import time

import RPi.GPIO as GPIO

import serial

from gpiozero import Servo, AngularServo

max\_distance = 40000

distance\_threshold = 0

#

ECHO\_PIN = 5

TRIGGER\_PIN = 6

#

# IR SENSOR PINS

IR\_SENSOR\_RIGHT = 26

IR\_SENSOR\_LEFT = 19

# SPEEDS

TURN\_SPEED = 100

TURN\_SPEED1 = 100

# MOTOR DRIVER PINS

ENABLE\_RIGHT\_MOTOR = 4

RIGHT\_MOTOR\_PIN1 = 27

RIGHT\_MOTOR\_PIN2 = 22

ENABLE\_LEFT\_MOTOR = 25

LEFT\_MOTOR\_PIN1 = 24

LEFT\_MOTOR\_PIN2 = 23

RIGHT\_IR\_SENSOR\_VALUE = 0

LEFT\_IR\_SENSOR\_VALUE = 0

ARM\_BASE = 11

ARM\_UPPER = 17

GRIPPER\_ROTATE = 18

GRIPPER\_GRAB = 20

LOW = 0

HIGH = 1

STORE\_NUMBER = 1

DIRECTION = {

"FORWARD": 1,

"REVERSE": 0

}

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

#GPIO.setup(ARM\_BASE, GPIO.OUT)

#GPIO.setup(ARM\_UPPER, GPIO.OUT)

#GPIO.setup(GRIPPER\_ROTATE, GPIO.OUT)

#GPIO.setup(GRIPPER\_GRAB, GPIO.OUT)

#servo\_arm\_base = AngularServo(ARM\_BASE, min\_pulse\_width=0.0006, max\_pulse\_width=0.0023)

#servo\_arm\_upper = AngularServo(ARM\_UPPER, min\_pulse\_width=0.0006, max\_pulse\_width=0.0023)

#servo\_gripper\_rotate = AngularServo(GRIPPER\_ROTATE, min\_pulse\_width=0.0006, max\_pulse\_width=0.0023)

#servo\_gripper\_grip = AngularServo(GRIPPER\_GRAB, min\_pulse\_width=0.0006, max\_pulse\_width=0.0023)

GPIO.setup(ECHO\_PIN,GPIO.IN)

GPIO.setup(TRIGGER\_PIN,GPIO.OUT)

GPIO.setup(IR\_SENSOR\_LEFT,GPIO.IN)

GPIO.setup(IR\_SENSOR\_RIGHT,GPIO.IN)

GPIO.setup(LEFT\_MOTOR\_PIN1,GPIO.OUT)

GPIO.setup(LEFT\_MOTOR\_PIN2,GPIO.OUT)

GPIO.setup(RIGHT\_MOTOR\_PIN1,GPIO.OUT)

GPIO.setup(RIGHT\_MOTOR\_PIN2,GPIO.OUT)

GPIO.setup(ENABLE\_LEFT\_MOTOR,GPIO.OUT)

GPIO.setup(ENABLE\_RIGHT\_MOTOR,GPIO.OUT)

GPIO.output(LEFT\_MOTOR\_PIN1,GPIO.LOW)

GPIO.output(LEFT\_MOTOR\_PIN2,GPIO.LOW)

GPIO.output(RIGHT\_MOTOR\_PIN1,GPIO.LOW)

GPIO.output(RIGHT\_MOTOR\_PIN2,GPIO.LOW)

MOTOR\_LEFT = GPIO.PWM(ENABLE\_LEFT\_MOTOR, 1000)

MOTOR\_RIGHT = GPIO.PWM(ENABLE\_RIGHT\_MOTOR, 2000)

MOTOR\_LEFT.start(75)

MOTOR\_RIGHT.start(75)

ser = serial.Serial('/dev/ttyAMA0', 9600, timeout=1)

ser.reset\_input\_buffer()

# Robot states

# STATE\_MOVE\_TO\_TRAILER

# SEARCH\_LINE\_1

# SEARCH\_LINE\_10

# SEARCH\_LINE\_2

# STORE\_ONE

# SEARCH\_LINE\_3

# DROPPING

# ROTATE150

# SEARCH\_LINE\_4

# BACK\_TO\_REST

#currentState = "TURN\_FINAL"

currentState = "STATE\_MOVE\_TO\_TRAILER"

def scan\_and\_pick():

ser.write(b'pick\n')

while not ser.readline():

line = ser.readline().decode('utf-8').rstrip()

print(line)

ser.flushInput()

ser.flushOutput()

ser.read\_all()

def drop():

ser.write(b'drop\n')

while not ser.readline():

line = ser.readline().decode('utf-8').rstrip()

print(line)

ser.flushInput()

ser.flushOutput()

ser.read\_all()

def Mforward(speed=75):

MOTOR\_LEFT.ChangeDutyCycle(speed)

MOTOR\_RIGHT.ChangeDutyCycle(speed)

GPIO.output(LEFT\_MOTOR\_PIN1,GPIO.HIGH)

GPIO.output(LEFT\_MOTOR\_PIN2,GPIO.LOW)

GPIO.output(RIGHT\_MOTOR\_PIN1,GPIO.HIGH)

GPIO.output(RIGHT\_MOTOR\_PIN2,GPIO.LOW)

while GPIO.input(IR\_LEFT) == 0 and GPIO.input(IR\_RIGHT) == 0:

print("BOTH 0")

break

while GPIO.input(IR\_LEFT) == 1 or GPIO.input(IR\_RIGHT) == 1:

print("ONE 1")

break

print("forward")

def Mreverse(speed=75):

MOTOR\_LEFT.ChangeDutyCycle(speed)

MOTOR\_RIGHT.ChangeDutyCycle(speed)

GPIO.output(LEFT\_MOTOR\_PIN1,GPIO.LOW)

GPIO.output(LEFT\_MOTOR\_PIN2,GPIO.HIGH)

GPIO.output(RIGHT\_MOTOR\_PIN1,GPIO.HIGH)

GPIO.output(RIGHT\_MOTOR\_PIN2,GPIO.LOW)

print("reverse")

def Mleft(speed=75, angle=90):

MOTOR\_LEFT.ChangeDutyCycle(speed)

MOTOR\_RIGHT.ChangeDutyCycle(speed)

GPIO.output(LEFT\_MOTOR\_PIN1,GPIO.LOW)

GPIO.output(LEFT\_MOTOR\_PIN2,GPIO.LOW)

GPIO.output(RIGHT\_MOTOR\_PIN1,GPIO.HIGH)

GPIO.output(RIGHT\_MOTOR\_PIN2,GPIO.LOW)

print("left")

def Mright(speed=75, angle=90):

MOTOR\_LEFT.ChangeDutyCycle(speed)

MOTOR\_RIGHT.ChangeDutyCycle(speed)

GPIO.output(LEFT\_MOTOR\_PIN1,GPIO.HIGH)

GPIO.output(LEFT\_MOTOR\_PIN2,GPIO.LOW)

GPIO.output(RIGHT\_MOTOR\_PIN1,GPIO.LOW)

GPIO.output(RIGHT\_MOTOR\_PIN2,GPIO.LOW)

print("right")

def Mstop(speed=100):

GPIO.output(LEFT\_MOTOR\_PIN1,GPIO.LOW)

GPIO.output(LEFT\_MOTOR\_PIN2,GPIO.LOW)

GPIO.output(RIGHT\_MOTOR\_PIN1,GPIO.LOW)

GPIO.output(RIGHT\_MOTOR\_PIN2,GPIO.LOW)

print("stop")

def rotate\_motor(right\_motor\_speed, left\_motor\_speed):

if right\_motor\_speed < 0:

GPIO.output(RIGHT\_MOTOR\_PIN1, GPIO.LOW)

GPIO.output(RIGHT\_MOTOR\_PIN2, GPIO.HIGH)

elif right\_motor\_speed > 0:

GPIO.output(RIGHT\_MOTOR\_PIN1, GPIO.HIGH)

GPIO.output(RIGHT\_MOTOR\_PIN2, GPIO.LOW)

else:

GPIO.output(RIGHT\_MOTOR\_PIN1, GPIO.LOW)

GPIO.output(RIGHT\_MOTOR\_PIN2, GPIO.LOW)

if left\_motor\_speed < 0:

GPIO.output(LEFT\_MOTOR\_PIN1, GPIO.LOW)

GPIO.output(LEFT\_MOTOR\_PIN2, GPIO.HIGH)

elif left\_motor\_speed > 0:

GPIO.output(LEFT\_MOTOR\_PIN1, GPIO.HIGH)

GPIO.output(LEFT\_MOTOR\_PIN2, GPIO.LOW)

else:

GPIO.output(LEFT\_MOTOR\_PIN1, GPIO.LOW)

GPIO.output(LEFT\_MOTOR\_PIN2, GPIO.LOW)

return 0

def adjust\_forward(t1=1,t2=0.2):

Mstop()

time.sleep(t1)

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

time.sleep(t2)

def move\_storeA():

global currentState

global STORE\_NUMBER

while True:

# STATE 1

if currentState == "STATE\_MOVE\_TO\_TRAILER":

while currentState == "STATE\_MOVE\_TO\_TRAILER":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else :

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.4)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

print("STATE 1 FINISHED")

currentState = "SEARCH\_LINE\_1"

# STATE 2

elif currentState == "SEARCH\_LINE\_1":

while currentState == "SEARCH\_LINE\_1":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

print("\_\_\_\_1")

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

print("\_\_\_\_2")

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

print("\_\_\_\_2\_\_\_\_1")

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

print("\_\_\_\_2\_\_\_\_2")

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

print("\_\_\_\_2\_\_\_\_3")

# Both sensors are on the line, stop

rotate\_motor(0, 0)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.4)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

print("STATE 1 FINISHED")

currentState = "SEARCH\_LINE\_10"

# STATE 3

elif currentState == "SEARCH\_LINE\_10":

print("ENTERING STATE 3")

while currentState == "SEARCH\_LINE\_10":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

print("\_\_\_\_3\_\_\_\_1")

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

print("\_\_\_\_3\_\_\_\_2")

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

print("\_\_\_\_3\_\_\_\_3")

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

print("\_\_\_\_3\_\_\_\_4")

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

print("\_\_\_\_3\_\_\_\_5")

rotate\_motor(0,0)

time.sleep(0.45)

currentState = "SEARCH\_LINE\_2"

# STATE 4

elif currentState == "SEARCH\_LINE\_2":

while currentState == "SEARCH\_LINE\_2":

print("ENTERING STATE 4")

Mstop()

print("MOTOR STOPPED")

scan\_and\_pick()

print("ARM CALLED")

print("ENTERING STATE 4")

currentState = "STORE\_ONE"

#break

# STATE 5

elif currentState == "STORE\_ONE":

while currentState == "STORE\_ONE":

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(2.0)

print("STATE STORE FINISHED")

currentState = "SEARCH\_LINE\_3"

#break

# STATE 6

elif currentState == "SEARCH\_LINE\_3":

print("ENTERING STATE 6")

while currentState == "SEARCH\_LINE\_3":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

currentState = "DROPPING"

#break

# STATE 7

elif currentState == "DROPPING":

print("ENTERING STATE 7")

while currentState == "DROPPING":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.7)

adjust\_forward()

currentState = "CHOOSE\_STORE"

#break

# STATE 8

elif currentState == "CHOOSE\_STORE":

print("ENTERING STATE CHOOSE\_STORE")

while currentState == "CHOOSE\_STORE":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

if STORE\_NUMBER == 1:

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

currentState = "GO\_TO\_STORE\_1"

elif STORE\_NUMBER == 2:

currentState = "SKIP\_STORE\_1"

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

# STATE 9

elif currentState == "GO\_TO\_STORE\_1":

print("ENTERING GO\_TO\_STORE\_1")

while currentState == "GO\_TO\_STORE\_1":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

#rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

#time.sleep(1.7)

print("BACK TO REST STATE FINISHED")

currentState = "STATE\_9\_10"

#break

# STATE 9\_10

elif currentState == "STATE\_9\_10":

while currentState == "STATE\_9\_10":

print("ENTERING STATE 9\_10")

Mstop()

print("MOTOR STOPPED")

drop()

print("ARM CALLED TO DROP")

print("DROPPED")

currentState = "180\_FROM\_STORE\_1"

# STATE 5

elif currentState == "180\_FROM\_STORE\_1":

while currentState == "180\_FROM\_STORE\_1":

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(2.0)

print("STATE 180\_FROM\_STORE\_1 FINISHED")

currentState = "TURN\_FROM\_STORE\_1"

#break

# STATE 10

elif currentState == "TURN\_FROM\_STORE\_1":

print("ENTERING TURN\_FROM\_STORE\_1")

while currentState == "TURN\_FROM\_STORE\_1":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(0, 0)

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

currentState = "TURN\_RIGHT\_FROM\_STORE\_1"

#break

# STATE 11

elif currentState == "TURN\_RIGHT\_FROM\_STORE\_1":

print("ENTERING TURN\_RIGHT\_FROM\_STORE\_1")

while currentState == "TURN\_RIGHT\_FROM\_STORE\_1":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

currentState = "PROCEED\_TO\_FINAL\_FROM\_1"

# TURN\_RIGHT\_STATE\_1

#break

# STATE 12

elif currentState == "PROCEED\_TO\_FINAL\_FROM\_1":

print("ENTERING PROCEED\_TO\_FINAL\_FROM\_1")

while currentState == "PROCEED\_TO\_FINAL\_FROM\_1":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

# check forward/reverse -/+

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

currentState = "TURN\_FINAL"

# TURN\_RIGHT\_STATE\_1

#break

# STATE 15

elif currentState == "TURN\_FINAL":

print("ENTERING\_STATE\_15")

while currentState == "TURN\_FINAL":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(2)

print("EXIT TURN FINAL")

print(currentState)

currentState = "INTERMEDIATE"

else:

break

def move\_storeB():

global currentState

global STORE\_NUMBER

while True:

# STATE 1

if currentState == "STATE\_MOVE\_TO\_TRAILER":

while currentState == "STATE\_MOVE\_TO\_TRAILER":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else :

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.4)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

print("STATE 1 FINISHED")

currentState = "SEARCH\_LINE\_1"

# STATE 2

elif currentState == "SEARCH\_LINE\_1":

while currentState == "SEARCH\_LINE\_1":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

print("\_\_\_\_1")

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

print("\_\_\_\_2")

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

print("\_\_\_\_2\_\_\_\_1")

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

print("\_\_\_\_2\_\_\_\_2")

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

print("\_\_\_\_2\_\_\_\_3")

# Both sensors are on the line, stop

rotate\_motor(0, 0)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.4)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

print("STATE 1 FINISHED")

currentState = "SEARCH\_LINE\_10"

# STATE 3

elif currentState == "SEARCH\_LINE\_10":

print("ENTERING STATE 3")

while currentState == "SEARCH\_LINE\_10":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

print("\_\_\_\_3\_\_\_\_1")

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

print("\_\_\_\_3\_\_\_\_2")

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

print("\_\_\_\_3\_\_\_\_3")

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

print("\_\_\_\_3\_\_\_\_4")

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

print("\_\_\_\_3\_\_\_\_5")

rotate\_motor(0,0)

time.sleep(0.45)

currentState = "SEARCH\_LINE\_2"

# STATE 4

elif currentState == "SEARCH\_LINE\_2":

while currentState == "SEARCH\_LINE\_2":

print("ENTERING STATE 4")

Mstop()

print("MOTOR STOPPED")

scan\_and\_pick()

print("ARM CALLED")

print("ENTERING STATE 4")

currentState = "STORE\_ONE"

#break

# STATE 5

elif currentState == "STORE\_ONE":

while currentState == "STORE\_ONE":

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(2.0)

print("STATE STORE FINISHED")

currentState = "SEARCH\_LINE\_3"

#break

# STATE 6

elif currentState == "SEARCH\_LINE\_3":

print("ENTERING STATE 6")

while currentState == "SEARCH\_LINE\_3":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

currentState = "DROPPING"

#break

# STATE 7

elif currentState == "DROPPING":

print("ENTERING STATE 7")

while currentState == "DROPPING":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.7)

adjust\_forward()

currentState = "CHOOSE\_STORE"

#break

# STATE 8

elif currentState == "CHOOSE\_STORE":

print("ENTERING STATE CHOOSE\_STORE")

while currentState == "CHOOSE\_STORE":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

if STORE\_NUMBER == 1:

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

currentState = "GO\_TO\_STORE\_1"

elif STORE\_NUMBER == 2:

currentState = "SKIP\_STORE\_1"

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

# STATE \_\_\_\_

elif currentState == "SKIP\_STORE\_1":

print("ENTERING SKIP\_STORE\_1")

while currentState == "SKIP\_STORE\_1":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

# check forward/reverse -/+

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

currentState = "GO\_TO\_STORE\_2"

# TURN\_RIGHT\_STATE\_1

#break

# STATE \_\_\_\_\_1

elif currentState == "GO\_TO\_STORE\_2":

print("ENTERING GO\_TO\_STORE\_2")

while currentState == "GO\_TO\_STORE\_2":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

#rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

#time.sleep(1.7)

print("BACK TO REST STATE FINISHED")

currentState = "STATE\_DROP\_STORE\_2"

#break

# STATE \_\_\_\_\_\_2

elif currentState == "STATE\_DROP\_STORE\_2":

while currentState == "STATE\_DROP\_STORE\_2":

print("ENTERING STATE\_DROP\_STORE\_2")

Mstop()

print("MOTOR STOPPED")

drop()

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(2.0)

print("ARM CALLED TO DROP AT STORE 2")

print("DROP")

currentState = "TURN\_FROM\_STORE\_2"

# STATE \_\_\_\_\_\_3

elif currentState == "TURN\_FROM\_STORE\_2":

print("ENTERING TURN\_FROM\_STORE\_2")

while currentState == "TURN\_FROM\_STORE\_2":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(0, 0)

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

currentState = "TURN\_RIGHT\_FROM\_STORE\_2"

# STATE \_\_\_\_\_\_\_\_4

elif currentState == "TURN\_RIGHT\_FROM\_STORE\_2":

print("ENTERING TURN\_RIGHT\_FROM\_STORE\_2")

while currentState == "TURN\_RIGHT\_FROM\_STORE\_2":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

currentState = "TURN\_FINAL"

# STATE 15

elif currentState == "TURN\_FINAL":

print("ENTERING\_STATE\_15")

while currentState == "TURN\_FINAL":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

rotate\_motor(0, 0)

time.sleep(2)

print("EXIT TURN FINAL")

print(currentState)

currentState = "INTERMEDIATE"

move\_storeA()

print("HERE\_DONE\_A")

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

print("HERE\_DONE\_B")

adjust\_forward(t2=0.5)

print("HERE\_DONE\_C")

print(currentState)

if currentState == "INTERMEDIATE":

print("ENTERING INTERMEDIATE")

while currentState == "INTERMEDIATE":

RIGHT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_RIGHT)

LEFT\_IR\_SENSOR\_VALUE = GPIO.input(IR\_SENSOR\_LEFT)

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == LOW:

print("\_\_\_\_3\_\_\_\_1")

# Both sensors are off the line, move forward

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

else:

# At least one sensor is on the line, adjust motor control

if RIGHT\_IR\_SENSOR\_VALUE == LOW and LEFT\_IR\_SENSOR\_VALUE == HIGH:

print("\_\_\_\_3\_\_\_\_2")

# Right sensor detects the line, turn right

rotate\_motor(TURN\_SPEED, -TURN\_SPEED)

elif RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == LOW:

print("\_\_\_\_3\_\_\_\_3")

# Left sensor detects the line, turn left

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

else:

print("\_\_\_\_3\_\_\_\_4")

# Both sensors are on the line, stop

rotate\_motor(TURN\_SPEED, TURN\_SPEED)

# Check if both sensors are off the line and transition to the next state

if RIGHT\_IR\_SENSOR\_VALUE == HIGH and LEFT\_IR\_SENSOR\_VALUE == HIGH:

print("ENTERING CYCLE 2")

Mstop()

time.sleep(0.9)

rotate\_motor(-TURN\_SPEED, TURN\_SPEED)

time.sleep(0.65)

adjust\_forward()

currentState = "STATE\_MOVE\_TO\_TRAILER"

move\_storeB()

GPIO.cleanup()

print("GPIO Clean up")