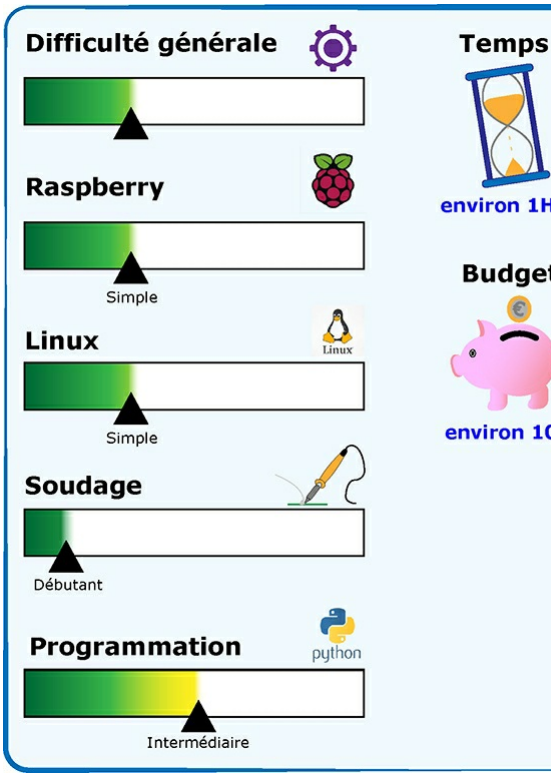
Découverte de la Pi Cap de Bare Conductive

Présentation

Dans cet article, nous allons vous présentez une carte programmable de la marque Bare Conductive : la Pi Cap.

La carte Pi CAP est basée sur 12 électrodes capacitives et permet d'ajouter des interactions tactiles à votre Raspberry Pi.

Cette carte est notamment adaptée pour une utilisation avec de la peinture conductrice mais fonctionne aussi avec des cordons crocodiles ou tout élément conducteur.





Initialisation

Vous aurez besoin d'une carte **Raspberry Pi** avec Raspbian d'installé [*Installation de Raspbian*]. Vous aurez également besoin de VNC pour administrer à distance Raspberry [*Installation de VNC*].

Dans les paramètres de Raspbian, vérifiez que le SSH est activé : **Menu Démarrer** (logo Raspberry), **Préférences**, **Configuration du Raspberry Pi**, onglet **Interfaces**, ligne **SSH** bouton **Activé** coché.

Lancez un Terminal, exécutez les commandes suivantes, les unes à la suite des autres en appuyant sur la touche ***Entrée*** :

- *sudo apt-get update*
- *sudo apt-get upgrade*
- *sudo apt-get install picap*
- *picap-setup*

Projet 1 : Capteur de proximité

Nous allons concevoir un programme permettant de reproduire un capteur de proximité grâce à la peinture conductrice. En passant la main au-dessus de la peinture, le système allumera une LED.

NB : pour ce projet nous utiliserons des pinces crocodiles pour relier la Pi Cap à nos connexions de peinture conductrice, mais vous pouvez très bien appliquer la peinture directement sur la carte.

-

Liste des produits :

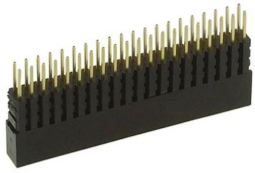
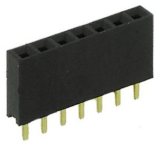
Raspberry Pi	Pi Cap	Peinture conductrice	Patron

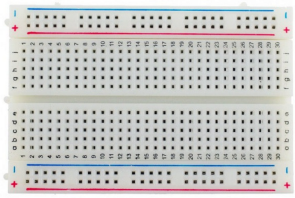








3 résistances de 100 ohms	3 LEDs 8mm	HE40 RPI	Connecteur 7 points
			

Plaque d'essai	Jumpers M/M	Pinces crocodiles	Connecteur crocodile-jumper
			

Application de la peinture :

Découpez le patron sur les pointillés.

Appliquez la peinture sur toutes les parties noires. Nous vous conseillons d'utiliser un pinceau et un peu d'eau pour diluer la peinture.

Inutile de faire une couche de peinture épaisse.

Attendez 5 minutes que la peinture sèche.



Montage :

Nous aurons besoin du nécessaire de soudage. Si vous n'avez pas le matériel nécessaire, nous vous conseillons ce kit de soudage.

Vous débutez en soudure ? Suivez le guide :

Le guide du soudage

Retournez la carte Pi Cap et soudez le connecteur 7 points.

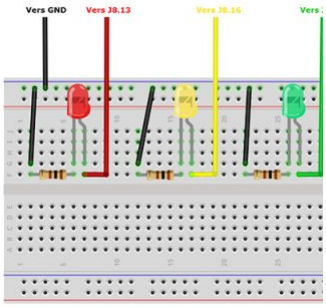


Insérez le connecteur HE40 RPI sur le port GPIO de votre Raspberry, puis insérez la Pi Cap sur le connecteur.

Sur la plaque de montage, montez les LEDs en série avec les résistances et en cathodes communes.

Reliez grâce aux jumpers :

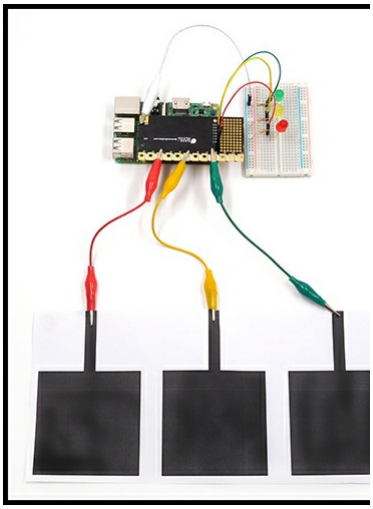
- l'anode de la LED **rouge** au point **J8.13** de la Pi Cap
- l'anode de la LED **jaune** au point **J8.16** de la Pi Cap
- l'anode de la LED **verte** au point **J8.22** de la Pi Cap



Reliez les cathodes communes à l'électrode **GND** de la Pi Cap grâce au câble crocodile-jumper.

Reliez les électrodes avec les pinces crocodiles:

- **E2** au carré **gauche** du patron
- **E5** au carré **central** du patron
- **E8** au carré **droit** du patron



Programme

Alimentez votre Raspberry. Lancez VNC pour naviguer dans Raspbian. Dans vos Documents,
créer un fichier vide ***proximite.py***.

Ouvrez le fichier et copiez-collez le programme suivant :

Broche Pi Cap	Numé
J8.12	
J8.13	
J8.15	
J8.16	
J8.18	
J8.22	
J8.36	

```
import RPi.GPIO as GPIO
import MPR121

sensor = MPR121.begin()
sensor.set_touch_threshold(4)
sensor.set_release_threshold(2)
#Plus la valeur du "threshold" est basse, plus on capte loin
#Valeurs par defaut : 40 ; 20

num_electrodes = 12

rouge = 27 #J8.13
jaune = 23 #J8.16
vert = 25 #J8.22

GPIO.setmode(GPIO.BCM)
GPIO.setup(rouge,GPIO.OUT)
GPIO.setup(jaune,GPIO.OUT)
GPIO.setup(vert,GPIO.OUT)

while True:

    if sensor.touch_status_changed(): #si une electrode a ete touchee
        sensor.update_touch_data() #mise a jour

        is_any_touch_registered = False

        for i in range(num_electrodes):
            if sensor.get_touch_data(i): #on identifie le numero de l'electrode
                is_any_touch_registered = True
            if sensor.is_new_touch(i):
                print ("Electrode : " + str(i))
                if i==2: #si on appuie sur l'electrode 2, alors on change l'etat de la LED rouge
                    GPIO.output(rouge, not GPIO.input(rouge)) #inversement de l'etat On/Off
                if i==5: #si on appuie sur l'electrode 5, alors on change l'etat de la LED jaune
                    GPIO.output(jaune, not GPIO.input(jaune)) #inversement de l'etat On/Off
                if i==8: #si on appuie sur l'electrode 8, alors on change l'etat de la LED verte
                    GPIO.output(vert, not GPIO.input(vert)) #inversement de l'etat On/Off
```

Dans un Terminal, tapez la commande suivante et appuyez sur la touche ***Entrée*** :

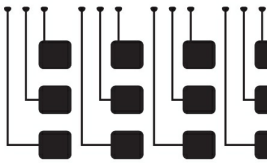
sudo python proximite.py

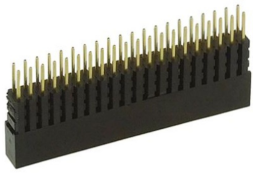
En passant votre main au-dessus d'un carré, la LED qui lui est raccordée changera d'état (allumée/éteinte).

Projet 2 : Sound Table :

Nous allons concevoir un programme permettant de reproduire une sound table. Chaque électrode jouera un son différent.

NB : pour ce projet nous utiliserons des pinces crocodiles pour relier la Pi Cap à nos connexions de peinture conductrice, mais vous pouvez très bien appliquer la peinture directement sur la carte.

Raspberry Pi	Pi Cap	Peinture conductrice	Patron
			

Pinces crocodiles	HE40 RPI	Enceinte
		

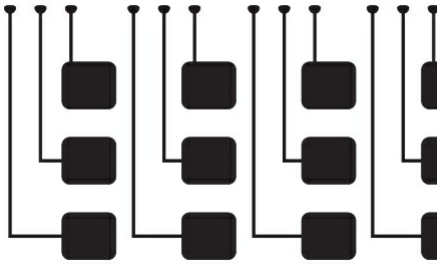
Application de la peinture :

Découpez le patron sur les pointillés.

Appliquez la peinture sur toutes les parties noires. Nous vous conseillons d'utiliser un pinceau et un peu d'eau pour diluer la peinture.

Inutile de faire une couche de peinture épaisse.

Attendez 5 minutes que la peinture sèche.

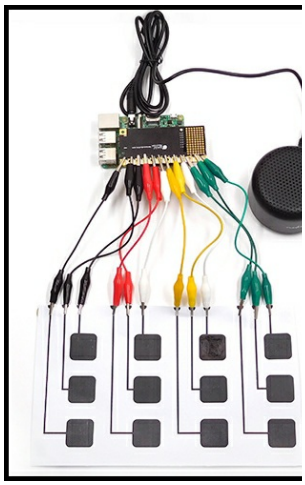


Montage :

Insérez le connecteur HE40 RPI sur le port GPIO de votre Raspberry, puis insérez la Pi Cap sur le connecteur.

Reliez à l'aide du câble Jack l'enceinte au port de Jack :

- de la carte Pi Cap si vous possédez une Raspberry Zero
- de votre Raspberry Pi si celle-ci est d'un autre modèle



Ici, nous utilisons une 3B, le Jack est donc relié à la Raspberry.

Reliez les électrodes aux boutons de la « sound table » avec les pinces crocodiles.

Gestion des fichiers audio :

On peut associer automatiquement une piste audio à une électrode. Pour cela, il faut que la piste audio soit nommée ***TRACK0XX*** où ***XX*** est le numéro de l'électrode. Exemple : la piste ***TRACK003*** sera jouée lorsque **l'électrode 3** sera appuyée. Il faut impérativement garder ce système de nommage pour assurer cet automatisme. Il n'est pas obligatoire d'avoir 12 fichiers audio : si une électrode n'est associée à aucun fichier, elle n'émettra pas de son si l'on appuie dessus.

Alimentez votre Raspberry. Lancez VNC pour naviguer dans Raspbian. Téléchargez la bibliothèque sonore [*Bare Conductive ®*], et faites extraire l'archive dans vos Documents.

Programme :

Lancez un Terminal et exécutez la commande ***alsamixer***.

La commande ***alsamixer*** permet de régler le volume de l'enceinte. Sur votre clavier, la touche « **Flèche du haut** » augmente le volume, la touche « **Flèche du bas** » diminue le volume, et la touche « **Echap** » valide et retourne au Terminal.

Nous pouvons débuter la programmation. Dans vos Documents, créer un fichier vide ***soundtable.py***. Ouvrez le fichier et copiez-collez le programme suivant :

```
import MPR121
from gpiozero import RGBLED
import subprocess
import pygame
from pygame.mixer import Sound
from glob import glob
from time import sleep

sensor = MPR121.begin()
sensor.set_touch_threshold(40)
sensor.set_release_threshold(20)

led = RGBLED(6, 5, 26, active_high=False)

num_electrodes = 12

# Convertion des fichier MP3 en WAV
led.blue = 1
subprocess.call("picap-samples-to-wav soundtable", shell=True) #soundtable = nom du dossier
led.off()

# init
pygame.mixer.pre_init(frequency=44100, channels=64, buffer=1024)
pygame.init()

sounds = [Sound(path) for path in glob("soundtable/.wavs/*.wav")] #soundtable = nom du dossier

def play_sounds_when_touched():
    if sensor.touch_status_changed(): #verifie si une electrode a ete touchee
        sensor.update_touch_data()

        is_any_touch_registered = False

        for i in range(num_electrodes): #regarde quelle electrode a ete touchee
            if sensor.get_touch_data(i):
                is_any_touch_registered = True
            if sensor.is_new_touch(i):
                print ("Son joue : " + str(i))
                sound = sounds[i] #joue le son dont le numero est le meme que celui de l'electrode
                sound.play()

        if is_any_touch_registered:
            led.red = 1
        else:
            led.off()

running = True
while running:
    try:
        play_sounds_when_touched()
    except KeyboardInterrupt:
        led.off()
        running = False
    sleep(0.01)
```

Dans un Terminal, tapez la commande suivante et appuyez sur la touche ***Entrée*** :

sudo python soundtable.py

Votre sound table est prête !

Démontration :

Voir la vidéo sur YouTube