Projets

Le détecteur de parent

Détecter les intrus avec un Raspberry Pi

Étape 1 Détecteur de parent

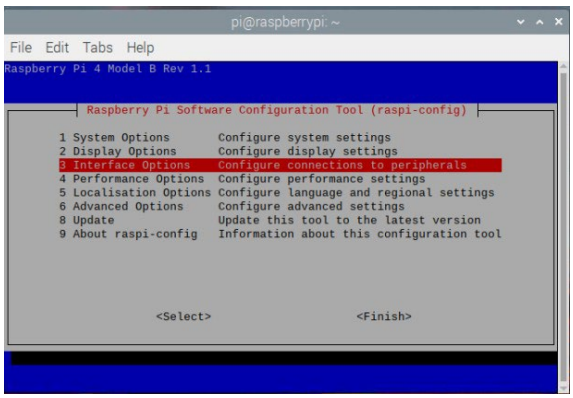
⚠ Module caméra Raspberry Pi – Notice du système d'exploitation

Le module Python Picamera n'est actuellement pas, par défaut, compatible avec la dernière version de Raspberry Pi OS (appelée Bullseye).

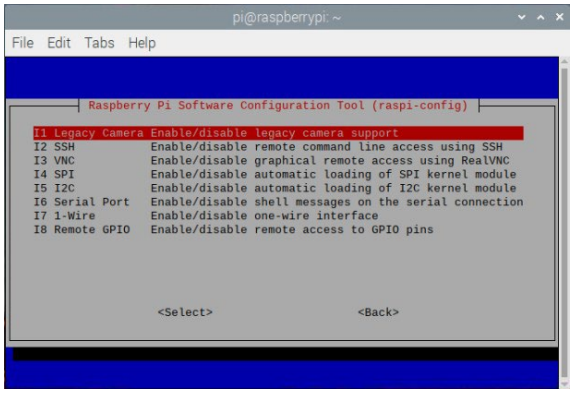
Pour utiliser le module Picamera, active la prise en charge de la caméra héritée.

Ouvre une fenêtre de terminal et entre la commande suivante :

Utilise les curseurs pour faire défiler vers le bas `Interface Options` et appuie sur la touche « Entrée ».

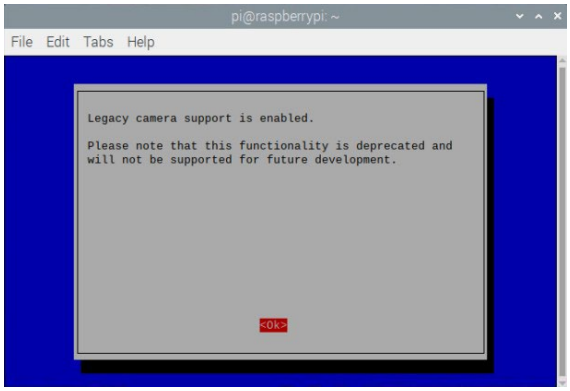


Assure-toi que « Legacy Camera Enable/disable legacy camera support » est sélectionné et appuie sur la touche « Entrée ».

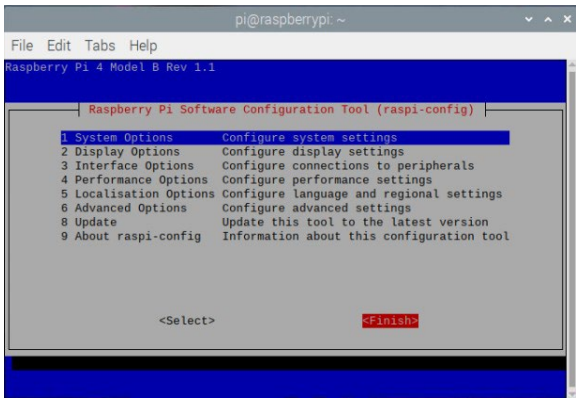


Utilise les clés du curseur pour sélectionner `<Yes>` et appuie sur la touche « Entrée ».

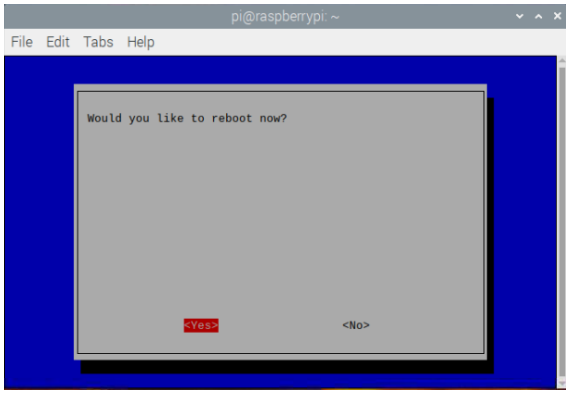
Appuie à nouveau sur « Entrée » pour confirmer



Utilise les touches curseur pour sélectionner `<Finish>`



Appuie sur « Entrée » pour redémarrer.



Comment utiliser un Raspberry Pi pour savoir qui est venu dans ta chambre ? Créer un détecteur de parent qui utilise un capteur de mouvement pour déclencher l'enregistrement vidéo via le module caméra Raspberry Pi.

Ce dont tu auras besoin

Ce dont tu auras besoin

Matériel

- Un Raspberry Pi
- Un module caméra Raspberry Pi
- Un module capteur de mouvement PIR (par exemple sur The Pi Hut (https://thepihut.com/products/pir-motion-sensor-module))
- 3 fils de connexion femelle-femelle ou plus (par exemple sur The Pi Hut (https://thepihut.com/products/adafruit-premium-female-female-jumper-wires-20-x-6-150mm))

Extras

- Camera Board 360 Gooseneck Mount (from modmypi.com (https://www.modmypi.com/raspberry-pi/camera/camera-cases/camera-board-360-gooseneck-mount)) ou équivalent

Informations supplémentaires pour les enseignants

Si vous avez besoin d'imprimer ce projet, veuillez utiliser la version imprimable (https://projects.raspberrypi.org/en/projects/parent-detector/print).

Vous pouvez trouver la solution pour ce projet ici (https://rpf.io/p/en/parent-detector-get).

Étape 2 Connecter le capteur de mouvement PIR

Pour ce projet, nous allons utiliser un capteur de mouvement infrarouge passif (PIR - passive infrared).

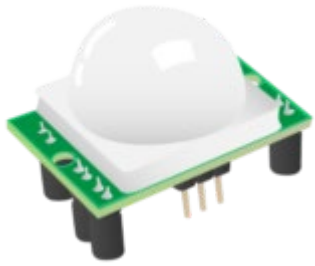
Qu'est-ce qu'un capteur de mouvement PIR ?

À propos des capteurs de mouvement PIR

Pour ce projet, nous allons utiliser un capteur de mouvement infrarouge passif (PIR - passive infrared). Tu en as probablement déjà vu auparavant, car ils sont souvent utilisés dans les systèmes d'alarme anti-effraction.

Si la température d'un objet ou d'un organisme est supérieure au zéro absolu (soit -273,15 °C !), celui-ci émet un rayonnement infrarouge. Les longueurs d'onde infrarouges ne sont pas visibles pour l'œil humain, mais elles peuvent être détectées par l'électronique à l'intérieur d'un capteur PIR.

Le capteur est considéré comme passif car il n'envoie aucun signal pour détecter les mouvements. Il fonctionne en s'ajustant à la signature infrarouge de la pièce dans laquelle il se trouve, puis en observant les variations. Tout objet se déplaçant dans la pièce perturbera la signature infrarouge, et le module PIR détectera cette perturbation.



Nous n'avons pas besoin de nous soucier du fonctionnement interne du capteur de mouvement. Ce qui nous intéresse, ce sont ses trois broches qui peuvent être utilisées pour le connecter au Raspberry Pi.

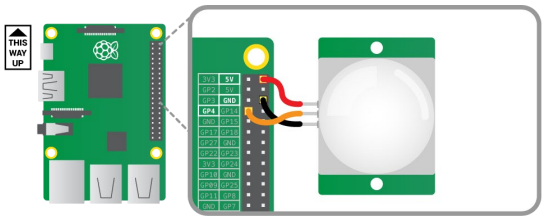
Connecte ton capteur PIR à la broche GPIO 4.

Connexion d'un capteur de mouvement PIR

Sur certains capteurs PIR, les broches sont clairement étiquetées sur le circuit imprimé. D'autres ont les étiquettes cachées sous leur capuchon. Si c'est le cas pour ton capteur, retire délicatement son capuchon afin de pouvoir distinguer les broches.

À l'aide de trois fils de connexion femelle-femelle, connecte chaque broche du capteur PIR aux broches correspondantes du Raspberry Pi :

- Connecte la broche du capteur PIR portant l'étiquette VCC à la broche 5 V du Raspberry Pi. Cela permet d'alimenter le capteur PIR.
- Connecte la broche étiquetée GND à une broche de masse sur le Pi (également étiquetée GND). Ceci permet de compléter le circuit.
- Connecte celle étiquetée OUT à n'importe quelle broche GPIO numérotée sur le Pi. Dans cet exemple, nous avons choisi GPIO 4. La broche OUT émettra une tension lorsque le capteur détectera un mouvement. La tension sera alors reçue par le Raspberry Pi.



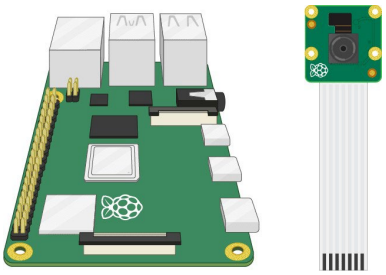
Sur le module PIR, tu peux voir deux composants orange avec des fiches adaptées à un tournevis cruciforme (voir ci-dessus). On les appelle des potentiomètres : ils te permettent d'ajuster la sensibilité et le temps de détection du capteur. Commence par définir le potentiomètre de sensibilité à son maximum et le potentiomètre de temps à son minimum. Tu pourras les modifier plus tard si tu le souhaites.

Étape 3 Connecter la caméra

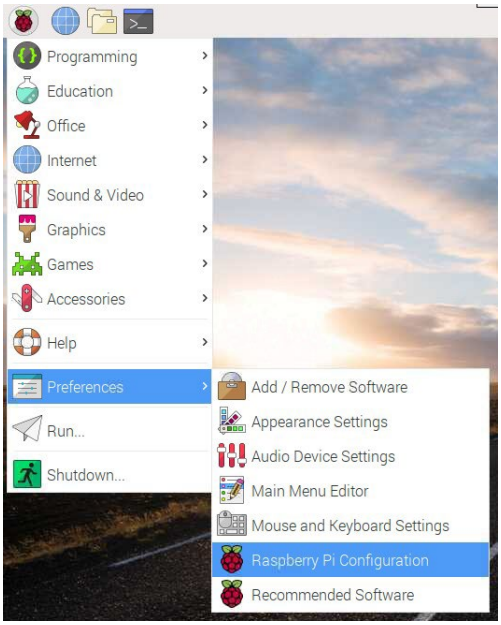
Avant d'allumer ton Raspberry Pi, connecte le module caméra à celui-ci.

Connecter un module caméra du Raspberry Pi

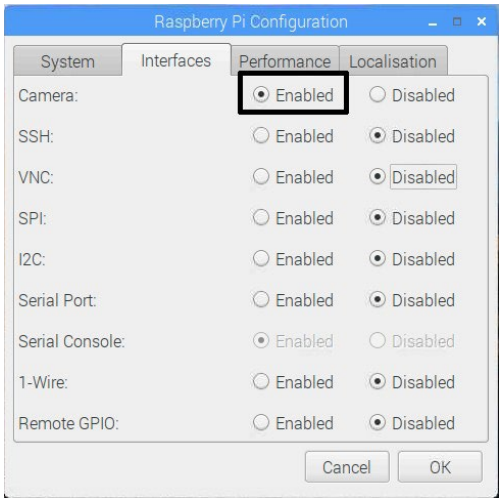
- Vérifie que le Raspberry Pi est éteint.
- Localise le port du module caméra.
- Tire doucement sur les bords du clip en plastique du port.
- Insère le câble ruban du module caméra ; vérifie que le câble est dans le bon sens.
- Remets en place le clip en plastique.



- Démarre le Raspberry Pi.
- Va dans le menu principal et ouvre l'outil de configuration du Raspberry Pi.



- Sélectionne l'onglet Interfaces et vérifie que la caméra est activée :



- Redémarre le Raspberry Pi.

Étape 4 Tester le capteur de mouvement PIR

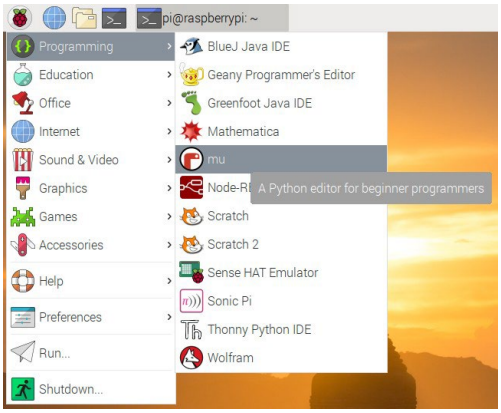
Nous allons écrire un code pour afficher « Motion detected! » (« Mouvement détecté ») lorsque le capteur PIR détecte un mouvement.

Ouvre Mu, crée un nouveau fichier et enregistre-le sous le nom de `parent_detector.py`.

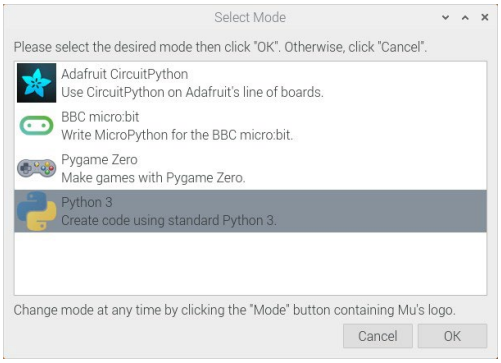
Comment ouvrir Mu

Comment ouvrir Mu

- Va dans le menu Programming et clique sur Mu.



- Choisis ensuite le mode dans lequel tu souhaites utiliser Mu. Choisis Python 3 si tu crées un nouveau script Python.



Monte ton capteur PIR sur GPIO 4.

J'ai besoin d'un indice

Voici le code complet

parent_detector.py

```
from gpiozero import MotionSensor

pir = MotionSensor(4)
```

-

Ajoute du code afin que « Motion detected! » (« Mouvement détecté ») s'affiche sur l'écran lorsque le capteur PIR détecte un mouvement.

J'ai besoin d'un indice

Voici le code :

parent_detector.py

```
from gpiozero import MotionSensor

pir = MotionSensor(4)

pir.wait_for_motion()
print("Motion detected!")
```

Enregistre ton code et clique sur Run pour l'exécuter. Tu devrais voir les mots `Motion detected!` apparaître à l'écran lorsque le capteur de mouvement est déclenché.

Pour le moment, ton code ne détecte le mouvement qu'une fois, puis le programme se termine. Place ton code à l'intérieur d'une boucle infinie afin que Python continue d'attendre un signal du capteur de mouvement et affiche `Motion detected!` chaque fois que le capteur est déclenché. Pour quitter ton programme, clique sur Stop.

En condition de boucle While True dans Python

L'objectif d'une boucle While est de répéter le code plusieurs fois tant qu'une condition est `True`. C'est pourquoi les boucles sont parfois appelées boucles conditionnelles.

L'exemple ci-dessous est une boucle While qui tourne indéfiniment : une boucle infinie. La boucle tourne indéfiniment car la condition est toujours `True`.

```
while True:
    print("Hello world")
```

Remarque : La ligne `while` indique la condition de la boucle. La ligne de code `print` en dessous est légèrement plus à droite. Il s'agit d'une indentation : la ligne est en retrait pour montrer qu'elle est à l'intérieur de la boucle. Tout code à l'intérieur de la boucle sera répété.

Une boucle infinie est utile lorsque tu souhaites effectuer les mêmes actions à plusieurs reprises, par exemple pour vérifier la valeur d'un capteur. Une boucle infinie comme celle-ci tournera indéfiniment, ce qui signifie que toutes les lignes de code écrites après la boucle n'arriveront jamais. Ceci est connu sous le nom de blocage : un programme bloque l'exécution de tout autre code.

J'ai besoin d'un indice

Tu dois mettre ton code permettant de détecter le mouvement dans une boucle.

parent_detector.py

```
from gpiozero import MotionSensor

pir = MotionSensor(4)

while True:
    pir.wait_for_motion()
    print("Motion detected!")
```

Étape 5 Configurer l'aperçu de la caméra

Au début de ton programme, importe la classe `Picamera` depuis la bibliothèque `picamera` afin de pouvoir utiliser du code pour contrôler le module caméra.

parent_detector.py

```
from gpiozero import MotionSensor
from picamera import PiCamera

pic = MotionSensor(4)

while True:
    pir.wait_for_motion()
    print("Motion detected!")
```

Ajoute une ligne de code pour créer un objet `PiCamera`. Assure-toi que cette ligne de code est au-dessus de la boucle infinie.

parent_detector.py

```
from gpiozero import MotionSensor
from picamera import PiCamera

pic = MotionSensor(4)
camera = PiCamera()

while True:
    pir.wait_for_motion()
    print("Motion detected!")
```

Complète ton code existant afin de démarrer l'aperçu de la caméra lorsque le capteur est activé et d'arrêter l'aperçu lorsqu'aucun mouvement n'est détecté.

Prendre une photo avec PiCamera

Tu peux utiliser Python et le module `picamera` pour prendre des photos avec le Raspberry Pi et son module caméra.

- Importe la classe `PiCamera` et crée un objet `camera`

```
from picamera import PiCamera

camera = PiCamera()
```

- Pour prendre une photo, tu peux utiliser la méthode `capture()`. Pour ce faire, tu dois indiquer à Python où tu souhaites enregistrer la photo et comment tu veux la nommer. Dans l'exemple ci-dessous, la photo sera nommée `selfie.png` et sera enregistrée dans le répertoire `/home/pi/`.

```
from picamera import PiCamera
```

```
camera = PiCamera()
camera.capture('home/pi/selfie.png')
camera.close()
```

- Exécute ton code, puis utilise le Gestionnaire de fichiers ou le Terminal pour rechercher le fichier `selfie.png`.

J'ai besoin d'un indice

Voici le code terminé :

parent_detector.py

```
from gpiozero import MotionSensor
from picamera import PiCamera

pir = MotionSensor(4)
camera = PiCamera()

while True:
    pir.wait_for_motion()
    print("Motion detected!")
    camera.start_preview()
    pir.wait_for_no_motion()
    camera.stop_preview()
```

Enregistre ton code et exécute-le. Vérifie que l'aperçu de la caméra apparaît lorsque le capteur de mouvement est activé et s'arrête lorsque le capteur de mouvement n'est plus actif.

Étape 6 Enregistrer la vidéo dans un fichier

Voir l'intrus sur l'écran dans un aperçu de la caméra pendant qu'ils sont dans la pièce ne t'aide pas beaucoup. Enregistrons plutôt une vidéo de l'intrus que tu pourras visionner plus tard lorsque tu rentreras chez toi.

Crée une variable appelée `filename` dans ta boucle infinie afin de stocker le nom du fichier vidéo.

parent_detector.py

```
from gpiozero import MotionSensor
from picamera import PiCamera

pir = MotionSensor(4)
camera = PiCamera()
filename = "intruder.h264"

while True:
    pir.wait_for_motion()
    print("Motion detected!")
    camera.start_preview()
    pir.wait_for_no_motion()
    camera.stop_preview()
```

Si tu te poses la question, `.h264` correspond au format de la vidéo.

Trouve la ligne de code qui démarre l'aperçu de la caméra et remplace-la par une ligne de code qui démarre un enregistrement vidéo.

parent_detector.py

```
from gpiozero import MotionSensor
from picamera import PiCamera

pic = MotionSensor(4)
camera = PiCamera()
filename = "intruder.h264"

while True:
    pir.wait_for_motion()
    print("Motion detected!")
    camera.start_recording(filename)
    pir.wait_for_no_motion()
    camera.stop_preview()
```

Trouve la ligne de code qui arrête l'aperçu de la caméra et remplace-la par une ligne de code qui arrête un enregistrement vidéo.

J'ai besoin d'un indice

Voici le code terminé :

```
from gpiozero import MotionSensor
from picamera import PiCamera

pir = MotionSensor(4)
camera = PiCamera()
filename = "intruder.h264"

while True:
    pir.wait_for_motion()
    print("Motion detected!")
    camera.start_recording(filename)
    pir.wait_for_no_motion()
    camera.stop_recording()
```

Enregistre ton programme et exécute-le. Vérifie qu'un fichier appelé `intruder.h264` apparaît dans le même dossier que ton fichier `parent-detector.py`.

Chaque fois qu'un nouvel intrus déclenche le capteur de mouvement, le fichier vidéo sera remplacé. Si tu as plusieurs parents ou frères et sœurs qui s'introduisent dans ta chambre, tu aimerais sûrement conserver des vidéos de chacun d'entre eux. Peux-tu écrire des lignes de code pour trouver automatiquement la date et l'heure actuelles, et les ajouter au nom du fichier vidéo ? Ensuite, chaque vidéo que tu enregistreras aura un nom de fichier différent.

Création d'horodatages avec Python

Il se peut que tu veuilles parfois obtenir la date et l'heure actuelles sous la forme d'une chaîne de caractères, par exemple pour donner des noms uniques à des fichiers ou à des données. Le module `datetime` est extrêmement utile pour créer de tels horodatages.

- Tu dois d'abord importer le module `datetime` , et plus particulièrement sa classe de `datetime`

```
from datetime import datetime
```

- Si tu souhaites utiliser l'horodatage pour nommer un fichier, tu peux utiliser ceci par exemple :

```
filename = "{0:%Y}-{0:%m}-{0:%d}".format(datetime.now())
```

  - Le `{ }` est utilisé comme paramètre fictif dans la chaîne.
  - Le `0` indique à la commande `print` d'utiliser le 0ème objet qui lui est transmis. Dans ce cas, l'objet est `now`
  - Le code `:%Y` indique à la commande `print` de prendre l'année complète de l'objet `datetime.now()`
  - Peux-tu deviner ce que les codes `:%m` et `:%d` désignent ?
- Jette un coup d'œil ici (http:/strftime.org/) pour d'autres codes que tu peux utiliser avec le module `datetime`

Étape 7 Visionner la vidéo

Lorsque tu retourneras dans ta chambre, tu pourras lire toutes les vidéos créées par ton détecteur de parent à l'aide du logiciel OMXPlayer.

Lire une vidéo avec OMXPlayer

Lire une vidéo avec OMXPlayer

1. Ouvre une fenêtre de terminal.

2. Saisis la commande suivante dans la fenêtre, en remplaçant `filename` par le nom du fichier que tu souhaites lire. Ensuite, appuie sur Entrée.

```
omxplayer filename
```

Par exemple, si ton fichier s'appelle `test.h264`, saisis cette commande et appuie sur Entrée.

```
omxplayer test.h264
```

Étape 8 : mode furtif

Tu as terminé ton détecteur de parents, mais pourquoi ne pas essayer de passer au niveau supérieur en l'utilisant en mode furtif ?

- Tu peux désactiver la LED rouge sur la carte de la caméra qui s'allume normalement lorsque tu démarres ton programme Python.

J'ai besoin d'un indice

Ouvre une fenêtre de terminal et saisis la commande suivante pour commencer à modifier le fichier : `config.txt`

```
sudo nano /boot/config.txt
```

Ajoute la ligne suivante à la fin du fichier :

```
disable_camera_led=1
```

Appuie sur `Ctrl + O` pour enregistrer et sur `Ctrl + X` pour quitter. Les modifications ne prendront effet qu'après un redémarrage : tape la commande suivante dans le terminal pour cela :

```
sudo reboot
```

- L'aperçu de la caméra peut indiquer à vos intrus qu'ils ont été repérés. Peux-tu supprimer le code d'aperçu de la caméra de ton script ?

Pour aller plus loin, consulte la page du projet Astro Pi sur https://esero.fr/projets/astro-pi/