Aydın Adnan Menderes University

Department of Computer Engineering

Spotter Security Systems

By

Onur Andiç/171805028

Ersu Karpuz/171805043

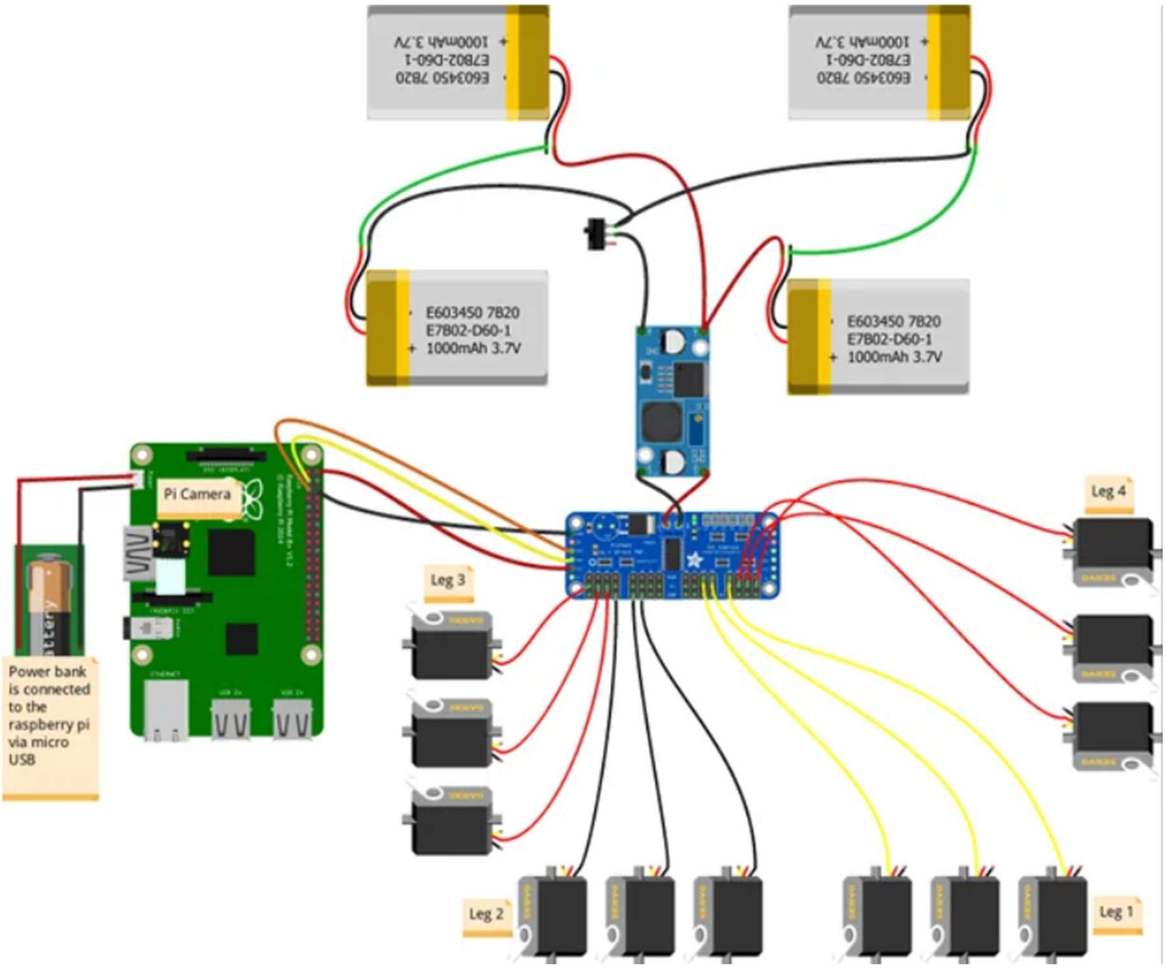
Atakan Üner/171805011

Uygar Utku Öztürk/171805046

Adem Enes Kesici/171805029

Supervisor: Ahmet Çağdaş Seçkin

1 - Sensor Node

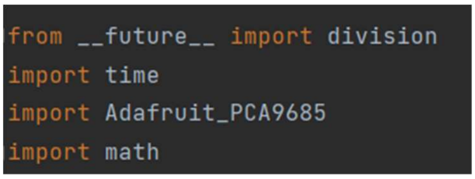


Firstly, we connected wires that we cut half to the output side and positive port of the buck converter with soldering tool. After that we connected the toogle switch. Then we connected batteries to the buck converter. We connected the servo motors to the ports of servo driver. These are the ports that used 0-5's and 10-16's. We connected buck converter to servo driver, positive to positive, negative to negative.Then we connected servo driver to Raspberry Pi 3B+ with wires; VCC pin to 5V or 3.3V pin , GND pin to Ground pin, SDA pin to SDA pin and SCL pin to SCL pin. Then we connected power bank and Pi camera module to Raspberry Pi.

2 - Node Code and Data Acquisition,

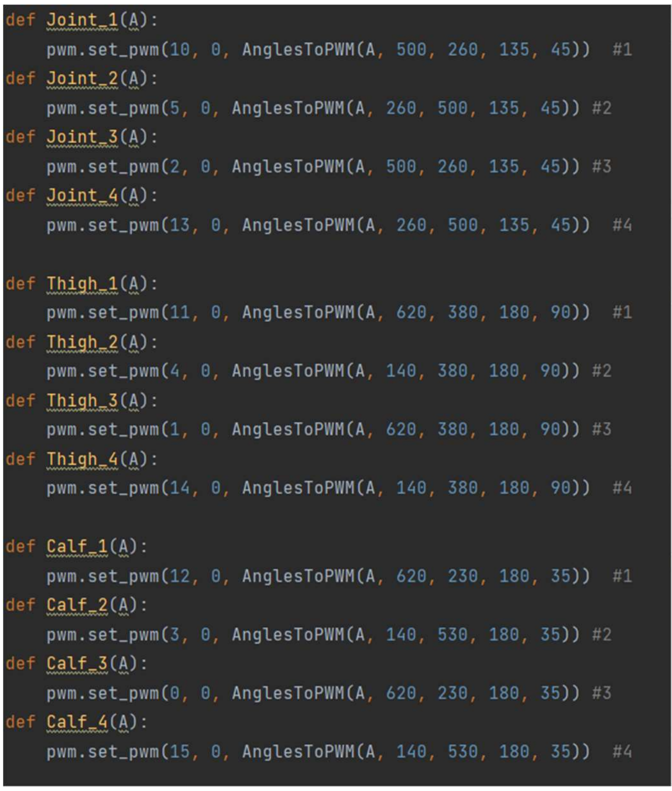
<u>movementlibrary.py;</u>

Importing necessary libraries.

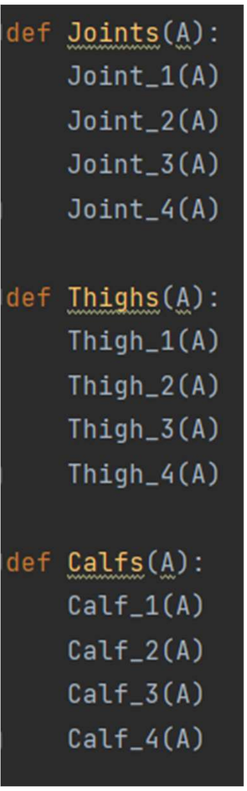


```python
from __future__ import division
import time
import Adafruit_PCA9685
import math
```

Setting up Joints, Thighs, Calfs and their coordinates which connected to servo motor each one.

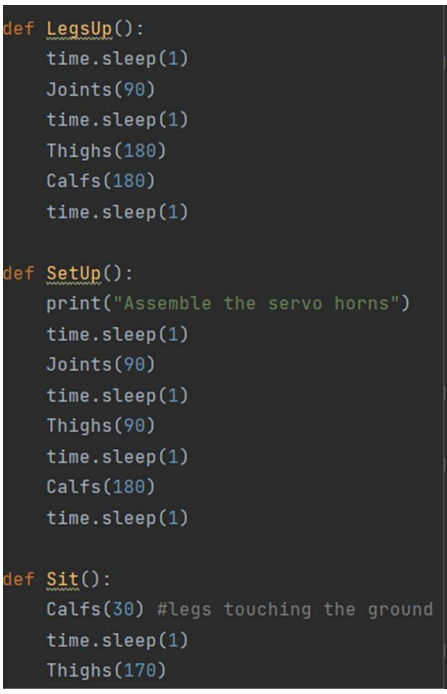


```python
def Joint_1(A):
    pwm.set_pwm(10, 0, AnglesToPWM(A, 500, 260, 135, 45))  #1
def Joint_2(A):
    pwm.set_pwm(5, 0, AnglesToPWM(A, 260, 500, 135, 45)) #2
def Joint_3(A):
    pwm.set_pwm(2, 0, AnglesToPWM(A, 500, 260, 135, 45)) #3
def Joint_4(A):
    pwm.set_pwm(13, 0, AnglesToPWM(A, 260, 500, 135, 45))  #4

def Thigh_1(A):
    pwm.set_pwm(11, 0, AnglesToPWM(A, 620, 380, 180, 90))  #1
def Thigh_2(A):
    pwm.set_pwm(4, 0, AnglesToPWM(A, 140, 380, 180, 90)) #2
def Thigh_3(A):
    pwm.set_pwm(1, 0, AnglesToPWM(A, 620, 380, 180, 90)) #3
def Thigh_4(A):
    pwm.set_pwm(14, 0, AnglesToPWM(A, 140, 380, 180, 90))  #4

def Calf_1(A):
    pwm.set_pwm(12, 0, AnglesToPWM(A, 620, 230, 180, 35))  #1
def Calf_2(A):
    pwm.set_pwm(3, 0, AnglesToPWM(A, 140, 530, 180, 35)) #2
def Calf_3(A):
    pwm.set_pwm(0, 0, AnglesToPWM(A, 620, 230, 180, 35)) #3
def Calf_4(A):
    pwm.set_pwm(15, 0, AnglesToPWM(A, 140, 530, 180, 35))  #4
```

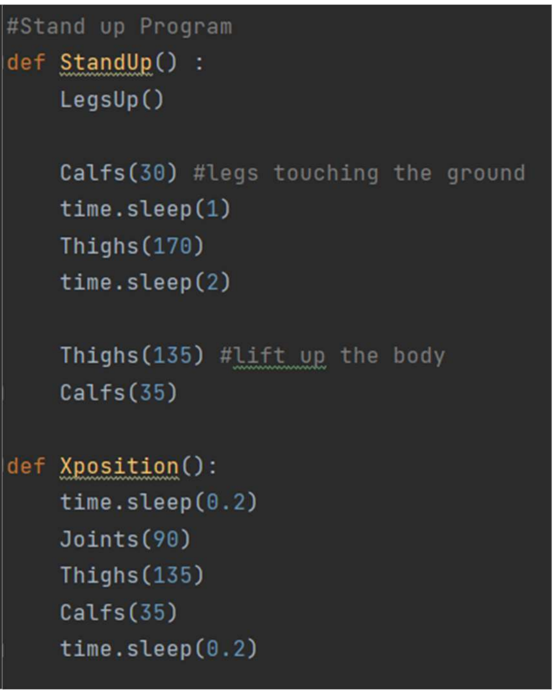


```python
def Joints(A):
    Joint_1(A)
    Joint_2(A)
    Joint_3(A)
    Joint_4(A)

def Thighs(A):
    Thigh_1(A)
    Thigh_2(A)
    Thigh_3(A)
    Thigh_4(A)

def Calfs(A):
    Calf_1(A)
    Calf_2(A)
    Calf_3(A)
    Calf_4(A)
```

Some movements that robot can do. Each movement transmits which servo motor will move at what angle via the servo driver.

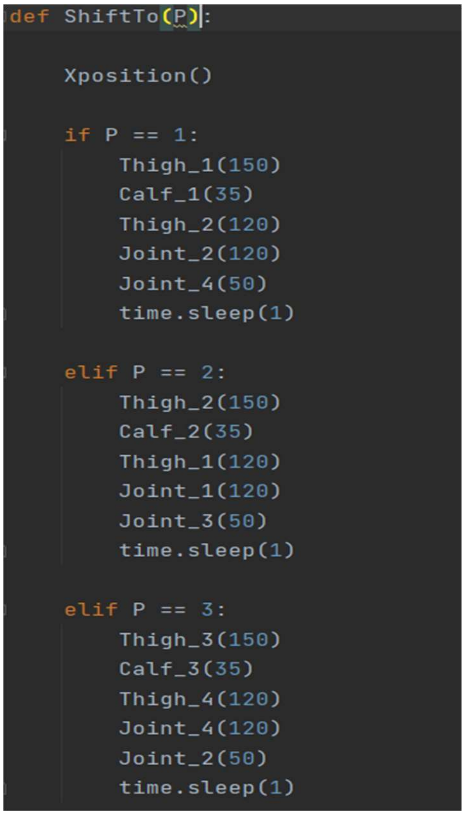


```
def LegsUp():
    time.sleep(1)
    Joints(90)
    time.sleep(1)
    Thighs(180)
    Calfs(180)
    time.sleep(1)

def SetUp():
    print("Assemble the servo horns")
    time.sleep(1)
    Joints(90)
    time.sleep(1)
    Thighs(90)
    time.sleep(1)
    Calfs(180)
    time.sleep(1)

def Sit():
    Calfs(30) #legs touching the ground
    time.sleep(1)
    Thighs(170)
```



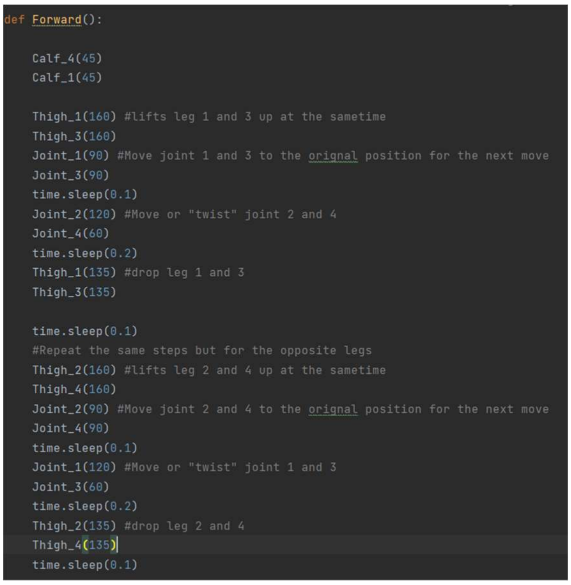
```
#Stand up Program
def StandUp() :
    LegsUp()

    Calfs(30) #legs touching the ground
    time.sleep(1)
    Thighs(170)
    time.sleep(2)

    Thighs(135) #lift up the body
    Calfs(35)

def Xposition():
    time.sleep(0.2)
    Joints(90)
    Thighs(135)
    Calfs(35)
    time.sleep(0.2)
```

Indicates in which position the servo motors can stay.



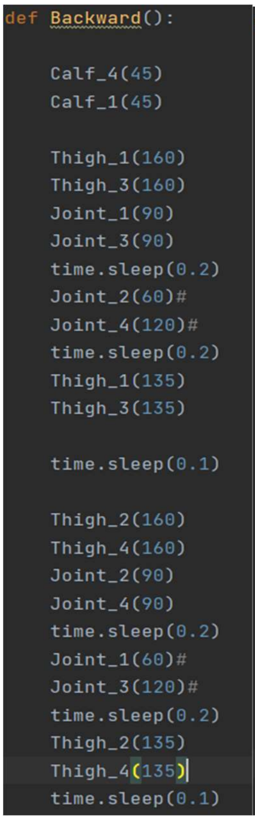
```
def ShiftTo(P):

    Xposition()

    if P == 1:
        Thigh_1(150)
        Calf_1(35)
        Thigh_2(120)
        Joint_2(120)
        Joint_4(50)
        time.sleep(1)

    elif P == 2:
        Thigh_2(150)
        Calf_2(35)
        Thigh_1(120)
        Joint_1(120)
        Joint_3(50)
        time.sleep(1)

    elif P == 3:
        Thigh_3(150)
        Calf_3(35)
        Thigh_4(120)
        Joint_4(120)
        Joint_2(50)
        time.sleep(1)
```

Going Forward code.

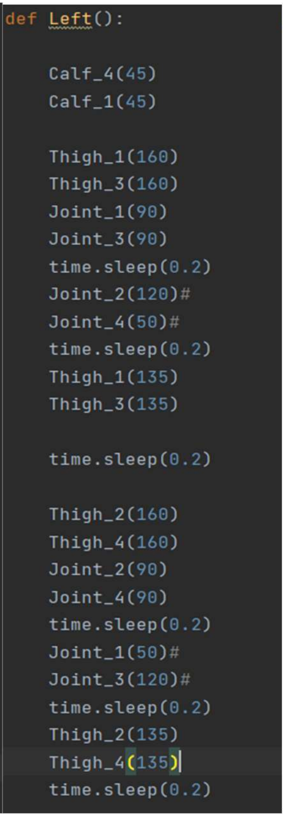


```
def Forward():

    Calf_4(45)
    Calf_1(45)

    Thigh_1(160) #lifts leg 1 and 3 up at the sametime
    Thigh_3(160)
    Joint_1(90) #Move joint 1 and 3 to the orignal position for the next move
    Joint_3(90)
    time.sleep(0.1)
    Joint_2(120) #Move or "twist" joint 2 and 4
    Joint_4(60)
    time.sleep(0.2)
    Thigh_1(135) #drop leg 1 and 3
    Thigh_3(135)

    time.sleep(0.1)
    #Repeat the same steps but for the opposite legs
    Thigh_2(160) #lifts leg 2 and 4 up at the sametime
    Thigh_4(160)
    Joint_2(90) #Move joint 2 and 4 to the orignal position for the next move
    Joint_4(90)
    time.sleep(0.1)
    Joint_1(120) #Move or "twist" joint 1 and 3
    Joint_3(60)
    time.sleep(0.2)
    Thigh_2(135) #drop leg 2 and 4
    Thigh_4(135)
    time.sleep(0.1)
```

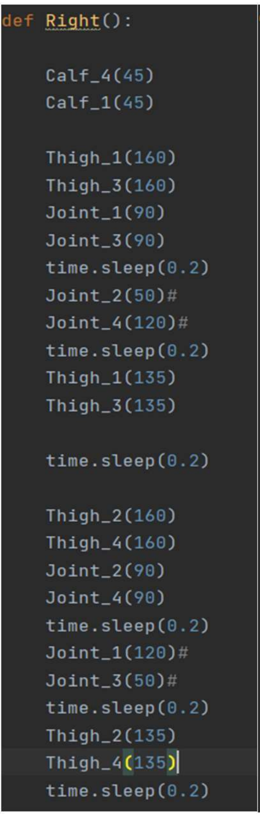
Going Backward, Left, Right, Counterclockwise and clockwise.



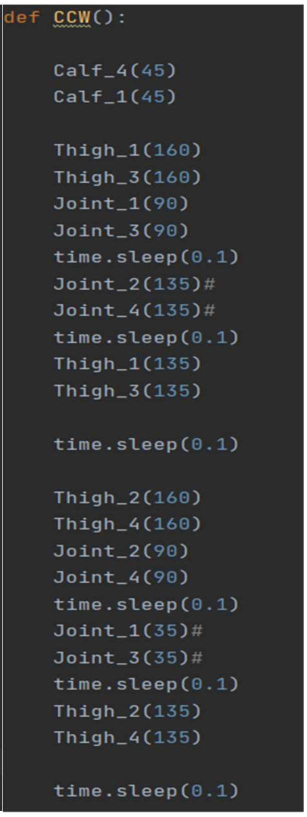
```
def Backward():

    Calf_4(45)
    Calf_1(45)

    Thigh_1(160)
    Thigh_3(160)
    Joint_1(90)
    Joint_3(90)
    time.sleep(0.2)
    Joint_2(60)#
    Joint_4(120)#
    time.sleep(0.2)
    Thigh_1(135)
    Thigh_3(135)

    time.sleep(0.1)

    Thigh_2(160)
    Thigh_4(160)
    Joint_2(90)
    Joint_4(90)
    time.sleep(0.2)
    Joint_1(60)#
    Joint_3(120)#
    time.sleep(0.2)
    Thigh_2(135)
    Thigh_4(135)
    time.sleep(0.1)
```



```
def Left():

    Calf_4(45)
    Calf_1(45)

    Thigh_1(160)
    Thigh_3(160)
    Joint_1(90)
    Joint_3(90)
    time.sleep(0.2)
    Joint_2(120)#
    Joint_4(50)#
    time.sleep(0.2)
    Thigh_1(135)
    Thigh_3(135)

    time.sleep(0.2)

    Thigh_2(160)
    Thigh_4(160)
    Joint_2(90)
    Joint_4(90)
    time.sleep(0.2)
    Joint_1(50)#
    Joint_3(120)#
    time.sleep(0.2)
    Thigh_2(135)
    Thigh_4(135)
    time.sleep(0.2)
```



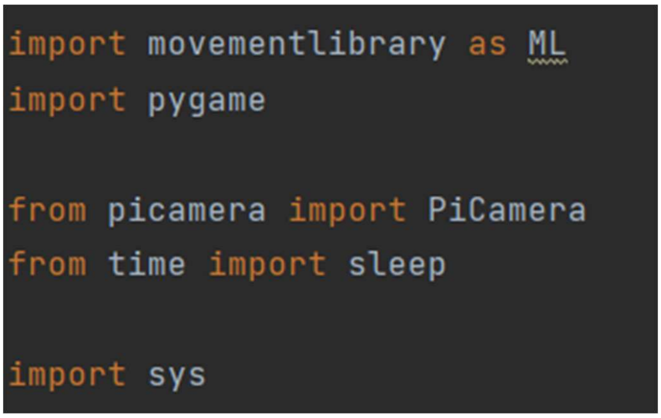
```
def Right():

    Calf_4(45)
    Calf_1(45)

    Thigh_1(160)
    Thigh_3(160)
    Joint_1(90)
    Joint_3(90)
    time.sleep(0.2)
    Joint_2(50)#
    Joint_4(120)#
    time.sleep(0.2)
    Thigh_1(135)
    Thigh_3(135)

    time.sleep(0.2)

    Thigh_2(160)
    Thigh_4(160)
    Joint_2(90)
    Joint_4(90)
    time.sleep(0.2)
    Joint_1(120)#
    Joint_3(50)#
    time.sleep(0.2)
    Thigh_2(135)
    Thigh_4(135)
    time.sleep(0.2)
```



```
def CCW():

    Calf_4(45)
    Calf_1(45)

    Thigh_1(160)
    Thigh_3(160)
    Joint_1(90)
    Joint_3(90)
    time.sleep(0.1)
    Joint_2(135)#
    Joint_4(135)#
    time.sleep(0.1)
    Thigh_1(135)
    Thigh_3(135)

    time.sleep(0.1)

    Thigh_2(160)
    Thigh_4(160)
    Joint_2(90)
    Joint_4(90)
    time.sleep(0.1)
    Joint_1(35)#
    Joint_3(35)#
    time.sleep(0.1)
    Thigh_2(135)
    Thigh_4(135)

    time.sleep(0.1)
```

Robot+Control+Program.py:

Imports that we used in code.

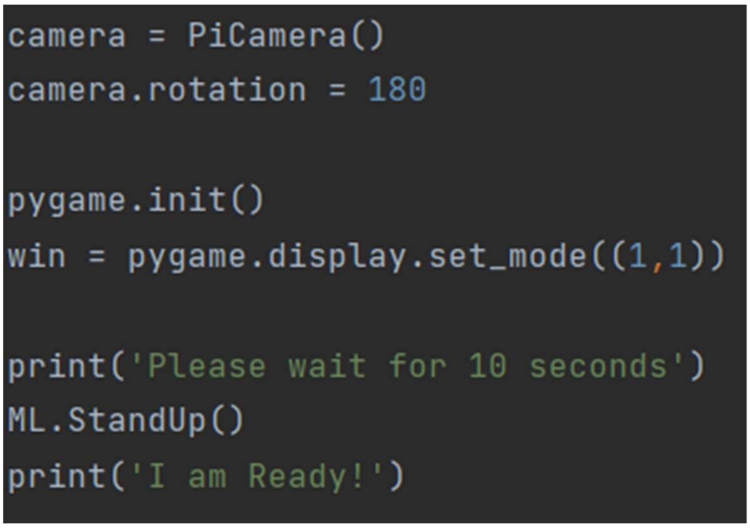


```
import movementlibrary as ML
import pygame

from picamera import PiCamera
from time import sleep

import sys
```

Connection to raspberry camera module and setting up its rotation.

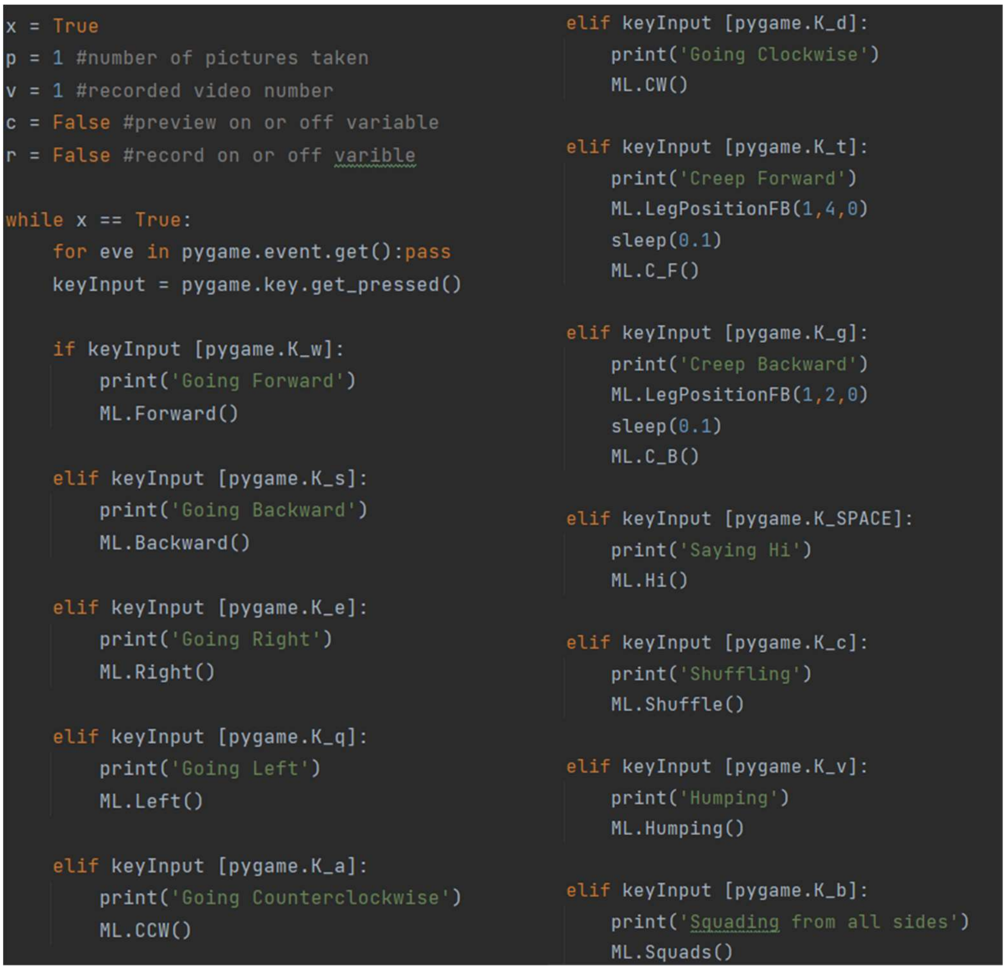
We used pygame to show preview on display.

First movement of the robot is the Stand up.



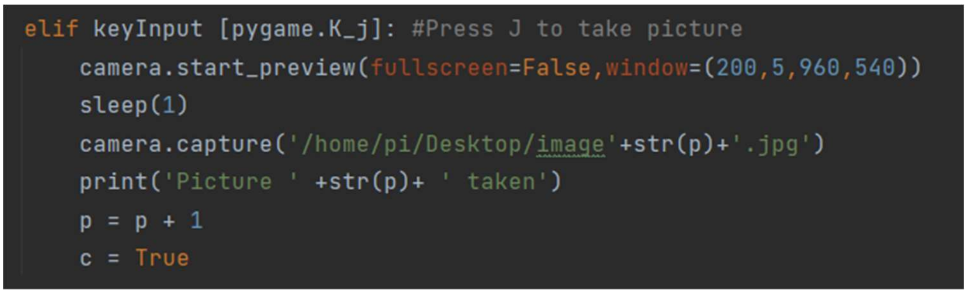
```
camera = PiCamera()
camera.rotation = 180

pygame.init()
win = pygame.display.set_mode((1,1))

print('Please wait for 10 seconds')
ML.StandUp()
print('I am Ready!')
```

To use movement functions we made key assignments.



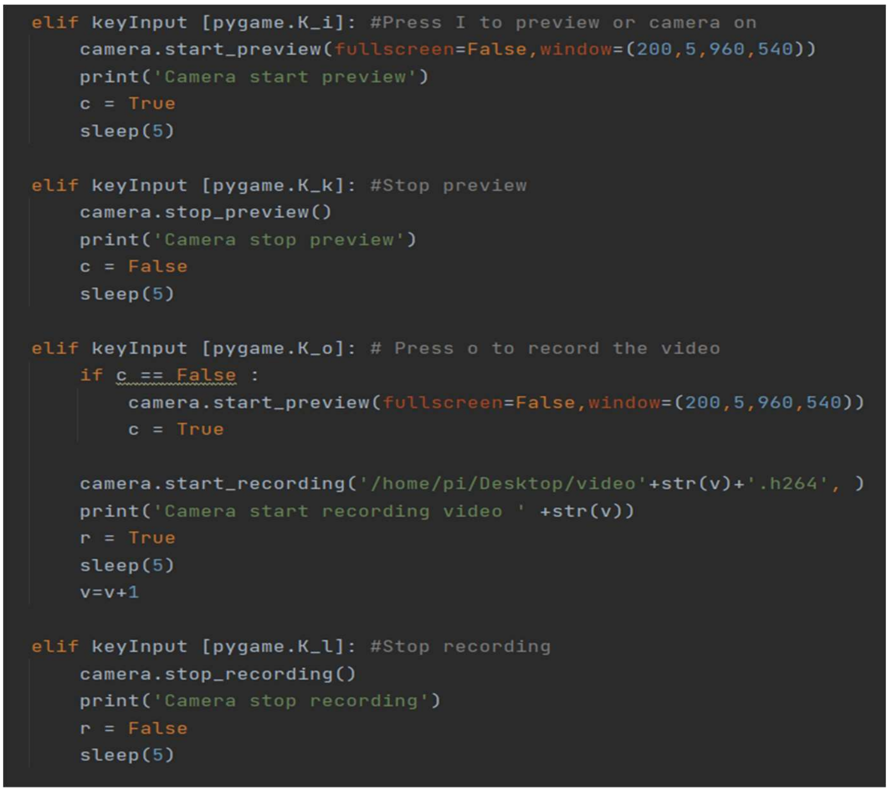
```
x = True
p = 1 #number of pictures taken
v = 1 #recorded video number
c = False #preview on or off variable
r = False #record on or off varible

while x == True:
    for eve in pygame.event.get():pass
    keyInput = pygame.key.get_pressed()

    if keyInput [pygame.K_w]:
        print('Going Forward')
        ML.Forward()

    elif keyInput [pygame.K_s]:
        print('Going Backward')
        ML.Backward()

    elif keyInput [pygame.K_e]:
        print('Going Right')
        ML.Right()

    elif keyInput [pygame.K_q]:
        print('Going Left')
        ML.Left()

    elif keyInput [pygame.K_a]:
        print('Going Counterclockwise')
        ML.CCW()
    elif keyInput [pygame.K_d]:
        print('Going Clockwise')
        ML.CW()

    elif keyInput [pygame.K_t]:
        print('Creep Forward')
        ML.LegPositionFB(1,4,0)
        sleep(0.1)
        ML.C_F()

    elif keyInput [pygame.K_g]:
        print('Creep Backward')
        ML.LegPositionFB(1,2,0)
        sleep(0.1)
        ML.C_B()

    elif keyInput [pygame.K_SPACE]:
        print('Saying Hi')
        ML.Hi()

    elif keyInput [pygame.K_c]:
        print('Shuffling')
        ML.Shuffle()

    elif keyInput [pygame.K_v]:
        print('Humping')
        ML.Humping()

    elif keyInput [pygame.K_b]:
        print('Squading from all sides')
        ML.Squads()
```

We can taking picture with pressing J. When we press J ,the code give an order to camera module to take a picture. ‘p’ variable in this code is the number of the pictures.



```
elif keyInput [pygame.K_j]: #Press J to take picture
    camera.start_preview(fullscreen=False,window=(200,5,960,540))
    sleep(1)
    camera.capture('/home/pi/Desktop/image'+str(p)+'.jpg')
    print('Picture ' +str(p)+ ' taken')
    p = p + 1
    c = True
```

With this code we can see from the camera and record the video.

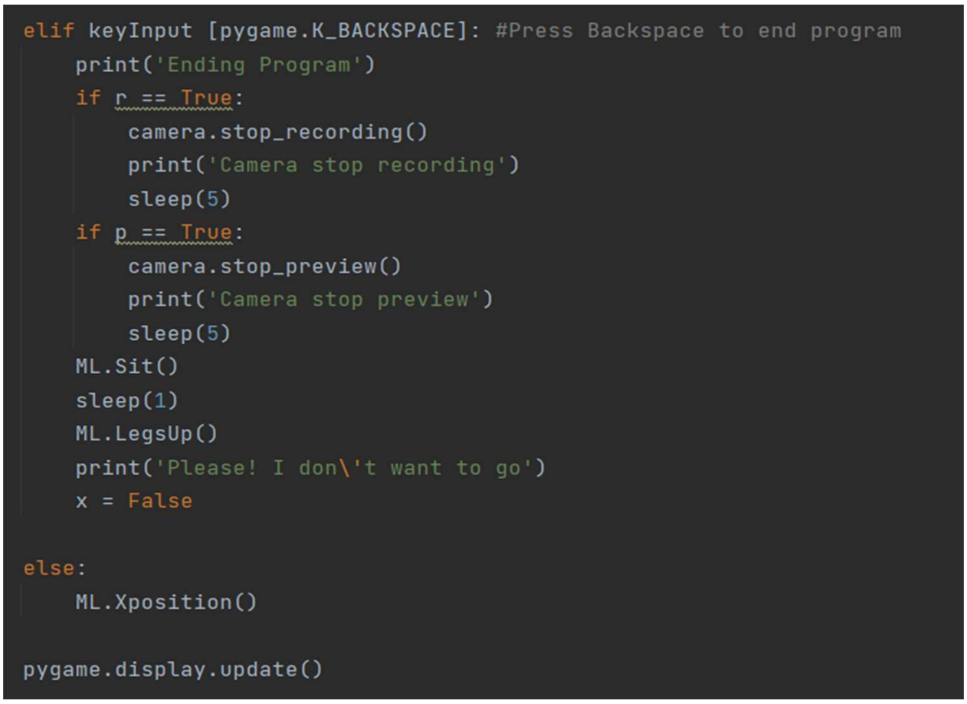


```
elif keyInput [pygame.K_i]: #Press I to preview or camera on
    camera.start_preview(fullscreen=False,window=(200,5,960,540))
    print('Camera start preview')
    c = True
    sleep(5)

elif keyInput [pygame.K_k]: #Stop preview
    camera.stop_preview()
    print('Camera stop preview')
    c = False
    sleep(5)

elif keyInput [pygame.K_o]: # Press o to record the video
    if c == False :
        camera.start_preview(fullscreen=False,window=(200,5,960,540))
        c = True

    camera.start_recording('/home/pi/Desktop/video'+str(v)+'.h264', )
    print('Camera start recording video ' +str(v))
    r = True
    sleep(5)
    v=v+1

elif keyInput [pygame.K_l]: #Stop recording
    camera.stop_recording()
    print('Camera stop recording')
    r = False
    sleep(5)
```

We can close the camera then end the program with this code block.

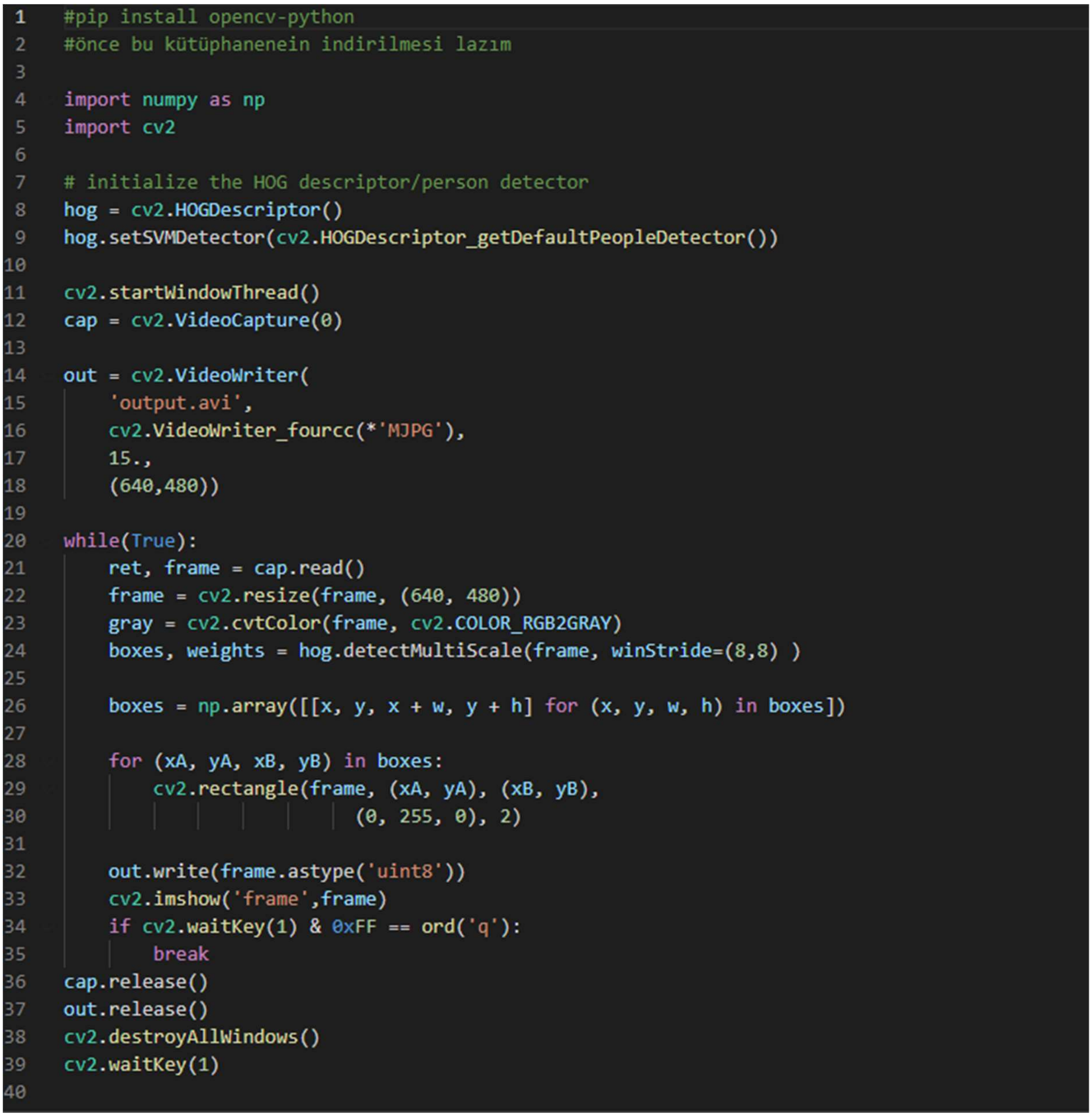
All key events prints in console what is going on currently

.



```
elif keyInput [pygame.K_BACKSPACE]: #Press Backspace to end program
    print('Ending Program')
    if r == True:
        camera.stop_recording()
        print('Camera stop recording')
        sleep(5)
    if p == True:
        camera.stop_preview()
        print('Camera stop preview')
        sleep(5)
    ML.Sit()
    sleep(1)
    ML.LegsUp()
    print('Please! I don\'t want to go')
    x = False

else:
    ML.Xposition()

pygame.display.update()
```

HumanBodyDedec.py:

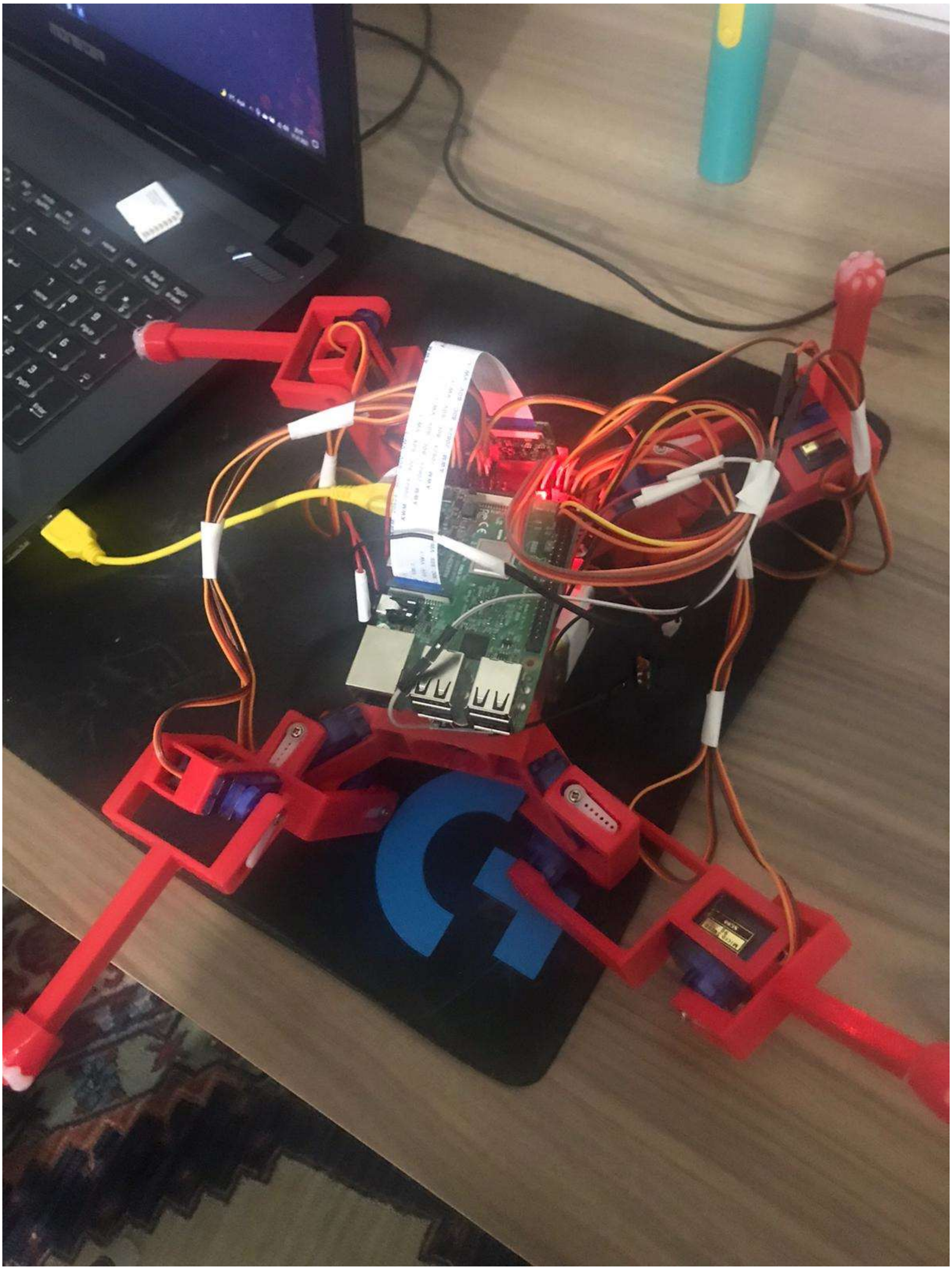


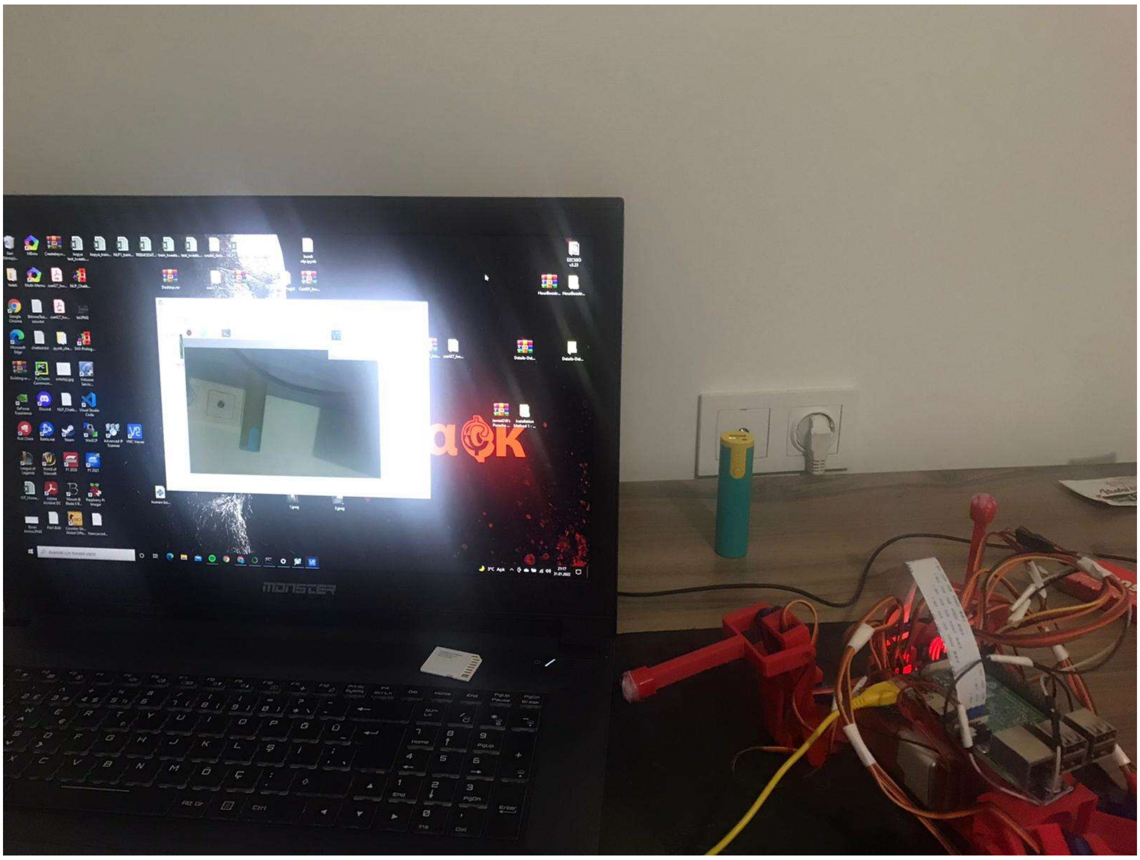
```python
#pip install opencv-python
#önce bu kütüphanenein indirilmesi lazım

import numpy as np
import cv2

# initialize the HOG descriptor/person detector
hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())

cv2.startWindowThread()
cap = cv2.VideoCapture(0)

out = cv2.VideoWriter(
    'output.avi',
    cv2.VideoWriter_fourcc(*'MJPG'),
    15.,
    (640,480))

while(True):
    ret, frame = cap.read()
    frame = cv2.resize(frame, (640, 480))
    gray = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)
    boxes, weights = hog.detectMultiScale(frame, winStride=(8,8) )

    boxes = np.array([[x, y, x + w, y + h] for (x, y, w, h) in boxes])

    for (xA, yA, xB, yB) in boxes:
        cv2.rectangle(frame, (xA, yA), (xB, yB),
                          (0, 255, 0), 2)

    out.write(frame.astype('uint8'))
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
out.release()
cv2.destroyAllWindows()
cv2.waitKey(1)
```

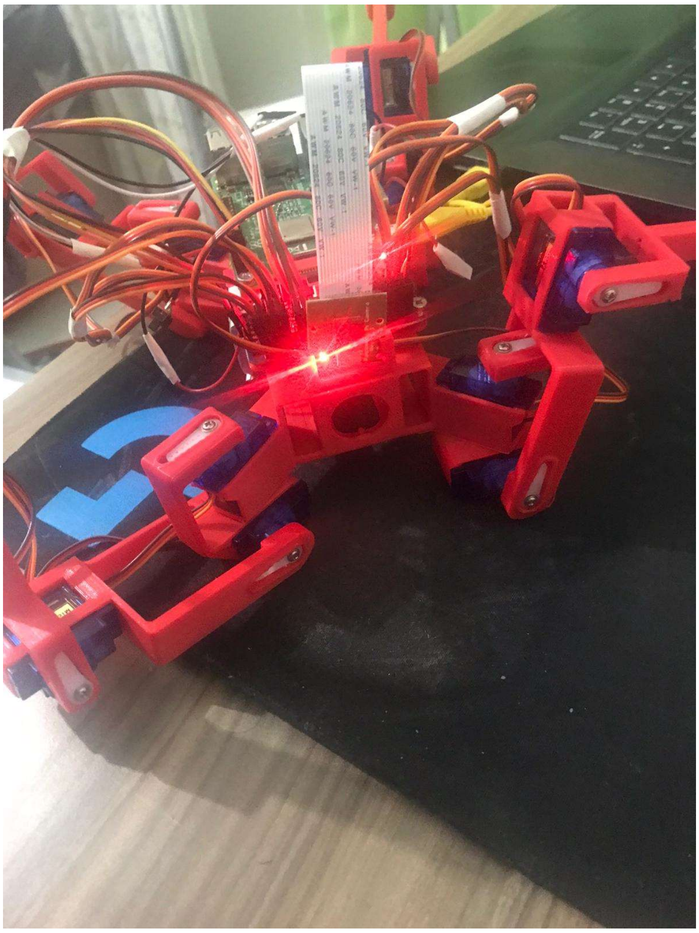
Firstly,We must install opencv libarary and numpy library.

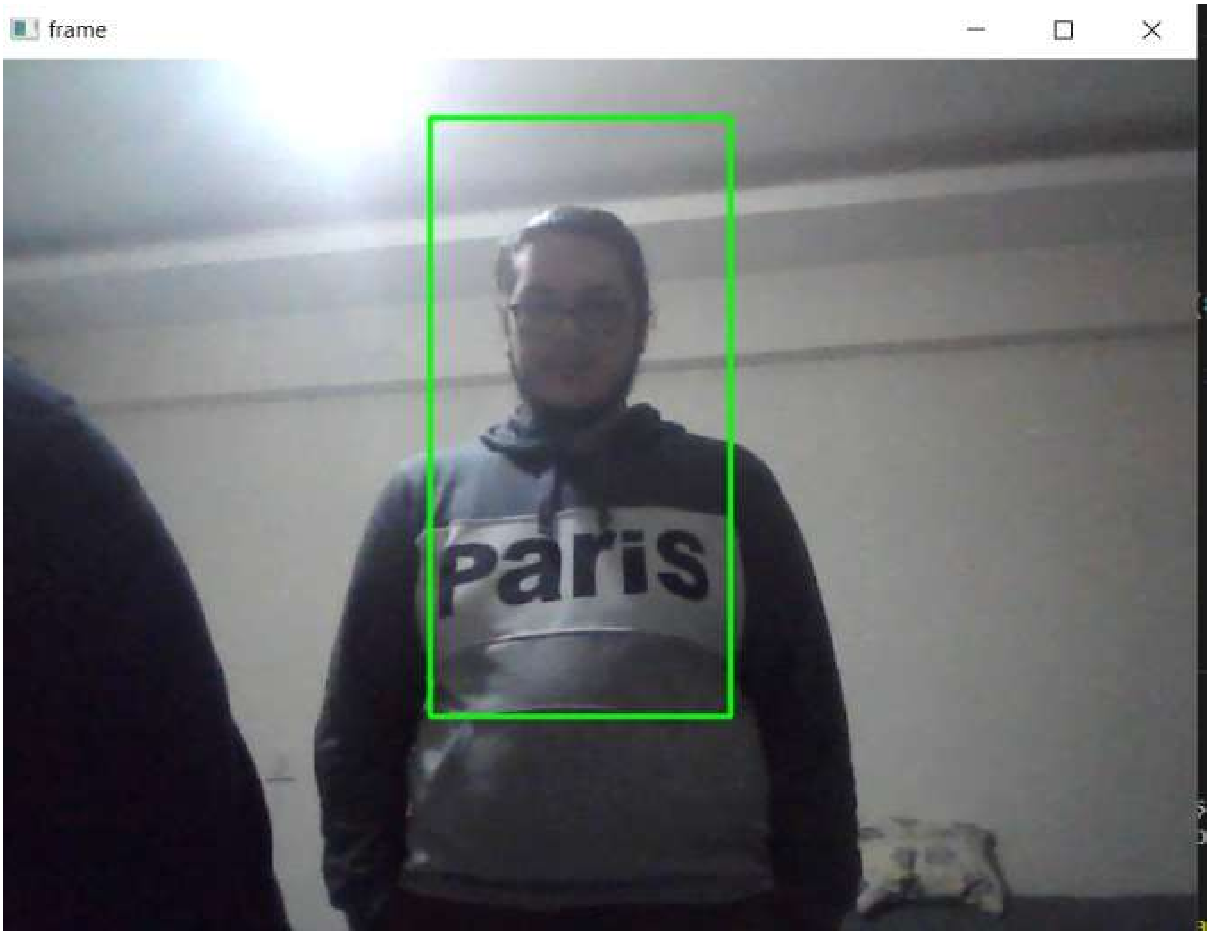
After that, This code allows us to detect people and enclose them in a rectangle.

Data Acquisition:








frame
Paris

3-)

We need to connect the installed raspberry to Wifi first. The thing to note is that Raspberry only works on 2.4g. So 5g can't connect to Wifi.

We create 1 config file to connect to Raspberry. Paste the code snippet below into the file.

ssid is our wifi name psk is our wifi password.

```
network={
  ssid="Test Wifi Network"
  psk="SecretPassword"
}
```

Computing:

After connecting the Raspberry to wifi, we reach the Raspberry IP address with the Advanced

IP Scanner.Our Raspberry IP address is 192.168.1.52.

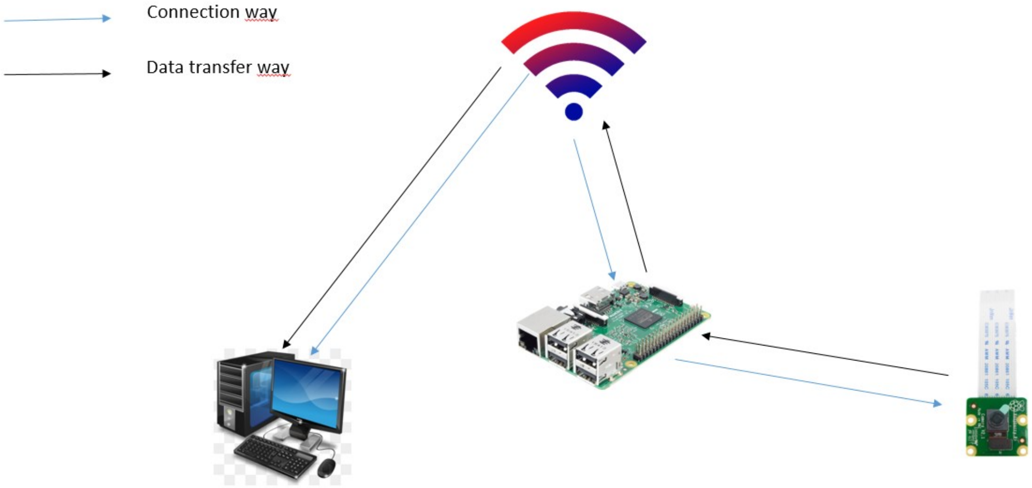
After that, We save the IP address we get from Advanced IP Scanner to VNC Viewer application. This is how we connect to our raspberry.

With WinSCP, we throw the codes we write on our computer into our Raspberry.

In this way, we can run the codes we throw on our Raspberry.

Thanks to the codes, we can control our robot remotely, and thanks to the camera on the robot, we can instantly save the photos of the images we view with a single click.

Our program runs as edge computing.



Service:

The sector served by our robot serves the security sector.