



**GDAŃSK UNIVERSITY
OF TECHNOLOGY**



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

Declared as Deemed-to-be University under sec. 3 of UGC Act, 1956

Karunya Nagar, Coimbatore 641 114, Tamil Nadu, India.

„Robotics in India” Intelligent Robot - Remote Car project

by Bartłomiej Borzyszkowski (Poland)

*Gdansk University of Technology (ETI)
Karunya Institute of Technology and Sciences (ET)*



GDĄŃSK UNIVERSITY
OF TECHNOLOGY



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

About the author



Bartłomiej Borzyszkowski

- Control Engineering and Robotics (BSc) student (ETI) at Gdansk University of Technology, Poland.
- Scientific intern with the IAESTE program at Karunya Institute of Technology and Sciences, Coimbatore - India.
- Gradient Science Club - artificial intelligence developer.
- IAESTE Poland IT Coordinator and Job Raising member.



"Robotics in India"

by Bartłomiej Borzyszkowski





GDĄŃSK UNIVERSITY
OF TECHNOLOGY



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

Gdańsk and Technology





GDĄŃSK UNIVERSITY
OF TECHNOLOGY



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

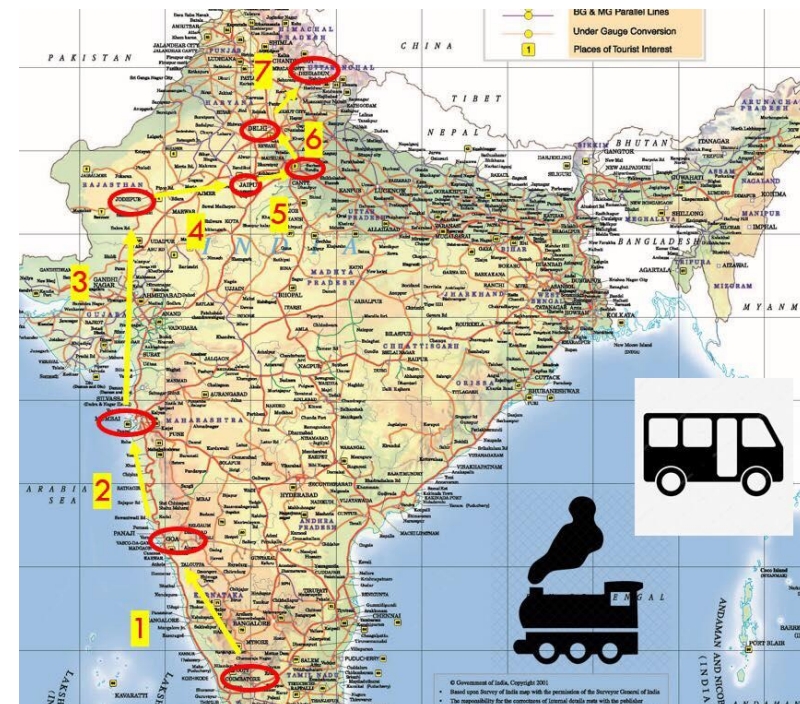
„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

Project introduction

- 8 week internship at the Karunya Electrical Department under the supervision of Dr. K. Rajasekaran with an assistance of Mr. J. Chinnadurai.
- Creating and developing of the intelligent robot, remote/self-driving car.
- Cooperation in the international engineering team.

Besides the work:

- Over 3 week travel adventure including: Goa, Mumbai, Jodhpur, Jaipur, Delhi, Agra, Haridwar, Rishikesh, Ganges, Himalayas.
- Exploration of India, discovery of the rich culture, unique cuisine, wild nature and diverse environment of the country.





GDĄŃSK UNIVERSITY
OF TECHNOLOGY



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

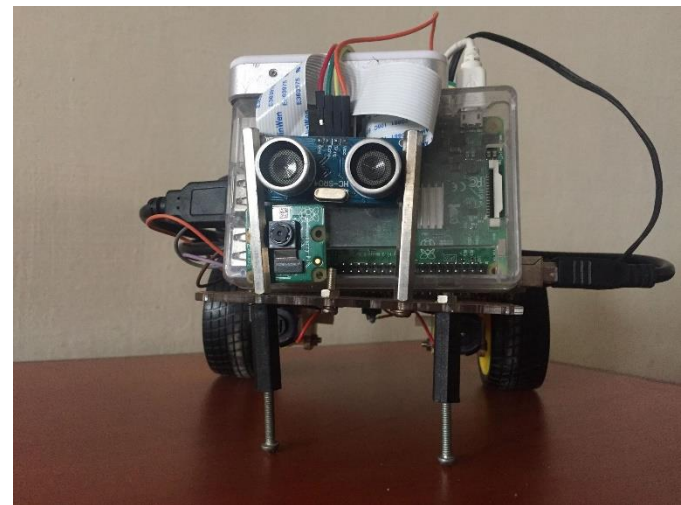
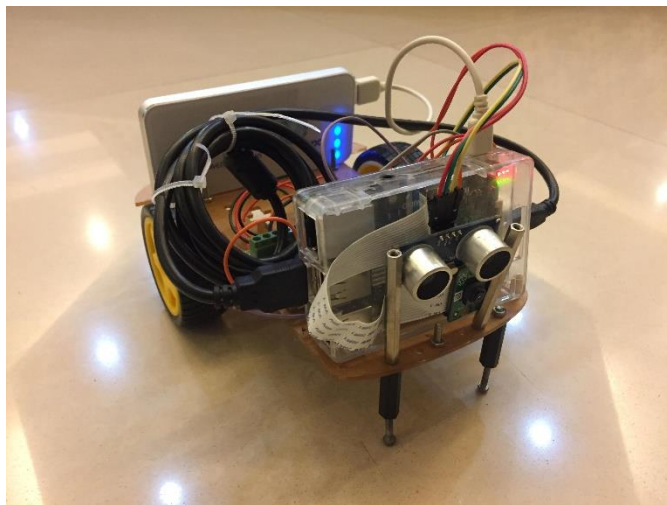
Project introduction

GitHub repository:

- github.com/Borzyszkowski/Robotics-in-India-Intelligent-Robot

Crowdfunding campaigns and the full internship description:

- zrzutka.pl/robotics (Polish)
- youcaring.com/robotics (English)



[Code](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)

Scientific internship at the Karunya Institute of Technology and Sciences in Coimbatore, India. Development of an intelligent, remote/self-driving car based on Raspberry Pi, ultrasonic sensors, camera and Arduino (Python + OpenCV with neural networks and digital image processing). Vehicle capable of recognizing traffic lights, road signs and obs...

[Edit](#)

[Add topics](#)

 84 commits  1 branch  0 releases  3 contributors  BSD-2-Clause

Branch: master ▾

[New pull request](#)


[Create new file](#)

[Upload files](#)

[Find file](#)

[Clone or download ▾](#)


















This branch is 50 commits ahead, 28 commits behind hamuchiwa:master.

 Pull request  Compare



Borzyszkowski scheme

Latest commit 5d89e7c an hour ago

 .idea	scheme	an hour ago
 Multimedia and Sources	scheme	an hour ago
 Traffic_signal	Traffic_Signal	2 years ago
 arduino tests programs by Bartek Borzyszkowski	Ultrasonic change	10 days ago
 arduino	Create README.md	3 years ago
 computer	keyboard server/client	12 days ago
 raspberryPi	keyboard client	11 days ago
 test	readme	a day ago
 venv	init	19 days ago
 .gitattributes	init	19 days ago
 Agenda for the workshop by Borzyszkowski Bartlomiej.jpg	multimedia	a day ago
 LICENSE.md	Create LICENSE.md	3 years ago
 README.md	scheme	an hour ago
 Scheme.png	scheme	an hour ago
 environment.yml	init	19 days ago
 robot1.jpg	robotpic	a day ago
 robot2.jpg	robotpic	a day ago



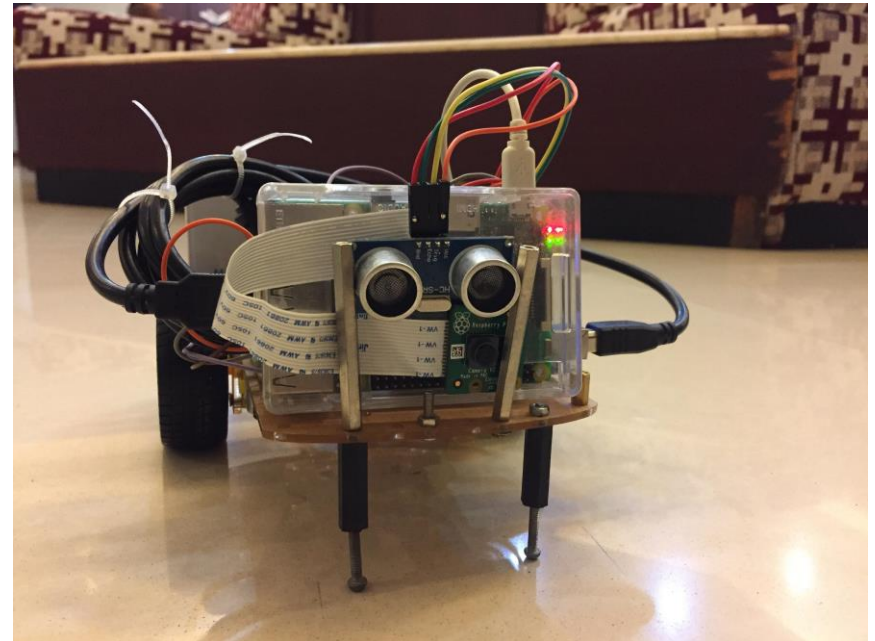
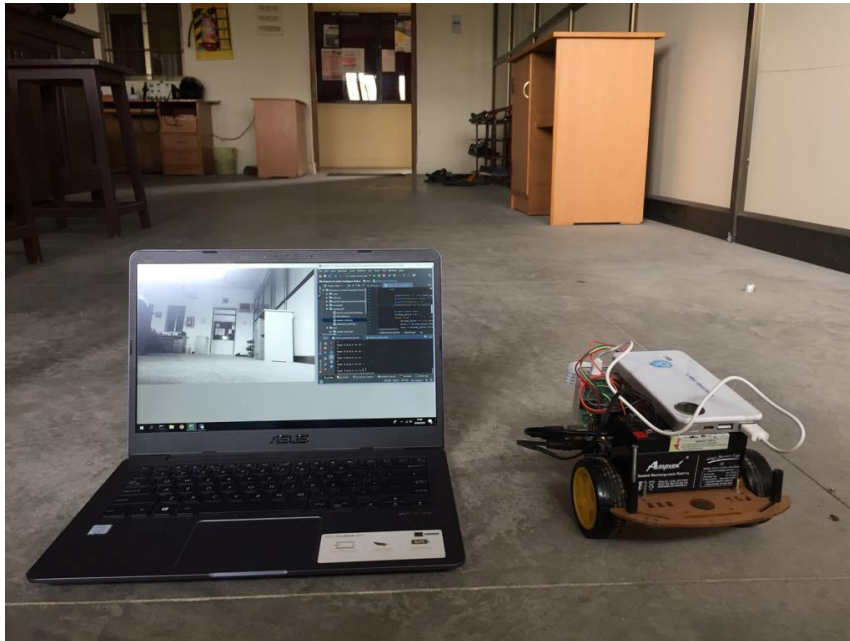
GDĄŃSK UNIVERSITY
OF TECHNOLOGY



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

Robot overview





Inspiration

Zheng Wang Auto RC Car:

- github.com/hamuchiwa/AutoRCCar
- zhengludwig.wordpress.com/projects/self-driving-rc-car/





GDĄŃSK UNIVERSITY
OF TECHNOLOGY

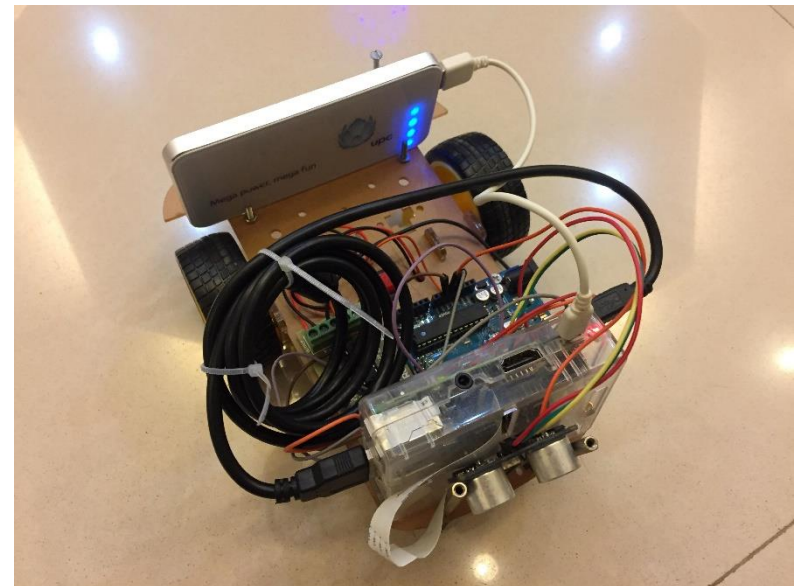
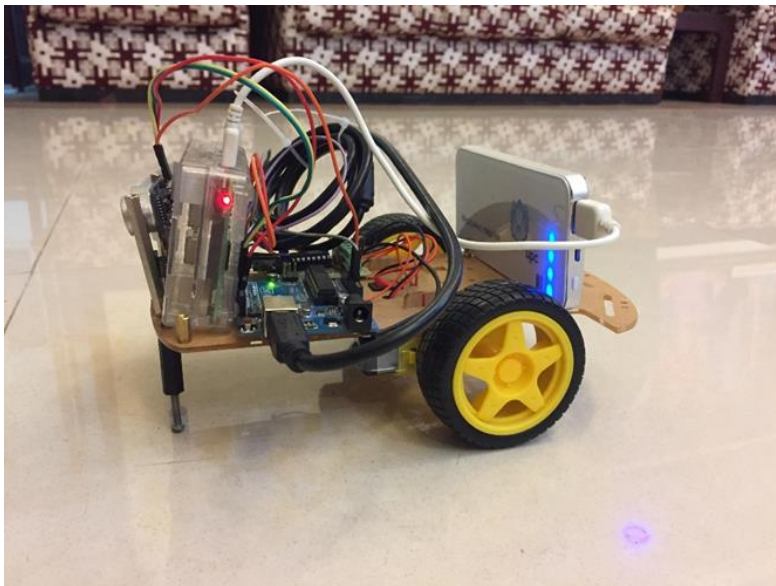


Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

Functionality overview

- Remote control using laptop keyboard
- Camera display on the laptop
- Distance measurement thanks to ultrasonic sensor
- Steering (Forward, Backward, Right, Left, Stop)

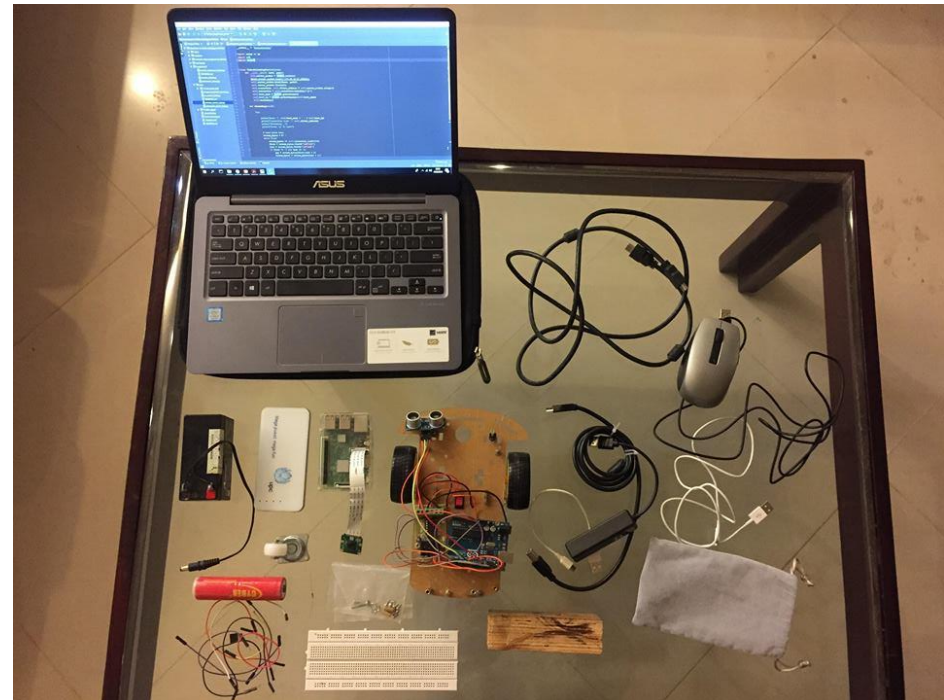




Construction details - components

Project elements:

- Raspberry Pi 3 Model B
- Camera desired for the Raspberry Pi
- Arduino UNO
- Laptop/PC
- Ultrasonic sensor HC-SR04
- Motor driver L293
- 2 separate motors and 2 wheels
- Power supply (Power Bank)
- Robot board – car frame
- Cords/wires and a switch button
- Soldering equipment
- Wi-Fi network
- External display
- Programming environment and software





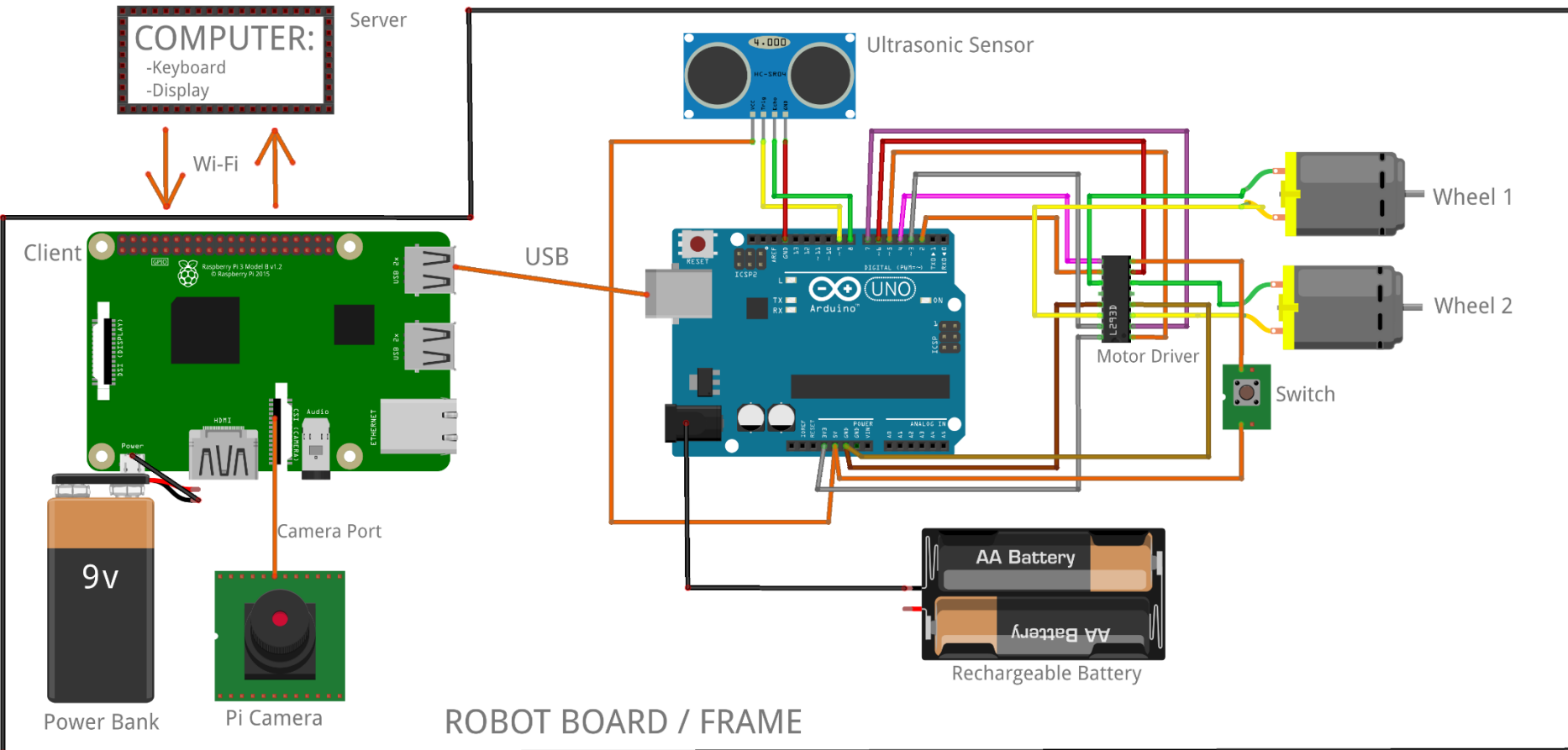
GDĄŃSK UNIVERSITY
OF TECHNOLOGY

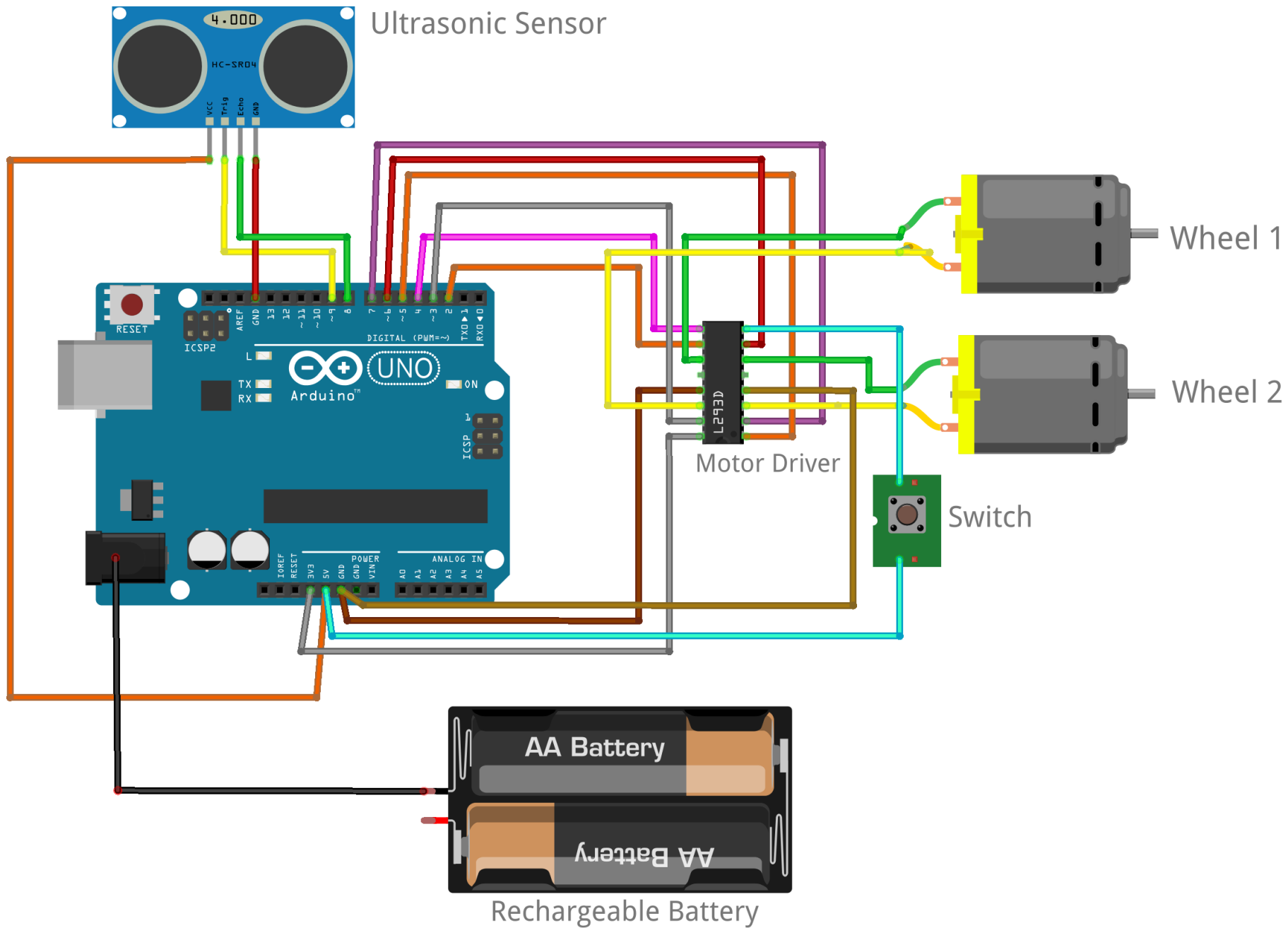


Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

Construction details - architecture



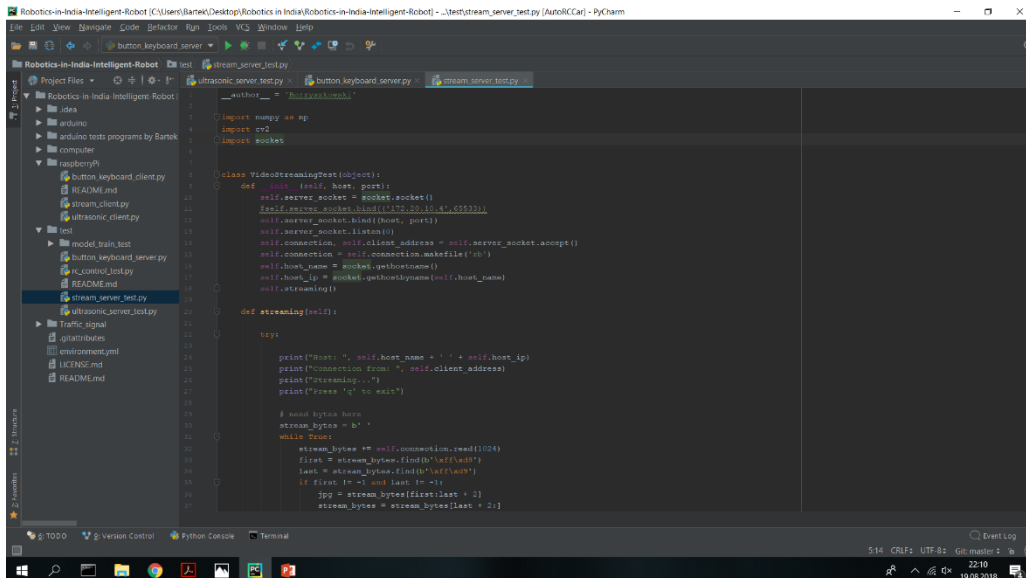




Construction details - Computer

Computer – laptop/PC:

- Working as a server for the Raspberry Pi (client) via Wi-Fi
- Collecting information about steering from the keyboard and forwarding it to the Raspberry Pi
- Receiving video from the Raspberry Pi and showing the display

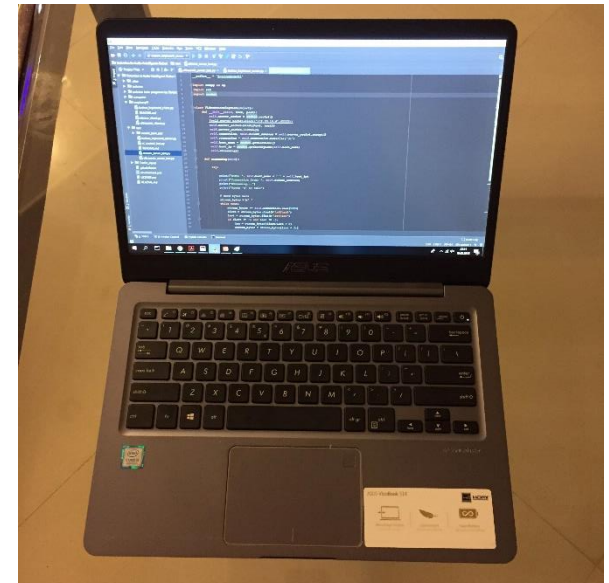


```
#!/usr/bin/env python3
import numpy as np
import cv2
import socket

class VideoStreamingTest(object):
    def __init__(self, host, port):
        self.server_socket = socket.socket()
        self.server_socket.bind((host, port))
        self.server_socket.listen()
        self.connection, self.client_address = self.server_socket.accept()
        self.connection.setsockopt(socket.IPPROTO_TCP, socket.TCP_NODELAY, 1)
        self.host_name = socket.gethostname()
        self.host_ip = socket.gethostbyname(self.host_name)
        self.streaming()

    def streaming(self):
        keys = None
        print("Host: ", self.host_name + " | " + self.host_ip)
        print("Connection from: ", self.client_address)
        print("Streaming...")
        print("Press 'q' to exit")

        # send bytes here
        stream_bytes = b''
        while True:
            stream_bytes += self.connection.recv(1024)
            first = stream_bytes.find(b'\xff\x00')
            last = stream_bytes.find(b'\xff\x00')
            if first != -1 and last != -1:
                sbytes = stream_bytes[first:last + 2]
                stream_bytes = stream_bytes[last + 2:]
```





Construction details - Raspberry Pi

Raspberry Pi:

- Working as a client for the computer (server) via Wi-Fi
- Connected to the Arduino via USB
- Connected with the Pi camera
- Collecting information about steering from the computer (keyboard) and forwarding it to the Arduino
- Collecting data from the camera and sending it to the computer

```
385 t = (s_data['T'] * p_data['T']) / 2.  
386 return {'R': R, 'T': T}  
387  
388 def position_resolved(layer, distance, coh_tmm_data):  
389 """  
390 Starting with output of coh_tmm(), calculate the Poynting vector,  
391 absorbed energy density, and E-field at a specific location. The  
392 location is defined by [layer, distance], defined the same way as in  
393 find_in_structure_with_inf(...).  
394  
395 Returns a dictionary containing:  
396 * poynt - the component of Poynting vector normal to the interfaces  
397 * absor - the absorbed energy density at that point  
398 * Ex and Ey and Ez - the electric field amplitudes, where  
399 z is normal to the interfaces and the light rays are in the x,z plane.  
400  
401 The E-field is in units where the incoming |E|=1; see  
402 https://arxiv.org/pdf/1603.02720.pdf for formulas.  
403 """  
404  
405 if layer > 0:  
406     v,w = coh_tmm_data['vw_list'][layer]  
407     else:  
408         v = 1  
409         w = coh_tmm_data['r']  
410     kz = coh_tmm_data['kz_list'][layer]  
411     th = coh_tmm_data['th_list'][layer]  
412     n = coh_tmm_data['n_list'][layer]  
413     n_0 = coh_tmm_data['n_list'][0]  
414     th_0 = coh_tmm_data['th_0']  
415     pol = coh_tmm_data['pol']  
416  
417     assert ((layer >= 1 and 0 <= distance <= coh_tmm_data['d_list'][layer])  
418            or (layer == 0 and distance <= 0))  
419  
420     # Amplitude of forward-moving wave is Ef, backwards is Eb  
421     Ef = v * exp(i) * kz * distance  
422     Eb = w * exp(-i) * kz * distance
```





Construction details - Arduino

Arduino:

- Connected to the Raspberry Pi via USB
- Connected to the Motor Driver
- Connected to the ultrasonic sensor
- Receiving data for steering from the Raspberry Pi and forwarding information to the Motor Driver
- Receiving distance from the ultrasonic sensor and stopping the car in case of obstacle detection (front)

```
4. ultrasonic (Arduino 1.8.1)
File Edit Sketch Tools Help

// Robotics in India by Bartłomiej Borzyszkowski, Karunya Institute of Technology and Sciences, IASSTP Internship 2018

#include <Wire.h>

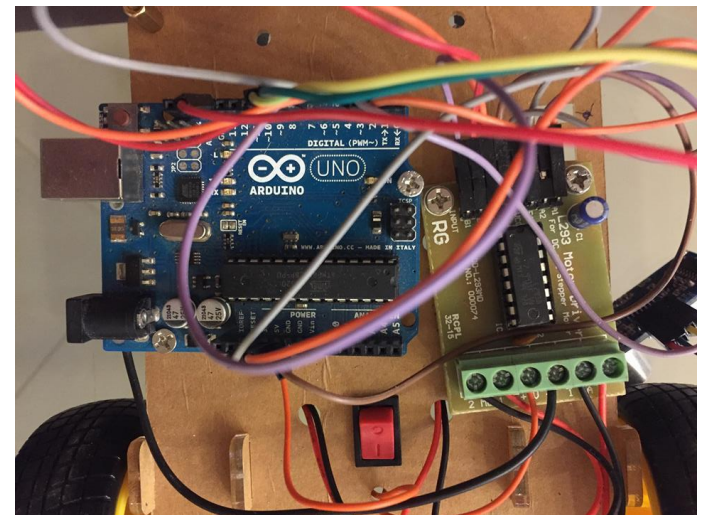
// Adding pinstates PWM:
const int wheel1_forward = 2; // digital 2
const int wheel1_reverse = 3; // digital 3
const int wheel1_enable = 4; // digital 4
const int wheel2_enable = 5; // digital 5
const int wheel2_forward = 6; // digital 6
const int wheel2_reverse = 7; // digital 7

const int echoPin = 8; //digital 8
const int trigPin = 9; //digital 9

// Adding variables
long duration;
int distance;

//initializing
void setup()
{
  pinMode(wheel1_forward, OUTPUT);
  pinMode(wheel1_reverse, OUTPUT);
  pinMode(wheel1_enable, OUTPUT);
  pinMode(wheel2_forward, OUTPUT);
  pinMode(wheel2_reverse, OUTPUT);
  pinMode(wheel2_enable, OUTPUT);

  pinMode(trigPin, OUTPUT); // Set the trigPin as an Output
  pinMode(echoPin, INPUT); // Set the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
}
```





Software details

In the current project version to achieve the required result five programs should run in the same time.

Computer:

- 1., „button_keyboard_server.py”
- 2., „stream_server_test.py”

Raspberry Pi:

- 3., „button_keyboard_client.py”
- 4., „stream_client.py”

Arduino:

- 5., „_7._steering.ino”

The screenshot shows an IDE window with the project structure on the left and the code for `stream_server_test.py` in the main editor. The project structure includes folders for `Robotics-in-India-Intelligent-Robot`, `arduino`, `computer`, `raspberrypi`, and `test`. The `test` folder contains `model_train_test`, `button_keyboard_server.py`, `rc_control_test.py`, `stream_server_test.py`, `ultrasonic_server_test.py`, `Traffic_signal`, `.gitattributes`, `Agenda for the workshop by Borzyszkowski Bartlomiej.jpg`, `environment.yml`, `LICENSE.md`, `README.md`, `robot1.jpg`, and `robot2.jpg`. The code in the main editor is as follows:

```
17 print("connection from: ", self.client_address)
18 print("streaming...")
19 print("Press 'q' to exit")
20
21 # need bytes here
22 stream_bytes = b' '
23 while True:
24     stream_bytes += self.connection.read(1024)
25     first = stream_bytes.find(b'\xff\xd8')
26     last = stream_bytes.find(b'\xff\xd9')
27     if first != -1 and last != -1:
28         jpg = stream_bytes[first:last + 2]
29         stream_bytes = stream_bytes[last + 2:]
30         image = cv2.imdecode(np.frombuffer(jpg, dtype=np.uint8), cv2.IMREAD_GRAYSCALE)
31         #image = cv2.imdecode(np.frombuffer(jpg, dtype=np.uint8), cv2.IMREAD_COLOR)
32         cv2.imshow('image', image)
33         if cv2.waitKey(1) & 0xFF == ord('q'):
34             break
35     finally:
36         self.connection.close()
37         self.server_socket.close()
38
39 if __name__ == '__main__':
40     # host, port
41     h, p = ('172.20.10.4', 65534)
42     VideoStreamingTest(h, p)
43
44 VideoStreamingTest > _init_()
```



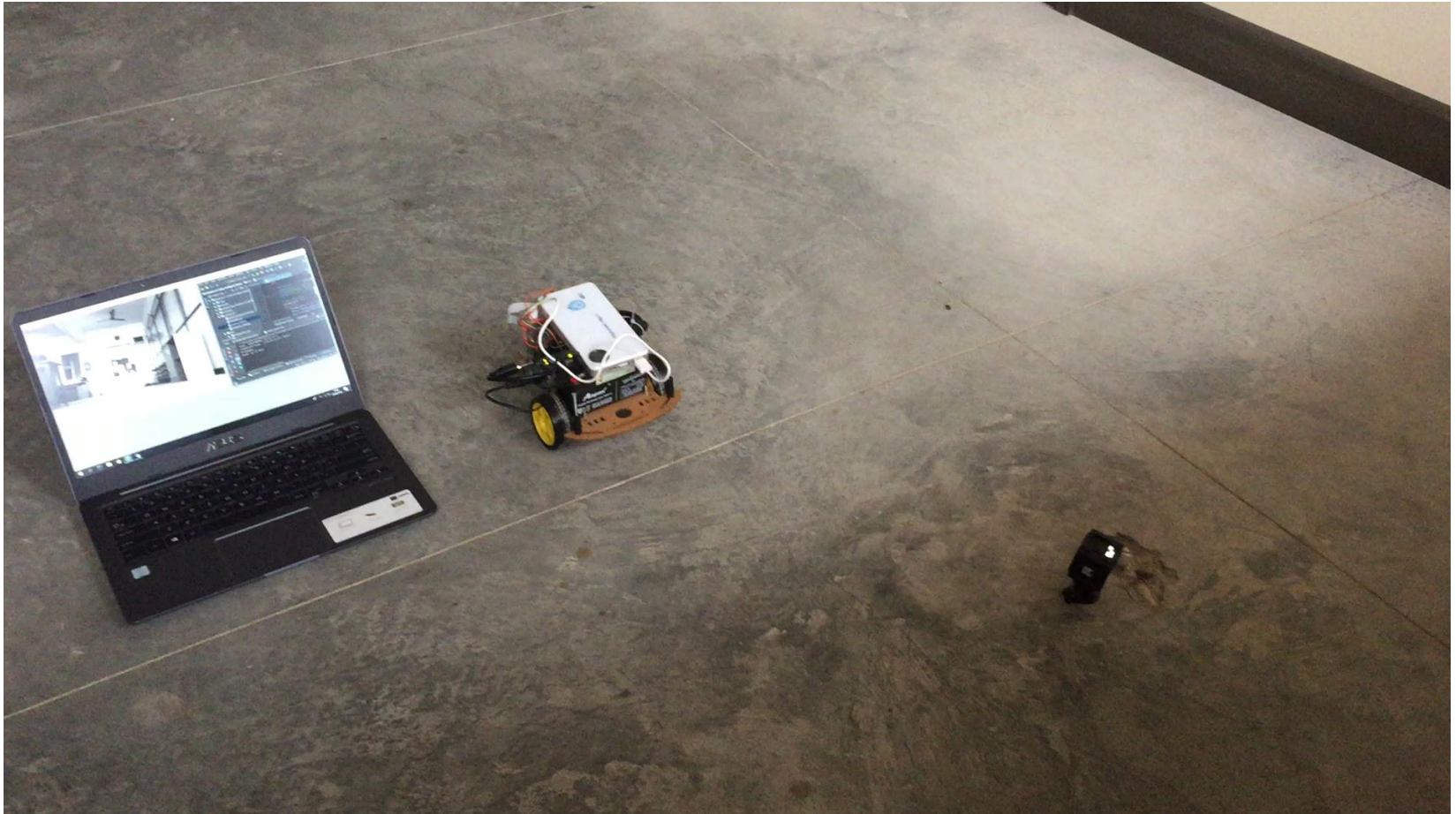

GDĄŃSK UNIVERSITY
OF TECHNOLOGY



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

Actual effect





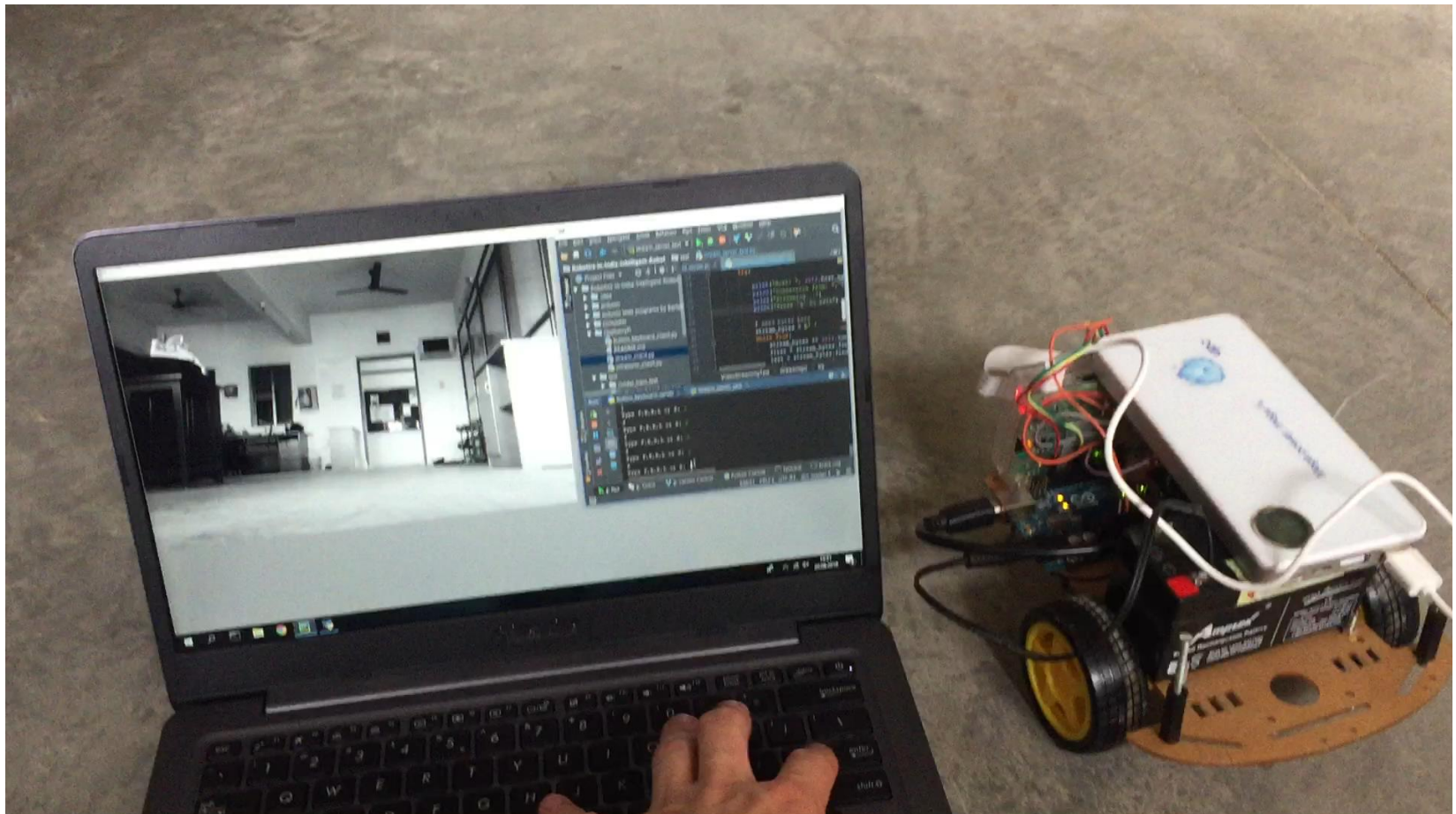
GDĄŃSK UNIVERSITY
OF TECHNOLOGY



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

Actual effect





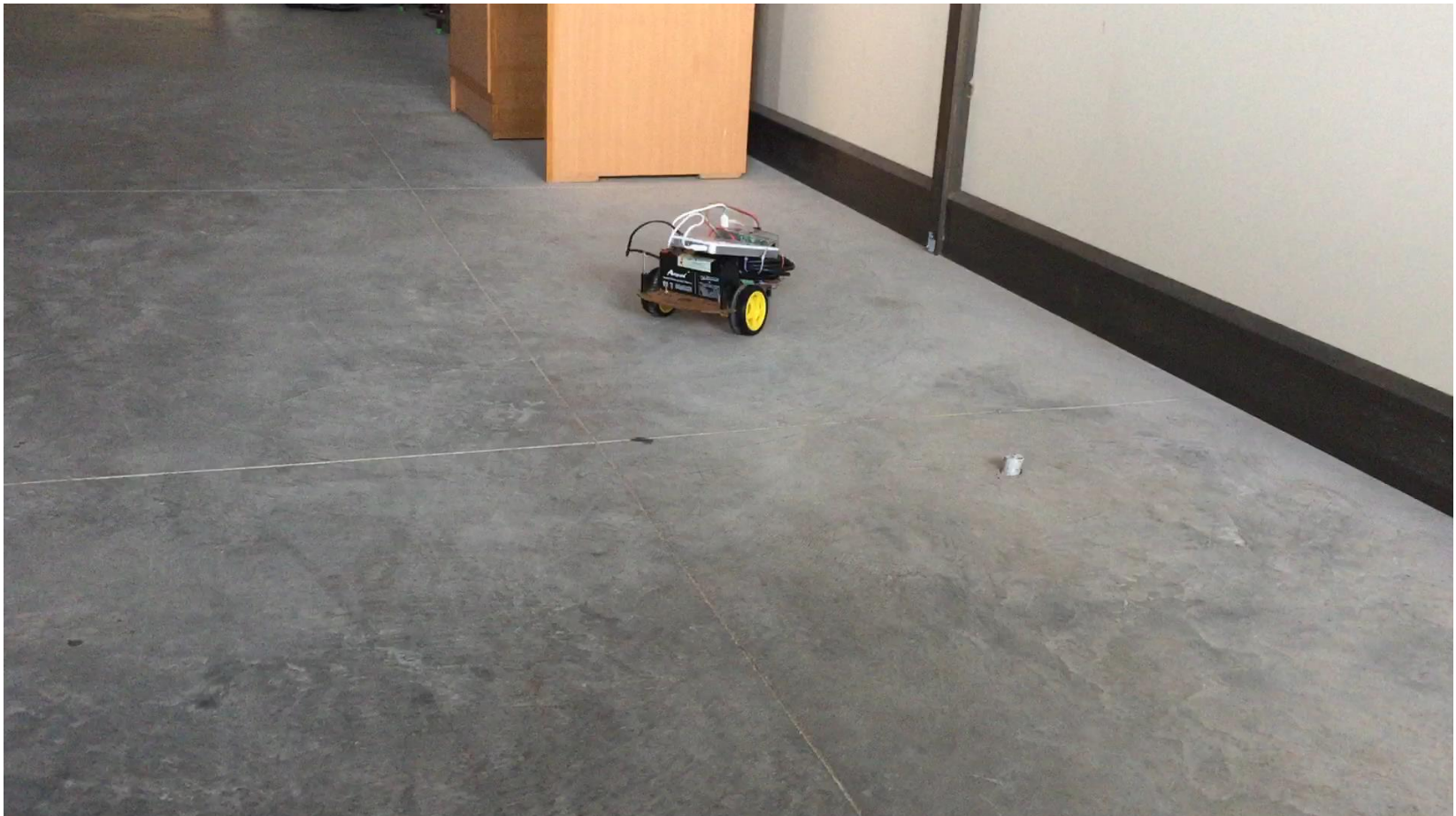
GDĄŃSK UNIVERSITY
OF TECHNOLOGY



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

Actual effect

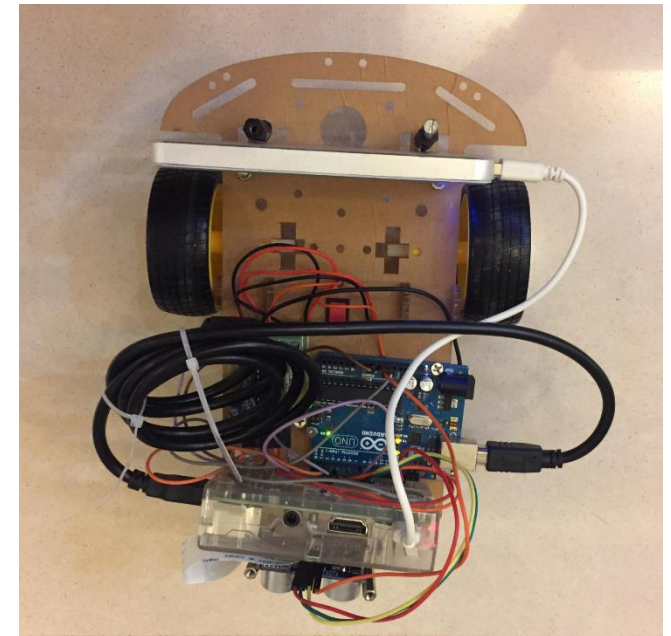
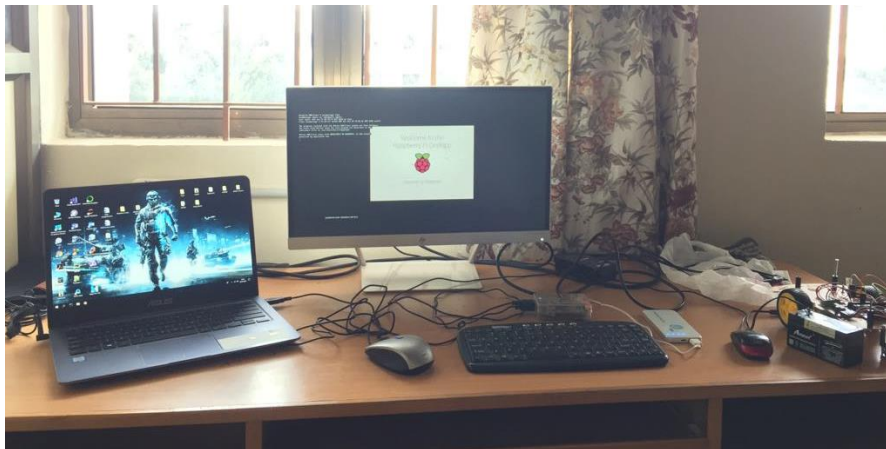




Difficulties

- Real time data transfer (a significant number of operations at the same time)
- Camera video streaming delay (scaling down the resolution is required)
- Multi-level architecture (long way for the information to cover)
- Complex software (big number of programs)

Despite difficulties the robot works!





GDĄŃSK UNIVERSITY
OF TECHNOLOGY



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

Further Develop

- Total autonomous
- Artificial intelligence, neural networks and image processing
- Road lines, signs and traffic lights recognition
- More advance steering
- Both kinds of control (self-drive and remote control) with balanced priorities





GDĄŃSK UNIVERSITY
OF TECHNOLOGY

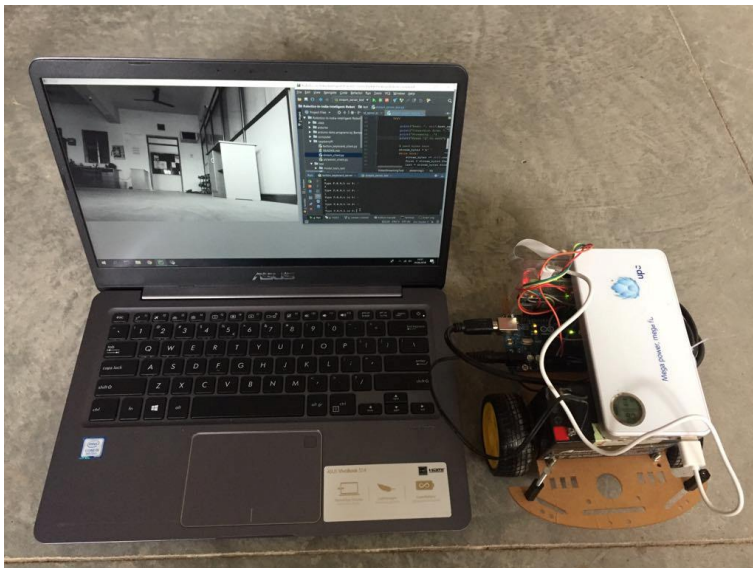


Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

„Robotics in India” – Intelligent Robot project by Bartłomiej Borzyszkowski

Project sum up

To conclude, the aims of this internship were achieved successfully and are posted above as a result and source for future extensions and uses. The design of the robot model allows to observe the processes necessary to construct an intelligent car, learn and draw the presented conclusions that will help in solving future potential problems.





Useful materials

Similar projects:

- by Zheng Wang
 1. zhengludwig.wordpress.com/projects/self-driving-rc-car/
 2. github.com/hamuchiwa/AutoRCCar
- by Ryan Zotti
 3. github.com/RyanZotti/Self-Driving-Car
- by Yazeed Alrubyli
 4. becominghuman.ai/building-self-driving-rc-car-series-5-serverless-control-using-computer-vision-fdf0d0136888
 5. github.com/yazeedalrubyli/SDRC
- by Multunus
 6. github.com/multunus/autonomous-rc-car



Useful materials

Guides and specifications:

L293 Motor Driver:

- ti.com/lit/ds/symlink/l293.pdf

HC-SR04 Sensor:

- components101.com/ultrasonic-sensor-working-pinout-datasheet
- tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/
- howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

Pi Camera:

- projects.raspberrypi.org/en/projects/getting-started-with-picamera
- opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html



**GDAŃSK UNIVERSITY
OF TECHNOLOGY**



**HISTORY IS WISDOM
FUTURE IS CHALLENGE**