VISION BASED NAVIGATION OF ARDUINO ROBOT

Project Number	Group members	Supervisor
44	Mustafa Göktürk Yandım 24134 Furkan Çelik 23631	Mustafa Ünel

INTRODUCTION

In this project, we have used two different micro-controllers called Raspberry Pi and Arduino. Raspberry Pi is in control of vision part of our project, which enables us to detect different colors and to analyze camera visuals. Raspberry Pi detects colored obstacles using OpenCV and then calculates the safe distance before any collision. When Raspberry Pi detects any possible collision with the obstacle, it sends a signal to the Arduino using an usb cable with serial communication procedure. Arduino controls the motors and changes the direction according to the received signal to avoid collision.

PURPOSE

To build an autonomous robot which can avoid obstacles by detecting them with a camera and help of vision algorithms.

ROBOT & PARTS

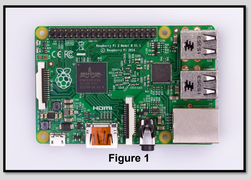


Figure 1

Raspberry Pi 2 Model B

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse (Pi Foundation, 2017). Raspberry Pi is powerful enough for vision apllications. We are using it with OpenCV libraries, which enables us to detect different colors in the environment.

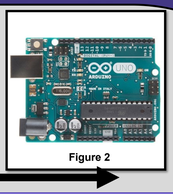
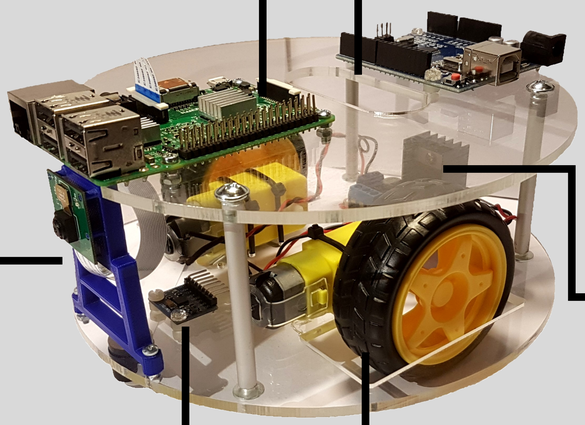


Figure 2

Arduino Uno R3

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs and send outputs (Arduino, 2017). We are using Arduino to control motors in this project.



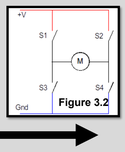


Figure 3.2

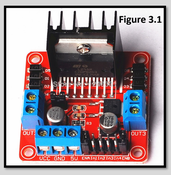
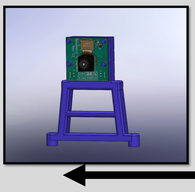


Figure 3.1

Ln298n Motor Driver

A circuit which enables to control motors, up to 2. It consists from two H-bridge circuits (Figure 3.2 , H-bridge circuit) which can change the direction of the voltage flow with providing to control motors in two directions, forward-backward. However, H-bridges creates too much heat that heat sinks are necessary for cooling (Figure 3.1 , black part).



Pi Camera

Pi Camera is an original product of Raspberry Pi Foundation. It's resolution is 5 megapixels and supports up to 90 frame rate. We designed a mount on Solidworks and printed it out with a 3D printer.

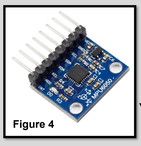


Figure 4

MPU6050 Gyroscope & Acceleration

MPU6050 calculates the position and the acceleration on x,y and z axises. Sensor works with 5 volts. This sensor helps the robot to navigate in a certain direction.

Figure 5

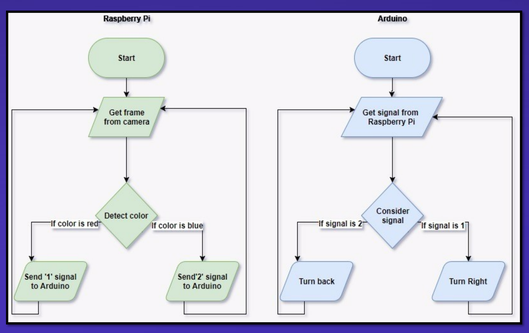
Figure 6

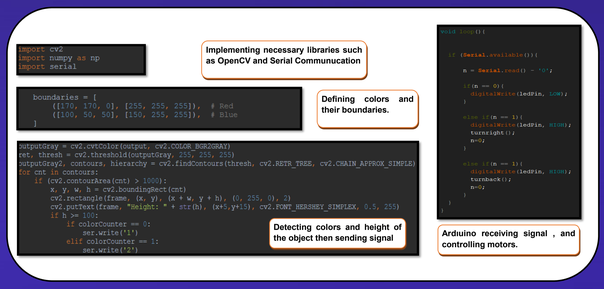
Motor & Wheels

- Two Dc motors with wheels (Figure 5) works between 3-12 volts. They can be controlled by Ln298n motor driver circuit (see above). Motors are fixed on the chasis ; we have designed the chasis using Solidworks and printed out with a laser cutter.

- Two ball caster wheels helps robot to stay balanced. Those wheels are consisting from a ball which can move in every direction freely (Figure 6).

HOW ROBOT WORKS?





OPENCV & COLOR DETECTION

OpenCV is Open Source Computer Vision Library which helps programmers to create vision based applications. OpenCV has more than 2500 optimized algorithms which can be used for different purposes. For instance, face, color and corner detection. We are using color detection algorithm since it does not require too much computing power and easier to use. This algorithm analyzes each pixels color code and decides whether it is inside the given boundary, or not. We have detected both blue and red colored items using this algorithm (Left side).

TIME TO COLLISION (TTC)

In order to avoid any collision, safe distance between robot and the obstacle should be calculated. However, cameras can not measure the distance or depth. Also we are not using any sensor for distance measurement. As a result, our supervisor suggested us to use TTC (time to collision) method. TTC can be found just looking for the height of an object, with knowing change in height , it is possible to calculate the the time remained for collision (Figure 7.1 & 7.2).

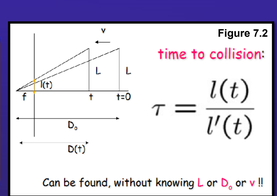


Figure 7.2

SERIAL COMMUNUCATION

We connected Arduino to the Raspberry Pi using an usb cable. Raspberry Pi sends different numbers for different situations. Both micro-controllers are using 9600 baud rate (modulation rate).

```
Serial.begin(9600);
```

```
ser = serial.Serial('/dev/USB0', 9600)
```

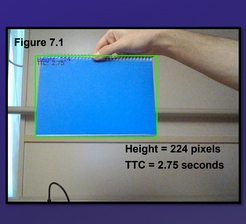


Figure 7.1

REFERENCES

(Figure 1) https://www.raspberrypi.org/app/uploads/2017/05/Raspberry-Pi-2-overhead-1-1576x1080.jpg

(Figure 2) http://www.tinyosshop.com/index.php?route=product/product&product_id=224

(Figure 3.1) https://www.clinchhub.com/wp-content/uploads/2016/12/motor-driver2.jpg

Figure 3.2) http://www.penguintutor.com/electronics/images/h-bridge-01.png

(Figure 4) https://images-na.ssl-images-amazon.com/images/I/61GYpJyVvSL._SL1000_.jpg

(Figure 5) https://ae01.alicdn.com/kf/HT81cnjSXXXXXXaXXXq6xXFXXX4.jpg

(Figure 6) https://www.bananarobotics.com/shop/image/cache/data/sku/BR/0/1/0/1/5/BR010357-Pololu-3-8-Inch-Metal-Ball-Caster-Wheel/Pololu-3-8-Inch-Metal-Ball-Caster-Wheel-600x600.jpg

(Figure 7.2) http://slideplayer.com/slide/8567857/

Arduino. (2017). Arduino - Introduction. Retrieved from https://www.arduino.cc/en/Guide/Introduction

Pi Foundation. (2017). What is a Raspberry Pi? Retrieved from https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/