

Polar01d

Fabriquer un appareil photo instantané comme le fameux POLAROID, qui reconnaît les visages et imprime ses photos sur papier thermique de ticket de caisse.

 Difficulté **Moyen**

 Durée **4 heure(s)**

 Catégories **Art, Électronique, Jeux & Loisirs, Recyclage & Upcycling**

 Coût **100 EUR (€)**

Sommaire

Introduction

Étape 1 - Installer Raspbian + OpenCV

Raspbian

Open CV

Outils logiciels

Étape 2 - Installer le code Polar01d_v1.0

Étape 3 - Souder, construire, essayer

Notes et références

Commentaires

Introduction

Mis au point pour remplacer l'appareil photo instantané réclamé par ma fille à Noël.

Ses photos ne coutent pas 1€ chacune, mais 1€ pour 4 mètres de pellicule (thermo & photosensible!)

L'exposition lumineuse est calibrée selon le temps d'appuie sur le bouton. La photo d'ensemble s'imprime dans la seconde, puis OpenCV se charge de détecter, zoomer et imprimer les visages et les QRcodes .

1. Installer Raspbian + OpenCV
2. Installer Polar01d_v1.0
3. Souder, construire, essayer

LISTE MATERIEL: <https://pod.p2p.legal/fr/p0lar01d>



Matériaux

- Raspberry Pi Zero W
- Carte micro SD (min 4Go)
- Module Camera Pi
- Mini Imprimante Thermique Portable (Bluetooth / USB) + cable USB
- Convertisseur Step Down DC-DC 5V
- Switch on/off, bouton, LED, résistance
- Boitier (tupperware)

Outils

- Fer à souder
- Cutter
- Colle à chaud + pistolet



Étape 1 - Installer Raspbian + OpenCV

Il s'agit du système que nous allons faire tourner sur le Rpi Zero.
Il sera installé sur une carte micro SD de 4 Go minimum

Raspbian

Je ne vais pas plagier les nombreux sites qui expliquent comment installer Raspbian sur Pi Zero.
Prendre soin d'activer la camera avec "`sudo raspi-config`"

Open CV

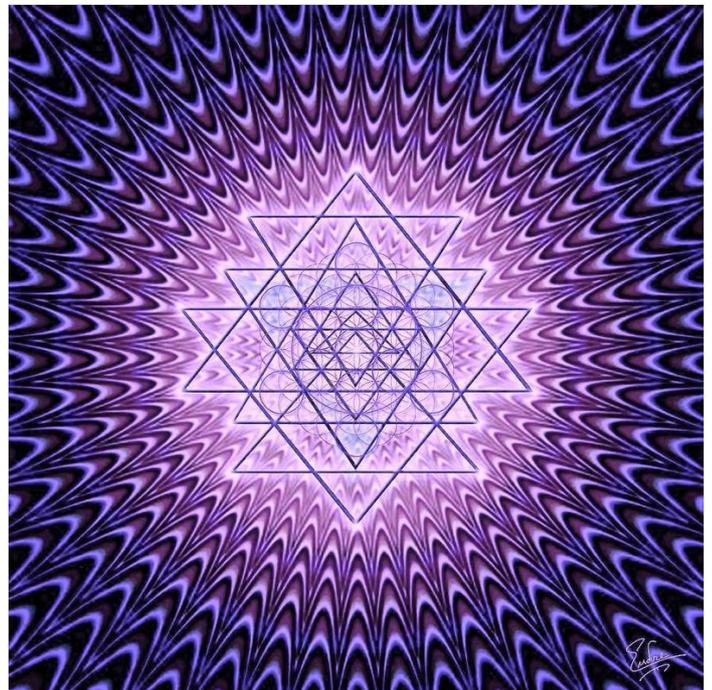
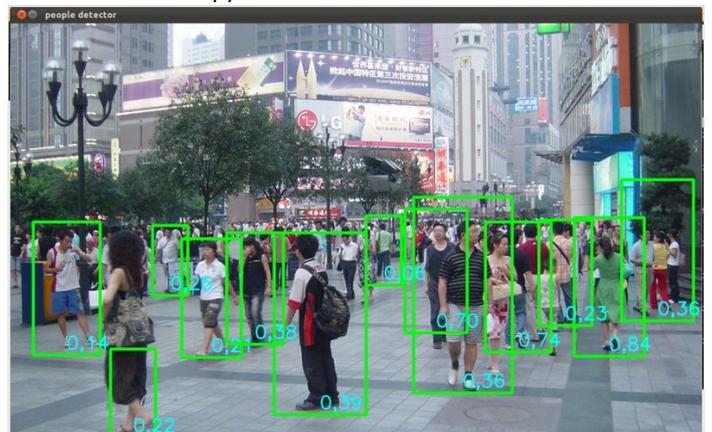
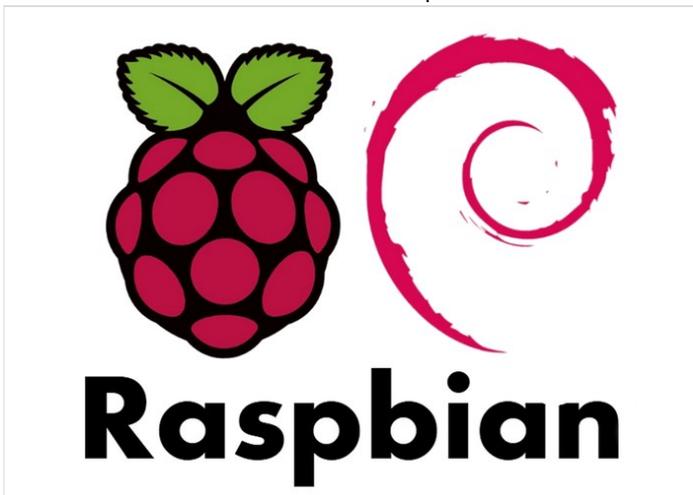
Pour faire tourner OpenCV3 sur le Pi Zero (seul opencv2 est disponible sur les dépôts) nous allons profiter du travail de yoursunny, et éviter des jours de compilation!

Outils logiciels

Ajouter les outils logiciels que nous allons utiliser

```
sudo apt-get -y install wiringpi imagemagick python-pil python-dev python3-rpi.gpio printer-driver-all cups libzbar0 python-zbar
```

Pour reconnaître les FlashCode suivez la procédure d'installation de Zbar et de ses bibliothèques python



Étape 2 - Installer le code Polar01d_v1.0

Copier ces fichiers dans /home/pi

<https://pad.p2p.legal/s/Polar01d>

Il comporte un script shell et un code python à lancer au démarrage (/etc/rc.local).

- Le shell est chargé de surveiller l'appui sur le bouton. Il déclenche 3 phases de prise de vue (Luminosité/Contraste: 80%, 40%, 0%). Lacher le bouton au moment souhaité. La photo d'ensemble s'imprime.

- La capture est placée dans un répertoire que surveille Facedetect.py qui procède à la détection des visages et Flashcode pour les extraire et les imprimer.

```
pi@emmap1: ~$ cat PolaEmma.sh
#!/bin/bash
LED=5
BUTTON=21
gpio -g mode $BUTTON up # Initialize GPIO states
gpio -g mode $LED out

ip=$(ifconfig wlan0 | grep "inet" | awk '{print $2}')"
echo "$(date +%Y-%m-%d-%H:%M:%S) emmap1: $ip" > /dev/usb/lp0

while :
do
# Forever
if [ $(gpio -g read $BUTTON) -eq 0 ]; then
gpio -g write $LED 1
# TAKE 1st shot (+++ brightness)
raspistill -n -th none -t 50 -fli auto -br 80 -co 80 -w 720 -h 480 -o /tmp/1st.jpg
sleep 0.25
# LONG PRESS ?
if [ $(gpio -g read $BUTTON) -eq 0 ]; then
# TAKE 2nd shot (darker)
raspistill -n -th none -t 50 -fli auto -br 40 -co 40 -w 720 -h 480 -o /tmp/1st.jpg
# Inform 2nd SHOT to printer
echo " " > /dev/usb/lp0
fi
sleep 0.25
# LONG PRESS ?
if [ $(gpio -g read $BUTTON) -eq 0 ]; then
# TAKE 2nd shot (darker)
raspistill -n -th none -t 50 -fli auto -w 720 -h 480 -o /tmp/1st.jpg
# Inform 2nd SHOT to printer
echo " " > /dev/usb/lp0
fi
# Send picture to FaceDetect (takes approx 4 seconds)
cp /tmp/1st.jpg /tmp/shot_images/shoot.jpg
# Print 1st shot
```

```
pi@emmap1: ~$ cat Facedetect.py
#!/usr/bin/env python
# find the barcodes in the image and decode each of the barcodes
barcodes = pyzbar.decode(image)
# loop over the detected barcodes
for barcode in barcodes:
(x, y, w, h) = barcode.rect
cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), 2)
barcodeData = barcode.data.decode("utf-8")
barcodeType = barcode.type
text = "{} ({}).format(barcodeData, barcodeType)
tmp = subprocess.check_output(['/home/pi/print.sh', 'TXT {}'.format(text)])

# Detect faces in the image
faces = faceCascade.detectMultiScale(
gray,
#scaleFactor=1.1,
minNeighbors=5,
minSize=(30, 30),
flags = cv2.CASCADE_SCALE_IMAGE#cv.CV_HAAR_SCALE_IMAGE
)

print("Found {} face(s)!".format(len(faces)))

#iterate through the found faces
count = 0
for (x, y, w, h) in faces:
x1 = int(x - w*0.05)
y1 = int(y - h*0.1)
x2 = int(x + w*1.1)
y2 = int(y + h*1.25)

#check if 'face' is really a face...
check_gray = gray[y1:y2,x1:x2]
check = faceCascade.detectMultiScale(
check_gray,
```

Étape 3 - Souder, construire, essayer

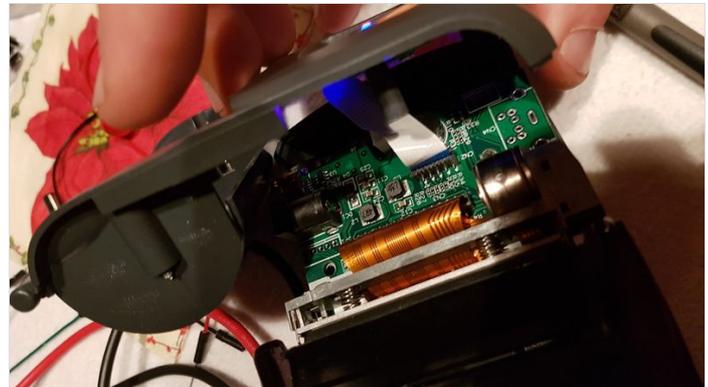
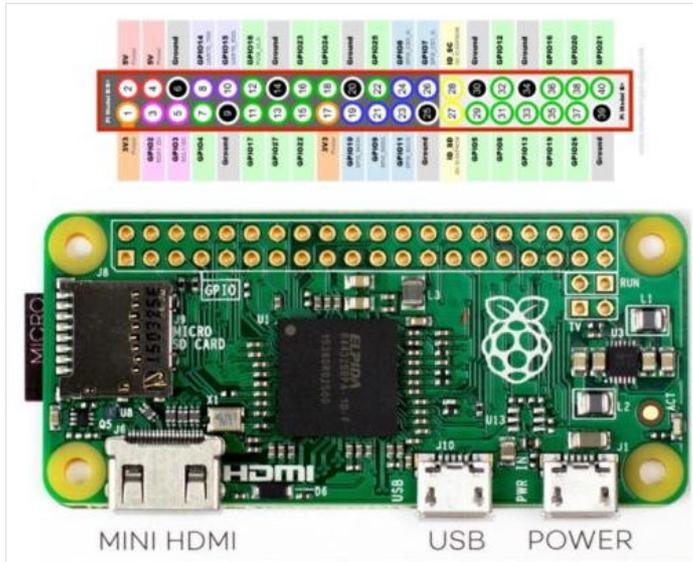
Maintenant que toute la partie logicielle est au point, il reste à assembler le tout.

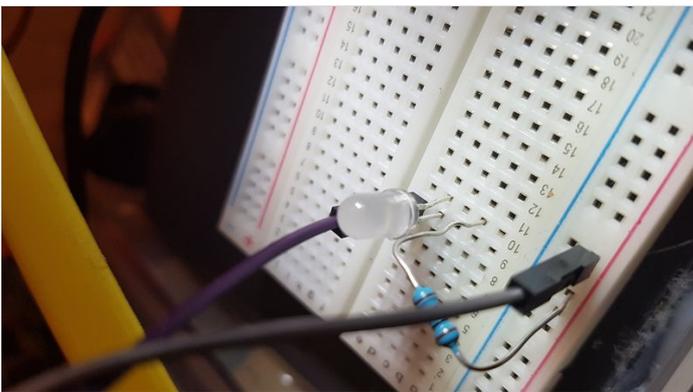
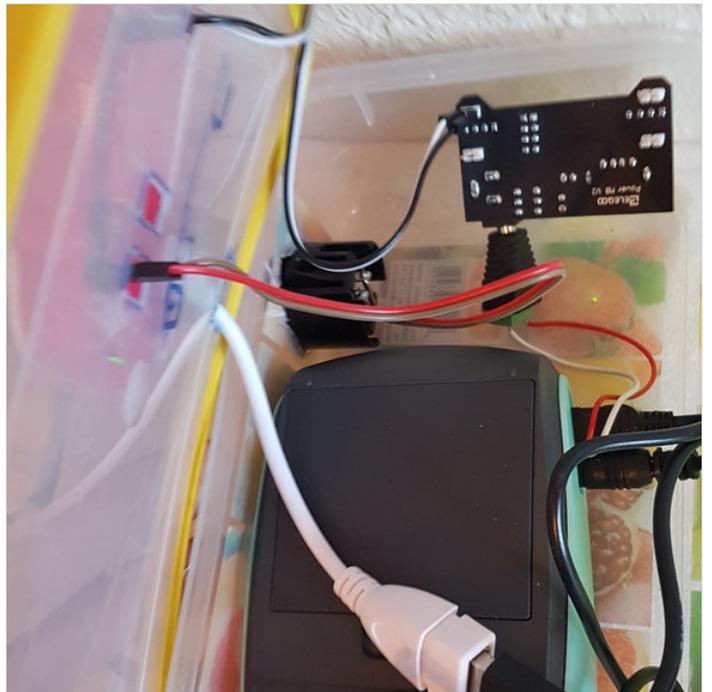
- Nous allons nous servir de la batterie (7,8v) de l'imprimante pour alimenter le Raspberry au travers d'un régulateur DC-DC 5v qui sera branché au PIN 5v et GRND du Rpi au travers d'un interrupteur.

- Le bouton sera connecté à la broches GPIO 21

- La LED et sa résistance adaptée au GPIO 5

On découpe le boîtier pour donner accès aux boutons puis on fixe le tout avec de la colle chaude.





Notes et références

Bidule réalisé par Fred - Membre forgeron Libre de #CopyLaRadio
Rejoignez-nous <https://www.copylaradio.com>