Code Annex

Arduino code:

```
char t;
char currentstate;//keep the current state of the car
int trig_f=6;
int echo_f=7;
int trig_b=8;
int echo_b=9;
int warningled=5;
int lighters=3;
int lightsensor=4;
int right_motors_forward=13;
int right_motors_reverse=12;
int left_motors_forward=11;
int left_motors_reverse=10;
int stopled=2;
int ct1=0, ct2=0;//for controlling the messages sent to the device

void setup() {

pinMode(lightsensor,INPUT);     //input from lightsensor
pinMode(trig_f,OUTPUT);    //trigger front output
pinMode(echo_f,INPUT);     //echo front input
pinMode(trig_b,OUTPUT);    //trigger back output
pinMode(echo_b,INPUT);     //echo back input
pinMode(left_motors_forward,OUTPUT);   //left motors forward
pinMode(left_motors_reverse,OUTPUT);   //left motors reverse
pinMode(right_motors_forward,OUTPUT);   //right motors forward
pinMode(right_motors_reverse,OUTPUT);   //right motors reverse
```

```
pinMode(warningled,OUTPUT);
pinMode(lighters,OUTPUT);
pinMode(stopled,OUTPUT);
Serial.begin(9600);

}

void loop() {
long duration1, distance1, duration2, distance2;

  digitalWrite(trig_f, LOW);
  delayMicroseconds(2);
  digitalWrite(trig_f, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_f, LOW);
  duration1 = pulseIn(echo_f, HIGH);
  distance1 = (duration1 / 2) / 29;//calculate distance to an obstacle
in front

  digitalWrite(trig_b, LOW);
  delayMicroseconds(2);
  digitalWrite(trig_b, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_b, LOW);
  duration2 = pulseIn(echo_b, HIGH);
  distance2 = (duration2 / 2) / 29;//calculate distance to an obstacle
in the back

if(Serial.available()){//establish Bluetooth connection
  t = Serial.read();//get the direction introduce (permanant change)
}
```

```
 int light=digitalRead(4);

  if (light==HIGH) {
    digitalWrite(lighters,HIGH);
  }
  else {
    digitalWrite(lighters,LOW);
  }

if (distance1 < 30){
  if (currentstate == 'F'){
    digitalWrite(left_motors_forward,LOW);
    digitalWrite(left_motors_reverse,LOW);
    digitalWrite(right_motors_forward,LOW);
    digitalWrite(right_motors_reverse,LOW);
    digitalWrite(stopled,HIGH);
    digitalWrite(warningled,HIGH);
    if (ct1==0){//control the rate of messages sent to the device
      Serial.println("Obstacle in front!");
      ct1=1;
    }
  }
}
else {
  digitalWrite(warningled,LOW);
  ct1=0;
}

if (distance2 < 30){
```

```
  if (currentstate == 'B'){
    digitalWrite(left_motors_forward,LOW);
    digitalWrite(left_motors_reverse,LOW);
    digitalWrite(right_motors_forward,LOW);
    digitalWrite(right_motors_reverse,LOW);
    digitalWrite(stopled,HIGH);
    digitalWrite(warningled,HIGH);
    if (ct2==0){//write only once through Bluetooth
      Serial.println("Obstacle in the back!");
      ct2=1;
    }
  }
}
else {
  if (distance1 >= 30){//  not interfere with the warning from front
obstacle
    digitalWrite(warningled,LOW);
    ct2=0;
  }
}


if(t == 'F'){           //move forward (all motors rotate in forward
direction)
  if (distance1 >= 30){
    digitalWrite(right_motors_reverse,LOW);
    digitalWrite(left_motors_reverse,LOW);
    digitalWrite(left_motors_forward,HIGH);
    digitalWrite(right_motors_forward,HIGH);
    digitalWrite(stopled,LOW);
    Serial.println("Move Forward");
    currentstate=t;
```

```
    }
}


if(t == 'B'){      //move backwards (all motors rotate in reverse
direction)
  if (distance2 >= 30){
    digitalWrite(right_motors_forward,LOW);
    digitalWrite(left_motors_forward,LOW);
    digitalWrite(left_motors_reverse,HIGH);
    digitalWrite(right_motors_reverse,HIGH);
    digitalWrite(stopled,LOW);
    Serial.println("Move backwards");
    currentstate=t;
  }
}


if(t == 'R'){      //turn right (left motors are HIGH. right motors
low)
  digitalWrite(left_motors_forward,LOW);
  digitalWrite(left_motors_reverse,LOW);
  digitalWrite(right_motors_reverse,LOW);
  digitalWrite(left_motors_forward,HIGH);
  digitalWrite(stopled,LOW);
  Serial.println("Turn Right");
  currentstate=t;
}


if(t == 'L'){      //turn left (right motors are HIGH, left motors are
low)
  digitalWrite(left_motors_reverse,LOW);
  digitalWrite(right_motors_forward,LOW);
```

```
  digitalWrite(right_motors_reverse,LOW);
  digitalWrite(right_motors_forward,HIGH);
  digitalWrite(stopled,LOW);
  Serial.println("Turn Left");
  currentstate=t;
}

if(t == 'S'){      //STOP (all motors stop)
  digitalWrite(left_motors_forward,LOW);
  digitalWrite(left_motors_reverse,LOW);
  digitalWrite(right_motors_forward,LOW);
  digitalWrite(right_motors_reverse,LOW);
  digitalWrite(stopled,HIGH);
  Serial.println("STOP");
  currentstate=t;
}
delay(100);
}
```