**I. Съдържание**

I. Съдържание  
II. Увод  
III. Преглед и подбор на елементната база  
 1. Arduino Uno микроконтролер  
 2. TL1838 Infrared Receiver

3. Ultrasonic Ranging Module HC - SR04

4. Gear мотори

5. Remote

6. 9V Battery  
IV. Реализация на устройството  
 1. Блок схеми на Hardware  
 2. Блок схеми на Software   
 3. Принципна електрическа схема

4. Софтуер - Код  
V. Същинска реализация на симулацията  
 1. Трудности   
 2. Успехи и неуспехи  
VI. Икономически аспект  
VII. Бъдещо развитие

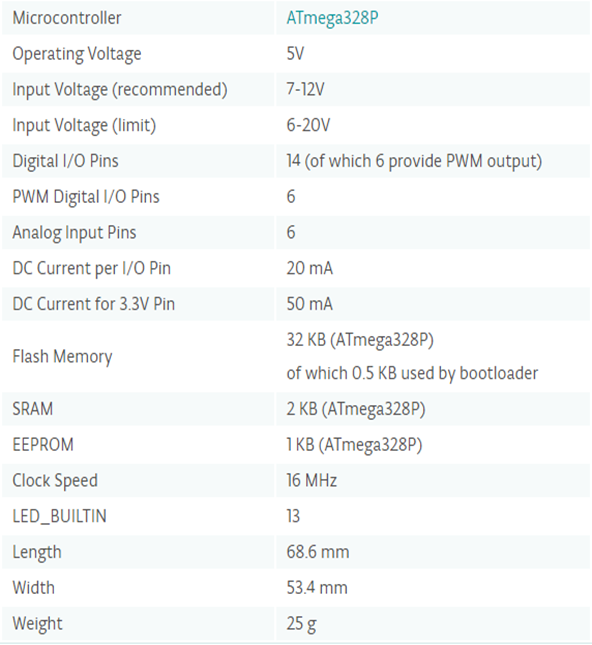
**II. Увод**

Този проект е сумо робот, чиято цел е да избута вражески машини извън предварително начертан кръг на земята. Той може да се контролира автоматично сам себе си чрез изключително прост AI(Artificial Intelligence) или така наречения Auto-Mode. В случаи когато се изисква по-голяма прецизност се използва Manual-Mode(Ръчен режим) чрез помощта на дистанционен предавател, който има 6 дефинирани бутона. Четири бутона за контролиране на посоката на движение, един бутон за спиране на всякакво движение и един бутон за смяна на режимите на работа.

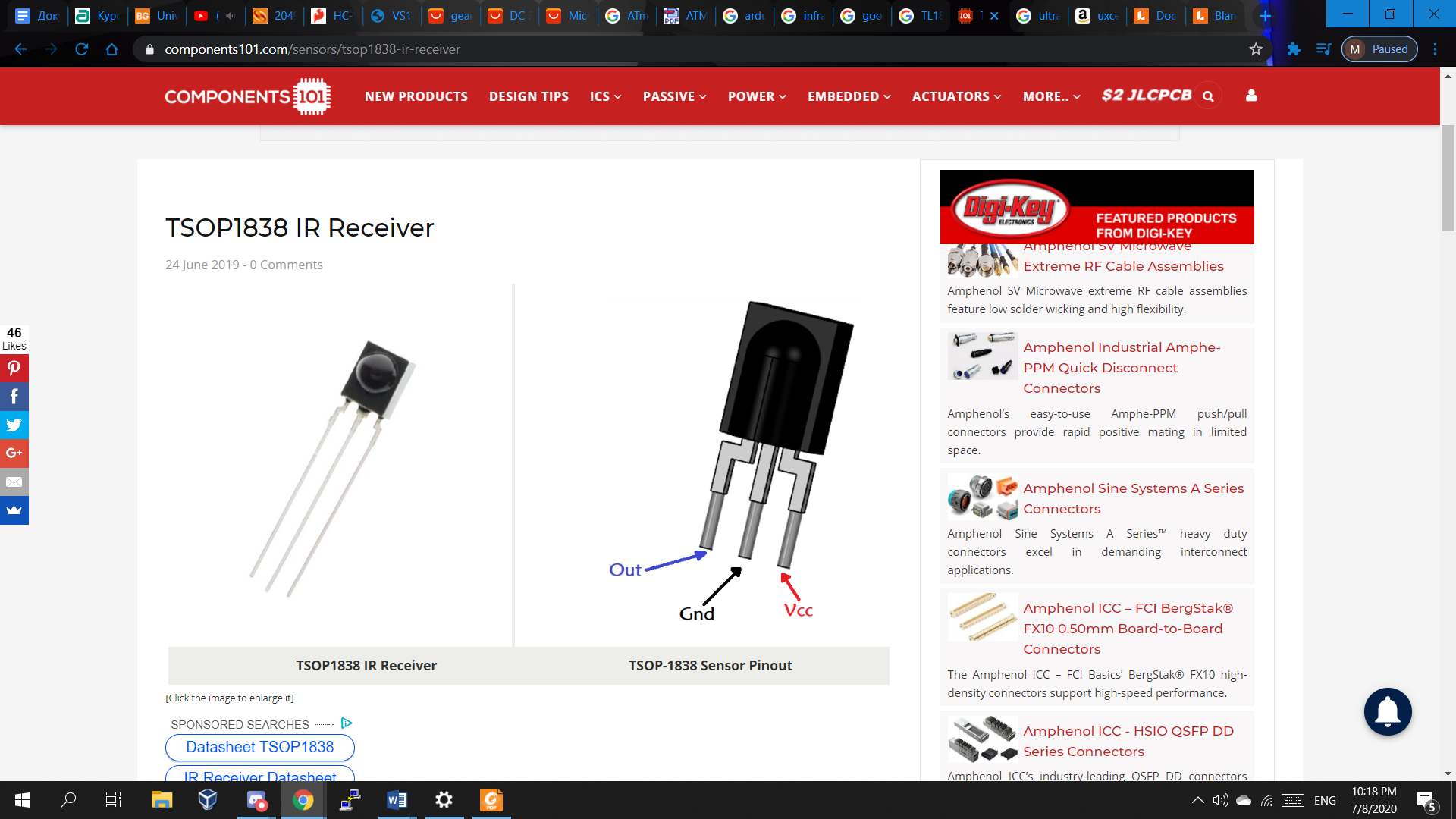
**III. Преглед и подбор на елементната база**

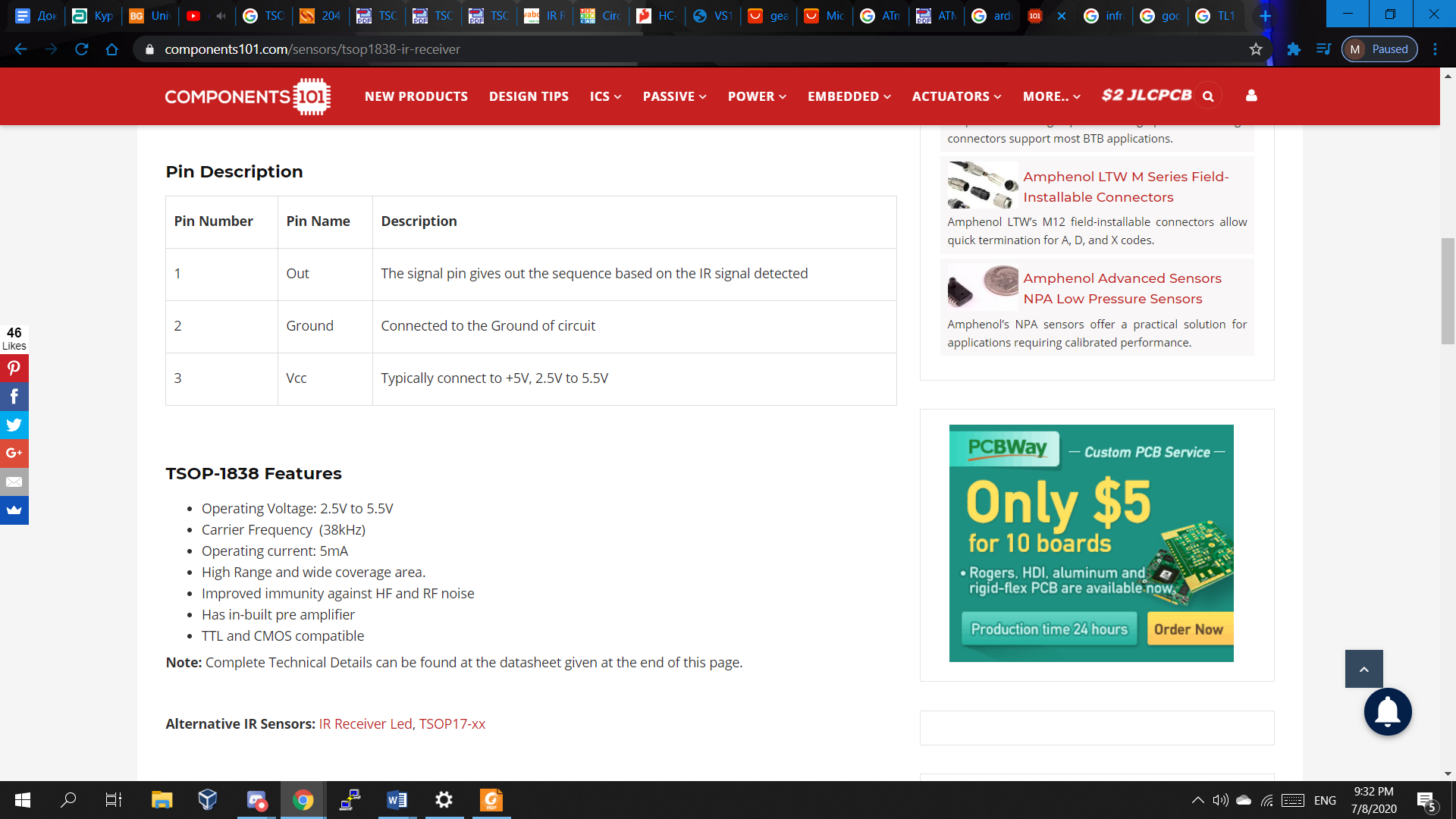
**1. Arduino Uno – Микроконтролер**, който e базиран е на процесора: ATmega328. Arduino-Uno има 14 входно/изходни пина, 6 аналогови входа, 4 серийна входа, 16MHz кварцов кристал, USB връзка, бутон за рестартиране и бутон за изтриване.



**Характеристики на процесора:**

**2. TSOP-1838 IR Receiver** – сензор за инфрачервени лъчи. Има 3 пина - VCC, GND и Out.

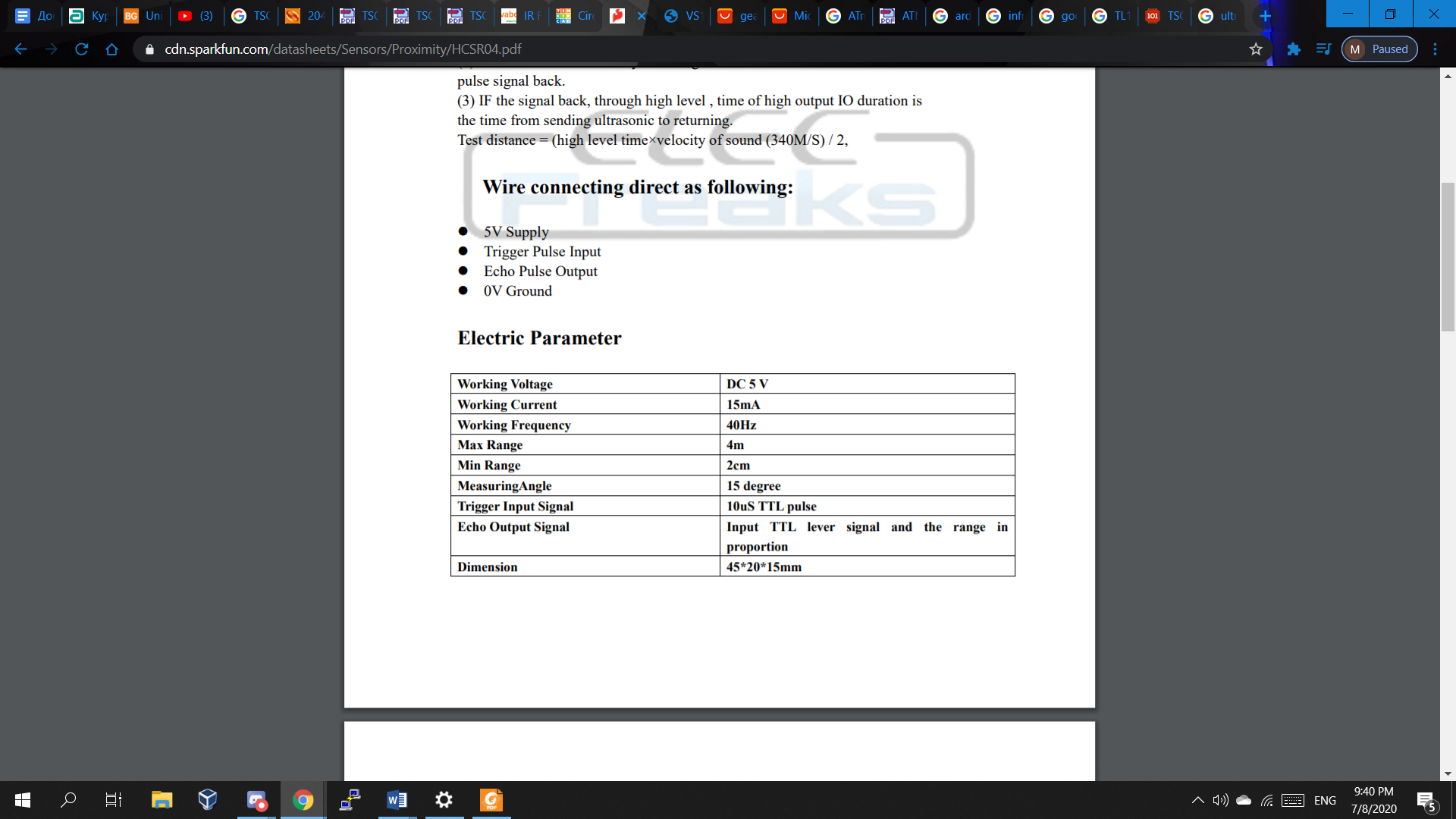


**Характеристики на IR сензора:** ****

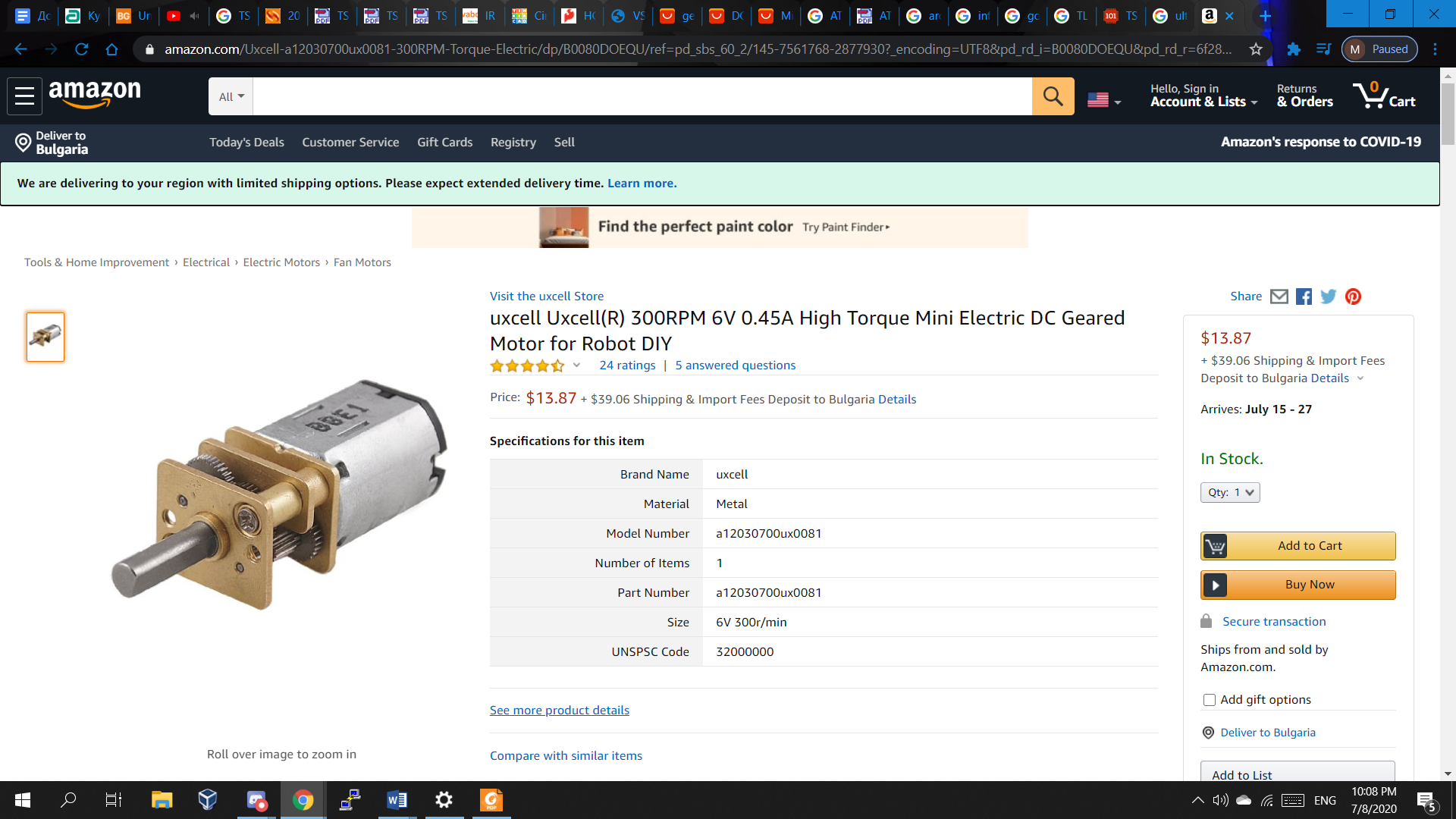
**3. Ultrasonic Ranging Module HC - SR04** - ултразвуков сензор, който осигурява 2-400cm функционалност за измерване на разстояние до обект с точност до 3mm.



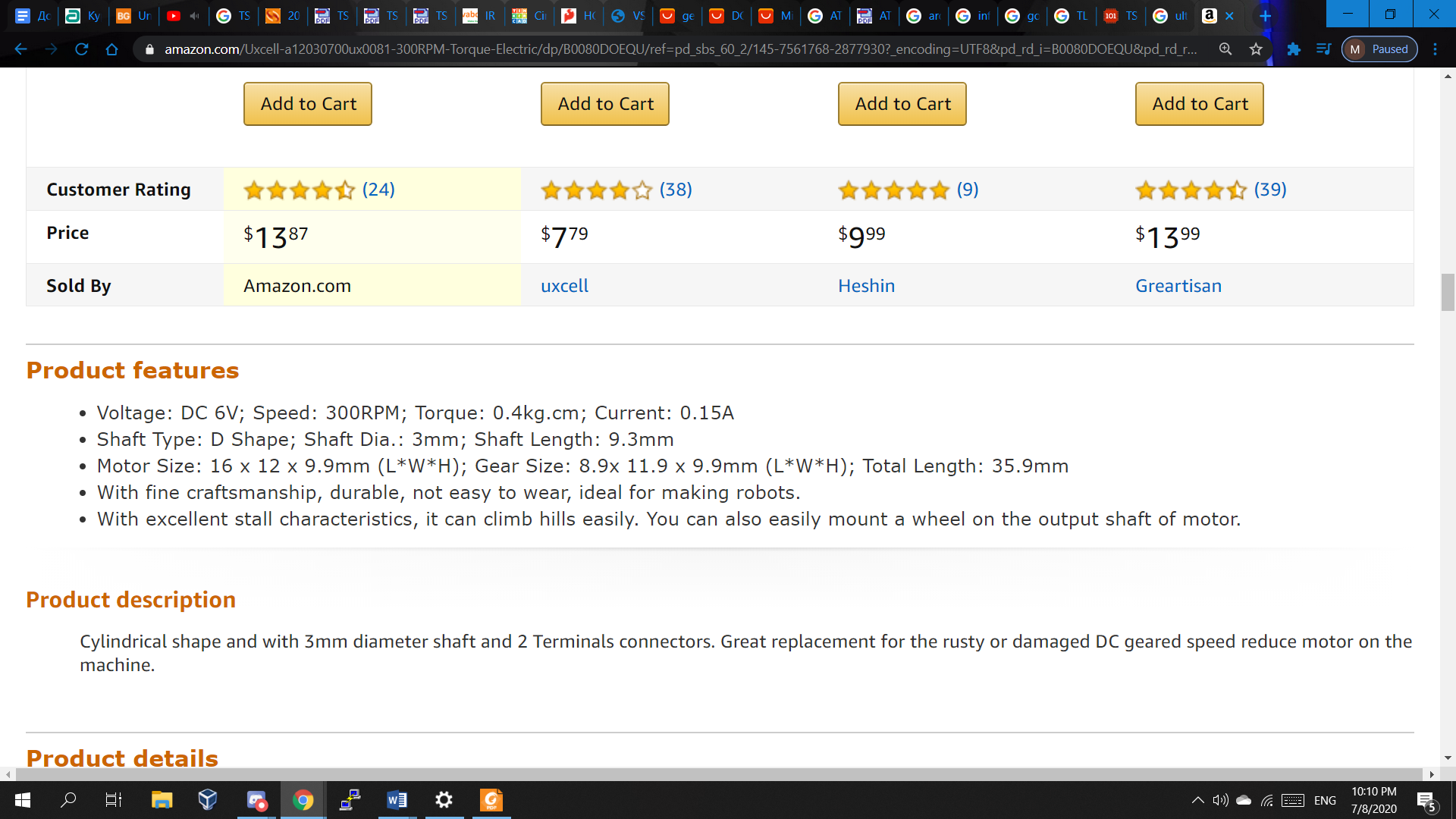
**Характеристики на ултразвуковия сензор:**

****

**4. Gearmotors -** uxcell Uxcell(R) DC Gear мотор с 6V захранване, 300RPM с висока степен на въртящ момент.

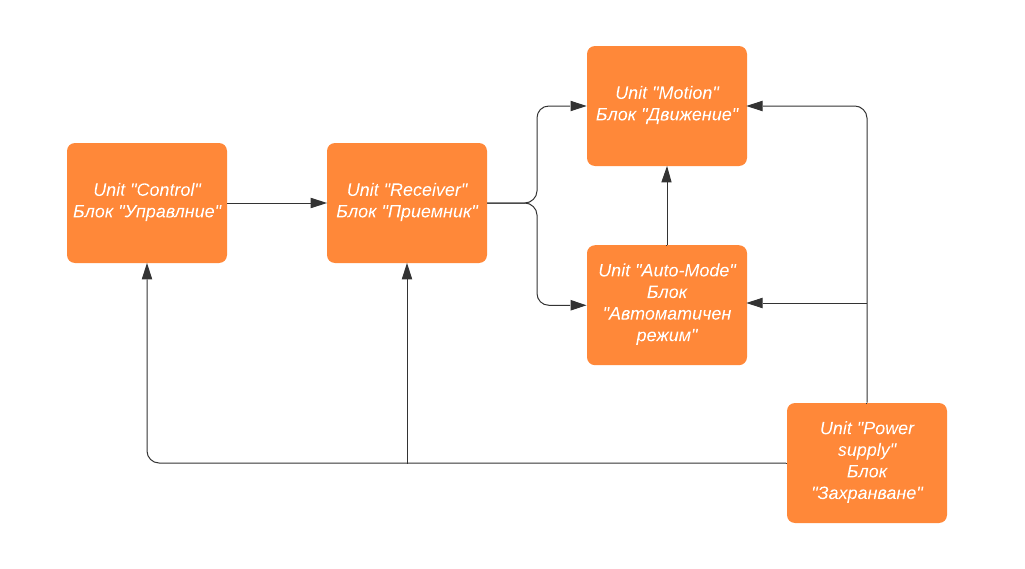


**Характеристики на мотора:**



**IV. Реализация на устройството**

**1. Блок схема на Хардуера**

****

Блок „Управление” – Реализиран, чрез Remote. През Remote-a се изпращат команди, които се приемат от блок “Приемник”.

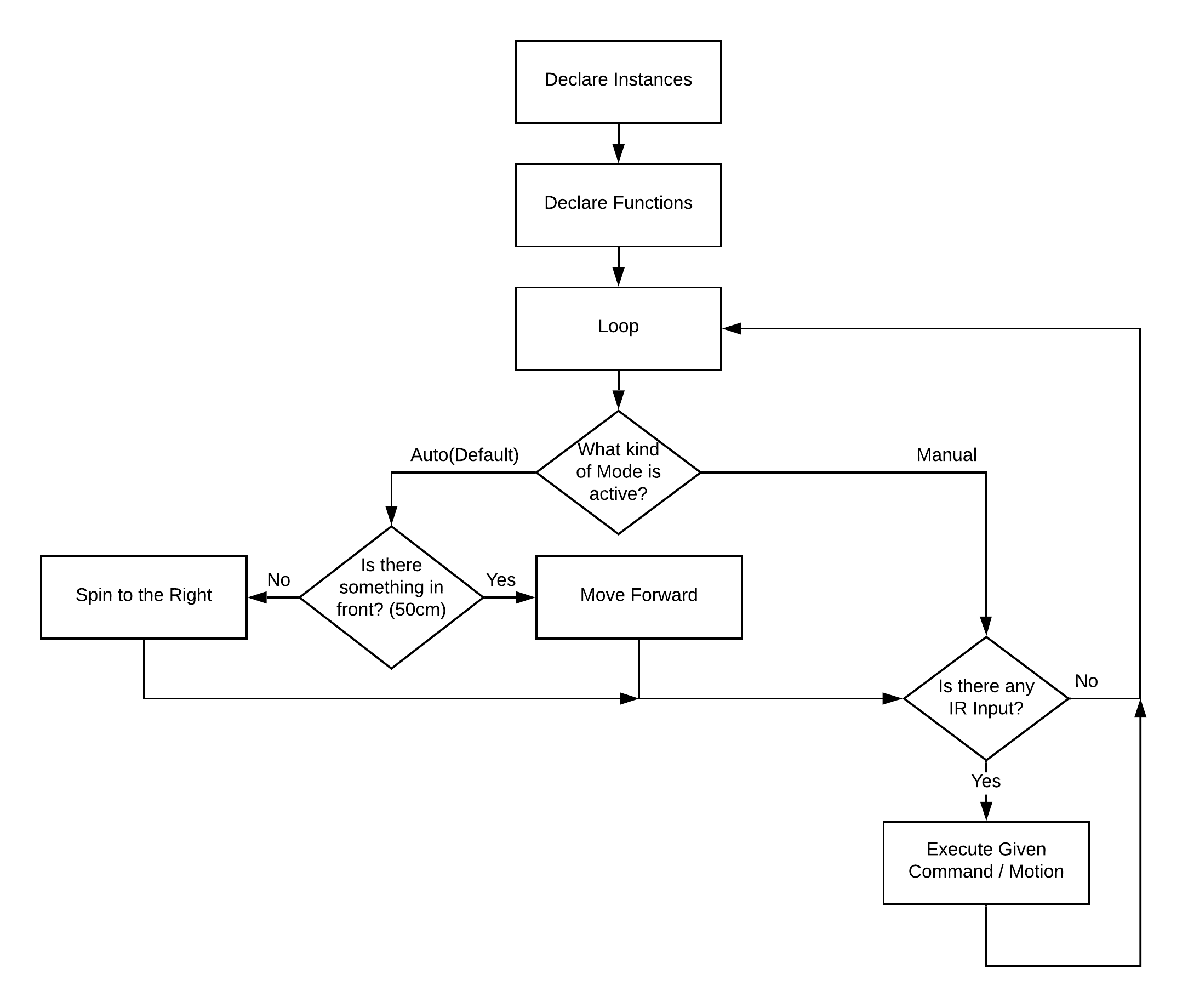
Блок “Приемник” - Реализиран чрез TL1838 Infrared Receiver. Той приема команди от блок “Управление”.

Блок „Auto-Mode” – Реализиран, чрез Ultrasonic Ranging Module HC - SR04. Сензора опитва да засече обект на разстояние 50 cm, като ако не намери такъв, цялото устройство започва да се върти.

Блок “Движение” - Реализиран, чрез Arduino Uno и 2 Gear мотори, като микроконтролера контролира движението на моторите

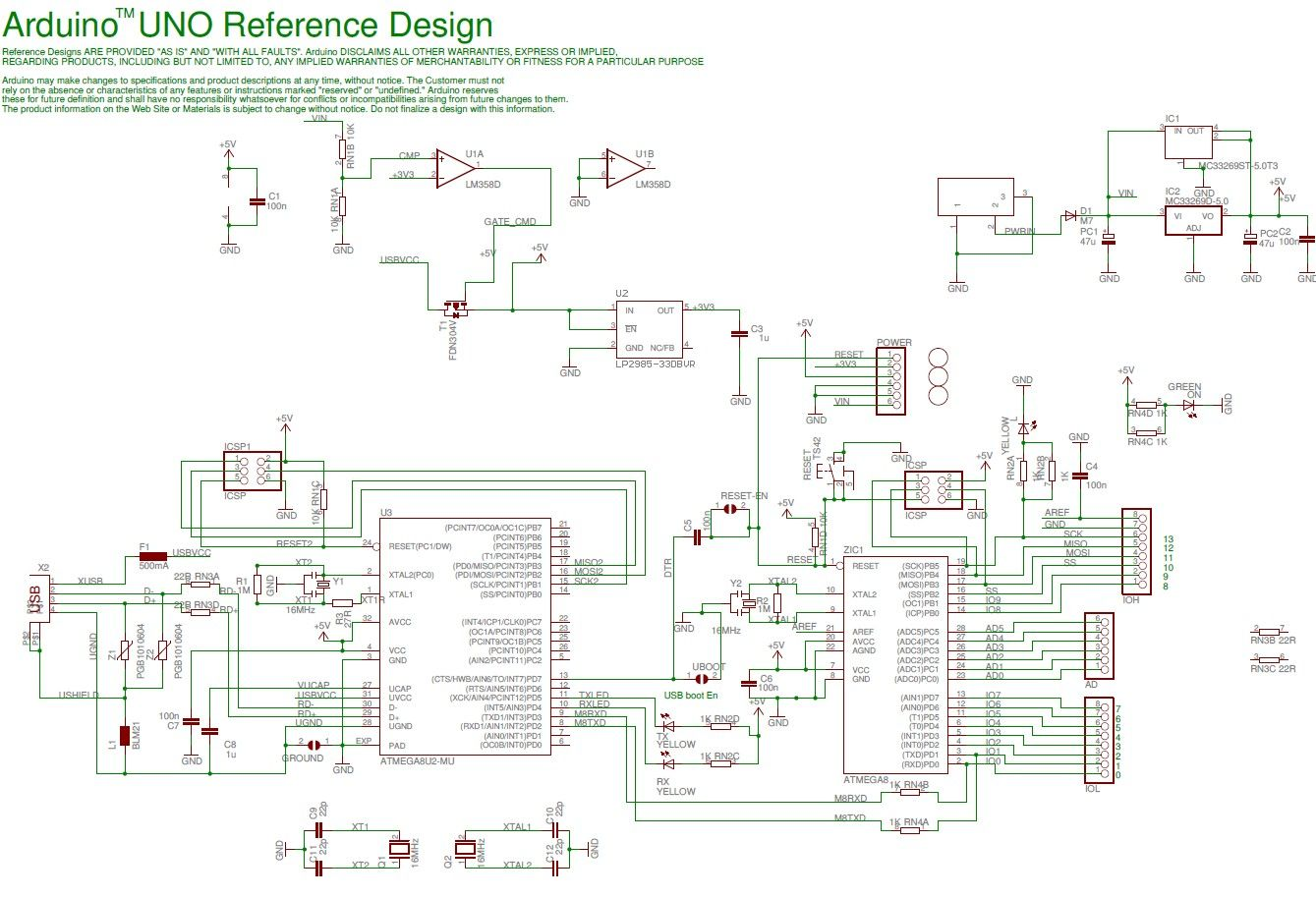
Блок „Захранване” – Подава захранване на цялата схема.

**2. Блок схема на кода**

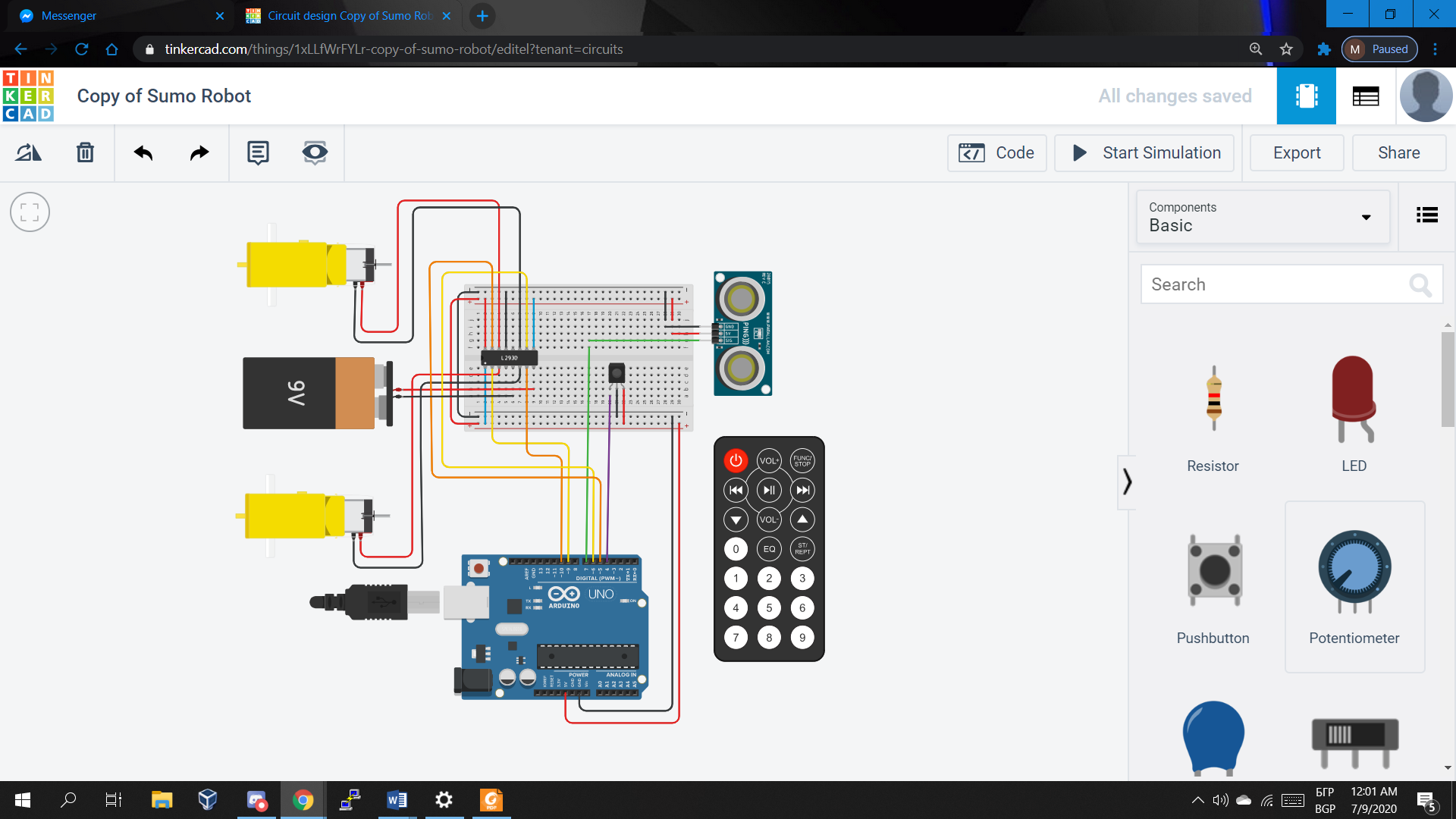


**3. Принципни Схеми**

**На Arduino Uno микроконтролер**



**На целия проект:**



**4. Софтуер - Код**

#include <IRremote.h>

#define RECV\_PIN 4

//Motor A

#define motorPin1 5

#define motorPin2 6

//Motor B

#define motorPin3 10

#define motorPin4 9

#define trigger 7

#define echo 7

int mode = 1;

int cm = 0;

int forwardBool = 0;

int rightBool = 0;

int reverseBool = 0;

int leftBool = 0;

int stopBool = 0;

//Declare instances of objects where we keep the results

IRrecv irrecv(RECV\_PIN);

decode\_results results;

long readUltrasonicDistance(int triggerPin, int echoPin)

{

pinMode(triggerPin, OUTPUT); // Clear the trigger

digitalWrite(triggerPin, LOW);

delayMicroseconds(2);

// Sets the trigger pin to HIGH state for 10 microseconds

digitalWrite(triggerPin, HIGH);

delayMicroseconds(10);

digitalWrite(triggerPin, LOW);

pinMode(echoPin, INPUT);

// Reads the echo pin, and returns the sound wave travel time in microseconds

return pulseIn(echoPin, HIGH);

}

void right(){

digitalWrite(motorPin1, HIGH);

digitalWrite(motorPin2, LOW);

digitalWrite(motorPin3, HIGH);

digitalWrite(motorPin4, LOW);

}

void left(){

digitalWrite(motorPin1, LOW);

digitalWrite(motorPin2, HIGH);

digitalWrite(motorPin3, LOW);

digitalWrite(motorPin4, HIGH);

}

void forward(){

digitalWrite(motorPin1, HIGH);

digitalWrite(motorPin2, LOW);

digitalWrite(motorPin3, LOW);

digitalWrite(motorPin4, HIGH);

}

void reverse(){

digitalWrite(motorPin1, LOW);

digitalWrite(motorPin2, HIGH);

digitalWrite(motorPin3, HIGH);

digitalWrite(motorPin4, LOW);

}

void stop(){

digitalWrite(motorPin1, LOW);

digitalWrite(motorPin2, LOW);

digitalWrite(motorPin3, LOW);

digitalWrite(motorPin4, LOW);

}

void setup()

{

Serial.begin(9600);

pinMode(motorPin1, OUTPUT);

pinMode(motorPin2, OUTPUT);

pinMode(motorPin3, OUTPUT);

pinMode(motorPin4, OUTPUT);

irrecv.enableIRIn(); // start the receiver

}

void loop()

{

// measure the cm

//cm is equal to the time (in microseconds) that it takes for the sound to send and then go back again.

//since the sound travels the distance twice, we then need to divide it by 2

//speed of sound is aprox. 340m/s which is 0.034cm/us

//0.034us divided by 2 (as stated above) is 0.017

//in order to get the distance we need to multiply the time by 0.017

cm = 0.017 \* readUltrasonicDistance(trigger, echo);

Serial.println(cm);

//do auto if mode is turned on

//[mode = 1 -> auto]

//[mode = 0 -> manual]

if(cm <= 50 && mode == 1){

//forward

forwardBool = 1;

reverseBool = 0;

leftBool = 0;

rightBool = 0;

stopBool = 0;

}

else if(mode == 1){

//right

forwardBool = 0;

reverseBool = 0;

leftBool = 0;

rightBool = 1;

stopBool = 0;

}

//execute the commands if a button is pressed

if(forwardBool == 1){

forward();

}

if(reverseBool == 1){

reverse();

}

if(rightBool == 1){

right();

}

if(leftBool == 1){

left();

}

if(stopBool == 1){

stop();

}

//prints the mode we are currently using (1 = auto, 0 = manual)

Serial.println(mode);

//IR remote

//decode results from remote

//IF we have received an IR signal, then do:

if (irrecv.decode(&results)){

Serial.println(results.value, HEX);

irrecv.resume(); //receive next value

//if button 2 is pushed, go forward

if(results.value==0xFD8877 && mode == 0){

Serial.println("forward");

forwardBool = 1;

reverseBool = 0;

leftBool = 0;

rightBool = 0;

stopBool = 0;

}

//if button 4 is pushed, go left

if(results.value==0xFD28D7 && mode == 0){

Serial.println("left");

forwardBool = 0;

reverseBool = 0;

leftBool = 1;

rightBool = 0;

stopBool = 0;

}

//if button 6 is pushed, go right

if(results.value==0xFD6897 && mode == 0){

Serial.println("right");

forwardBool = 0;

reverseBool = 0;

leftBool = 0;

rightBool = 1;

stopBool = 0;

}

//if button 8 is pushed, go backwards

if(results.value==0xFD9867 && mode == 0){

Serial.println("reverse");

forwardBool = 0;

reverseBool = 1;

leftBool = 0;

rightBool = 0;

stopBool = 0;

}

//if button 5 is pushed, stop all movements

if(results.value==0xFDA857 && mode == 0){

Serial.println("stop");

forwardBool = 0;

reverseBool = 0;

leftBool = 0;

rightBool = 0;

stopBool = 1;

}

//if ON/OFF button is pushed, change auto modes

if(results.value==0xFD00FF){

Serial.println("switch mode");

forwardBool = 0;

reverseBool = 0;

leftBool = 0;

rightBool = 0;

stopBool = 1;

if(mode == 1) { mode = 0; }

else { mode = 1; }

}

}

}

**V. Същинска реализация на симулацията**

**1. Трудности при разработката на проекта**

Трудностите, които срещнахме при изработката на проекта са:

1.1 Ограниченост на елементите

TinkerCAD има ограничен брой елементи за използване. Първоначално желаехме да направим този робот с bluetooth управление, но поради този проблем трябваше да се сменим към инфрачервена технология.

1.2. Приемане на информацията от дистанционния предавател

Приемане на информацията от дистанционното бе голям проблем поради това, че не знаехме какъв вид hexadecimal инфрачервена информация излъчва. Този проблем беше разрешен с Serial.println(), който показва hexadecimal числото което се изпраща при натискане на даден бутон от дистанционния предавател.

**2. Успехи и неуспехи**

Успешно са реализирани:  
 - Подбор на елементна база  
 - Свързаност на елементите  
 - Програмиране на микроконтролера  
 - Запазена е идеята и тематиката

Неуспехи:  
 - Имплементиране на Bluetooth комуникация между машината и външно устройство.

**VI. Икономически Аспект**

1. Arduino Uno микроконтролер - $2.50;  
 2. TL1838 Infrared Receiver - $0.35/10 pieces;

3. Ultrasonic Ranging Module HC - SR04 - $0.70;

4. Gear мотори - $13.87\*2;

5. Remote - $0.50;

6. 9V Battery - $0.97;

Обща цена - $32.44;

# 

**VII. Бъдещо развитие**

**Бъдещото развитие на проекта може да се разгледа в няколко точки.**

1. Физическа реализация

2. Увеличаване на мощността на моторите

3. Подобряване на автоматичния и ръчния режим на работа

4. Допълнително форматиране на кода за по-голяма производителност