

Mohamed Ashraf – Omar ehab- omarr osman- ahmed hosni-Khaled Soliman

R.D.IEEEHTI@gmail.com

Ball follwer robot

IEEE High Technological institute

2020

**Contents**

•About US……………………………………………………………………..2

•Introduction…………………………………………………………………4

•Circuit diagram

• Project Code

•Hardware description

•Project components with their prices

**About IEEE HTI-SB:**

In 1999 IEEE HTI-SB was the first founded IEEE student branch in the Egypt Section. The student branch was considered one of the best branches worldwide, and surely the best in Egypt, the branch made several academic sessions and Events.

**Our Vision:**

Bridging the gap between academic studies and real productive industry. Since technology leads in every part of the world. Creativity is also required which is why IEEE is very persistent on this event. Allowing students and giving them motivation, advanced skills and materials to meet the requirements of today's market.

**About R&D Committee:**

Ranked Committee working on bridging the technical problems with academic solutions based on searching, testing, debugging, and enhancing and developing new models & solutions.

Targeting engineering department (Mechatronics, Electrical, Mechanical, Biomedical and computer).

As we are part of the IEEE Global community, we advance technology for humanity.

Our objectives (benefits reflected on members) building experience fulfilled with achievement by simulating corporate tasks & trends helping him hunt jobs related to his activities as a result.

Like a training field to find his technical interests and get prepared for bigger deals.

**Our Goals:**

* Technical (Papers/Reports/Proposals), about scientific topics for students.
* Online tutorials of project demos & open source technical materials.
* Providing workshops based on new self learnt new technology trends (AI/IoT, etc.)
* Project lab producing technical innovative solutions of new technology trends.
* Technical assistance in Events / Competitions /Graduation projects.
* Participating in YP activities (EED/MIE/Nasa Space apps/FC)
* Calibrating /controlling parts, Eliminating Errors developing products.
* Community educational tracks online course materials & Webinars.
* Supporting the market & Startups as freelancers solving industrial problems.
* write researches/papers for world’s professionals.
* publish in the IEEE global using IEEE Spectrum Magazine.

**Introduction:**

**Overview:**

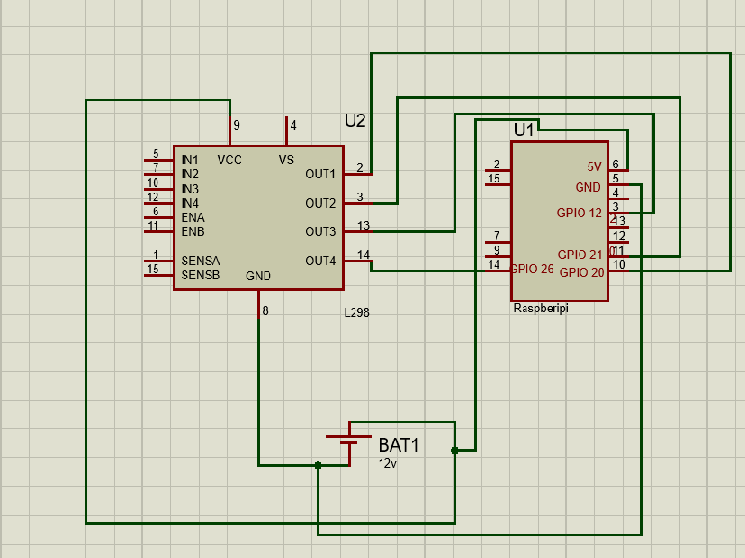
**In this project of ball follower car. We use image processing to make car able to recognize the ball color and run forward it. to make image processing on the camera photo we use Python programming language, we use “Open-Cv” library and we use Raspberry pi to write the code of image processing on it and give motor signal to move to the ball position.**

**In this project we have learned open-cv python library to make image processing on photos and we have learned how to use raspberry pi and how to use Gpio to control to motors .**

**Project Objective:**

1. **Raspberry Pi**
2. **Camera**
3. **Robot Chassis**
4. **Gear motors with wheel**
5. **L298N H-Bridge Motor driver**
6. **Power bank or any other portable power source**

**Circuit diagram:**

****

**Project Code**:

import cv2

import numpy as np

import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(26,GPIO.OUT)

GPIO.setup(12,GPIO.OUT)

GPIO.setup(20,GPIO.OUT)

GPIO.setup(21,GPIO.OUT)

def left():

GPIO.output(21,GPIO.HIGH)

GPIO.output(20,GPIO.LOW)

def right():

GPIO.output(20,GPIO.HIGH)

GPIO.output(21,GPIO.LOW)

def stop():

GPIO.output(20,GPIO.LOW)

GPIO.output(21,GPIO.LOW)

GPIO.output(12,GPIO.LOW)

GPIO.output(26,GPIO.LOW)

def forward():

GPIO.output(20,GPIO.HIGH)

GPIO.output(21,GPIO.HIGH)

cap = cv2.VideoCapture(0)

cv2.namedWindow('mask')

def nothing():

pass

#Trackbar for measure pixel value easily

cv2.createTrackbar('b','mask',0,255,nothing)

cv2.createTrackbar('g','mask',0,255,nothing)

cv2.createTrackbar('r','mask',0,255,nothing)

cv2.createTrackbar('B','mask',0,255,nothing)

cv2.createTrackbar('G','mask',0,255,nothing)

cv2.createTrackbar('R','mask',0,255,nothing)

while True:

\_,frame = cap.read()

hsv = cv2.cvtColor(frame,cv2.COLOR\_BGR2HSV)

b = cv2.getTrackbarPos('b','mask')

g = cv2.getTrackbarPos('g','mask')

r = cv2.getTrackbarPos('r','mask')

lower = np.array([10,55,255])

ubber = np.array([ 22,255,255])

mask = cv2.inRange (hsv, lower, ubber)

contours = cv2.findContours(mask.copy(),cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_SIMPLE)

if len(contours)>0:

contour = max(contours, key=cv2.contourArea)

(x,y,w,h) = cv2.boundingRect(contour)

if cv2.contourArea(contour)>14000:

print('stop')

stop()

elif x<640/3:

print('left')

left()

elif x>(640/3)\*2:

print('Right')

right()

elif x>640/3 and x <(640/3)\*2:

print('forward')

forward()

else:

stop()

k = cv2.waitKey(5)

if k == 27:

break

cap.release()

cv2.destroyAllWindows()

**Project components:**



