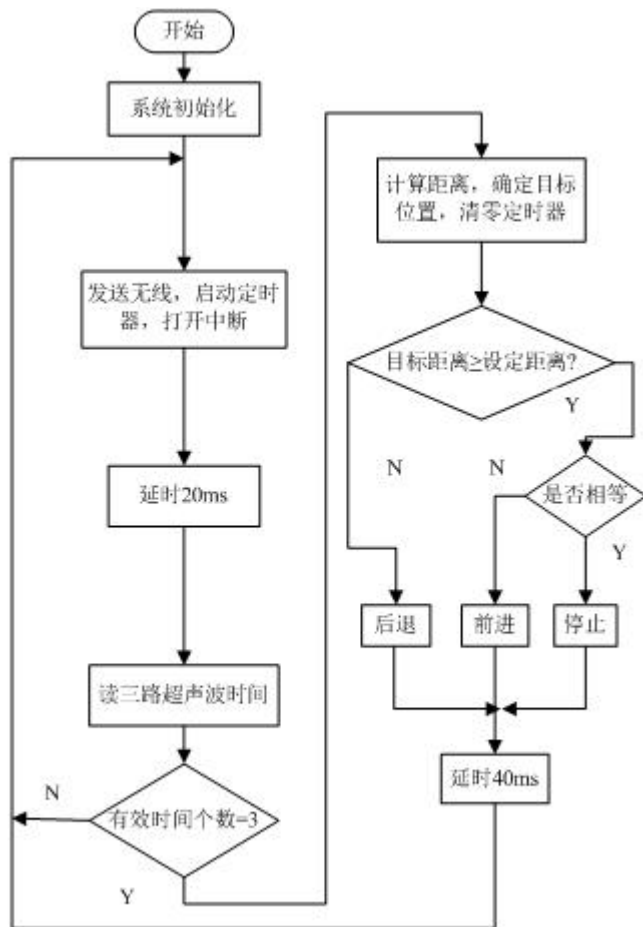
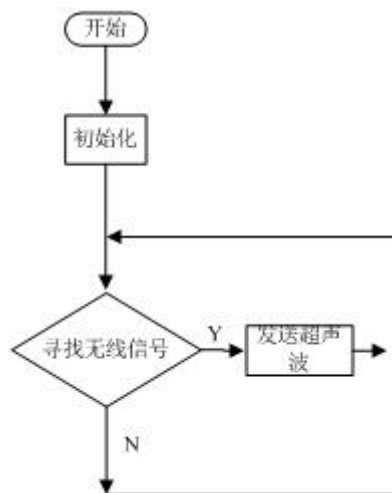


小车控制流程图



目标携带装置控制流程图:



程序展示:

```
#define DEBUG 0 // set to 1 to print to serial monitor, 0 to disable
#include <Servo.h>
Servo headservo; //
```

```

    // Constants
const int EchoPin = 2; //
const int TrigPin = 3; //
const int leftmotorpin1 = 4; //
const int leftmotorpin2 = 5;
const int rightmotorpin1 = 6;
const int rightmotorpin2 = 7
const int HeadServopin = 9; //
const int Sharppin = 11; //
const int maxStart = 800; //run dec time
// Variables
int isStart = maxStart; //
int currDist = 0; //
boolean running = false;
void setup() {
    Serial.begin(9600); //
    pinMode(EchoPin, INPUT);
    pinMode(Sharppin, INPUT);
    for (int pinindex = 3; pinindex < 11; pinindex++) {
        pinMode(pinindex, OUTPUT); // set pins 3 to 10 as outputs
    }
    //
    headservo.attach(HeadServopin);
    //
    headservo.write(70);
    delay(2000);
    headservo.write(20);
    delay(2000);
}
void loop() {
    if(DEBUG){
        Serial.print("running:");
        if(running){
            Serial.println("true");
        }
        else{

```

```

    Serial.println("false");
  }
}
if (isStart <= 0) {
  if(running){
    totalhalt(); //

}
int findsomebody = digitalRead(Sharppin);
if(DEBUG){
  Serial.print("findsomebody:");
  Serial.println(findsomebody);
}
if(findsomebody > 0) {
  isStart = maxStart;
}
delay(4000);
return;
}
isStart--;
delay(100);
if(DEBUG){
  Serial.print("isStart: ");
  Serial.println(isStart);
}
currDist = MeasuringDistance(); //
if(DEBUG){
  Serial.print("Current Distance: ");
  Serial.println(currDist);
}
if(currDist > 30) {
  nodanger();
}
else if(currDist < 15){

```

```

    backup();
    randTrun();
}
else {
    //whichway();
    randTrun();
}
}
/
long MeasuringDistance() {
    long duration;
    //pinMode(TrigPin, OUTPUT);
    digitalWrite(TrigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(TrigPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(TrigPin, LOW);
    //pinMode(EchoPin, INPUT);
    duration = pulseIn(EchoPin, HIGH)
    return duration / 29 / 2;
}
//
void nodanger() {
    running = true;
    digitalWrite(leftmotorpin1, LOW);
    digitalWrite(leftmotorpin2, HIGH);
    digitalWrite(rightmotorpin1, LOW);
    digitalWrite(rightmotorpin2, HIGH);
    return;
}
//
void backup() {
    running = true;
    digitalWrite(leftmotorpin1, HIGH);
    digitalWrite(leftmotorpin2, LOW);
    digitalWrite(rightmotorpin1, HIGH);

```

```

digitalWrite(rightmotorpin2, LOW);
delay(1000);
}
//
void whichway() {
  running = true;
  totalhalt(); // first stop!
  headservo.write(20);
  delay(1000);
  int lDist = MeasuringDistance(); // check left distance
  totalhalt(); //
  headservo.write(120); // turn the servo right
  delay(1000);
  int rDist = MeasuringDistance(); // check right distance
  totalhalt(); //
  if(lDist < rDist) {
    body_lturn();
  }
  else{
    body_rturn();
  }
  return;
}
//
void totalhalt() {
  digitalWrite(leftmotorpin1, HIGH);
  digitalWrite(leftmotorpin2, HIGH);
  digitalWrite(rightmotorpin1, HIGH);
  digitalWrite(rightmotorpin2, HIGH);
  headservo.write(70); // set servo to face forward
  running = false;
  return;
  delay(1000);
}
//
void body_lturn() {

```

```

    running = true;
    digitalWrite(leftmotorpin1, LOW);
    digitalWrite(leftmotorpin2, HIGH);
    digitalWrite(rightmotorpin1, HIGH);
    digitalWrite(rightmotorpin2, LOW);
    delay(1500);
    totalhalt();
}
//
void body_rturn() {
    running = true;
    digitalWrite(leftmotorpin1, HIGH);
    digitalWrite(leftmotorpin2, LOW);
    digitalWrite(rightmotorpin1, LOW);
    digitalWrite(rightmotorpin2, HIGH);
    delay(1500);
    totalhalt();
}
void randTrun(){
    long randNumber;
    randomSeed(analogRead(0));
    randNumber = random(0, 10);
    if(randNumber > 5) {
        body_rturn();
    }
    else
    {
        body_lturn();
    }
}
}

```