



Chapitre N°9.) Détecter les mouvements

I. Le capteur de mouvement PIR	147
1. Deux fentes	147
2. Lentilles de Fresnel	147
3. Plusieurs lentilles	148
4. Réglage de la sensibilité	149
5. Principe de fonctionnement	150
6. Des questions quand même.....	150
II. Montage	150
1. Réglage de la sensibilité	150
2. Câblage.....	150
3. Programmation AVEC gpiozero.....	150
a. La documentation gpiozero	150
b. La classe MotionSensor	151
c. Les méthodes disponibles.....	151
d. Codage	151
e. Avec des fonctions.....	152
III. Surveillance d'une piece	152
1. Cahier des charges	152
2. Première version : envoi d'un mail lors d'une détection de mouvement	152
a. Envoi de mails	152
b. Envoi mail sur détection de mouvement	153
3. Envoyer une image	153
Annexe I : autoriser les applications à utiliser gmail	154
Étape N°1 : accès à mon compte	154
Étape N°2 : Connexion et sécurité	154
Étape N°3 : Cliquez sur Activer l'accès même si c'est déconseillé	155
Annexe II : fonctions récupération du nom et de l'adresse IP	156



BTS SN – EC

Support de formation Raspberry Pi

Détecter les mouvements



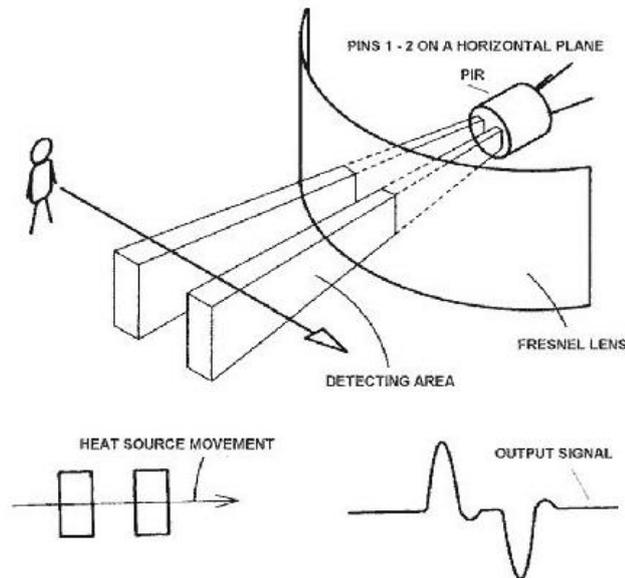
Annexe III : envoi d'une image..... 157



I. Le capteur de mouvement PIR

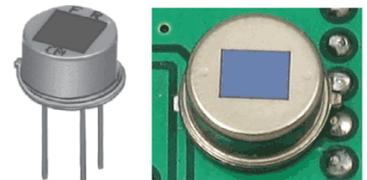
1. Deux fentes

(D'après <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>)



Le capteur PIR (PIR = Pyroelectric InfraRed) se compose deux fentes, chacune d'elle est constituée d'un matériau sensible à l'infrarouge. Lorsque le capteur est au repos, les deux fentes détectent la même quantité infra-rouge, celui de la pièce, des murs ou de l'extérieur. Quand un corps chaud comme un être humain ou un animal passe, la première fente détecte mais pas la seconde : le différentiel est positif. Inversement, lorsque le corps chaud quitte la zone de détection, le différentiel est négatif. Ce sont ces deux changements d'impulsions qui permettent de détecter un mouvement.

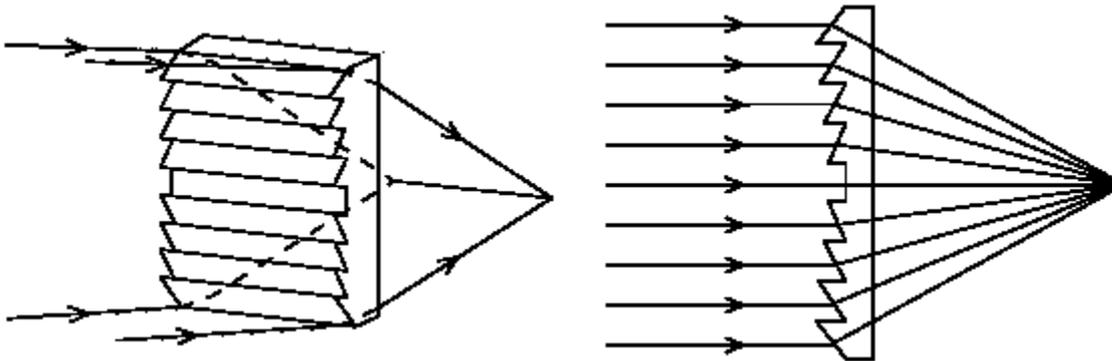
Le capteur IR est lui-même logé dans un métal scellé hermétiquement pour augmenter l'immunité au bruit/ température / humidité. Au-dessus, se trouve une fenêtre laissant passer les infrarouges.



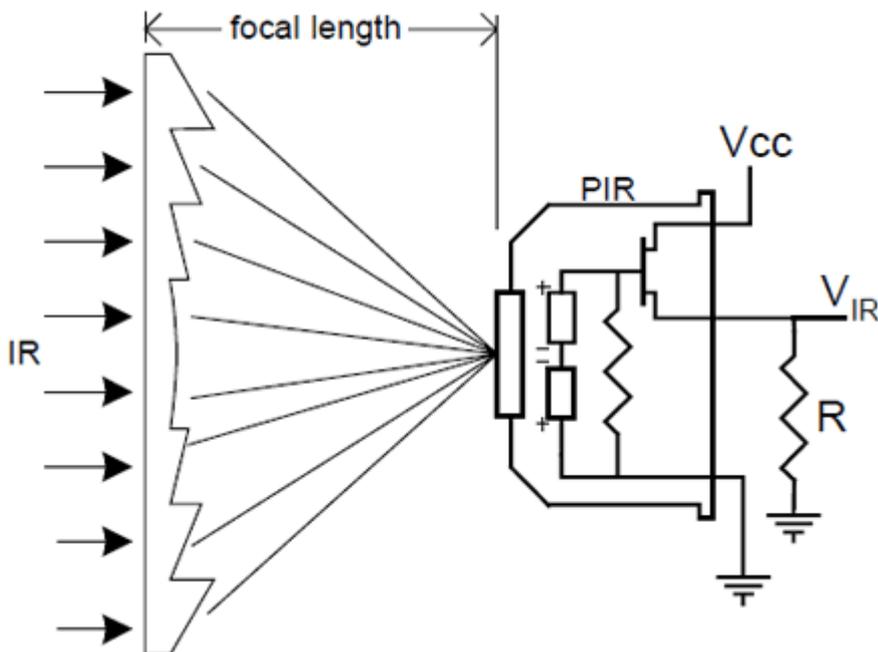
2. Lentilles de Fresnel

Pour l'instant, la détection ne se fait que sur deux petits rectangles. Afin d'avoir une plus grande zone de détection, au dessus se trouve une lentille de Fresnel qui condense les ondes infrarouges.

La lentille de Fresnel condense la lumière, offrant une plus grande gamme de IR au capteur.



[Image de BHIens.com](#)



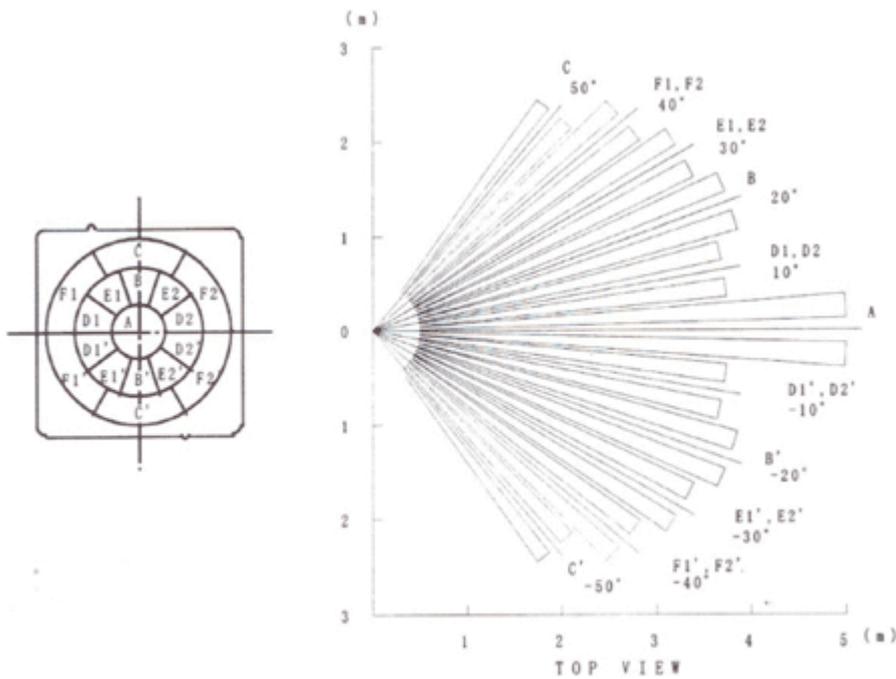
[Image de Cypress appnote 2105](#)

3. Plusieurs lentilles

Pour l'instant nous n'avons que deux grandes zones de détection et nous voudrions une multitude de petites zones. Ainsi la lentille est en fait plusieurs sections de lentilles de Fresnel



Bien sûr, plus le mouvement se trouvera dans l'axe, meilleur sera la détection :



4. Réglage de la sensibilité

Deux vis permettent de régler la sensibilité ainsi que le temps pendant lequel il est à l'état haut lors d'une détection

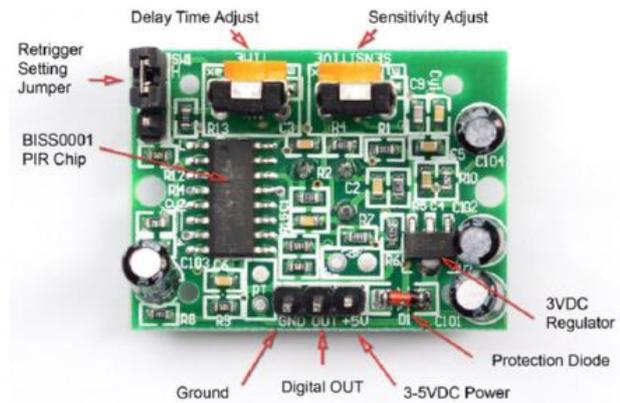
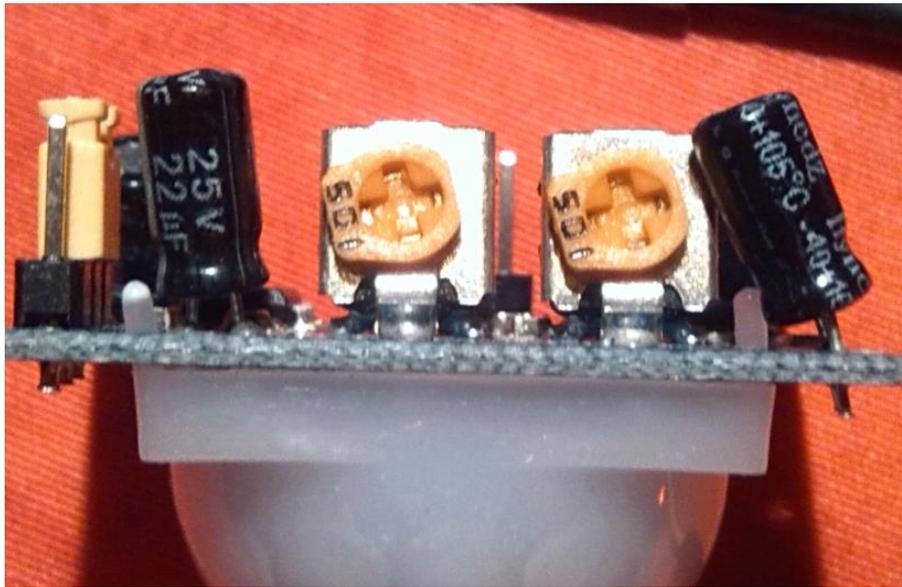


Figure 4 : à utiliser pour une question difficile

Le problème avec les PIRs, c'est leur stabilité : il peut alterner des états haut/bas plusieurs fois à la suite... Après de nombreux essais, la configuration qui apporte le plus de stabilité est la suivante :



5. Principe de fonctionnement

Dès que le PIR détecte un mouvement, il met sa broche OUT à l'état haut pendant un certain temps (réglable avec une des deux vis).

6. Des questions quand même...

🔧 Ce capteur fonctionne-t-il à 3.3V ou à 5 ? A 4 volts ?

En plus compliqué, utilisez le schéma du Réglage de la [4.Réglage de la sensibilité](#)

🔧 Quelle est la tension qui est fourni à la sortie du capteur ?

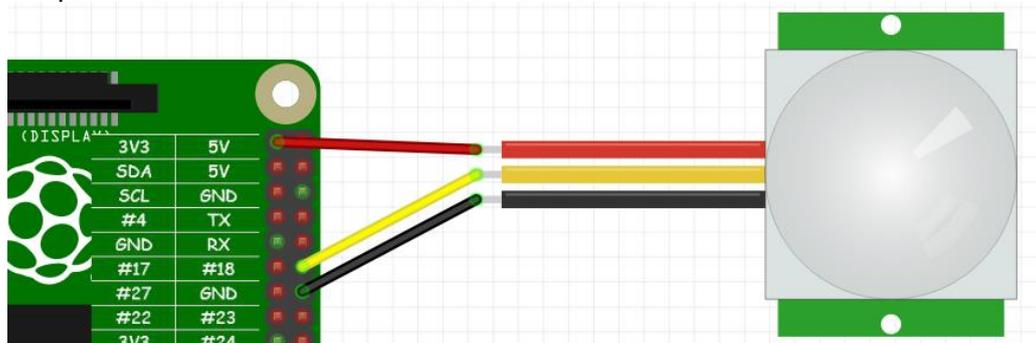
I. Montage

1. Réglage de la sensibilité

🔧 En suivant les indications dans la fiche ressource, régler correctement la sensibilité

2. Câblage

Le montage est simple : le PIR s'allume en 3.3V ou en 5V.

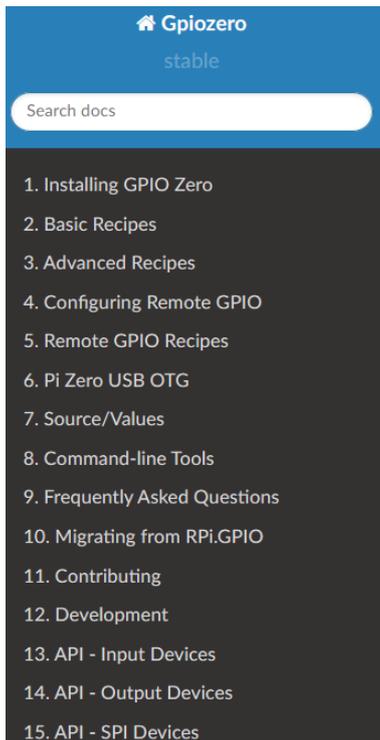


3. Programmation AVEC gpiozero

4. La documentation gpiozero

Elle se trouve à <https://gpiozero.readthedocs.io/en/stable/index.html>.

Pour l'instant elle est en anglais, on verra les prochaines grandes vacances pour la traduire.



Sur la gauche, soit que vous utilisez la recherche soit vous allez directement à la documentation du composant.

Le PIR est-il une sortie ou une entrée ? Est-ce donc API – Input Devices ou API – Output Devices ?

Allez sous **Regular Classes** et retrouvez la documentation de *MotionSensor*

Testez l'exemple donné sous fond vert après avoir effectué le montage

5. La classe *MotionSensor*

Elle est importée de gpiozero est demande comme argument la broche sur laquelle elle est branchée :

```
pir = MotionSensor(4)
```

6. Les méthodes disponibles

Vous allez utiliser un premier attribut de la classe *MotionSensor*

motion_detected : retourne True si le capteur est actif sinon False

7. Codage

Soit le code Arduino suivant :

```
/*
 * PIR
 */
int Led = 13; int PIR = 2; // le PIR est câblé sur la broche 2
int etatPrecedent = LOW; // au départ, l'état précédent est bas
void setup() {
  pinMode(Led, OUTPUT); // declare LED as output
  pinMode(PIR, INPUT); // declare sensor as input
  Serial.begin(9600);
  Serial.println("Demarrage");
}
void loop(){
  if (digitalRead(PIR) == HIGH) { // état haut sur la sortie du PIR ?
    /* si l'état précédent est bas et que le nouvel état est haut
    cela veut dire qu'il vient de détecter un mouvement
    sinon on ne vas pas encore écrire Mouvement détecté */
    if (etatPrecedent == LOW) {
      digitalWrite(Led, HIGH); // Allume la Led
      Serial.println("Mouvement détecté");
      etatPrecedent = HIGH;
    }
  }
}
```



```

}
}
else {
    digitalWrite(Led, LOW); // éteint la led
    if (etatPrecedent == HIGH)
    {
        // comme l'état précédent était Haut alors c'est la fin de la détection
        Serial.println("Détection terminée\n");
        etatPrecedent = LOW;
    }
}
}
}

```

☞ Le traduire en python pour le RPI avec le module gpiozero en utilisant `motion_detected`

8. Avec des fonctions

L'attribut `when_motion` fait appel à une fonction quand le capteur passe de l'état bas à haut.

`when_no_motion` fait la même chose pour l'état haut vers bas. Leur fonctionnement est identique aux boutons (`button.when_pressed = led.on`)

☞ Réécrire le plus simplement possible le programme précédent

II. Surveillance d'une pièce

1. Cahier des charges

Un chef d'entreprise voudrait faire surveiller son entrepôt à l'aide d'un détecteur de présence mis à des points d'entrées possibles (portes, fenêtres). Il a déjà un tel système à base de caméra mais c'est très souvent une fausse alerte. C'est pourquoi il voudrait être prévenu par mail dès qu'une intrusion est produite avec une photo.

2. Première version : envoi d'un mail lors d'une détection de mouvement

3. Envoi de mails

Soit le programme qui envoie un mail de l'adresse toto.branly@gmail.com vers tomczak.poubelle@outlook.fr

```

from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import smtplib

print('Envoi en cours')
msg = MIMEMultipart()
msg['From'] = 'toto.branly@gmail.com'

```



...)
sms :
alerte.

se

mail



```
msg['To'] = 'tomczak.poubelle@outlook.fr'
msg['Subject'] = 'coucou'
message = 'On y va'

msg.attach(MIMEText(message))

mailserver = smtplib.SMTP('smtp.gmail.com', 587)
mailserver.ehlo()
mailserver.starttls()
mailserver.ehlo()
mailserver.login('toto.branly@gmail.com', 'totobranlyfaitsnec')
mailserver.sendmail('toto.branly@gmail.com', 'tomczak.poubelle@outlook.fr',
msg.as_string() )
mailserver.quit()

print('Message envoyé')
```

-  Créer une adresse mail Gmail ou utilisez la vôtre
-  Configurez-la comme indiqué dans [Annexe 1 : autoriser les applications à utiliser gmail](#)
-  Choisissez un destinataire pour votre mail
-  Modifiez le code ci-dessus avec la nouvelle adresse créée
-  Ecrivez une fonction qui prend trois paramètres **def** EnvoyerMail(destinataire, objet, message): et l'utiliser

4. Envoi mail sur détection de mouvement

-  Lorsque le PIR détecte un mouvement, envoyez un mail avec ces caractéristiques :
 - Le sujet sera 'Mvt détecté le ' + la date et l'heure
 - Le corps du mail sera le nom de la raspberryPi et l'adresse IP (utilisez l'[Annexe II : fonctions récupération du nom et de l'adresse IP](#))

5. Envoyer une image

-  En vous inspirant de l'annexe III, rejoindre à votre mail, une photo (n'importe laquelle)

6. Prendre une photo d'une webcam

-  En suivant <https://www.raspberrypi.org/documentation/usage/webcams/> installez et configurez une webcam

La commande :

```
fswebcam image.jpg
```

prend une photo.

-  A l'aide de la fonction `os.system` (<https://www.journaldev.com/16140/python-system-command-os-subprocess-call>) , écrivez un programme python qui prend une seule photo

-  Envoyez un mail avec la photo prise par la webcam



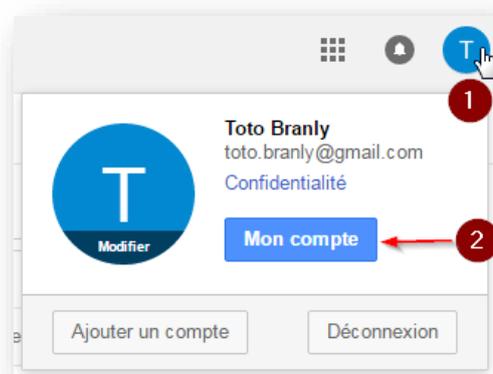
III. Annexe I : autoriser les applications à utiliser gmail

Par défaut, la configuration de gmail ne permet pas l'accès aux applications « moins sécurisées ». Néanmoins il est possible de le modifier.

Dans cet exemple le compte s'appelle toto.branly@gmail.com

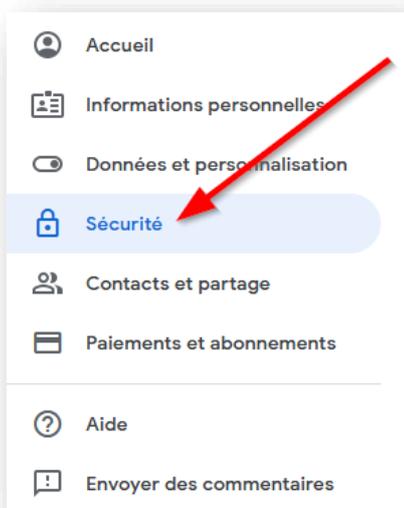
1. Étape N°1 : accès à mon compte

🔒 ❶ Clic droit sur l'initial de votre compte en haut à droite puis ❷ sur **Mon compte** :

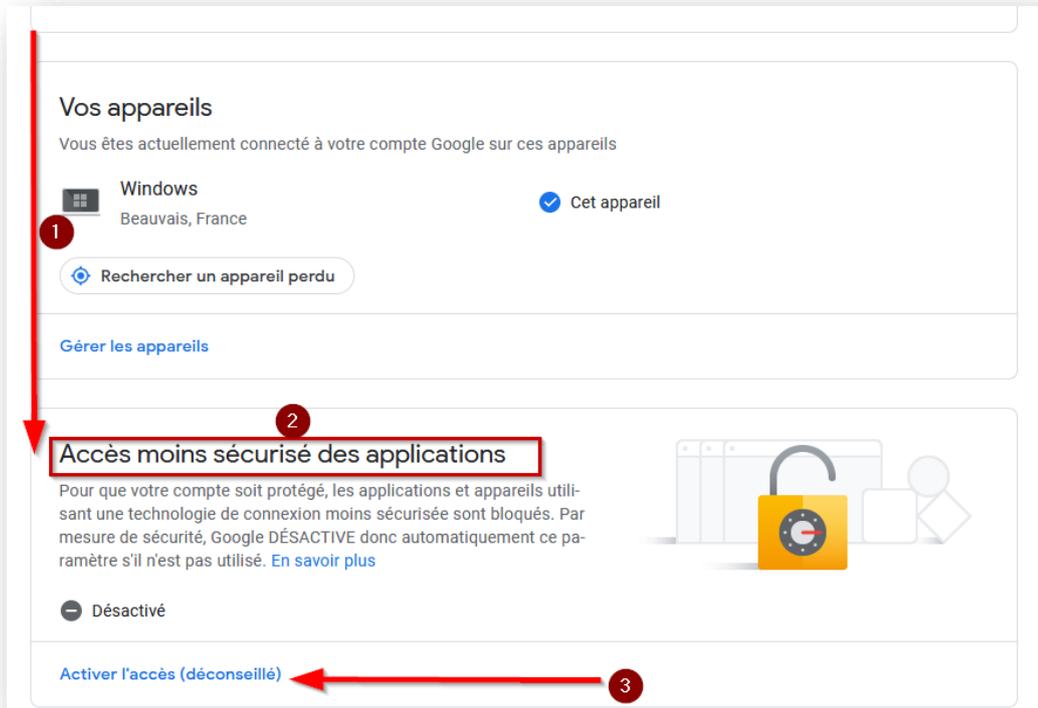


2. Étape N°2 : Connexion et sécurité

🔒 Choisissez Sécurité



🔒 Allez tout en bas de cette page :



3. Étape N°3 : Cliquez sur Activer l'accès même si c'est déconseillé

 Autorisez les applications moins sécurisées

Paramètre "Autoriser les applications moins sécurisées" désactivé 

Paramètre "Autoriser les applications moins sécurisées" activé 



 Attendre une dizaine de secondes pour que la configuration soit active.



IV. Annexe II : fonctions récupération du nom et de l'adresse IP

```
import socket # pour récupérer le nom de la machine
from datetime import datetime # pour l'heure

# Permet de récupérer l'adresse d'une des interfaces
# eth0 correspond à l'ethernet, wlan0 à la connexion wifi

def getMac(interface):
    #lit le fichier /sys/class/net/eth0/address
    #contenant l'adresse mac de l'ethernet
    #idem avec wlan0 pour le wifi
    mac = open('/sys/class/net/'+interface+'/address').readline()
    return mac[0:17]

def getAddressIP():
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    s.connect(("8.8.8.8", 80))# ouvre une connection n'importe laquelle
    return s.getsockname()[0] #récupère l'adresse

print(getMac('eth0')) #adresse MAC

hostname = socket.gethostname() #nom de la machine hostname
print(hostname)

print(getAddressIP()) #adresse IP

dateHeure = datetime.now()
print(str(dateHeure))
Message = 'Mvt détecté le '+str(dateHeure)[0:10]+' à '+str(dateHeure)[11:19]
print(Message)
```



V. Annexe III : envoi d'une image

```
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import smtplib
from email.mime.base import MIMEBase
from email import encoders

print('Envoi en cours')
msg = MIMEMultipart()
msg['From'] = 'toto.branly@gmail.com'
msg['To'] = 'tomczak@hotmail.fr'
msg['Subject'] = 'coucou'
message = 'On y va'

msg.attach(MIMEText(message))

# with permet la gestion plus simple en cas d'erreur (pas besoin d'utiliser try:
except///
with open('toto.jpg', 'rb') as fichier:
    # MIME = Multipurpose Internet Mail Extensions
    # est un médias qui sera join au mail/
    # dans notre cas, il s'appelle toto, c'est un jpg
    mime = MIMEBase('image', 'jpg', filename='toto.jpg')
    # ajout des entêtes
    mime.add_header('Content-Disposition', 'attachment', filename='toto.jpg')
    mime.add_header('X-Attachment-Id', '0')
    mime.add_header('Content-ID', '<0>')
    # le contenu est le fichier
    mime.set_payload(fichier.read())
    # encodage en base 64
    encoders.encode_base64(mime)
    # on ajoute l'objet MIMEBase object à l'objet MIMEMultipart (bref on joint le
    fichier image)
    msg.attach(mime)

mailserver = smtplib.SMTP('smtp.gmail.com', 587)
mailserver.ehlo()
mailserver.starttls()
mailserver.ehlo()
mailserver.login('toto.branly@gmail.com', 'totobranlyfaitsnec')
mailserver.sendmail('toto.branly@gmail.com', 'tomczak@hotmail.fr', msg.as_string() )
mailserver.quit()

print('Message envoyé')
```