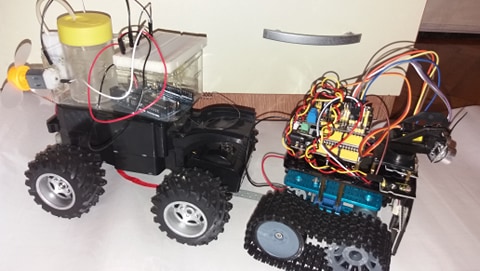
НАЦИОНАЛНА ПРОГРАМА "ОБУЧЕНИЕ ЗА ИТ КАРИЕРА" – БУРГАС

ПРОЕКТ

RoboCleaner

**Изготвили:** Ръководител:

Александър Маринов Петър Петров

Александър Станков

Бургас, 2020

# Съдържание

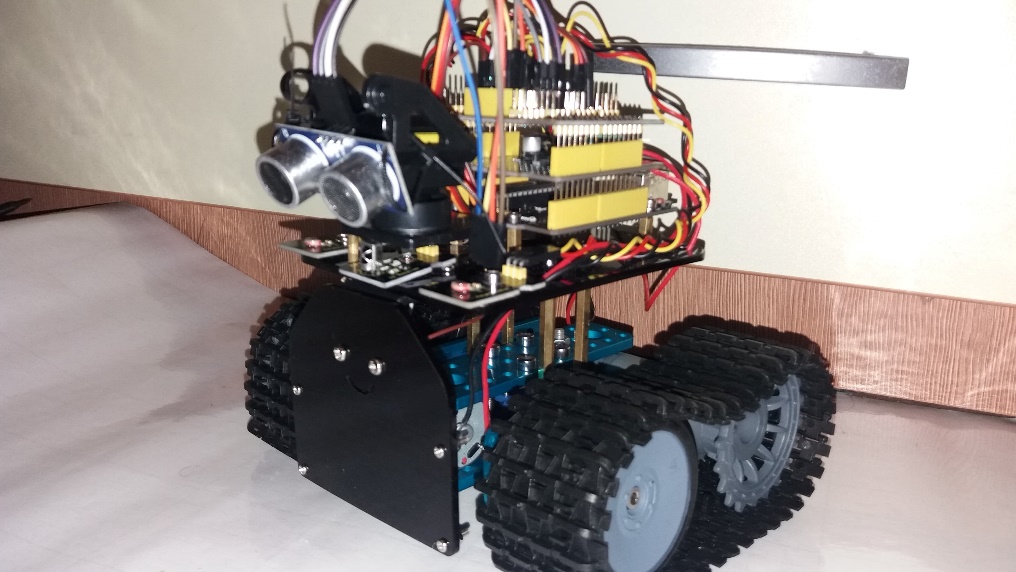
* Описание на проекта 3
* Необходими елементи 4
* Схема на свързване 5
* Програмиране на робота 6
* Код 10
* Заключение 12

# Описание на проекта

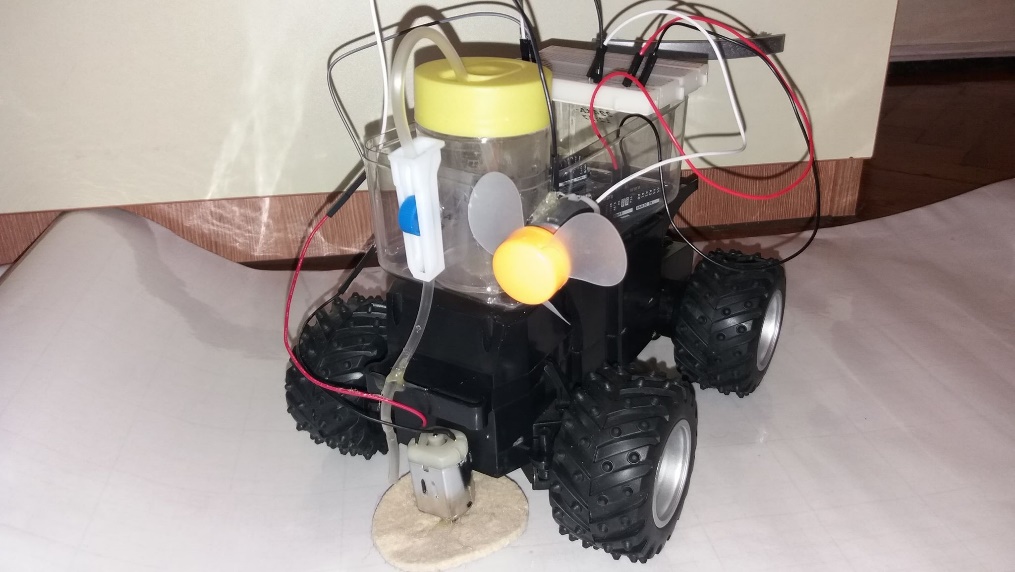
Проектът „ RoboCleaner“ е предназначен за дезинфекциране на замърсени повърхности. Съставен от два роботизирани елемента – единият за придвижване, а другият за почистване. Роботът танк се движи самостоятелно, като използва Ultrasonic Sensor, за да избягва препятствия. Той служи също така за придвижване и на другата част, чрез която се реализира почистването с помощта на два DC мотора – единият се използва за почистване на повърхностите, а другият за изсушаване.

## Начин на действие:

При включване на робота, танкът започва да се движи по замърсената повърхност. С помощта на система се подава дезинфекцираща течност към въртящо се кече, свързано с DC мотор. С друг DC мотор, на който е поставена перка, се реализира допълнително изсушаване на почистената повърхност. Когaто срещне препятствие, танкът спира, придвижва се леко назад и с помощта на серво двигател ултразвуковия сензор „се оглежда“ и търси посока без препятствия. Когато открие такава посока, с помощта на двигателите, задвижващи веригите, танкът се обръща и продължава в избраната посока.



*Танк*



*Чистач*

# Необходими елементи

**За танка:**

keyestudio UNO R3 Controller \* 1

keyestudio L298P Shield \* 1

keyestudio V5 Sensor Shield \* 1

HC-SR04 Ultrasonic Sensor \* 1

keyestudio Bluetoot h Module (HC-06) \* 1

Metal Holder \* 4

Plastic Platform \*1

Servo Motor (black)\* 1

Transparent Acrylic Board \* 1

Tank Driver Wheel \*2

Tank Load-bearing Black Wheel \* 2

Caterpillar Band (Black) \* 2

Metal Motor\* 2

Copper Coupler \* 2

18650 2-cell Battery Case \*1

Copper Bush \* 2

Flange Bearing \* 4

Hexagon Copper Bush (M3\*10MM) \*8

Hexagon Copper Bush (M3\*45MM) \* 4

M3\*10MM Flat Head Screw \* 4

Inner Hexagon Screw (M3\*8MM) \* 25

Inner Hexagon Screw (M3\*25MM) \* 6

Inner Hexagon Screw (M4\*35MM) \* 5

Inner Hexagon Screw (M4\*50MM) \* 3

Inner Hexagon Screw (M4\*12MM) \* 5

M3 Nut \* 4

M4 Self-locking Nut \* 3

M4 Nut \* 15

Black Wire(150mm) \*2

Red Wire(150mm) \*2

Black Wire(200mm) \*2

Black Wire(200mm) \*2

F-F Dupont Wire (20CM, 4Pin) \* 1

Supporting Part (27\*27\*16MM, Blue) \* 2

Winding Wire \* 1

Tapping Screw (M1.2\*5MM) \* 6

**За чистача:**

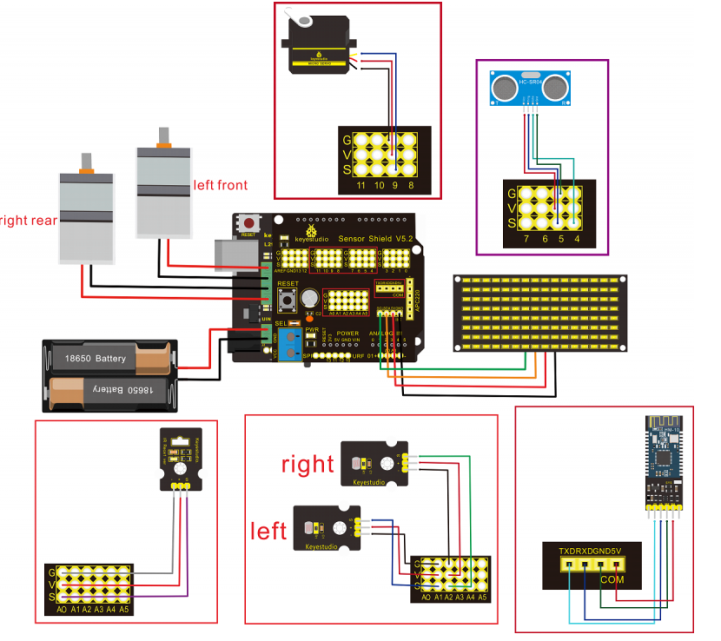
Arduino Uno R3 \* 1

Lipo Battery 32 mAh \* 1

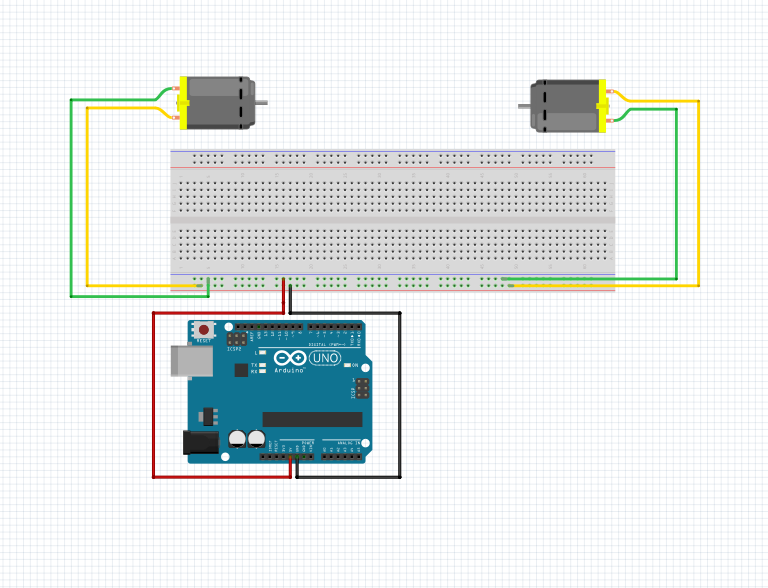
DC motor \* 2

Jumper cables \* 2

# Схеми на свързване



*Схема 1 – танк*

**

*Схема 2 – чистач*

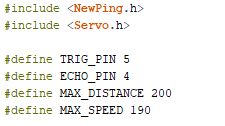
# Програмиране на робота

Задачата на робота е да тегли ремарке върху което има поставен контейнер с дезинфектант (или някакъв почистващ препарат) и тръбичка с устройство за регулиране на струята, която пуска върху парчето вътящ се плат докосвайки земята. Главното тяло на робота е програмиран да засича обекти и да избягва контакт с тях. Това се извършва с ултразвуков модул, който е прикрепен към servo мотор с който се оглежда наляво и надясно за да прецени към коя посока има по-голяма дистанция до обект и да продължи към нея.

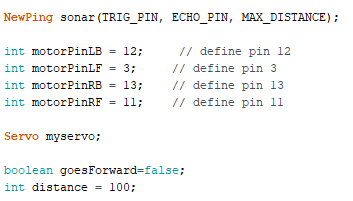
Използваме 2 готови библиотеки за в нашия код за по-точното изпълнение на дадени модули. Библиотеката NewPing се използва за да направи работата на модула HC-SR04 по-точна и без lag, който забелязахме че го има в други библиотеки за контролиране на ултразвукови модули, ако не се получи информация за дистанция от измерване.  
Втората e обикновена Servo библиотека за използване на мотора към който е прикачен ултразвуковия модул.  
  
Библиотеката е изтеглена от този github клон и е добавена към нашата Arduino lib директория.

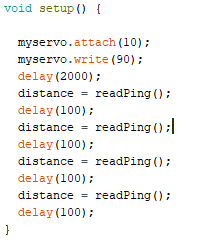
https://github.com/livetronic/Arduino-NewPing

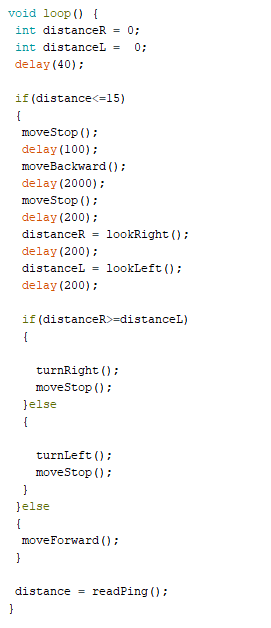
Започваме първо с добавянето на нужните ни библиотеки.   
След това задаваме пиновете на ултразвуковия модул, максималната дистанция за измерване и максималната скорост на DC моторите.



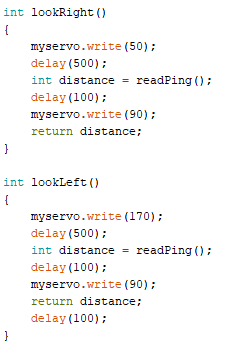
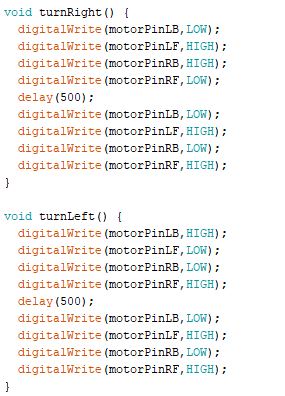
Казваме на ултразвуковия модул с кои стойности ще работи и след това задаваме пиновете на DC моторите.  
Декларираме че ще използваме servo мотор и слагаме дистанция за HC-SR04, с която ще оперира докато не засече обект пред себе си за ново измерване.  
Булевата goesForward е нужна за да определим дали мотора се движи напред за да може да изпълнява и други движения които ще обясним в методите за контрол.



В Setup() функцията казваме на кой пин сме свързали мотора и задаваме начални градуси към които да е обърнат.  
И след това чете дистанцията 4 пъти през 100 милисекунди.  


Следва Loop() функцията където има if проверка ако ще се сблъска с обект (по-малко от 15cm) да направи ход назад и да измери разстоянието наляво и надясно. След това да прецени по удачната посока в зависимост дали там има или няма обект.  


Накрая сме сложили прости методи за определяне на дистанцията и подаване на ток за правилното движение на DC моторите.

# Код

#include <NewPing.h>

#include <Servo.h>

#define TRIG\_PIN 5

#define ECHO\_PIN 4

#define MAX\_DISTANCE 200

#define MAX\_SPEED 190

NewPing sonar(TRIG\_PIN, ECHO\_PIN, MAX\_DISTANCE);

int motorPinLB = 12; // define pin 12

int motorPinLF = 3; // define pin 3

int motorPinRB = 13; // define pin 13

int motorPinRF = 11; // define pin 11

Servo myservo;

boolean goesForward=false;

int distance = 100;

void setup() {

Serial.begin(9600);

myservo.attach(10);

myservo.write(90);

delay(2000);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

distance = readPing();

delay(100);

}

void loop() {

int distanceR = 0;

int distanceL = 0;

delay(40);

if(distance<=15)

{

Serial.println("ln");

moveStop();

delay(100);

moveBackward();

delay(2000);

moveStop();

delay(200);

distanceR = lookRight();

delay(200);

distanceL = lookLeft();

delay(200);

if(distanceR>=distanceL)

{

turnRight();

moveStop();

}else

{

turnLeft();

moveStop();

}

}else

{

moveForward();

}

distance = readPing();

}

int lookRight()

{

myservo.write(50);

delay(500);

int distance = readPing();

delay(100);

myservo.write(90);

return distance;

}

int lookLeft()

{

myservo.write(170);

delay(500);

int distance = readPing();

delay(100);

myservo.write(90);

return distance;

delay(100);

}

int readPing() {

delay(70);

int cm = sonar.ping\_cm();

if(cm==0)

{

cm = 25;

}

return cm;

}

void moveStop() {

digitalWrite(motorPinLB,LOW);

digitalWrite(motorPinLF,LOW);

digitalWrite(motorPinRB,LOW);

digitalWrite(motorPinRF,LOW);

}

void moveForward() {

if(!goesForward)

{

goesForward=true;

digitalWrite(motorPinLB,LOW);

digitalWrite(motorPinLF,HIGH);

digitalWrite(motorPinRB,LOW);

digitalWrite(motorPinRF,HIGH);

}

}

void moveBackward() {

goesForward=false;

digitalWrite(motorPinLB,HIGH);

digitalWrite(motorPinLF,LOW);

digitalWrite(motorPinRB,HIGH);

digitalWrite(motorPinRF,LOW);

}

void turnRight() {

digitalWrite(motorPinLB,LOW);

digitalWrite(motorPinLF,HIGH);

digitalWrite(motorPinRB,HIGH);

digitalWrite(motorPinRF,LOW);

delay(500);

digitalWrite(motorPinLB,LOW);

digitalWrite(motorPinLF,HIGH);

digitalWrite(motorPinRB,LOW);

digitalWrite(motorPinRF,HIGH);

}

void turnLeft() {

digitalWrite(motorPinLB,HIGH);

digitalWrite(motorPinLF,LOW);

digitalWrite(motorPinRB,LOW);

digitalWrite(motorPinRF,HIGH);

delay(500);

digitalWrite(motorPinLB,LOW);

digitalWrite(motorPinLF,HIGH);

digitalWrite(motorPinRB,LOW);

digitalWrite(motorPinRF,HIGH);

}

# Заключение

RoboCleaner може да се използва за почистване на замърсени повърхности. Наличието на втората платка Arduino Uno дава възможност да се задават различни режими на работа на двигателите на „чистача“, както и да се добавят различни други функционалности като например меренето на температурата и влажността в почистваното помещение.