Report of “Weletubbies” Carrying Robot Vehicles

Team 6:

Qi Zeng

Muchen Li

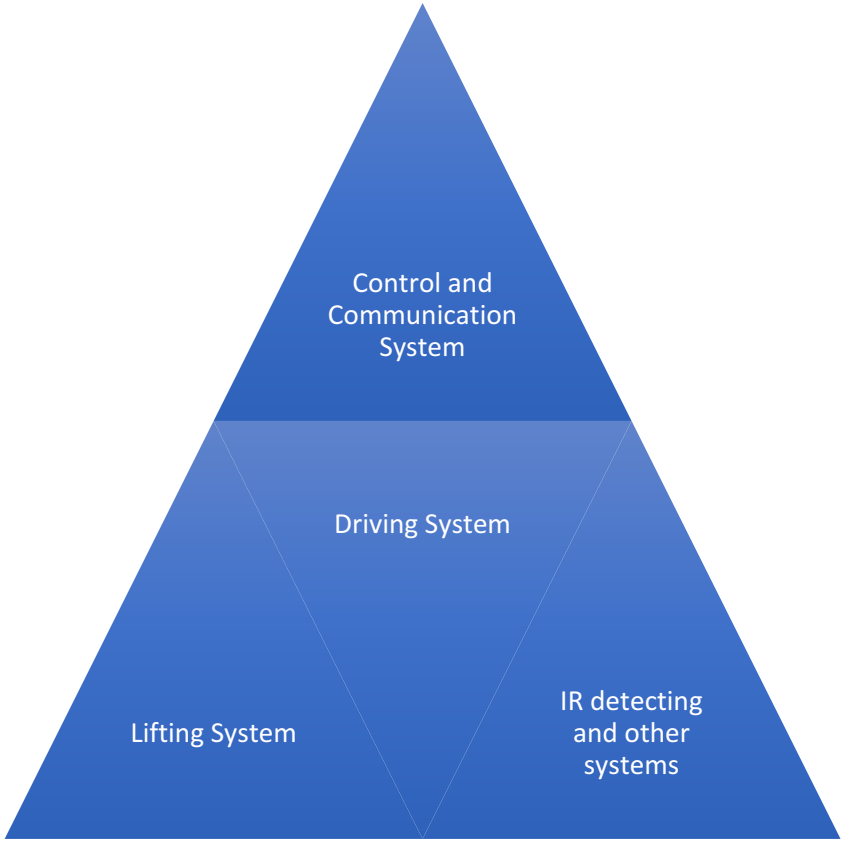
Dian Liu

Mingcong Shi

Instructors: Prof. Adam Wickenheiser, Kyle Crandall

Summary

- This robot vehicle can move freely and carry any well-shaped objects.
- Innovation of IR remote control system.
- Perfectly designed size, appearance and attractive looking.
- Optimized lifting system by shortening the arm of force.
- Intentional low wheel speed for easy control.
- Any number of vehicles can cooperate by demand.



Contents

Table of Contents

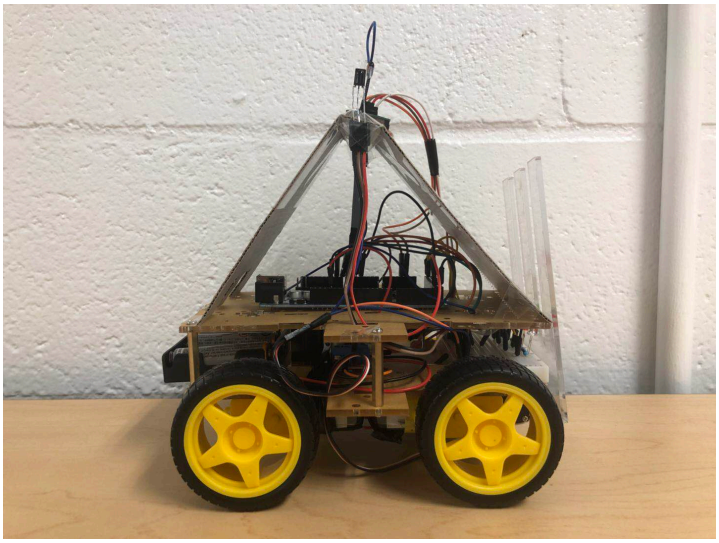
Structure and Appearance

Motivation of design

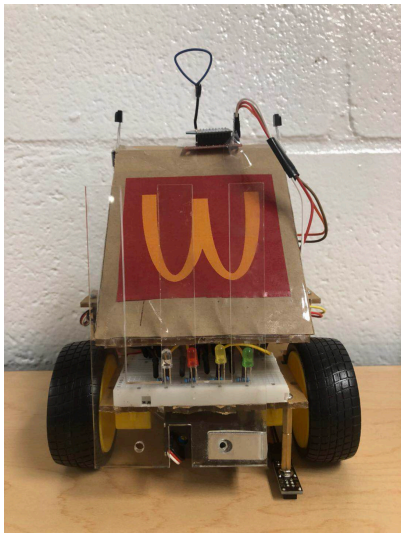
- Compact for size requirement but enough room for all components
- Reliable and appealing
- A special design, roof, to protect wiring and show neat outlook. Also provide platform of Xbee and IR sensors

Details of design

The structure of cars contains two decks of acrylic chassis and one roof.
Below first deck: Driving system including four motors and wheels, QTI sensor, whole lifting system
First deck: Driving system including power supply, H bridge. Bread board with indicating LEDs
Second deck: Arduino
Roof: XBee and IR sensors. Designed Teletubbies antenna.



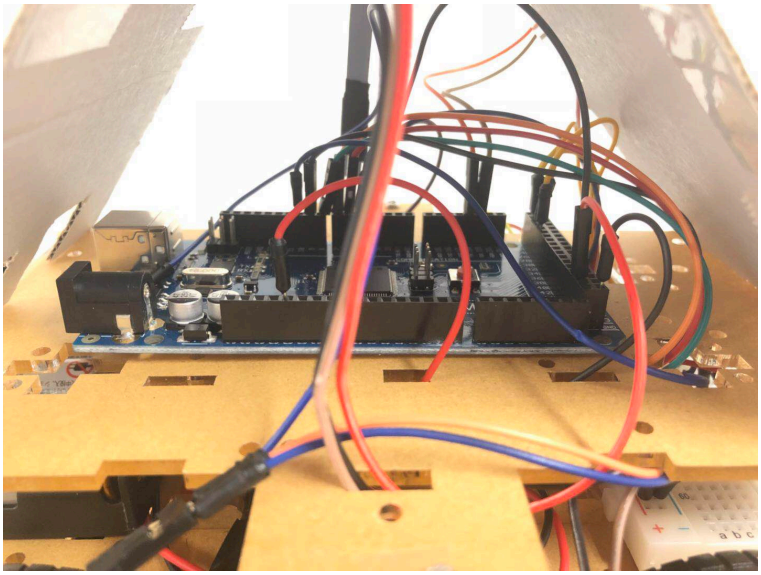
Side view



Front view

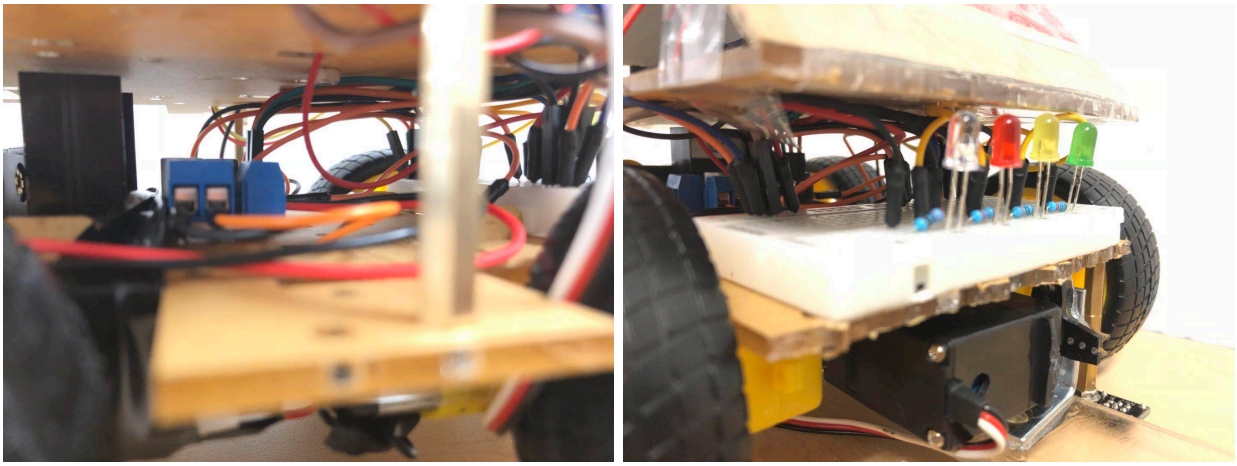
The roof provides protection of Arduino board and wiring. It's also a platform for XBee transmission and IR receivers to meet the height of IR sender. The roof hides most chaotic wires, making the whole vehicle looking much better. Outside the roof, we have large area available for team logo and vehicle number.

Second deck consists of Arduino board only, providing convenient wiring through holes on the acrylic chassis and very neat looking.



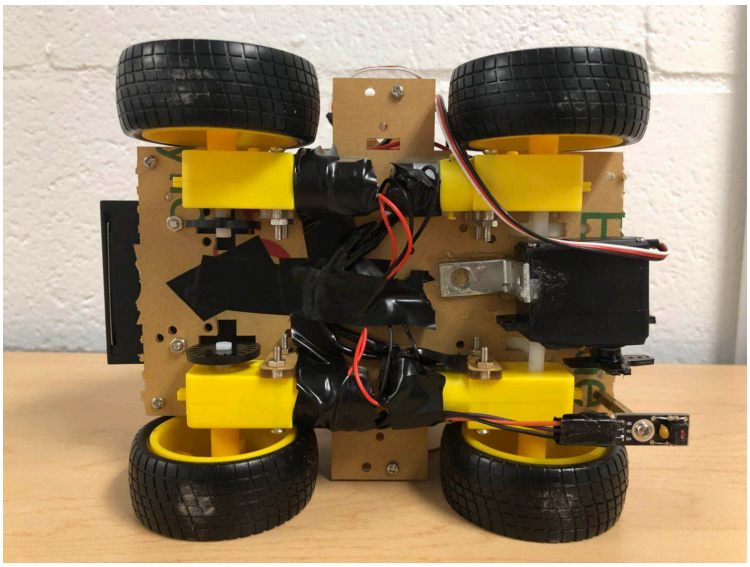
Second deck

First deck is where we put most functional components. Power supply is at the rear of this deck. The most important driving system component, H bridge, is on the middle of this deck, connecting Arduino above and motors below. Also, the LEDs are placed at the front of this deck, which can be very clearly observed, not hiding in the wires. Since this deck has most components and wires but least room, the assembly can be hard but vehicles can have good and neat looking.



First deck, H bridge(middle) and LEDs(front)

Below first deck, two motors each side are connected parallelly to ensure their simultaneous rotation. QTI sensor is put at the front, next to the servo motor of lifting system.

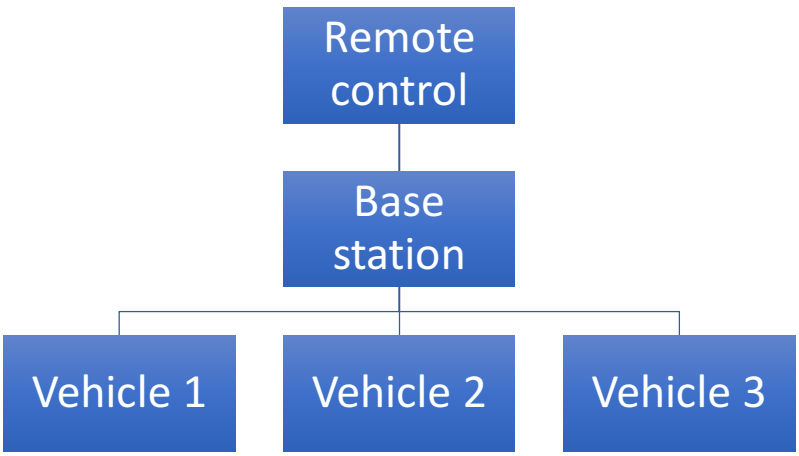


Below first deck

Control and Communication System

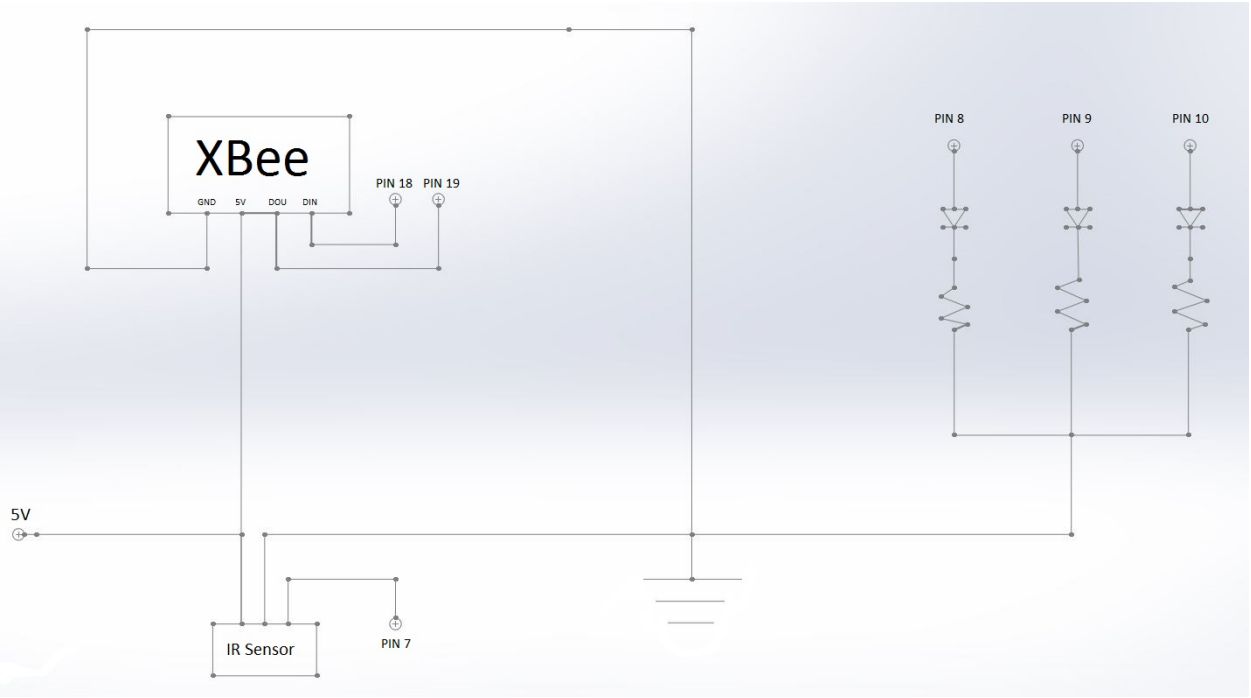
Motivation of design

When designing the control method of the robot, we thought of two kinds of control tools, namely the joystick and the remote controller. However, we found that the joystick does not accurately control the robot's actions and interference may occur between the signals during the testing process. Therefore, we chose the remote control as our control tool.



Details of design

Our robots receive signals through infrared sensors. We write code to make each button on the remote control correspond to a different signal. For example, we can use the number buttons 1, 2 and 3 to represent each robot's number. The robot's actions are also controlled by the up, down, left, right and stop buttons.



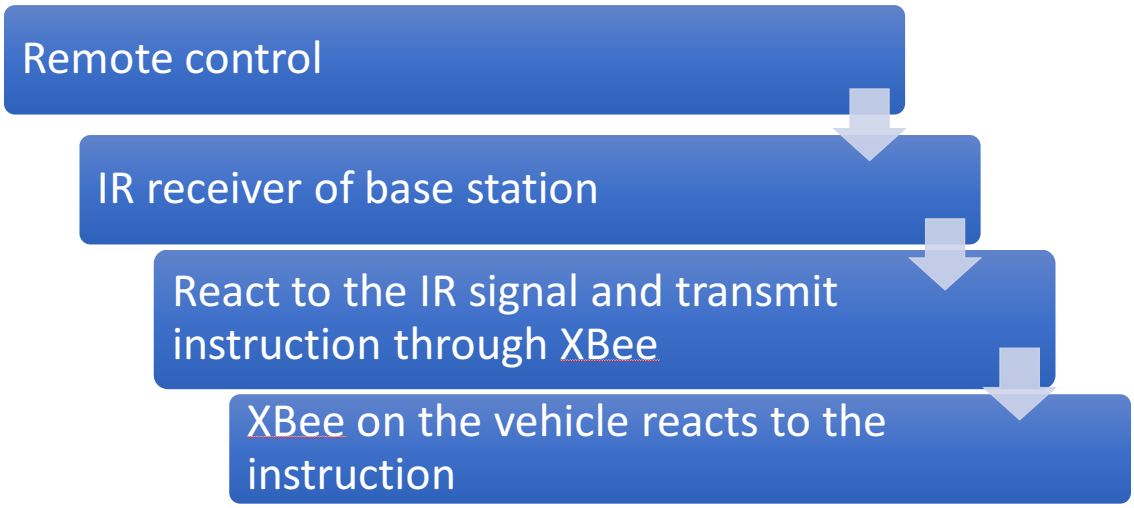
Circuit design of base station

Advantage

- We can control each robot's actions very clearly and accurately.
- easy control of any vehicles moving together or designated relationship
- Controller can stay apart from the base station.

Disadvantage

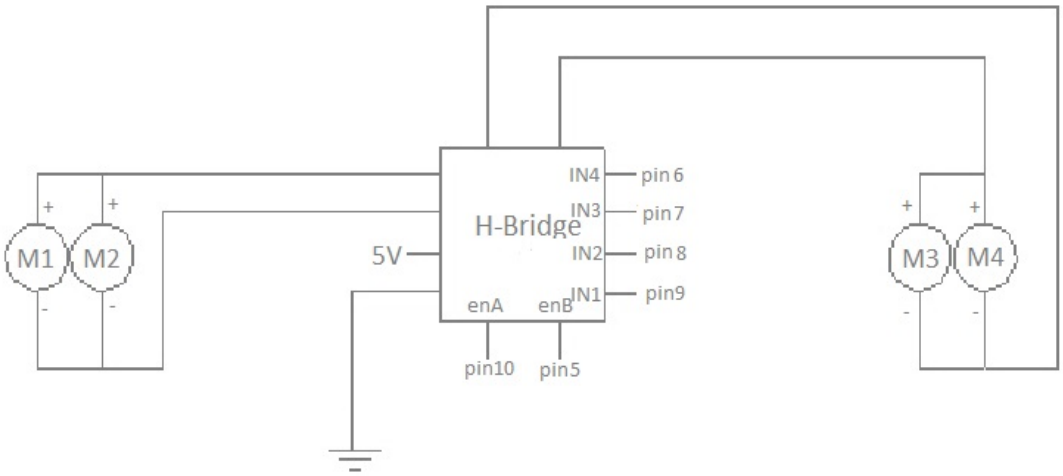
- The signal transmission could be unstable during some conditions.
- Sometimes remote control is not very responsible if it’s away from base station.



Driving System

Motivation of design

- The signal detection requires the robot can make a turn in place. Our design can be used to make it by controlling directions of two-sides motors.
- In order to reduce the errors in the direction and distance in the controlling process. The speeds of motors can be controlled by the design.



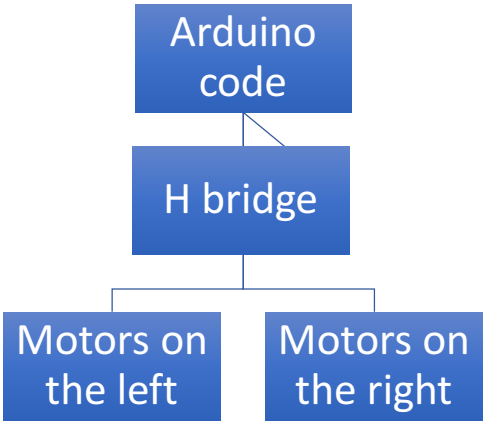
Circuit design of driving system

Advantage

The design can increase the stability of the system and can decrease the complexity of the circuit.

Disadvantage

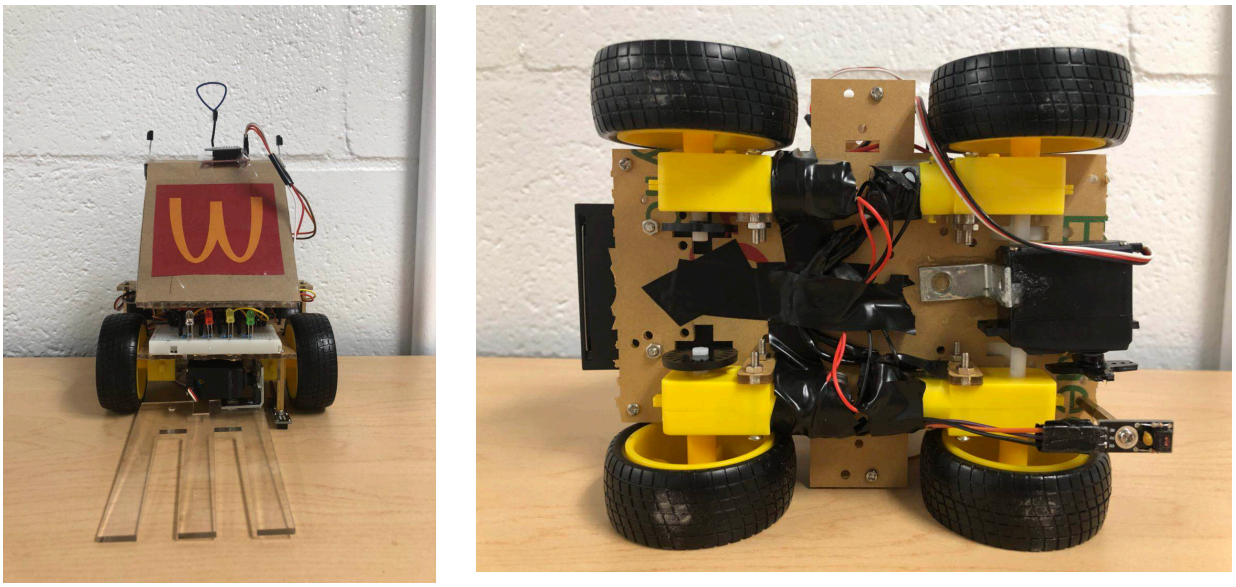
The maximum speed of motor is limited by the module.



Lifting System

Motivation of design

Regarding to the lifting system, we need to design a fork that not only meets the requirements for carrying heavy objects but also does not exceed the specified length. We found that the fork with a vertical lifting system is not satisfied with the size requirements. Therefore, we chose to use a motor to control the rotation of the fork.



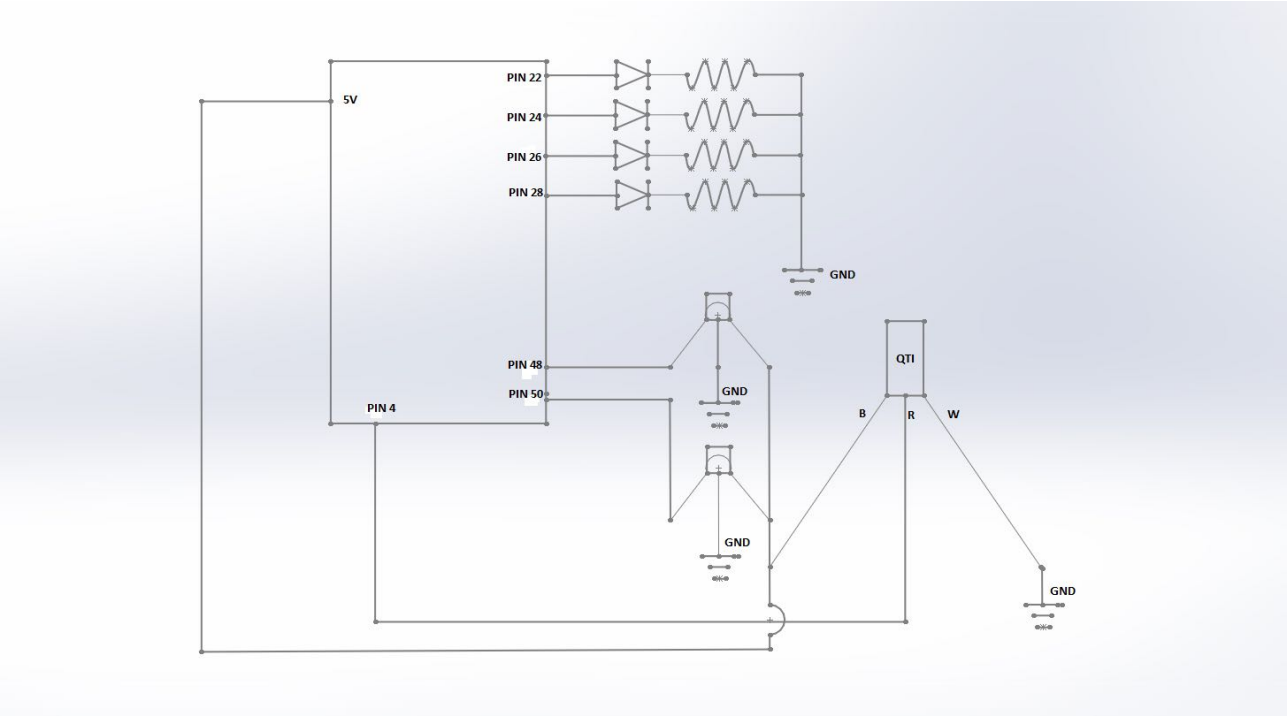
Lifting system at bottom of chassis

Details of design

As for the structural part of the fork. First of all, we got the fork by laser-cutting the acrylic board, and then we designed a connecting device to connect the chassis and the motor and the fork. After that we write the code to control the angle of rotation and rotation speed of the fork. The advantages of this design are listed as below.

The advantage of this approach is that we can rotate the fork perpendicular to the ground for making the fork close to the robot when we don't need to use it.
We have significantly saved the cost, made a fork with a good weight bearing with a lower budget. Secondly, We can make the fork more weight-bearing by positioning the servo into a very crowded space and close to the carrying weight as possible but not exceeding maximum size requirement.

IR Signal Detecting and Other Systems



Circuit design of IR and other systems

Sensors used in this vehicle include:

IR receiver

Highly sensitive to ambient light and other interference.
Sending pulses at a given frequency allows receivers to discriminate from other sources.
High detection capability.

QTI sensor

The QTI module is designed for close-proximity infrared light detection.
We place the QTI sensor at the bottom of the vehicle near the ground to detect the black tape.
The QTI module can be one-button disabled for quick reset and driving out of arena area.

Thoughts on future work

Regarding to the robot's future work. Firstly, we can replace a more powerful motor to meet the larger load-bearing demand in the future. Furthermore, use 3D printing technology to print a more stable connection structure, rather than the self-made connection parts we are using now. What's more, we can find a way to make the transmission and reception of the infrared signals more stable, because signal problem happened to us at competition and wasted one turn. Also, we should write some new codes to get more action instructions that allows us to control the robot for an extended use of this robot vehicle.

Last but not the least, automation is a huge problem. We tried to add ultrasonic sensors to allow the vehicle avoiding obstacles and looking for the target object. However, we found that even the vehicles can find the object successfully, we cannot ensure the position of the two vehicles. So they cannot cooperate with each other to move the object. Therefore, the most recent problem to be solved for automation is the communication and positioning system between these two vehicles. So they may be able to line up in opposite direction or shoulder to shoulder together to move the object.

Appendix

1 Annotated Arduino Code

Code for vehicles (They are similar with the only difference of control numbers)

```
#include <Servo.h>
Servo servoRight;

int a;
int b;
int QTI = 4;
// motor one
const int enA = 10;
const int in1 = 9;
const int in2 = 8;
// motor two
const int enB = 5;
const int in3 = 7;
const int in4 = 6;
//LEDs
const int LED_g = 22;
const int LED_y = 24;
const int LED_r = 26;
const int LED_w = 28;

float V_r;
float V_l;
float t;

//IR sensor on the right
const int sensor_ri = 50;
//IR sensor on the left
const int sensor_l = 48;
```

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial1.begin(9600);

  //connect servo to pin 13
  servoRight.attach(13);

  //H bridge pins
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  pinMode(sensor_ri, INPUT);
  pinMode(sensor_l, INPUT);

  pinMode(LED_g, OUTPUT);
  pinMode(LED_y, OUTPUT);
  pinMode(LED_r, OUTPUT);
  pinMode(LED_w, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  int data = Serial1.read();
  if (data > 49 && data < 54) {
    a = data; //input range of car selection
  }
  else if (data >= 54) {
    b = data; //input range of car control
  }
```

```
if (b != 69){ //b==69 to disable QTI sensor
  int val = QTISensor(QTI);

if (val == 1) { //if QTI sensor is activated, stop the car and light up red LED
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  digitalWrite(LED_r, HIGH);
}}
else{
  digitalWrite(LED_r, LOW);
}
if (a == 53) {//control button "4" for controling car 1 and 2 simultaneously
  digitalWrite(LED_w, HIGH);
  switch (b) {
    case 54: //control button "up" for going stright
      digitalWrite(in1, LOW);
      digitalWrite(in2, HIGH);
      analogWrite(enA, 160);
      digitalWrite(in3, LOW);
      digitalWrite(in4, HIGH);
      analogWrite(enB, 160);
      break;
    case 55: //control button "left" for turning left
      digitalWrite(in1, LOW);
      digitalWrite(in2, HIGH);
      analogWrite(enA, 180);
      digitalWrite(in3, HIGH);
      digitalWrite(in4, LOW);
      analogWrite(enB, 180);
      break;
    case 56: //control button "right" for turning right
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      analogWrite(enA, 200);
      digitalWrite(in3, LOW);
```

```
    digitalWrite(in4, HIGH);
    analogWrite(enB, 200);
    break;
   case 65: //control button "down" for going backwards
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    analogWrite(enA, 200);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    analogWrite(enB, 200);
    break;
   case 66:  //control button "OK" for stopping
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    break;
   case 67: //control button "*" for putting down the fork
    servoRight.writeMicroseconds(1540);
    break;
   case 68: //control button "#" for lifting the fork
    servoRight.writeMicroseconds(1300);
    break;
  }
}
if (a == 50) { //control button "4" for controling car 1
  digitalWrite(LED_w, HIGH);
  switch (b) {
   case 54:
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    analogWrite(enA, 160);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    analogWrite(enB, 160);
    break;
   case 55:
```

```
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    analogWrite(enA, 230);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    analogWrite(enB, 230);
    break;
  case 56:
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    analogWrite(enA, 230);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    analogWrite(enB, 230);
    break;
  case 65:
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    analogWrite(enA, 160);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    analogWrite(enB, 160);
    break;
  case 66:
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    break;
  case 67:
    servoRight.writeMicroseconds(1540);
    break;
  case 68:
    servoRight.writeMicroseconds(1200);
    break;
 }
}
```

```
  t = millis();

  //detecting frequency of IR signal
  int H_ri = pulseIn(sensor_ri,LOW,100000);
  int Freq_ri = 250000/(H_ri);
  int H_l = pulseIn(sensor_l,LOW,100000);
  int Freq_l = 250000/(H_l);
  //while (Serial1.available()){
  //  data = Serial1.read();
  //}
  if ((abs(Freq_ri - 10) <= 5)||(abs(Freq_l - 10) <= 5) ){
    //judging if it's true signal
    digitalWrite(LED_g,HIGH);
    digitalWrite(LED_y,LOW);
  }
  else{
    digitalWrite(LED_g,LOW);
    digitalWrite(LED_y,HIGH);
  }

}

int QTISensor(int QTI) {//QTI sensor
  pinMode(QTI, OUTPUT);
  digitalWrite(QTI, HIGH);
  delay(1);
  pinMode(QTI, INPUT);
  delay(1);
  int sensor_QTI = digitalRead(QTI);
  delay(1);
  return sensor_QTI;
}
```

Code for base station

```
#include <IRremote.h> //library for IR remote control

const int RECV_PIN = 7; //recieving IR signal
const int LED_r = 8; //LED for vehicle 1
const int LED_g = 9; //LED for vehicle 2
const int LED_y = 10; //LED for vehicle 3
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup(){
  //LEDs output
  pinMode(LED_r, OUTPUT);
  pinMode(LED_g, OUTPUT);
  pinMode(LED_y, OUTPUT);
  Serial.begin(9600);
  Serial1.begin(9600);
  irrecv.enableIRIn();
  irrecv.blink13(true);
}

void loop(){
  if (irrecv.decode(&results)){
       Serial.println(results.value, HEX);
       switch(results.value){
         case 0xFFA25D: //control button "1"
         digitalWrite(LED_r, HIGH);
         digitalWrite(LED_g, LOW);
         digitalWrite(LED_y, LOW);
         Serial1.print(2);
         break;
         case 0xFF629D: //control button "2"
         digitalWrite(LED_r, LOW);
         digitalWrite(LED_g, HIGH);
         digitalWrite(LED_y, LOW);
         Serial1.print(3);
         break;
         case 0xFFE21D: //control button "3"
```

```
digitalWrite(LED_r, LOW);
digitalWrite(LED_g, LOW);
digitalWrite(LED_y, HIGH);
Serial1.print(4);
break;
case 0xFF22DD: //control button "4"
digitalWrite(LED_r, HIGH);
digitalWrite(LED_g, HIGH);
digitalWrite(LED_y, LOW);
Serial1.print(5);
case 0xFF18E7: //control button "up"
Serial1.print(6);
break;
case 0xFF10EF: //control button "left"
Serial1.print(7);
break;
case 0xFF5AA5: //control button "right"
Serial1.print(8);
break;
case 0xFF4AB5: //control button "OK"
Serial1.print("A");
break;
case 0xFF38C7: //control button "down"
Serial1.print("B");
break;

case 0xFF6897: //control button "*"
Serial1.print("C");
break;
case 0xFFB04F: //control button "#"
Serial1.print("D");
break;

case 0xFF9867: //control button "0"
Serial1.print("E");
break;
```

```
        case 0xFFFFFFFF:

        Serial1.print(9);
        delay(1);
        Serial1.print(1);
        break;
      }
      irrecv.resume();
  }
}
```

2 Purchase List

Product	Number	Total Price
Gorilla Epoxy 5MIN .850Z	1.	6.99
Cornr Brace 1-1/2''Zn4pk	1.	3.99
Cornr Brace 1''X1/2''Zn4pk	2.	7.58
Wingoneer New Style Infraed IR Wireless Remote Control Sensor Module Kits for Arduino	1.	6.99
LGDehome 3PCS L298N H-Bridge Motor Drive Controller Board	1.	13.99
LewanSoul LD-20MG Full Metal Gear Standard Digital Servo With 20kg High Torque	3.	53.97
HC-SR04 Ultransonic Sensor Distance Module.	1.	9.79

(5pcs) for Arduino UNO		
Makerfire 4-wheel Robot Smart Car Chassis Kits Car Model with Speed Encoder for Arduino	3.	60
Switch GH7455-ND	1.	19.13
Switch EG2381-ND	3.	9.99
Total		**192.42**