Sensor Systems Lab Project

Remote Controllable

Obstacle Avoiding Mobile Robot

Group G8

Aytekin Dönmez 190412029

Umut Korkmaz 190412071

Berkehan Akyüz 190412052

Introduction

In this project, we tried to build a remote control robot platform. The robot platform has a 2 wheel and we used 6V DC gear motor to generate power to these wheels. Also, the platform has a caster wheel on the front to steer the robot and the other wheels attached to at the back side of chassis of the robot. For the chassis, we used 3D printer. The chassis carries the electrical components. In addition, there is gap at the middle of the chassis to get rid of wiring mess.

Components

DC Gear Motor & Wheel

A gear motor is an all-in-one combination of a motor and gearbox. The addition of a gear head to a motor reduces the speed while increasing the torque output

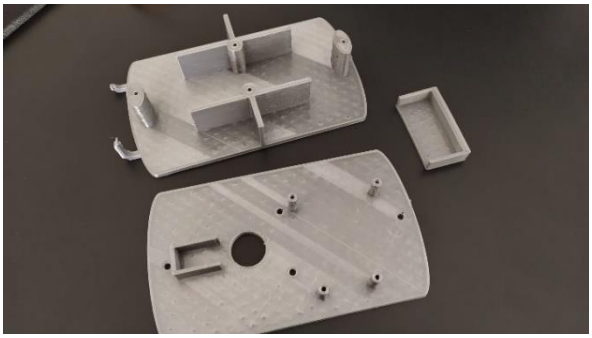
HC- SR04 Ultrasonic Sensor

The ultrasonic sensor works on the principle of SONAR and RADAR system which is used to determine the distance to an object. An ultrasonic sensor generates the high-frequency sound (ultrasound) waves.

Arduino

Arduino is an open-source electronics platform that is based on beginner-level hardware and software. The hardware component of an Arduino board is a programmable circuit board that is also known as a microcontroller

Chassis



We used PLA for the chassis material.

L298N Motor Shield

L298 2Amp Motor Driver Shield for Arduino is based on L298 motor driver integrated circuit, a full- bridge motor driver. It can drive two seperate 2A DC motors or 1 2A step motor. Motor's velocity and directions can be controlled separately and also there are 6 connectors connected to Arduino analog pins.

HM10 BLUETOOTH MODULE

The HM10 is a Bluetooth 4.0 module and HM10 is ideal for creating simple connections and it is easy to use.

Chester Wheel

A caster is an undriven wheel that is designed to be attached to the bottom of a larger object to enable that object to be moved.

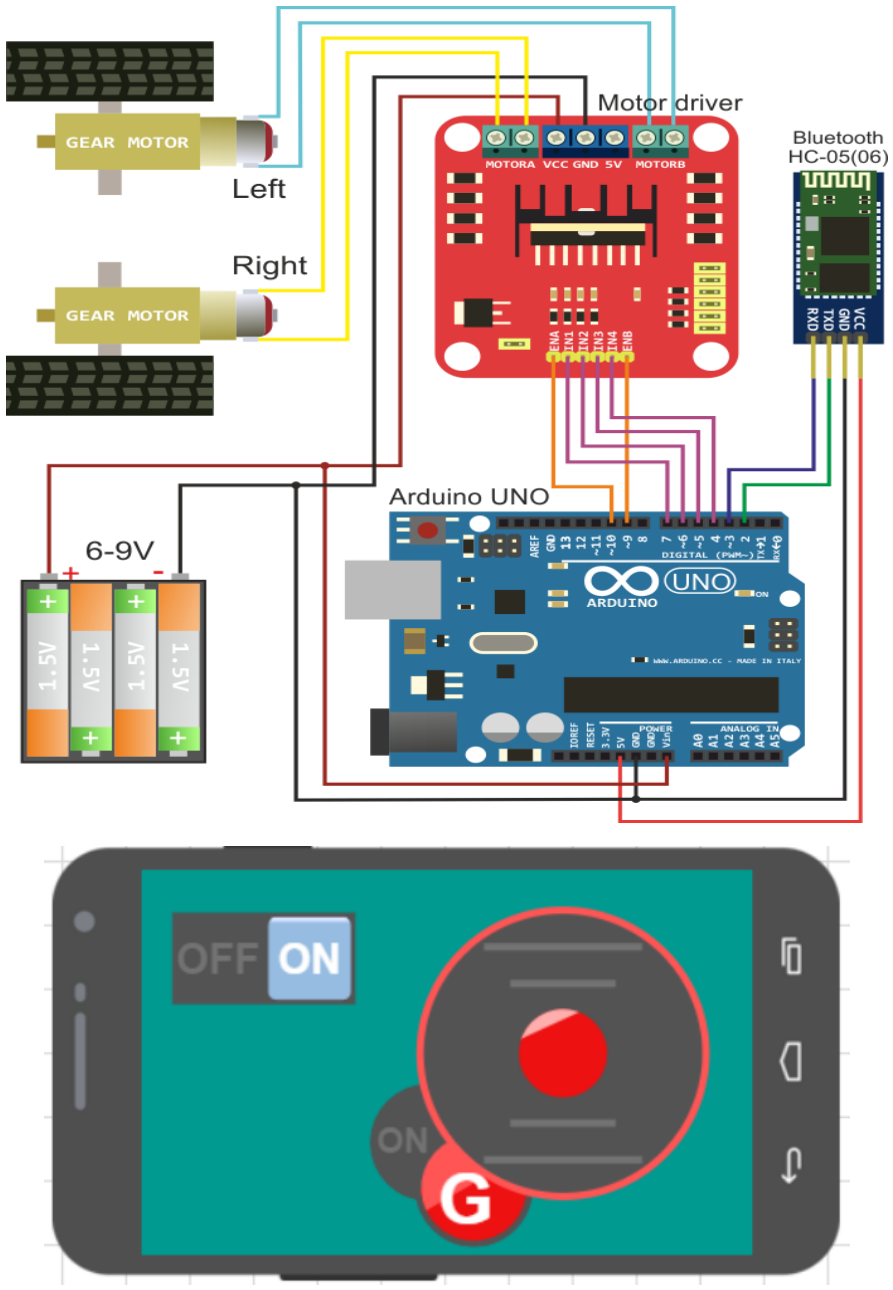
Lithium-Ion Battery

Lithium-ion batteries are more effective and prevalent than lithium-polymer batteries.

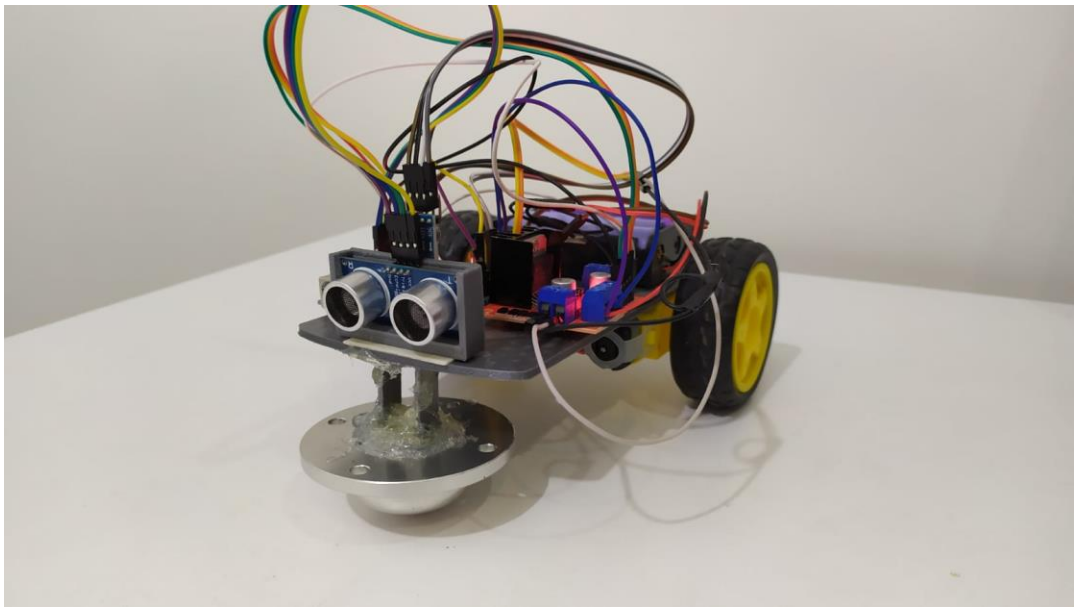
Working Principle of The Robot Platform

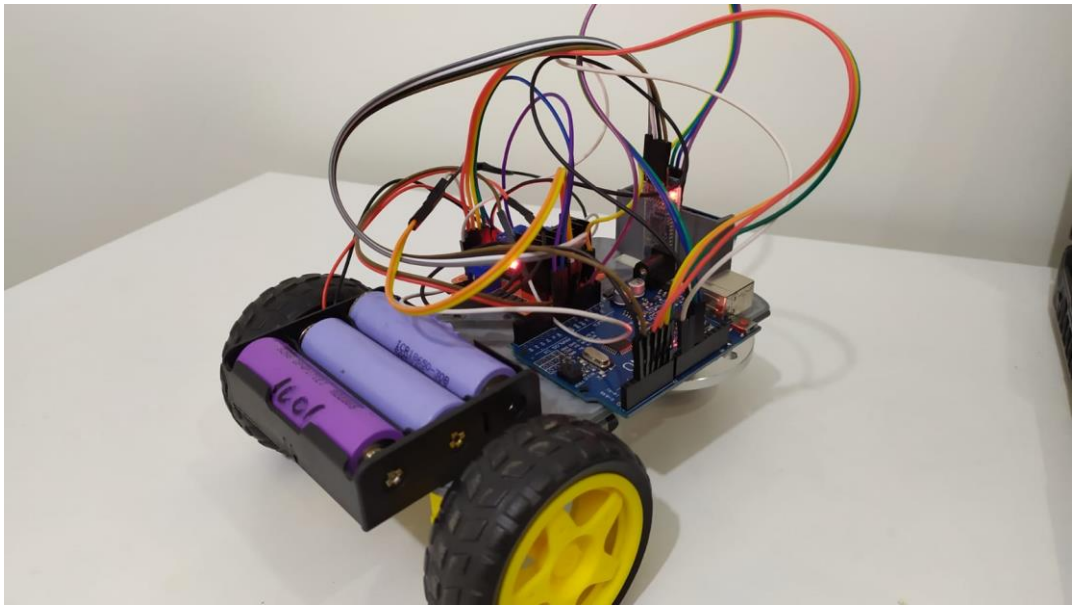
DC motors on the mobile platform are controlled by L298N Arduino Shield. We connected a bluetooth module to Arduino Shield to receive this data from the Remote XY program. We chose Remote XY because it's easy to use and customizable. We put 2 button as a choice on the screen to control the robot. It can be controlled by the user with the joystick or Gyro sensor could be active by using a on-off button. Also, The robot determines a surface it encounters according to the data it receives from the ultrasonic sensor and stops the motors from running.

Circuit Diagram



FINAL DESİGN





Arduino Code

```
/* RemoteXY select connection mode and include library */
#define REMOTEXY_MODE__SOFTWARESERIAL
#include <SoftwareSerial.h>
#include <RemoteXY.h>

/* RemoteXY connection settings */
#define REMOTEXY_SERIAL_RX 2
#define REMOTEXY_SERIAL_TX 3
#define REMOTEXY_SERIAL_SPEED 9600

/* RemoteXY configurate  */
unsigned char RemoteXY_CONF[] =
  { 3,0,23,0,1,5,5,15,41,11
  ,43,43,1,2,0,6,5,27,11,5
  ,79,78,0,79,70,70,0 };

/* this structure defines all the variables of your control interface */
struct {

    /* input variable */
  signed char joystick_1_x; /* =-100..100 x-coordinate joystick position */
  signed char joystick_1_y; /* =-100..100 y-coordinate joystick position */
  unsigned char switch_1; /* =1 if switch ON and =0 if OFF */

    /* other variable */
  unsigned char connect_flag;  /* =1 if wire connected, else =0 */

} RemoteXY;

////////////////////////////////////////////
//           END RemoteXY include          //
////////////////////////////////////////////
/* defined the right motor control pins */
#define PIN_MOTOR_RIGHT_UP 7
#define PIN_MOTOR_RIGHT_DN 6
#define PIN_MOTOR_RIGHT_SPEED 10
#define echoPin 12
```

```
#define trigPin 11

long sure, uzaklik;

/* defined the left motor control pins */
#define PIN_MOTOR_LEFT_UP 5
#define PIN_MOTOR_LEFT_DN 4
#define PIN_MOTOR_LEFT_SPEED 9

/* defined the LED pin */
#define PIN_LED 13


/* defined two arrays with a list of pins for each motor */
unsigned char RightMotor[3] =
  {PIN_MOTOR_RIGHT_UP, PIN_MOTOR_RIGHT_DN, PIN_MOTOR_RIGHT_SPEED};
unsigned char LeftMotor[3] =
  {PIN_MOTOR_LEFT_UP, PIN_MOTOR_LEFT_DN, PIN_MOTOR_LEFT_SPEED};

/*
   speed control of the motor
   motor - pointer to an array of pins
   v - motor speed can be set from -100 to 100
*/
void Wheel (unsigned char * motor, int v)
{
  if (v>100) v=100;
  if (v<-100) v=-100;
  if (v>0) {
    digitalWrite(motor[0], HIGH);
    digitalWrite(motor[1], LOW);
    analogWrite(motor[2], v*2.55);
  }
  else if (v<0) {
    digitalWrite(motor[0], LOW);
    digitalWrite(motor[1], HIGH);
    analogWrite(motor[2], (-v)*2.55);
  }
  else {
```

```
void setup()
{
  /* initialization pins */
  pinMode (PIN_MOTOR_RIGHT_UP, OUTPUT);
  pinMode (PIN_MOTOR_RIGHT_DN, OUTPUT);
  pinMode (PIN_MOTOR_LEFT_UP, OUTPUT);
  pinMode (PIN_MOTOR_LEFT_DN, OUTPUT);
  pinMode (PIN_LED, OUTPUT);
  pinMode (echoPin, INPUT);
  pinMode (trigPin, OUTPUT);

  /* initialization module RemoteXY */
  RemoteXY_Init ();

}

void loop()
{
  /* event handler module RemoteXY */
  RemoteXY_Handler ();

  /* manage LED pin */
  digitalWrite (PIN_LED, (RemoteXY.switch_1==0)?LOW:HIGH);

  /* manage the right motor */
  Wheel (RightMotor, RemoteXY.joystick_1_y - RemoteXY.joystick_1_x);
  /* manage the left motor */
  Wheel (LeftMotor, RemoteXY.joystick_1_y + RemoteXY.joystick_1_x);

  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  sure = pulseIn(echoPin, HIGH);
  uzaklik = (sure*2) * 0.0343;
  Serial.println(uzaklik);

  if (uzaklik < 15) // Uzaklık 15'den küçük ise,
  {
    analogWrite(RightMotor[2], 0);
    analogWrite(LeftMotor[2], 0);
  }
}
```

Conclusion

As a conclusion, we built a 3-wheeled robot platform with Arduino. In the building part, we used silicon to glue wheels, Arduino, and DC motors. We soldered the pins using male headers to the Arduino to connect with a motor driver. The challenging part was commanding the robot using the ultrasonic sensor. To get rid of the cable mess we taped them to the body. Thus, we prevented the contactless between cables. When we finished the wire connections and coding, we powered the robot using 3 lithium ion battery. To sum up, everything worked as expected except the ultrasonic sensor. The problem could been happened because of the connection problem or coding error.