Making Open Source Hardware for Retrogaming on Raspberry Pi

Leon Anavi
Konsulko Group
leon.anavi@konsulko.com
leon@anavi.org
FOSDEM 2019

whoami | disclaimer

- ✗ Gamer
- ✗ Retrocomputing expert
- ✓ Open source enthusiast
- ✓ Software engineer

Agenda

- Designing a retro gamepad as a simple Raspberry Pi HAT
- Open source hardware certification
- Creating a device-tree overlay with fragments for mapping keys and flashing it in an EEPROM
- Turing Raspberry Pi into retro-gaming machine with RetroPie
- Conclusions

Raspberry Pi

- Low cost single board computer developed in the UK by the Raspberry Pi Foundation
- With Broadcom ARM SoC
- Available with size of credit card (85x56mm), even smaller (65x56mm and 65x30mm) or as a industrial compute module
- As of mid 2018 more than 19 million units have been sold worldwide
- Hardware Attached on Top (HAT) specifications

Gamepad

- *“A gamepad, joypad, or simply controller is a type of game controller held in two hands, where the fingers (especially thumbs) are used to provide input.”* **Wikipedia**
- Let’s design a low-cost open source hardware gamepad as a Raspberry Pi add-on

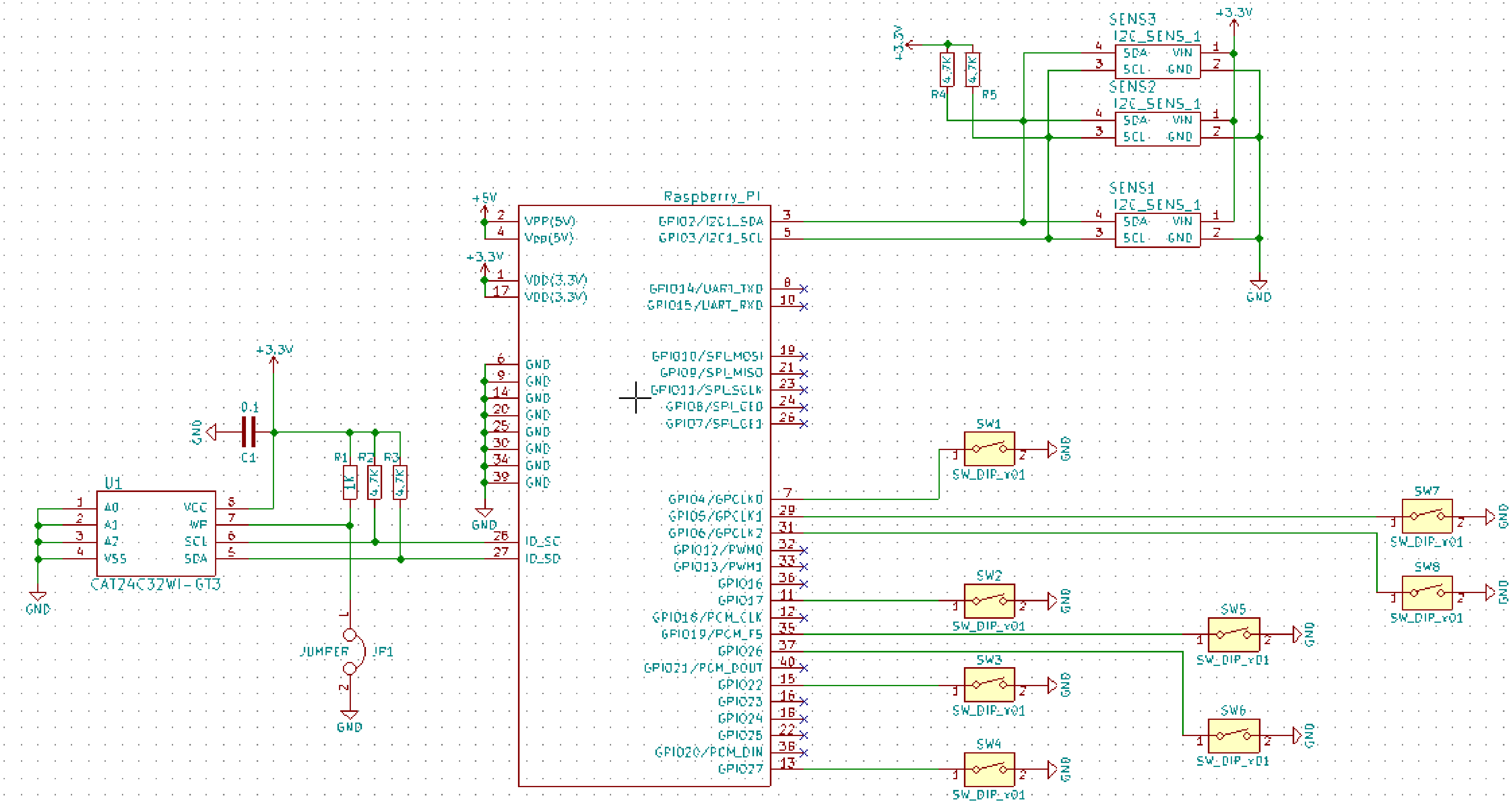
KiCAD

- Free & open source EDA software (GPLv3+)
- Cross platform (works on GNU/Linux distributions, MS Windows and Mac OS X)
- Integrated 3D viewer
- Up to 32 copper layers + 14 fixed-purpose technical layers
- Contributions from CERN developers
- Already adopted by the industry
- http://kicad-pcb.org/

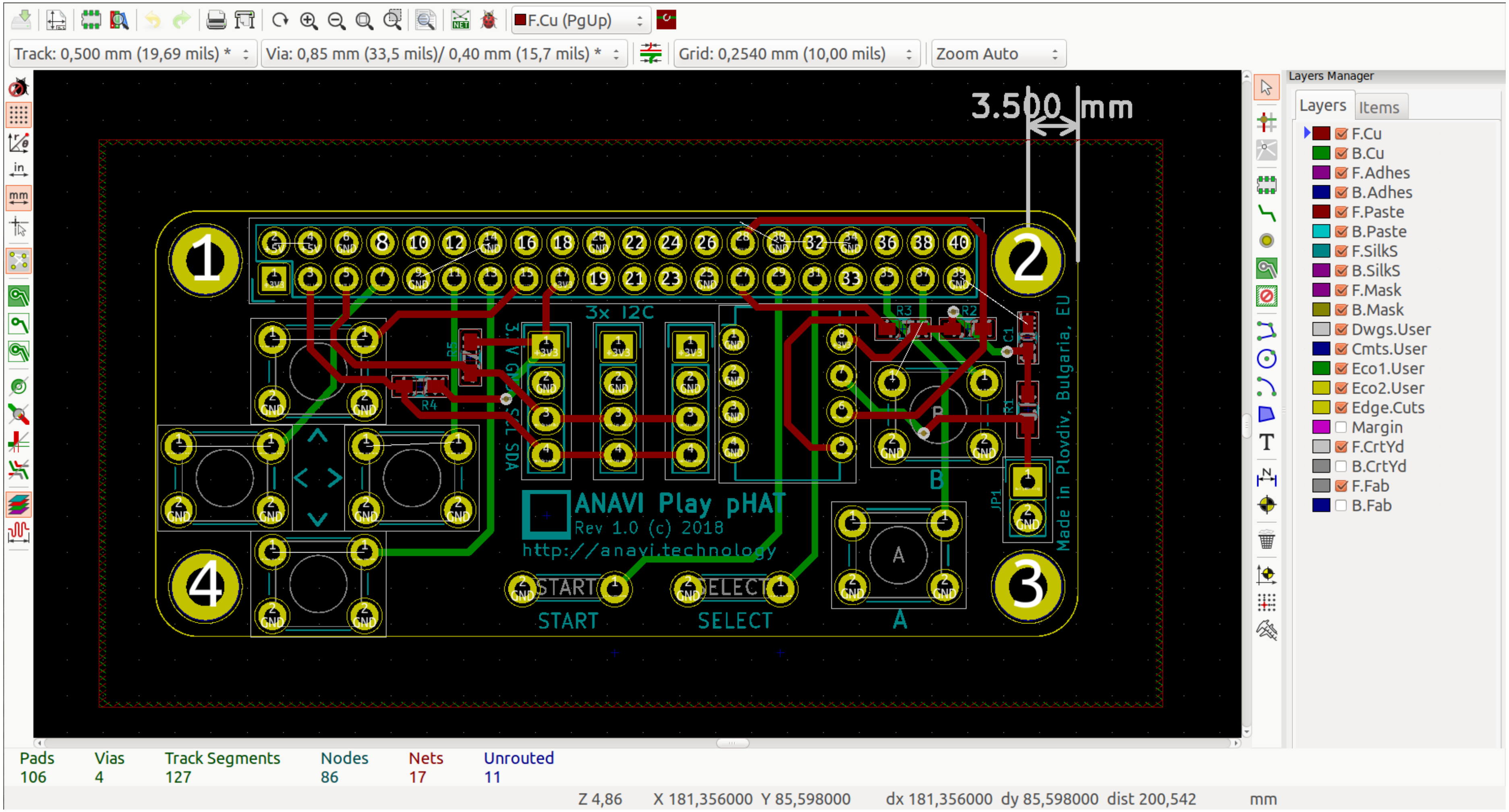
Ingredients for a retro gamepad

- 4 tact buttons for movement
- 2 tact buttons for **select** and **start**
- 2 tact buttons for **A** and **B**
- EEPROM connected to the secondary I2C (pins 27 and 28)
- Bonus: several slots for sensors on the first I2C (pins 3 and 5) to make the board useful for home automation while not playing retro games

Schematics


SENS3
I2C_SENS_1
SDA
VIN
SCL
GND
SENS2
I2C_SENS_1
SENS1
I2C_SENS_1
+3.3V
4.7K
R4
R5
GND
Raspberry_PI
+5V
VPP(5V)
Vpp(5V)
VDD(3.3V)
VDD(3.3V)
GPIO2/I2C1_SDA
GPIO3/I2C1_SCL
GPIO14/UART_TXD
GPIO15/UART_RXD
GPIO10/SPI_MOSI
GPIO9/SPI_MISO
GPIO11/SPI_SCLK
GPIO8/SPI_CE0
GPIO7/SPI_CE1
ID_SC
ID_SD
GPIO4/GPCLK0
GPIO5/GPCLK1
GPIO6/GPCLK2
GPIO12/PWM0
GPIO13/PWM1
GPIO16
GPIO17
GPIO18/PCM_CLK
GPIO19/PCM_FS
GPIO26
GPIO21/PCM_DOUT
GPIO22
GPIO23
GPIO24
GPIO25
GPIO20/PCM_DIN
GPIO27
0.1
C1
R1
R2
R3
1K
U1
A0
A1
A2
VSS
VCC
WP
SCL
SDA
CAT24C32WI-GT3
JUMPER
JP1
SW1
SW2
SW3
SW4
SW5
SW6
SW7
SW8
SW_DIP_x01

Pcbnew



F.Cu (PgUp)

Track: 0,500 mm (19,69 mils) *

Via: 0,85 mm (33,5 mils)/ 0,40 mm (15,7 mils) *

Grid: 0,2540 mm (10,00 mils)

Zoom Auto

Layers Manager

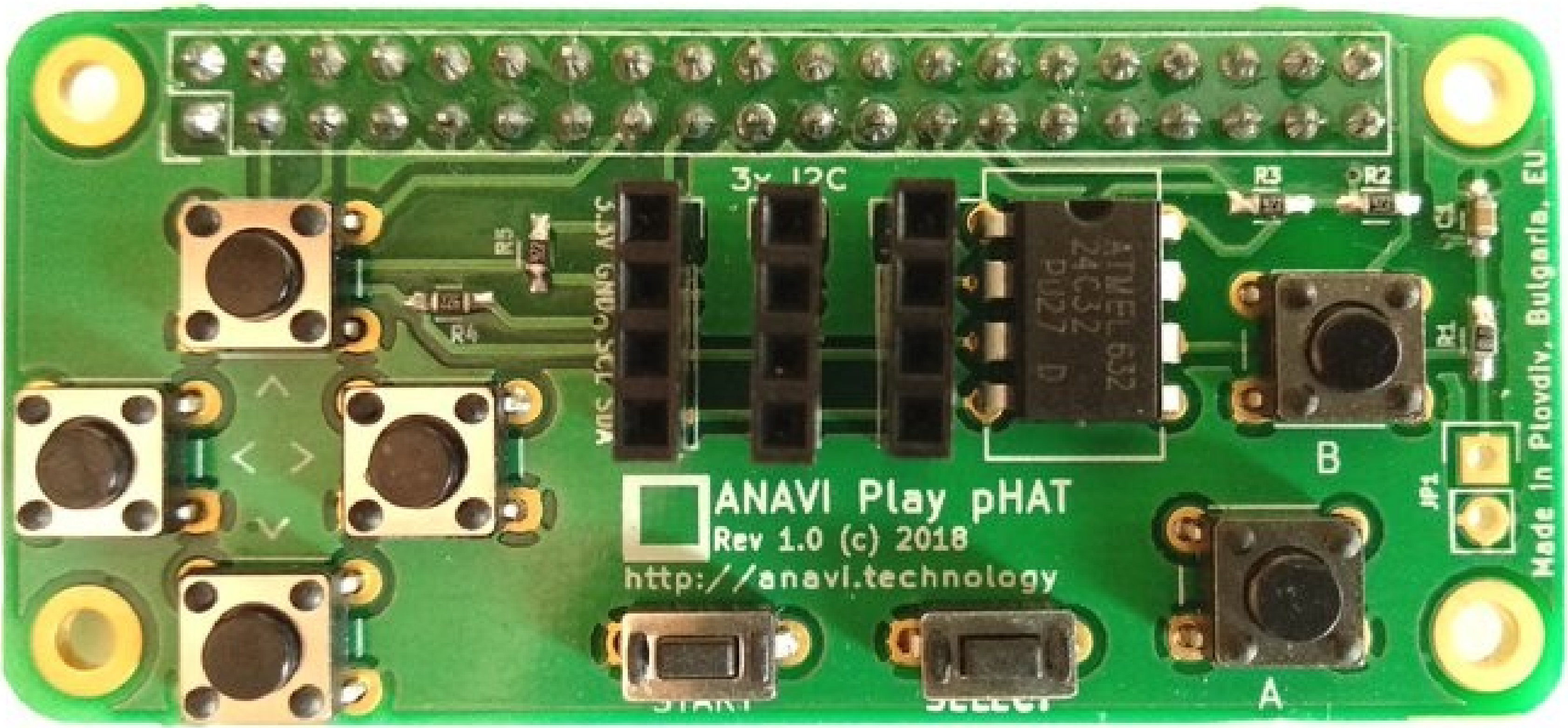
Layers | Items

- ☑ F.Cu
- ☑ B.Cu
- ☑ F.Adhes
- ☑ B.Adhes
- ☑ F.Paste
- ☑ B.Paste
- ☑ F.SilkS
- ☑ B.SilkS
- ☑ F.Mask
- ☑ B.Mask
- ☑ Dwgs.User
- ☑ Cmts.User
- ☑ Eco1.User
- ☑ Eco2.User
- ☑ Edge.Cuts
- ☐ Margin
- ☑ F.CrtYd
- ☐ B.CrtYd
- ☑ F.Fab
- ☐ B.Fab

Pads	Vias	Track Segments	Nodes	Nets	Unrouted
106	4	127	86	17	11

Z 4,86 X 181,356000 Y 85,598000 dx 181,356000 dy 85,598000 dist 200,542 mm

The Real Hardware



Open Source Hardware

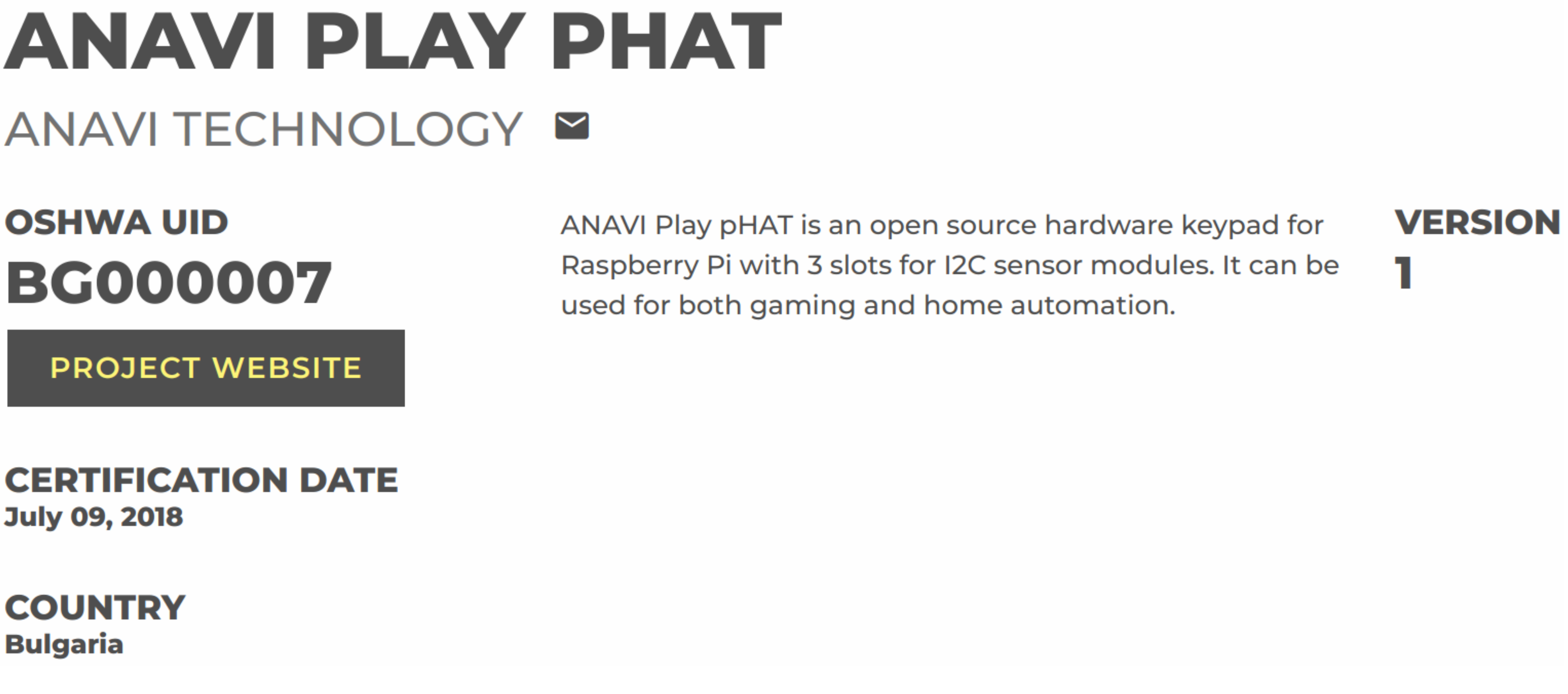
- Design of physical objects that is publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design
- To be really open source hardware the project needs 4 main elements: hardware, software, documentation, branding

Open Source Hardware Certification

Open Source Hardware Association(OSHWA):

- Maintains Certified Projects Directory
- Ensures that the definition of “open source hardware” used by a specific project matches the community definition of open source hardware
- Provides an unique ID for certified hardware based on the country code and a number, for example: BG000007
- Provides an unique logo with for the certified ID

Open Source Hardware Certified!



ANAVI PLAY PHAT

ANAVI TECHNOLOGY ✉

OSHWA UID

BG000007

PROJECT WEBSITE

ANAVI Play pHAT is an open source hardware keypad for Raspberry Pi with 3 slots for I2C sensor modules. It can be used for both gaming and home automation.

VERSION

1

CERTIFICATION DATE

July 09, 2018

COUNTRY

Bulgaria

https://certification.oshwa.org/bg000007.html

Device Tree

- Specification for a software data structure that describes hardware components
- Allows compiled Linux kernel to support different hardware configurations within a wider architecture family
- The device tree is compiled as an external file from the Linux kernel binary files called **DTB** (Device Tree Blob) from source code in **DTS** (Device Tree Source)
- Device Tree Compiler (DTC) compiles the source into a binary
- For more details:

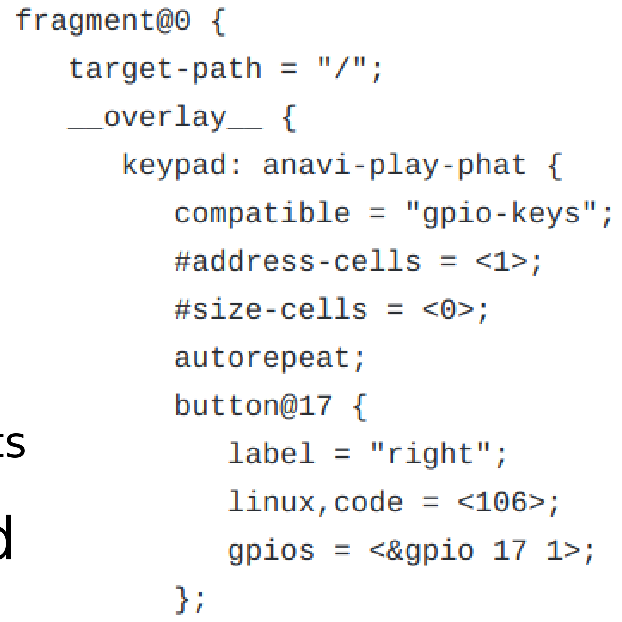
  https://www.devicetree.org/

  https://elinux.org/images/f/f9/Petazzoni-device-tree-dummies_0.pdf

Device Tree Overlay (DTO)

- DTO enables a central **DTB** (device tree blob) to be overlaid on the device tree
- DTO includes a number of fragments
- With the default bootloader on Raspberry Pi, DTO can be set at **config.txt** on the FAT partition with **device_tree=**
- The EEPROM on a Raspberry Pi add-on board, connected to the secondary I2C bus on pins 27 and 28, should include the overlay required to automatically enable the board
- On Raspbian and other GNU/Linux distributions for Raspberry Pi the information in the EEPROM can be seen from user-space at **/proc/device-tree/hat/**

Mapping Keys in DTS

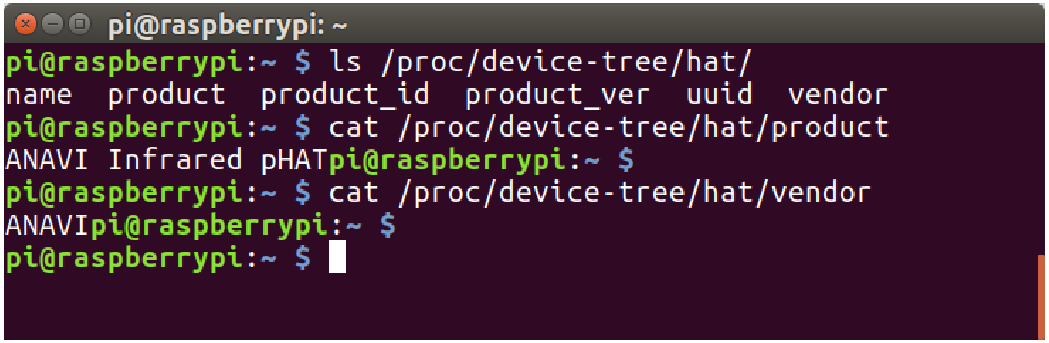


```
fragment@0 {
   target-path = "/";
   __overlay__ {
      keypad: anavi-play-phat {
         compatible = "gpio-keys";
         #address-cells = <1>;
         #size-cells = <0>;
         autorepeat;
         button@17 {
            label = "right";
            linux,code = <106>;
            gpios = <&gpio 17 1>;
         };
```

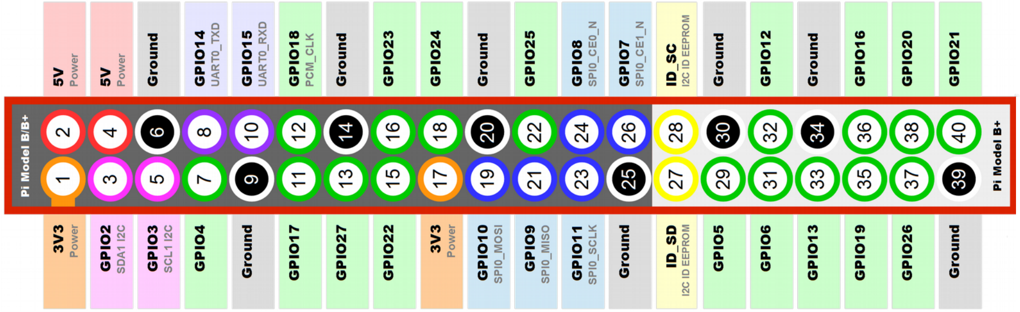
- Linux key codes as defined in /usr/include/linux/input-event-codes.h
- Compile **DTBO**:
  dtc -I dts -O dtb -o anavi-play-phat.dtbo anavi-play-phat.dts
- Create settings.txt file for the EEPROM and specify the DTO
- Create binary **.eep** file from the .txt file using **eepmake**
  ./eepmake settings.txt settings.eep anavi-play-phat.dtbo
- DTS for the DTO of Play pHAT:
  https://github.com/AnaviTechnology/hats/blob/anavi/eepromutils/anavi-play-phat.dts

Flashing an EEPROM for a HAT

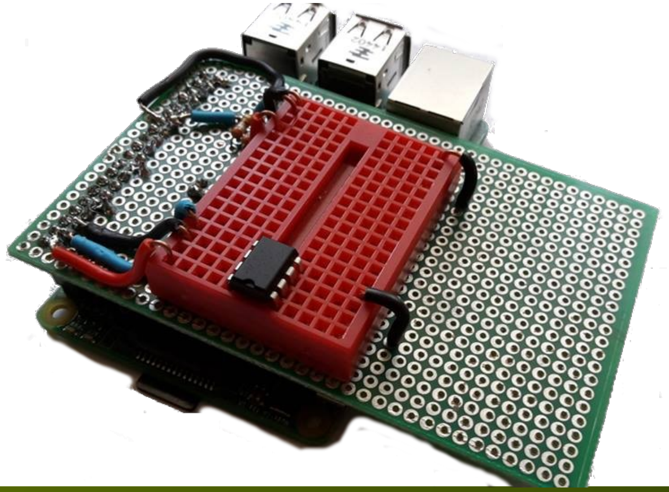
- Writing to the EEPROM: pin 2 and 3
- Reading from the EEPROM: pin 27 and 28
- Flash the **.eep** file with **eepflash**: https://github.com/raspberrypi/hats



```
pi@raspberrypi:~ $ ls /proc/device-tree/hat/
name   product   product_id   product_ver   uuid   vendor
pi@raspberrypi:~ $ cat /proc/device-tree/hat/product
ANAVI Infrared pHATpi@raspberrypi:~ $
pi@raspberrypi:~ $ cat /proc/device-tree/hat/vendor
ANAVIpi@raspberrypi:~ $
pi@raspberrypi:~ $ 
```

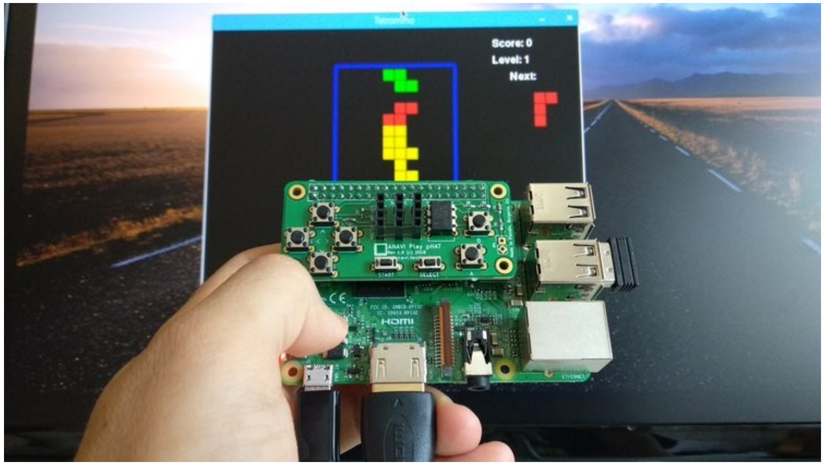


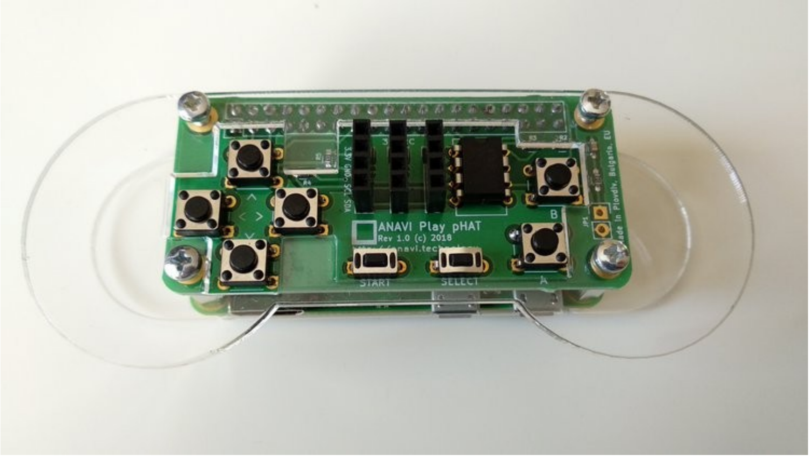
3V3 Power	1	2	5V Power
GPIO2 SDA1 I2C	3	4	5V Power
GPIO3 SCL1 I2C	5	6	Ground
GPIO4	7	8	GPIO14 UART0_TXD
Ground	9	10	GPIO15 UART0_RXD
GPIO17	11	12	GPIO18 PCM_CLK
GPIO27	13	14	Ground
GPIO22	15	16	GPIO23
3V3 Power	17	18	GPIO24
GPIO10 SPI0_MOSI	19	20	Ground
GPIO9 SPI0_MISO	21	22	GPIO25
GPIO11 SPI0_SCLK	23	24	GPIO8 SPI0_CE0_N
Ground	25	26	GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM	27	28	ID_SC I2C ID EEPROM
GPIO5	29	30	Ground
GPIO6	31	32	GPIO12
GPIO13	33	34	Ground
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
Ground	39	40	GPIO21



Testing with Open Source Games in Raspbian GNU/Linux distribution

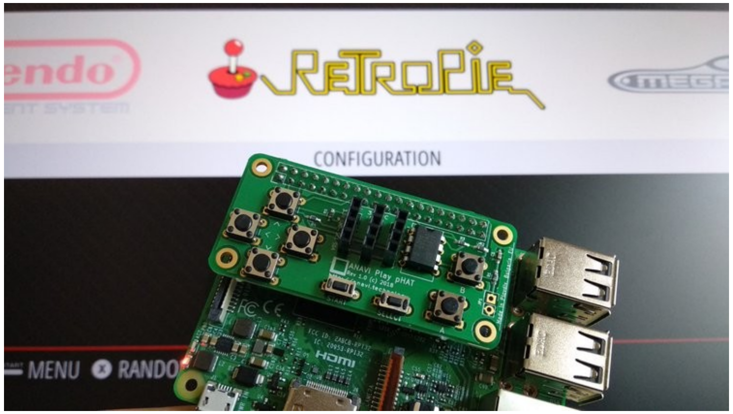
- Verify that Raspberry Pi boots with the attached Play pHAT and the information from the EEPROM in user-space is available
- Verify that the keys on Play pHAT are properly set
- Start a simple game included in the Raspbian Desktop image to test Play pHAT for gaming





RetroPie

- GNU/Linux distribution optimized for playing retro games that combines RetroArch and Emulation Station
- Available for Raspberry Pi, Odroid C1/C2 or personal computer (with Debian or Ubuntu)
- Provides emulators for loading ROMs (digital version of game cartriges), no ROMs are included in RetroPie due to copyright issues
- Useful links:
  https://retropie.org.uk
  https://retropie.org.uk/docs/
  https://github.com/RetroPie
  https://www.mamedev.org/roms/



Further adjustments for emulators

- The problem: the gamepad works fine within the menus of RetroPie but fails after launching some games and emulators
- The solution: add a Python script for software emulation of the keys that works will the game emulators

```
sudo apt-get update
sudo apt-get install -y python-pip
sudo pip install evdev
cd ~
git clone https://github.com/AnaviTechnology/anavi-examples.git
```

- Add the following lines to **/etc/rc.local**

```
sudo python /home/pi/anavi-examples/anavi-play-phat/anavi-play-gamepad.py &
```

Conclusions

- Retro gaming on Raspberry Pi is fun and provides an excellent opportunity for learning modern open source technologies like device tree
- RetroPie is an awesome free and open source Linux distribution for playing retro games on Raspberry Pi
- KiCad is high-quality free and open source software for designing hardware
- Certified open source hardware is a fair way to share schematics and knowledge about physical objects to the community

Thank you!

- https://www.raspberrypi.org/blog/introducing-raspberry-pi-hats/
- https://retropie.org.uk/
- https://github.com/AnaviTechnology/hats/blob/anavi/eepromutils/anavi-play-phat.dts
- https://github.com/AnaviTechnology/anavi-play-phat
- https://www.crowdsupply.com/anavi-technology/anavi-play-phat