Documentation

Sas-Miresan Andrei

**Summary:**  
In this project I created a robot which through a bluetooth module it is controlled and has two modes:

* A manually controlled robot/car through a bluetooth app.
* An obstacle avoiding robot which is able to avoid any obstacle which it faces when it moves. Simply, when it meets an obstacle (while moving forward) it automatically stops moving forward and takes a step back. Then it looks to its two sides (left & right) and starts to move in another direction: either in left direction if there is another obstacle in right or in right direction if there is another obstacle in left side.

**Components:**

1. Two DC Motors

DC Motors are the best option for movement as they have power and are cheap.

1. L298n motor driver

It was used as it offers complete control of the forward and backward movement of the 2 connected motors.

1. Ultrasonic Sonar Sensor

It was used as it is the best type of sensor to detect objects at a resonable distance compared to infrared sensors or other options.

1. Tower Pro micro servo

It was used as it was a cheap and easy to obtain servo motor.

1. Bluetooth module HC-05

Is easier to use to control the robots movements compared to the wifi alternative.

1. Arduino MEGA

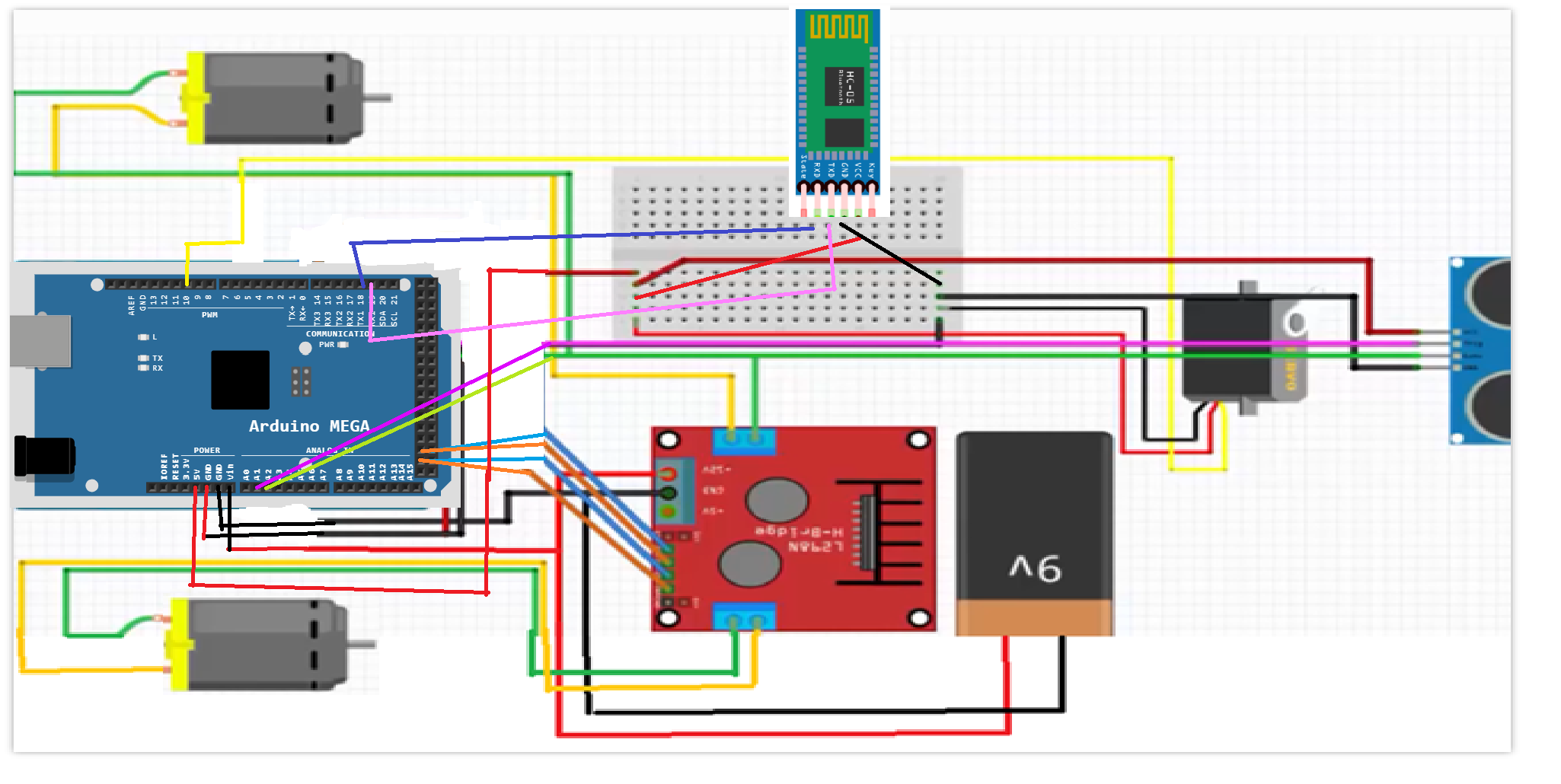
Initially the project was designed for Arduino UNO but for ease of use when paired to an external communication module MEGA is a better option.

1. 9V Battery

To keep the robot on even after it has been disconnected from the computer and for the motors to have power to move the whole robot

1. Wires
2. Breadboard

**Schematic:**



**Program explanation:**

First in the setup part we fix each output and input to the desired pins and set the communication path of the Arduino to the bluetooth on Serialize1.

After that the main loop checks continuosly for input from the bluetooth to see if it changes the mode the robot is functioning in and then calling the corresponding function to the mode it runs on.

The controlled function checks what input gets from the bluetooth and then calls the function corresponding to the direction selected to start the motors.

The autonomus function runs the motors on forward as long as the ultrasonic sensor doesn’t detect anything too close after that it stops the robot and makes him turn to left or right depending on which direction it detects more free space.  
  
**Code:**

#include <Servo.h>

#include <NewPing.h>

const int LeftMotorForward = 51;

const int LeftMotorBackward = 50;

const int RightMotorForward = 53;

const int RightMotorBackward = 52;

#define trig\_pin A1

#define echo\_pin A2

#define max\_distance 200

boolean goesForward = false;

int distance = 100;

boolean controlMode = true;

NewPing sonar(trig\_pin, echo\_pin, max\_distance);

Servo servo\_motor;

void setup() {

Serial.begin(9600); // Serial 0 interface for PC

Serial1.begin(9600); // Serial 1 interface for Bluetooth module

pinMode(LeftMotorForward, OUTPUT);

pinMode(LeftMotorBackward, OUTPUT);

pinMode(RightMotorForward, OUTPUT);

pinMode(RightMotorBackward, OUTPUT);

servo\_motor.attach(10);

servo\_motor.write(90);

delay(2000);

distance = readPing();

delay(100);

distance = readPing();

}

void loop() {

char mode = 'c';

if (Serial1.available()) { // Read from Bluetooth and send to PC

mode = Serial1.read();

if (mode == 'a')

controlMode = false;

else if (mode == 'c')

controlMode = true;

}

if (controlMode)

controled();

else

autonomus();

}

void controled() {

char comand = '0';

if (Serial1.available()) { // Read from Bluetooth and send to PC

comand = Serial1.read();

}

if (comand == '0')

moveStop();

if (comand == '1')

moveForward();

if (comand == '2')

moveBackwards();

if (comand == '3')

turnRight();

if (comand == '4')

turnLeft();

delay(100);

}

void autonomus() {

//Serial.write("autonom\n");

int distanceRight = 0;

int distanceLeft = 0;

delay(50);

if (distance <= 20) {

moveStop();

delay(300);

moveBackwards();

delay(400);

moveStop();

distanceRight = LookRight();

delay(300);

distanceLeft = LookLeft();

delay(300);

if (distance >= distanceLeft) {

turnRight();

moveStop();

}

else {

turnLeft();

moveStop();

}

}

else {

moveForward();

}

distance = readPing();

}

int readPing() {

delay(70);

int cm = sonar.ping\_cm();

if (cm == 0)

cm = 250;

return cm;

}

int LookLeft() {

servo\_motor.write(170);

delay(500);

int distance = readPing();

delay(100);

servo\_motor.write(90);

return distance;

}

int LookRight() {

servo\_motor.write(10);

delay(500);

int distance = readPing();

delay(100);

servo\_motor.write(90);

return distance;

}

void moveStop() {

digitalWrite(LeftMotorForward, LOW);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

}

void moveForward() {

//Serial.write("Front\n");

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(RightMotorBackward, LOW);

}

void moveBackwards() {

//Serial.write("Back\n");

digitalWrite(LeftMotorForward, LOW);

digitalWrite(LeftMotorBackward, HIGH);

digitalWrite(RightMotorForward, LOW);

digitalWrite(RightMotorBackward, HIGH);

}

void turnLeft() {

//Serial.write("Right\n");

digitalWrite(LeftMotorForward, LOW);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorForward, HIGH);

digitalWrite(RightMotorBackward, LOW);

}

void turnRight() {

//Serial.write("Left\n");

digitalWrite(LeftMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);

digitalWrite(RightMotorForward, LOW);

digitalWrite(RightMotorBackward, LOW);

}