```
‘’’
Luis Alejandro
‘’’
#import all the libraries that we need
#1. tkinter is used to create the GUI
#2. Pillow/PIL is used to manage the video data and image
#3. openCV/cv2 is used to identify objects, faces and others
#4. imutils are a series of convinient functions to make basic image processing finctions like rotation, rezising images
#5. GPIO activate the inputs and outputs of the RPi

from tkinter import *
from tkinter import filedialog
from PIL import Image
from PIL import ImageTk
import cv2
import imutils
import time
import RPi.GPIO as GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)#relay VFD 240VAC
GPIO.setup(18, GPIO.OUT)#number of steps the steppermotor
GPIO.setup(23, GPIO.OUT)#Enavel or desable the drive // it prevent the motor from overheating
GPIO.setup(24, GPIO.OUT)#Direction of the stepperMotor

#function to setup the video
def visualizar():
    if cap is not None:
```

```
        ret, frame = cap.read()
        if ret == True:
            frame = imutils.resize(frame,width=640)
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

            im = Image.fromarray(frame)
            img = ImageTk.PhotoImage(image=im)

            lblVideo.configure(image=img)
            lblVideo.image = img
            lblVideo.after(10, visualizar)
    else:
        lblVideo.image = ""
        cap.release()

#funtion link to the btnIniciar
#start the camera by calling visualizar()
def iniciar():
    global cap
    cap = cv2.VideoCapture(0)
    visualizar()

#funtion link to the btnFinalizar
#Stop the camera
def finalizar():
    global cap
    cap.release()

cap = None
```

```
#function to activate contactor VFD 240v
vfdON = False
def vfdON():
    # change the state of the contactor. if it is On turn Off and viseversa
    global vfdON
    vfdON = not vfdON
    GPIO.output(17, GPIO.HIGH if vfdON else GPIO.LOW)

    if (17, GPIO.HIGH):
        buttonVFD["bg"]="green"
    else:
        buttonVFD["bg"]="red"

#create funtion to rotate to the camera to the left
def left():

    GPIO.output(24, GPIO.HIGH)#input to the stepper drive tocontrol direction
    GPIO.output(23, GPIO.LOW)#enable stepper drive

    for step in range (200) :
        GPIO.output(18, GPIO.HIGH)
        time.sleep(0.001)
        GPIO.output(18, GPIO.LOW)
        time.sleep(0.001)
    GPIO.output(23, GPIO.HIGH)#disable stepper drive
```

```
#create funtion to rotate to the camera to the right
def right():

    GPIO.output(24, GPIO.LOW)# change direction in stepper drive
    GPIO.output(23, GPIO.LOW)#enable stepper drive

    for step in range (200) :
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(18, GPIO.OUT)
        GPIO.output(18, GPIO.HIGH)
        time.sleep(0.001)
        GPIO.output(18, GPIO.LOW)
        time.sleep(0.001)
    GPIO.output(23, GPIO.HIGH)#disable stepper drive



#Start the tkinter part
root = Tk()

#Window name
root.title("Troubleshooting")

#create button iniciar
btnIniciar = Button(root, text="Start", width = 45, command = iniciar)
btnIniciar.grid(column=0, row=0, pady=5)
```

```
#create button finalizar
btnFinalizar = Button(root, text="Finish", width = 45, command = finalizar)
btnFinalizar.grid(column=1, row=0, pady=5)

#create a space for the camera in column0-row1
lblVideo = Label(root)
lblVideo.grid(column=0, row=1, columnspan=2)

#create button to rotate the camera to the left
btnTLeft = Button(root, text="<<", width=25, command=left)
btnTLeft.grid(column=0, row=10)

#create button to rotate the camera to the right
btnTRight = Button(root, text='>>', width=25,command=right)
btnTRight.grid(column=1, row=10)

#Create a control for the vfd contactor
buttonVFD = Button(root, text= "VFD Start/Stop", width=25 , bg="red", command=vfdON)
buttonVFD.grid(column=0, row=11)




root.mainloop()
```