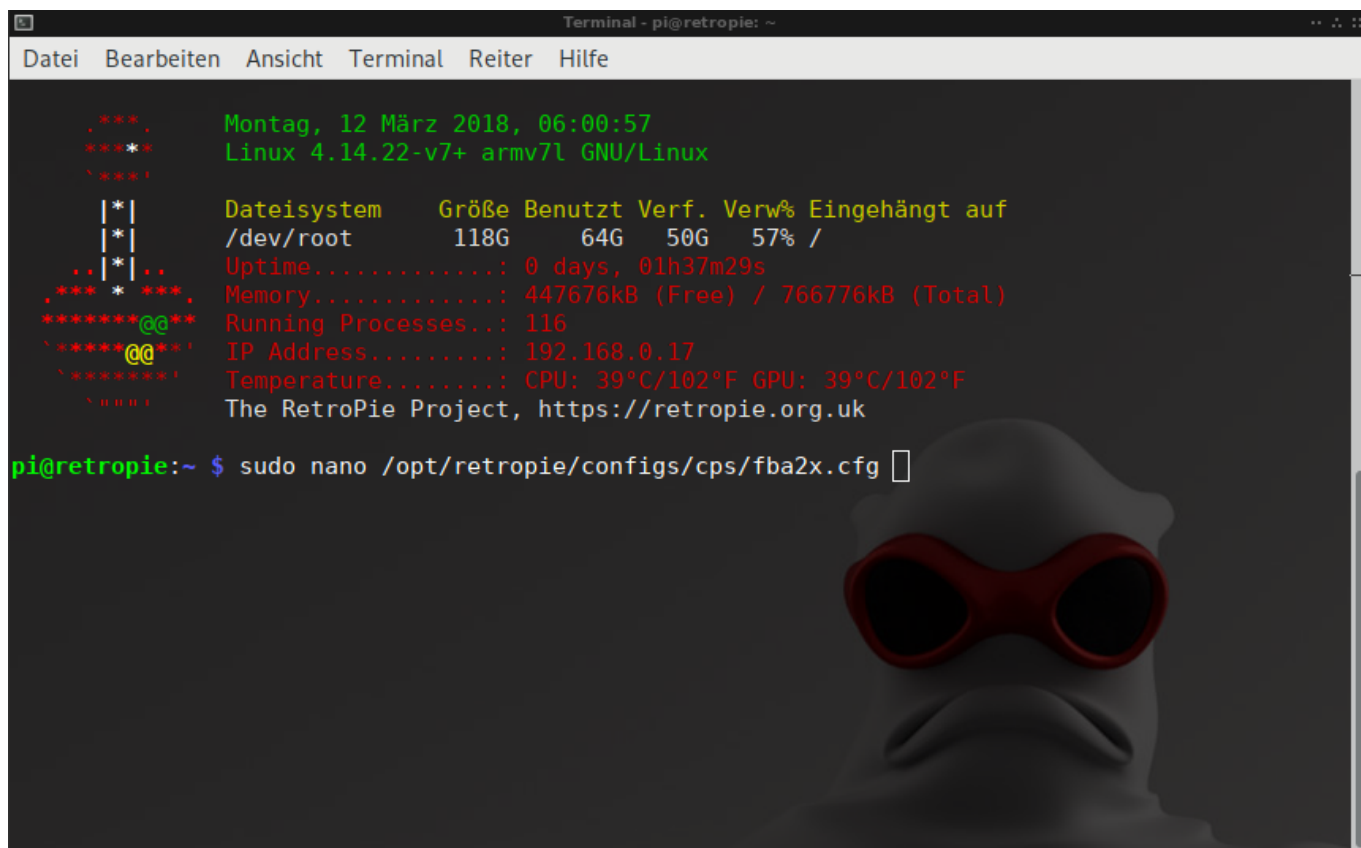


# RetroPie Capcom Joypad Mapping



Gerade Bluetooth-Joypads sind in der Hinsicht etwas zickig, wenn ein Emulator mal nicht von der Emulationsstation betreut wird. Ein Beispiel dafür ist Capcom, wenn man es als einzel System in der Emulationstation einbindet. Ebenfalls ist es nicht möglich vom Spiel aus in die retroarch zu kommen. Nun werden einige sagen, "Momentmal, das wird doch mit fba ausgeführt!" Ja das ist richtig und dennoch ist es bei RetroPie der "pifba" Emulator der diese Spiele starten tut und dieser hat noch keine Anbindung zu retroarch.

## Mapping per Hand



DATEI: /opt/retropie/configs/cps/fba2x.cfg

```
...
[Joystick]
# Get codes from "jstest /dev/input/js0"
# from package "joystick"
# Defaults to 8Bitdo SF30 Pro Controller
A_1=1
B_1=0
X_1=3
Y_1=2
L_1=4
R_1=5
START_1=9
SELECT_1=8
#Joystick axis
JA_LR=0
JA_UD=1
...

[collapse]
```

Anzupassen wäre hier Start/Select und die Aktionstasten A,B,X,Y,L,R.

Um eine Bewegung sichtbar zu machen haben die Tasten und Analogen Joysticks Zahlenwerte und genau diese Tastennummern werden bei dieser Konfiguration angegeben.

## **Tasten-Nummern anzeigen lassen**

Mit der eingabe von "ls /dev/input/" lassen sich die angeschlossenen Geräte auflisten. Joypads, ob nun über USB oder Bluetooth, werden mit "jsX" gekennzeichnet, das X steht für eine Ziffer (zB. js0 für Joypad 1 und js1 für den zweiten).

Mit dem Befehl "jstest /dev/input/js0" startet man den Test der Kalibrierung.

```

Terminal - pi@retroPie: ~
Datei Bearbeiten Ansicht Terminal Reiter Hilfe

****
**** Montag, 12 März 2018, 06:11:47
****
**** Linux 4.14.22-v7+ armv7l GNU/Linux
****

|*|
|*| Dateisystem Größe Benutzt Verf. Verw% Eingehängt auf
|*| /dev/root 118G 64G 50G 57% /
|*|
|*| Uptime.....: 0 days, 01h48m19s
****
**** * Memory.....: 447324KB (Free) / 766776KB (Total)
****
**** * Running Processes..: 119
****
**** * IP Address.....: 192.168.0.17
****
**** * Temperature.....: CPU: 39°C/102°F GPU: 39°C/102°F
****
**** * The RetroPie Project, https://retroPie.org.uk
****

pi@retroPie:~$ ls /dev/input/
by-id by-path event0 event1 event2 event3 js0 js1 mice
pi@retroPie:~$ jstest /dev/input/js0
Driver version is 2.1.0.
Joystick (8Bitdo SF30 Pro) has 8 axes (X, Y, Z, Rz, Gas, Brake, Hat0X, Hat0Y)
and 16 buttons (BtnX, BtnY, BtnZ, BtnTL, BtnTR, BtnTL2, BtnTR2, BtnSelect, BtnStart, BtnMode, BtnThumbL, BtnThumbR, ?, ?, ?, ?).
Testing ... (interrupt to exit)
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: -32767 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: -32767 1: -32767 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: -32767 1: -32767 2: -32767 3: 0 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: -32767 1: -32767 2: -32767 3: -32767 4: 0 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: -32767 1: -32767 2: -32767 3: -32767 4: -32767 5: 0 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: -32767 1: -32767 2: -32767 3: -32767 4: -32767 5: -32767 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
Axes: 0: -32767 1: -32767 2: -32767 3: -32767 4: -32767 5: -32767 6: 0 7: 0 Buttons: 0:off 1:off 2:off 3:off 4:off 5:off 6:
off 7:off 8:off 9:off 10:off 11:off 12:off 13:off 14:off 15:off █

```

Wie auf dem Bild zu sehen ist, stehen alle Tasten auf "off". Wird nun eine Taste gedrückt und gehalten, so zeigt diese den Status "on" an. Beim normalen Tastendruck (drücken und wieder loslassen), springt die jeweilige Zahl kurz auf "on" und gleich wieder auf "off".

Dementsprechend sucht man sich die Tasten heraus die man benötigt und notiert sich die Nummer dazu. Bei den Analogjoysticks ist es nicht anders, nur das dort 1 und 0 eine Rolle spielt. Da haben Links/Rechts (0) und Hoch/Runter (1) jeweils die gleiche Ziffer. Das Steuerkreuz bedarf kein extra Mapping, da diese Standard sind.