

```

#include <Ultrasonic.h>
#include <Servo.h>

Servo armservo;
Servo gripperservo;

Ultrasonic ultrasonic(10,11);

int motor_pin1 = 2;
int motor_pin2 = 3;
int motor_pin3 = 4;
int motor_pin4 = 5;
int motor_pin5 = 6;
int motor_pin6 = 7;
int motor_pin7 = 8;
int motor_pin8 = 9;
int BuzzPin   = 12;
int Light     = 13;

int distance;

int state;

void setup(){
  Serial2.begin(9600);
  armservo.attach(46);
  gripperservo.attach(45);
  pinMode(BuzzPin, OUTPUT);
  pinMode(Light, OUTPUT);
  pinMode(motor_pin1,OUTPUT);
  pinMode(motor_pin2,OUTPUT);
  pinMode(motor_pin3,OUTPUT);
  pinMode(motor_pin4,OUTPUT);
  pinMode(motor_pin5,OUTPUT);
  pinMode(motor_pin6,OUTPUT);
  pinMode(motor_pin7,OUTPUT);
  pinMode(motor_pin8,OUTPUT);
  armservo.write(180);
  gripperservo.write(0);

  delay(100);
}

void loop(){

  distance = ultrasonic.Ranging(CM);

  if (distance <= 20){
    brake();
    delay(100);
    backward();
    delay(100);
    brake();
  }
}

```

```

    delay(100);
}
else{
    state = Serial2.read();
    if (state == 'F'){
        forward();
    }
    else if (state == 'B'){
        backward();
    }
    else if (state == 'L'){
        left();
    }
    else if (state == 'G'){
        fleft();
    }
    else if (state == 'I'){
        fright();
    }
    else if (state == 'R'){
        right();
    }
    else if (state == 'S'){
        brake();
    }
    else if (state == 'V'){
        BuzzON();
    }
    else if (state == 'v'){
        BuzzOFF();
    }
    else if (state == 'W'){
        LightON();
    }
    else if (state == 'w'){
        LightOFF();
    }
    else if (state == 'U'){
        gripperservo.write(140);
    }
    else if (state == 'u'){
        gripperservo.write(0);
    }
    else if (state == 'X'){
        armservo.write(130);
    }
    else if (state == 'x'){
        armservo.write(180);
    }
}
distance = ultrasonic.Ranging(CM);
}
}

void forward(){

```

```
digitalWrite(motor_pin1,LOW);
digitalWrite(motor_pin2,HIGH);
digitalWrite(motor_pin3,HIGH);
digitalWrite(motor_pin4,LOW);
digitalWrite(motor_pin5,HIGH);
digitalWrite(motor_pin6,LOW);
digitalWrite(motor_pin7,LOW);
digitalWrite(motor_pin8,HIGH);
}
```

```
void fleft(){
digitalWrite(motor_pin1,LOW);
digitalWrite(motor_pin2,HIGH);
digitalWrite(motor_pin3,HIGH);
digitalWrite(motor_pin4,LOW);
digitalWrite(motor_pin5,HIGH);
digitalWrite(motor_pin6,LOW);
digitalWrite(motor_pin7,LOW);
digitalWrite(motor_pin8,HIGH);
digitalWrite(motor_pin1,LOW);
digitalWrite(motor_pin2,HIGH);
digitalWrite(motor_pin3,HIGH);
digitalWrite(motor_pin4,LOW);
digitalWrite(motor_pin5,LOW);
digitalWrite(motor_pin6,HIGH);
digitalWrite(motor_pin7,HIGH);
digitalWrite(motor_pin8,LOW);
}
```

```
void fright(){
digitalWrite(motor_pin1,LOW);
digitalWrite(motor_pin2,HIGH);
digitalWrite(motor_pin3,HIGH);
digitalWrite(motor_pin4,LOW);
digitalWrite(motor_pin5,HIGH);
digitalWrite(motor_pin6,LOW);
digitalWrite(motor_pin7,LOW);
digitalWrite(motor_pin8,HIGH);
digitalWrite(motor_pin1,HIGH);
digitalWrite(motor_pin2,LOW);
digitalWrite(motor_pin3,LOW);
digitalWrite(motor_pin4,HIGH);
digitalWrite(motor_pin5,HIGH);
digitalWrite(motor_pin6,LOW);
digitalWrite(motor_pin7,LOW);
digitalWrite(motor_pin8,HIGH);
}
```

```
void backward(){
digitalWrite(motor_pin1,HIGH);
digitalWrite(motor_pin2,LOW);
digitalWrite(motor_pin3,LOW);
digitalWrite(motor_pin4,HIGH);
digitalWrite(motor_pin5,LOW);
}
```

```
digitalWrite(motor_pin6,HIGH);
digitalWrite(motor_pin7,HIGH);
digitalWrite(motor_pin8,LOW);
}
```

```
void left(){
digitalWrite(motor_pin1,LOW);
digitalWrite(motor_pin2,HIGH);
digitalWrite(motor_pin3,HIGH);
digitalWrite(motor_pin4,LOW);
digitalWrite(motor_pin5,LOW);
digitalWrite(motor_pin6,HIGH);
digitalWrite(motor_pin7,HIGH);
digitalWrite(motor_pin8,LOW);
}
```

```
void right (){
digitalWrite(motor_pin1,HIGH);
digitalWrite(motor_pin2,LOW);
digitalWrite(motor_pin3,LOW);
digitalWrite(motor_pin4,HIGH);
digitalWrite(motor_pin5,HIGH);
digitalWrite(motor_pin6,LOW);
digitalWrite(motor_pin7,LOW);
digitalWrite(motor_pin8,HIGH);
}
```

```
void brake(){
digitalWrite(motor_pin1,LOW);
digitalWrite(motor_pin2,LOW);
digitalWrite(motor_pin3,LOW);
digitalWrite(motor_pin4,LOW);
digitalWrite(motor_pin5,LOW);
digitalWrite(motor_pin6,LOW);
digitalWrite(motor_pin7,LOW);
digitalWrite(motor_pin8,LOW);
}
```

```
void BuzzON(){
digitalWrite(BuzzPin, HIGH);
}
```

```
void BuzzOFF(){
digitalWrite(BuzzPin, LOW);
}
```

```
void LightON(){
digitalWrite(Light, HIGH);
}
```

```
void LightOFF(){
digitalWrite(Light, LOW);
}
```