

# Système de détection faciale de la sécurité

Fait par : Hamza Fijahi-SF030T  
Themes : Enjeux Sociétaux

# PLAN :

## I. Introduction générale :

- Problématique
- Objectif

## II. Etude fonctionnelle :

- Analyse de besoin
- Diagramme d'exigence
- Etude de l'approche

## III. Problématique

- Comment peut-on réaliser un tel système

# I. Introduction générale :

- Problématique

Il s'agit d'étudier des approches mathématique afin d'obtenir un suivi d'un cible d'une manière lisse, et donner une présentation pratique de ce projet par l'élaboration des constituants facile à utiliser.

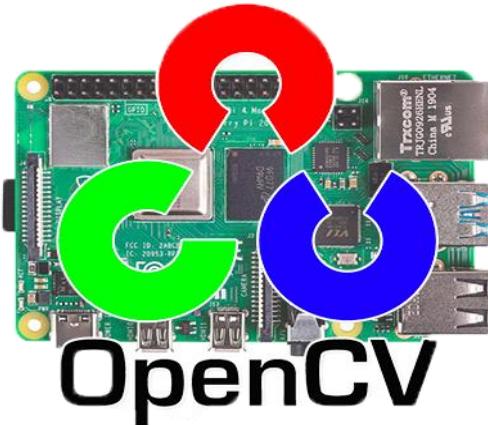
Application:

Dans le cas où un intrus sera détecter après l'activation par le propriétaire et l'envoyer un message



- Objectif

On étudie une expérience mathématique qui prédict le déplacement d'un cible et le temps nécessaire pour le fonctionnement de la camera par des approches mathématiques



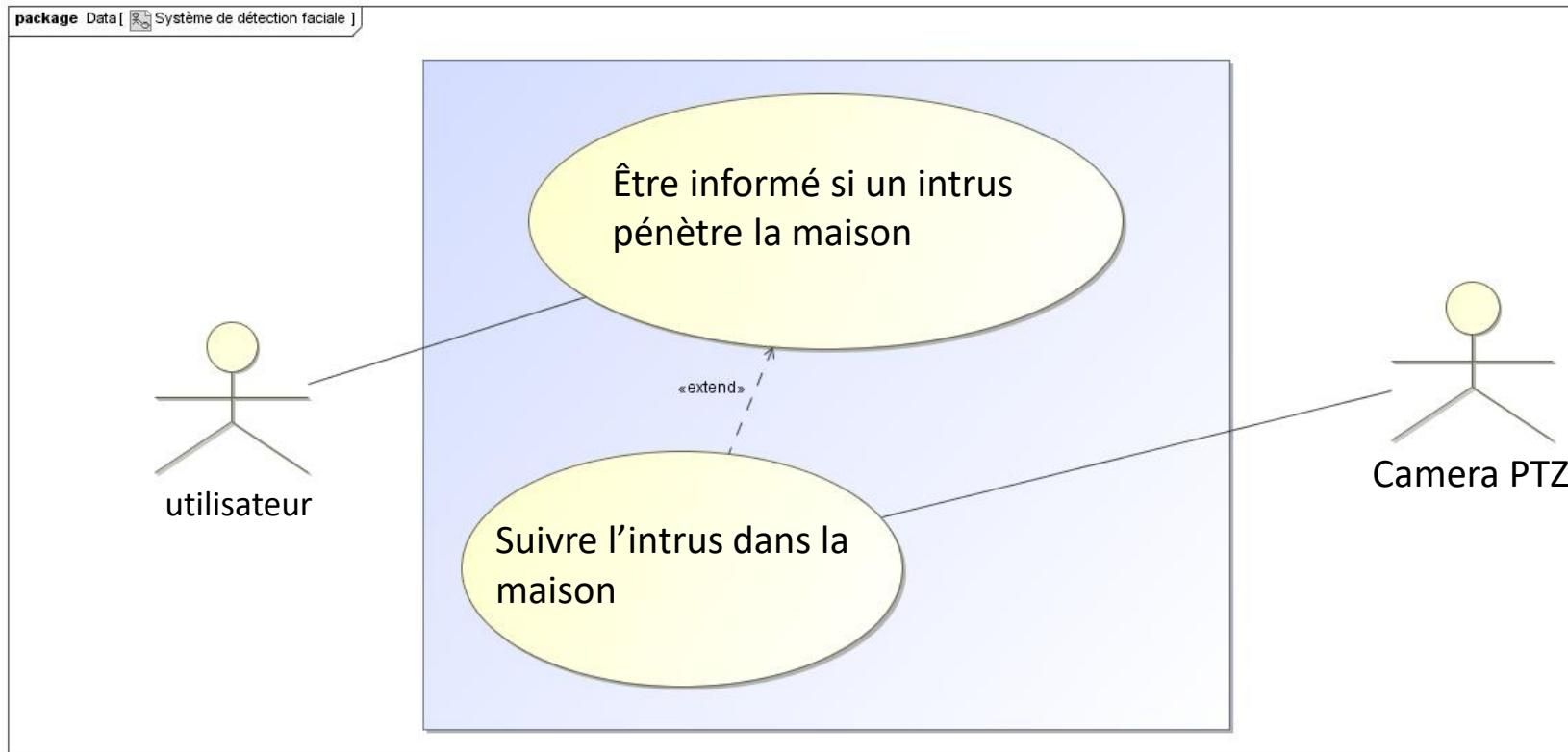
Puis, on l'applique dans une camera PTZ pour vérifier l'efficacité de cette étude ou les composants par d'autre approches.



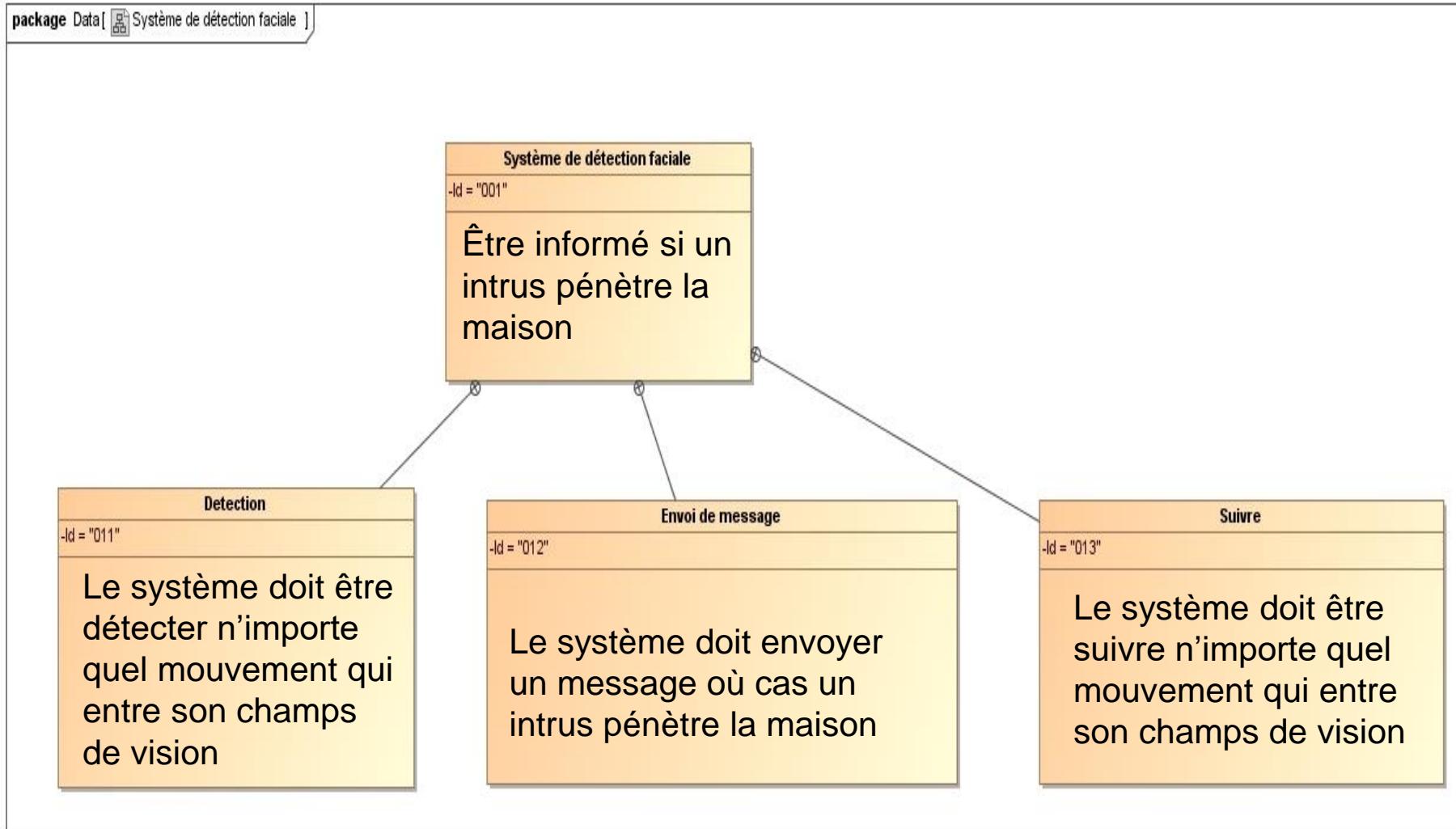
Enfin, une petite démonstration par Raspberry Pi qui nous permettre de réaliser un tel system complexe.

# II. Etude fonctionnelle :

- Analyse de besoin



## • Diagramme d'exigence



## • Etude de l'approche

Dans cette approche on distingue 2 phases :

Phase d'attente:

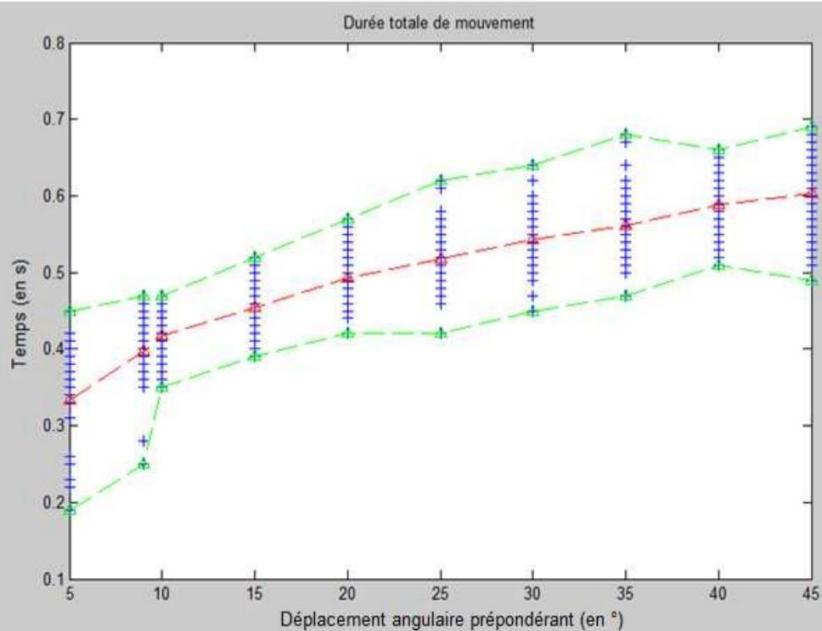


Figure 1

Phase de déplacement effectif:

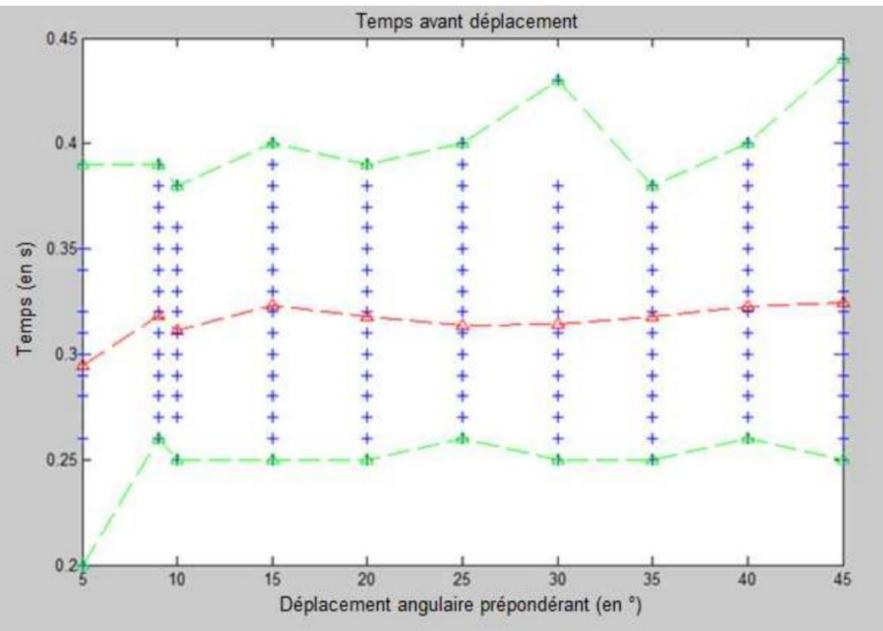


Figure 2

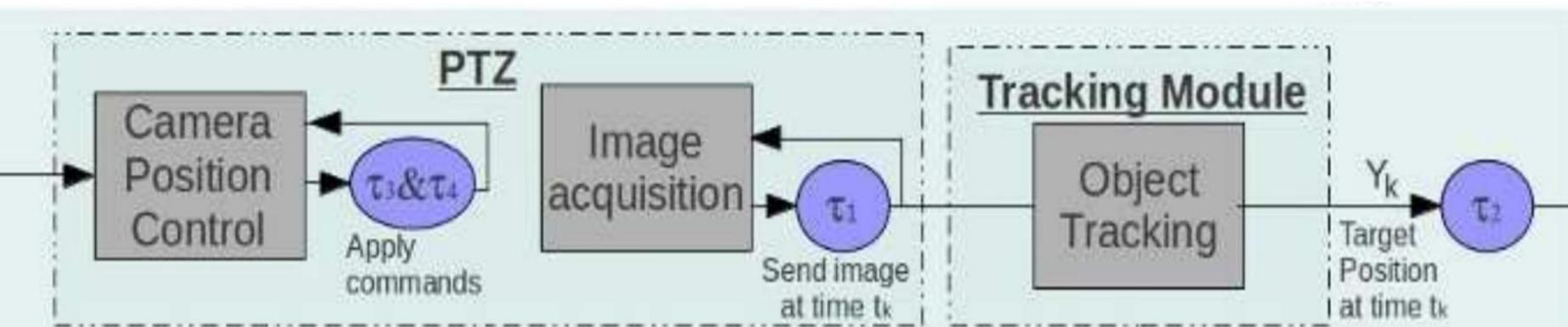
Ce qui nous amène à considérer 4 délais dans un commande complet pour la camera PTZ :

$\tau_1$  : temps de transmission de l'image entre la caméra PTZ et notre unité de traitement

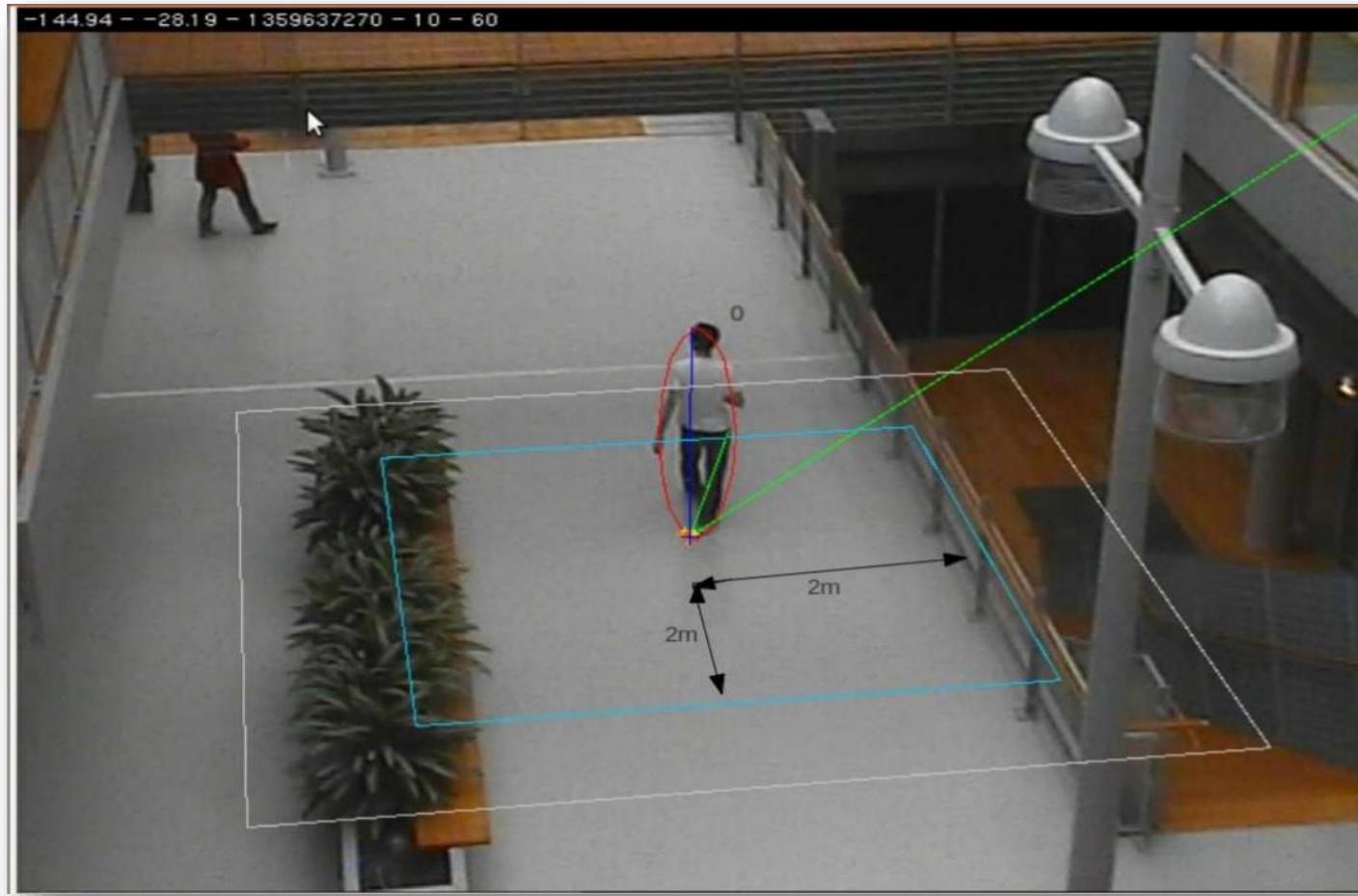
$\tau_2$  : temps de traitement

$\tau_3$  : temps d'attente de la caméra, étudie figure 1

$\tau_4$  : temps d'exécution réel de la commande envoyée à la caméra PTZ



Donnant un exemple de traitement d'image par le traqueur :



En adoptant par la suite les notions suivantes :

$X_k$  est la position de la cible au sol à l'instant  $t_k$  (vecteur d'état)

$Y_k$  est la position estimée par le traqueur de  $X_k$  (observation),

$\hat{X}_{k+1|k}$  est la position prédictive de  $X_{k+1}$  à  $t_k$ .

Avec :

$$t_{k+1} = t_k + \sum_{i=1}^4 \tau_i$$

Alors pour la prédite de la positon du cible on donne la relation suivante :

$$\hat{X}_{k+1|k} = Y_k + \bar{\nu} \cdot \hat{\tau}_{\text{move}}$$

$$\text{où } \bar{\nu} = \frac{\sum_{i=1}^D \Delta X_i}{\sum_{i=1}^D \tau_1^i + \tau_2^i}$$

Et avec :

$$\hat{\tau}_{\text{move}} = \hat{\tau}_3 + \hat{\tau}_4$$

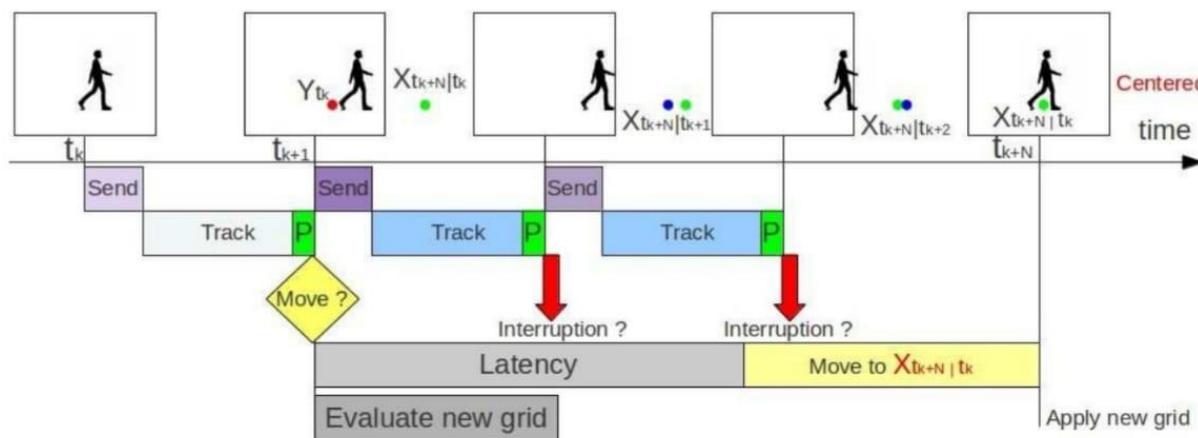
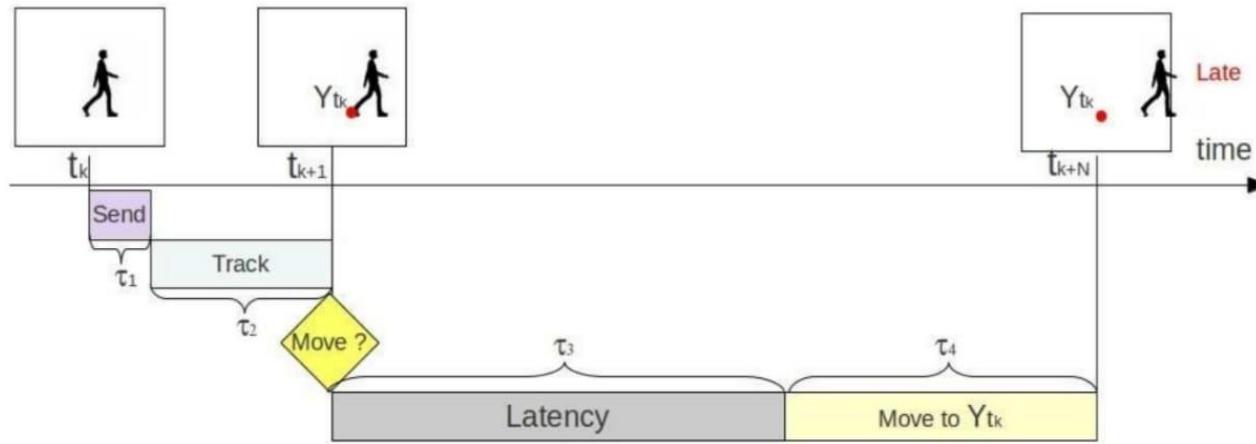
Dans cette approche pour la bien décision on donne la relation suivante :

$$\|\hat{X}_{k+1|k} - X_{ref}\|_\infty < \epsilon$$

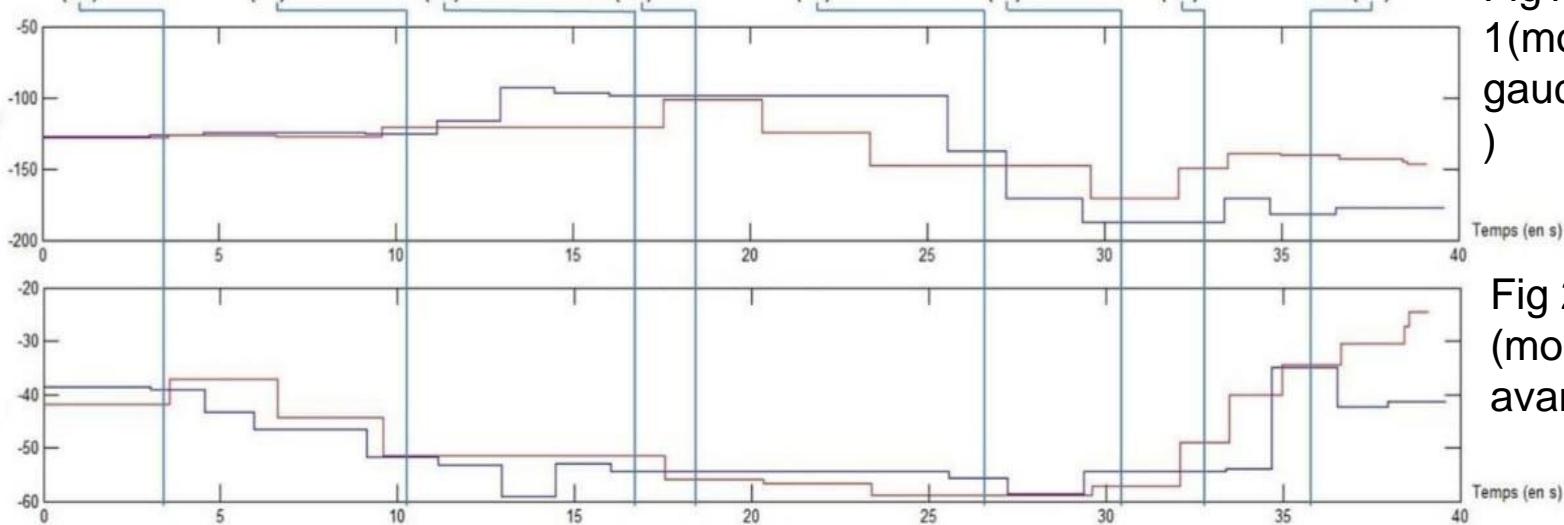
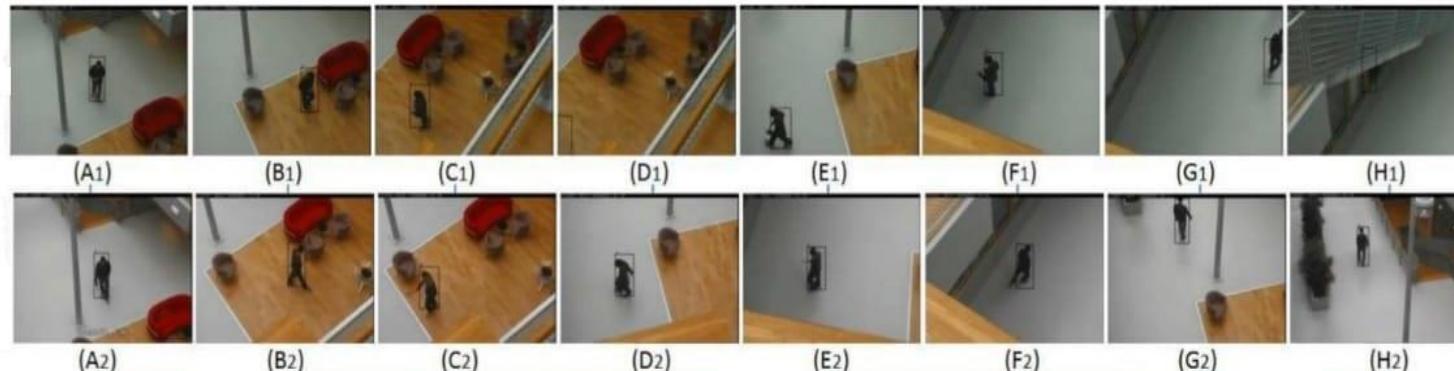
Avec dans la pratique :

$$\epsilon = 2m$$

Comparaison de synoptique entre la méthode classique et cette approche :



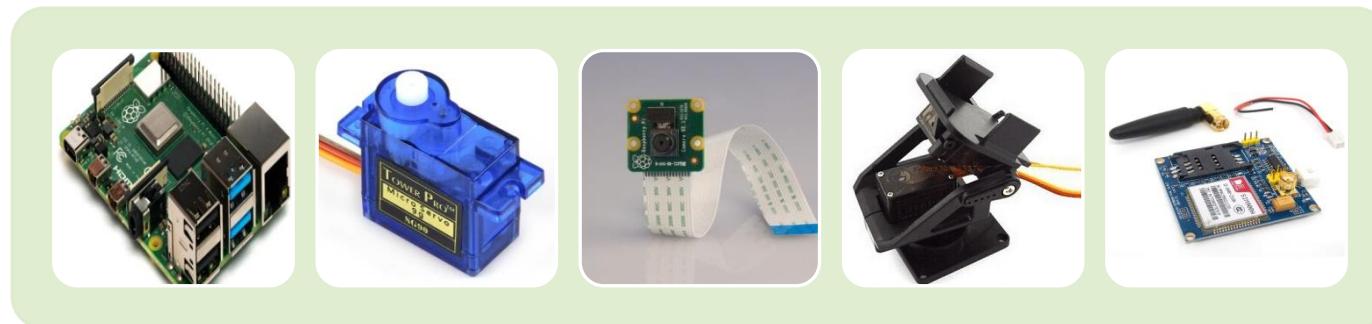
## Résultats :



### III. Problématique :

- Comment peut-on réaliser un tel système :

- Dans ce projet on a utilisée plusieurs composantes pour bien traiter et réaliser notre problématique c'est pour cela afin de la comprendre on a l'initiation d'utiliser ces objets suivant :



Raspberry  
Pi 2B

Servo  
moteur  
SG90

Raspberry  
Pi 8mp  
Camera  
Module  
v2

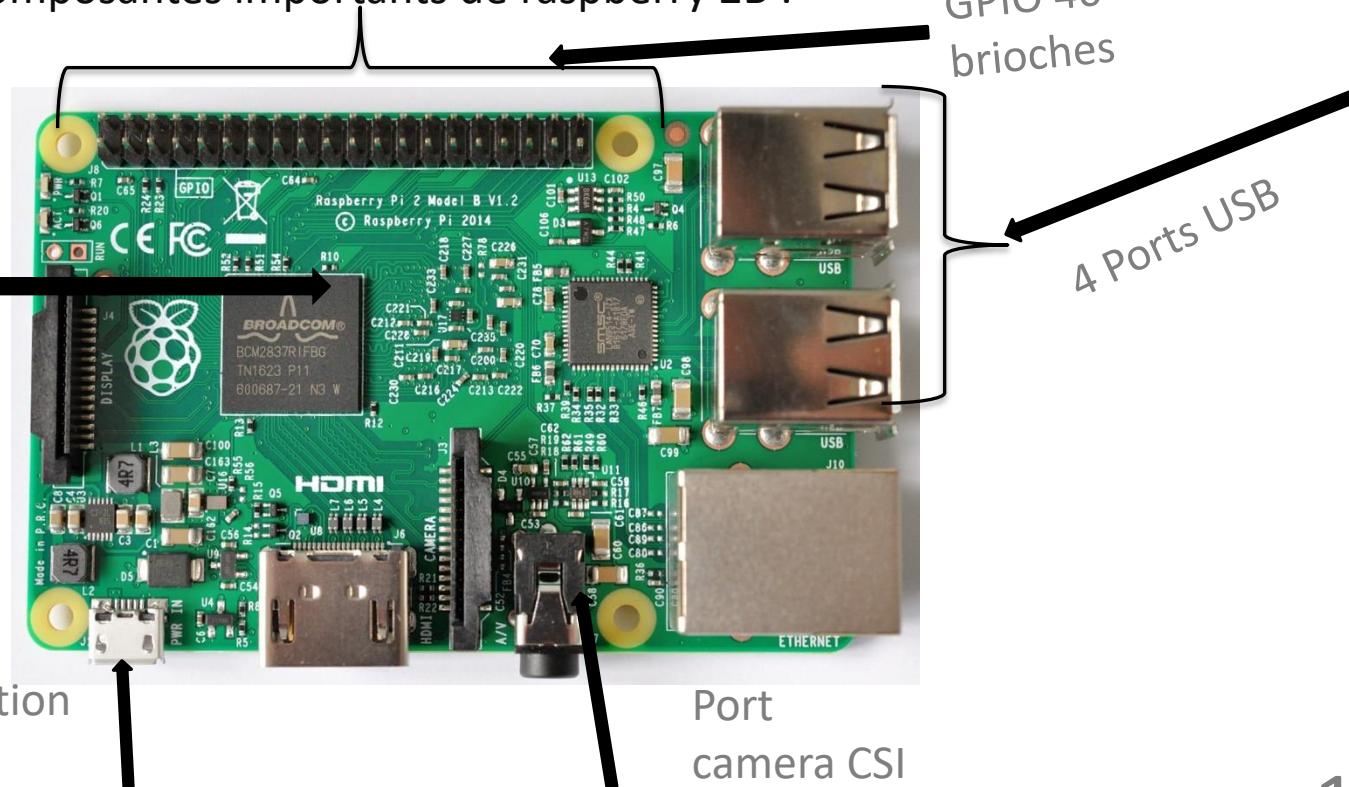
“Pan Tilt”  
Support  
pour les  
servo  
moteur

sim900a  
gsm  
module

## Raspberry Pi 2B :

- C'est un nano-ordinateur qui fonctionne avec un système d'exploitation (Rasbian OS très utiliser) et sert a faire des tests électroniques , écrire des script python , créer des projets électrique et grâce a la mémoire vive(RAM) on pourra faire des projets de l'intelligence artificiel
- Les composantes importants de raspberry 2B :

Processeur  
CPU/GPU



- Pour un bien fonctionnement de la Raspberry pi pour ce projet il faut que :

1. Installer Rasbian OS.

2. Dans la tache Terminal on tape la commande suivante :

```
sudo apt-get update  
sudo apt-get upgrade
```

3. Installer les différentes bibliothèque a utiliser :

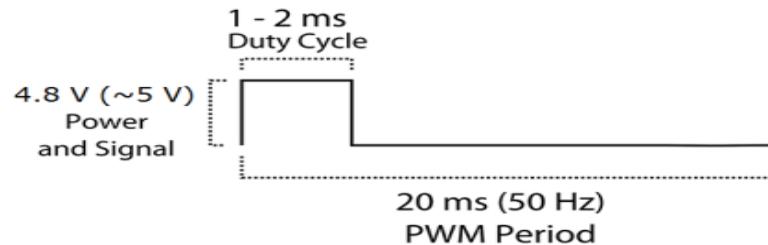
```
sudo apt-get install rpi.gpio  
sudo apt install libopencv-dev opencv-doc  
sudo apt-get install python-picamera  
sudo apt-get install numpy
```

4. Activer la camera dans le systeme : (dans Terminal)

```
sudo raspi-config
```

On déplace dans le menu jusqu'à "camera" et on l'active.

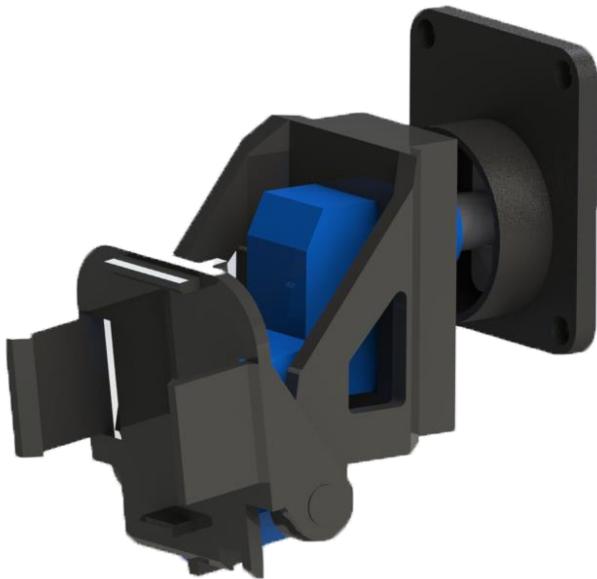
## Servo moteur SG90 :



- D'après cette image on remarque que le PWM signal générer doit être une fréquence de 50Hz , et pour varier la rotation de moteur on donne une valeur de Duty Cycle entre 1ms et 2ms , par exemple : si on veut 90 de rotation on donne une valeur de Duty Cycle 1,5ms.
- Dans ce cas pour bien controller notre servo moteur on utilisera Raspberry pi avec une commandation par python :

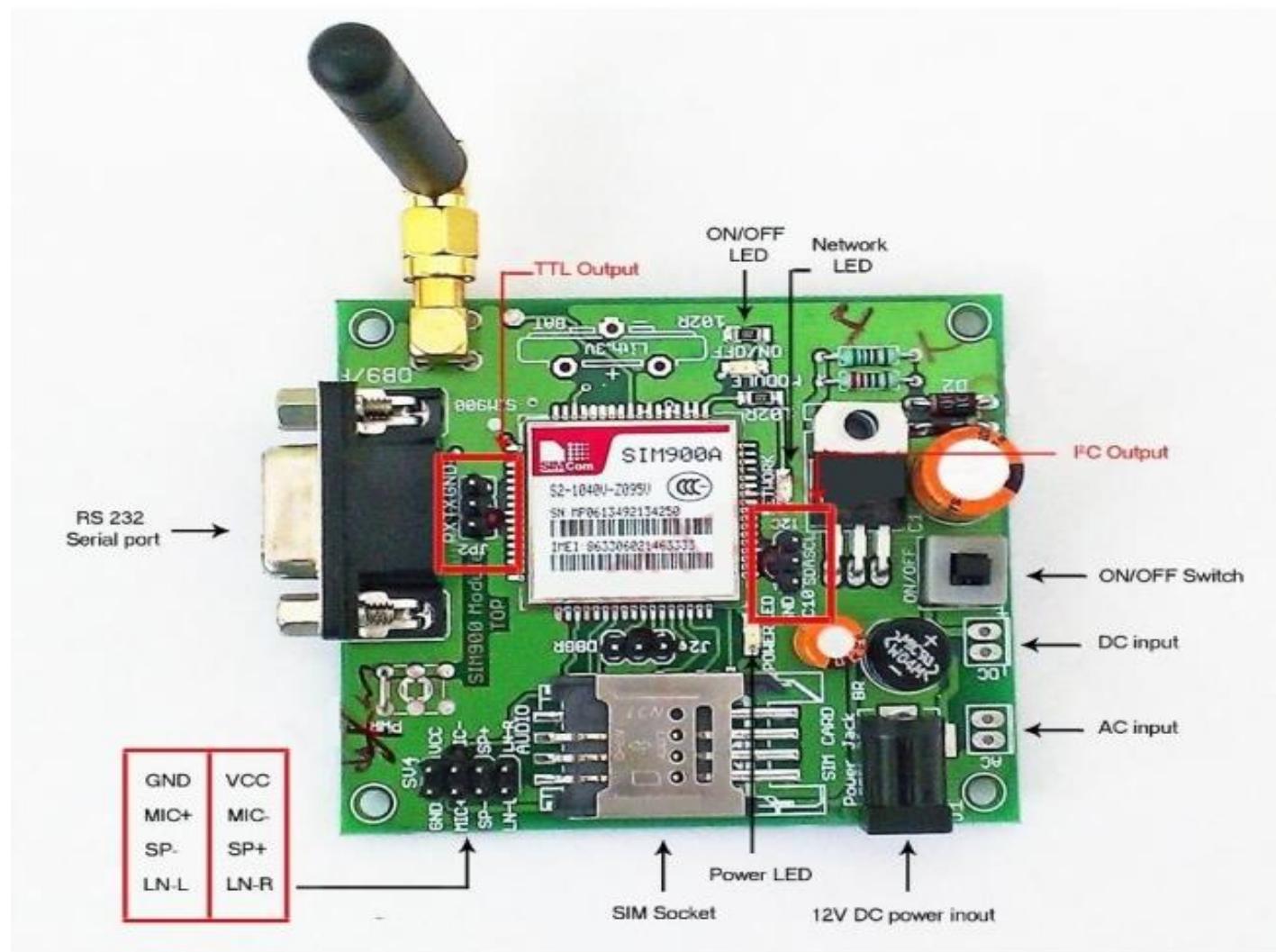
```
- import RPi.GPIO as GPIO
- import time
- servoPIN = PIN #le nombre pin où on a branche le servo dans le raspberry
- GPIO.setmode(GPIO.BCM)
- GPIO.setup(servoPIN, GPIO.OUT)
- p = GPIO.PWM(servoPIN, 50)
- p.start(2.5)
- p.ChangeDutyCycle(5)
```

Le support de la camera de Raspberry pi pour le fonctionnement Pan-Tilt :

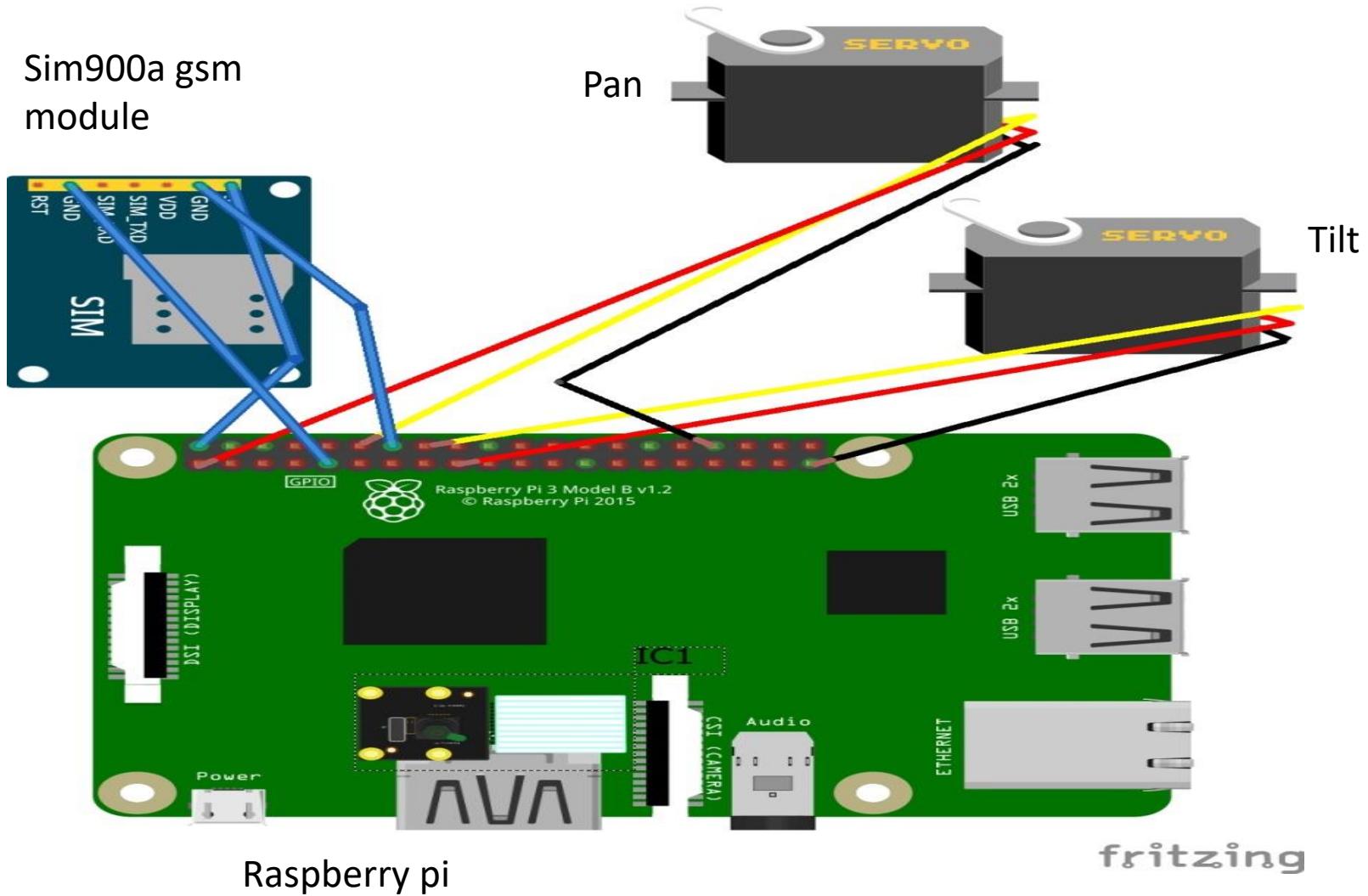


  
**SOLIDWORKS**

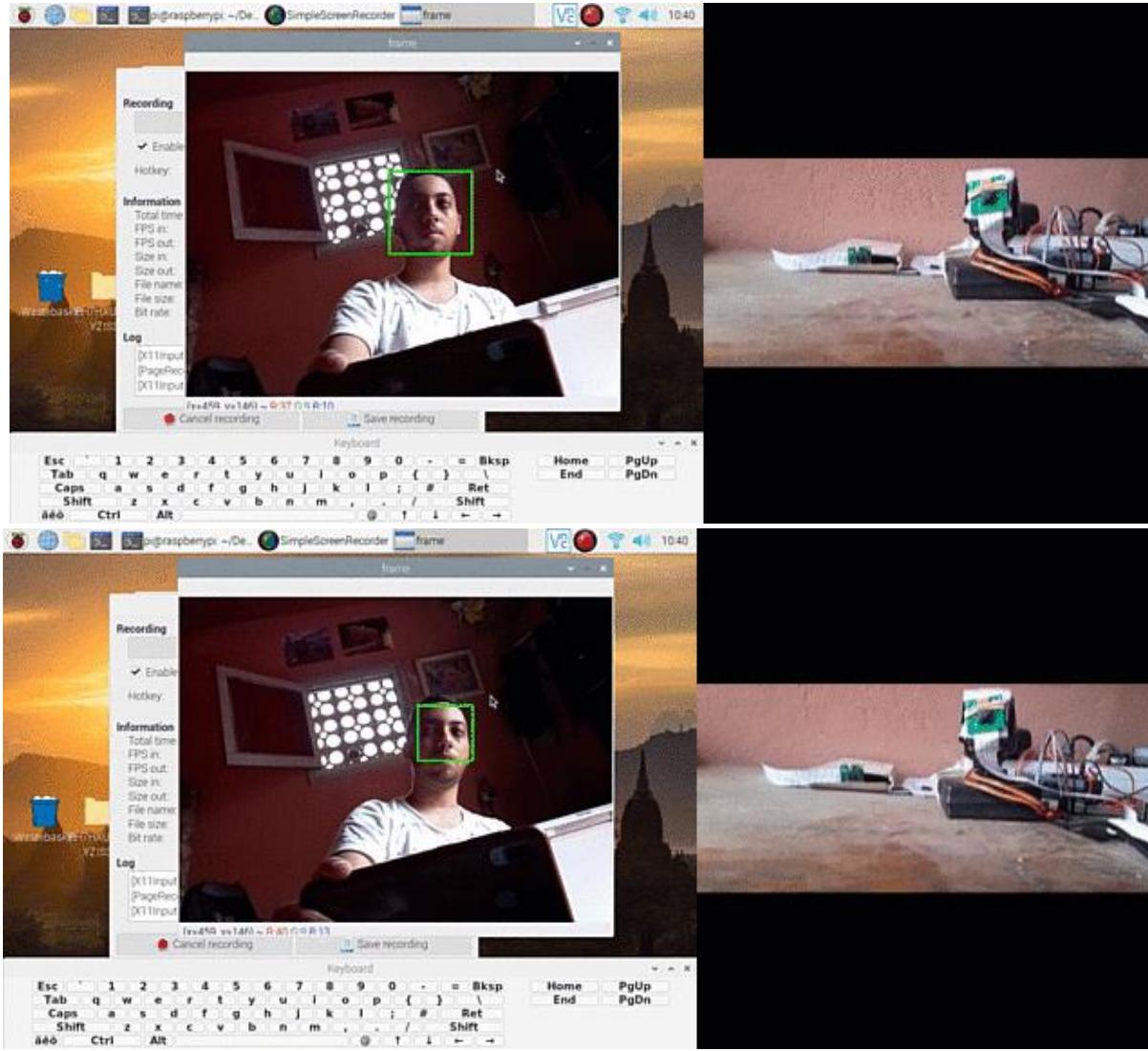
## Sim900a gsm module



Sim900a gsm  
module



# Résultats :



Je vous remercie pour votre  
attention