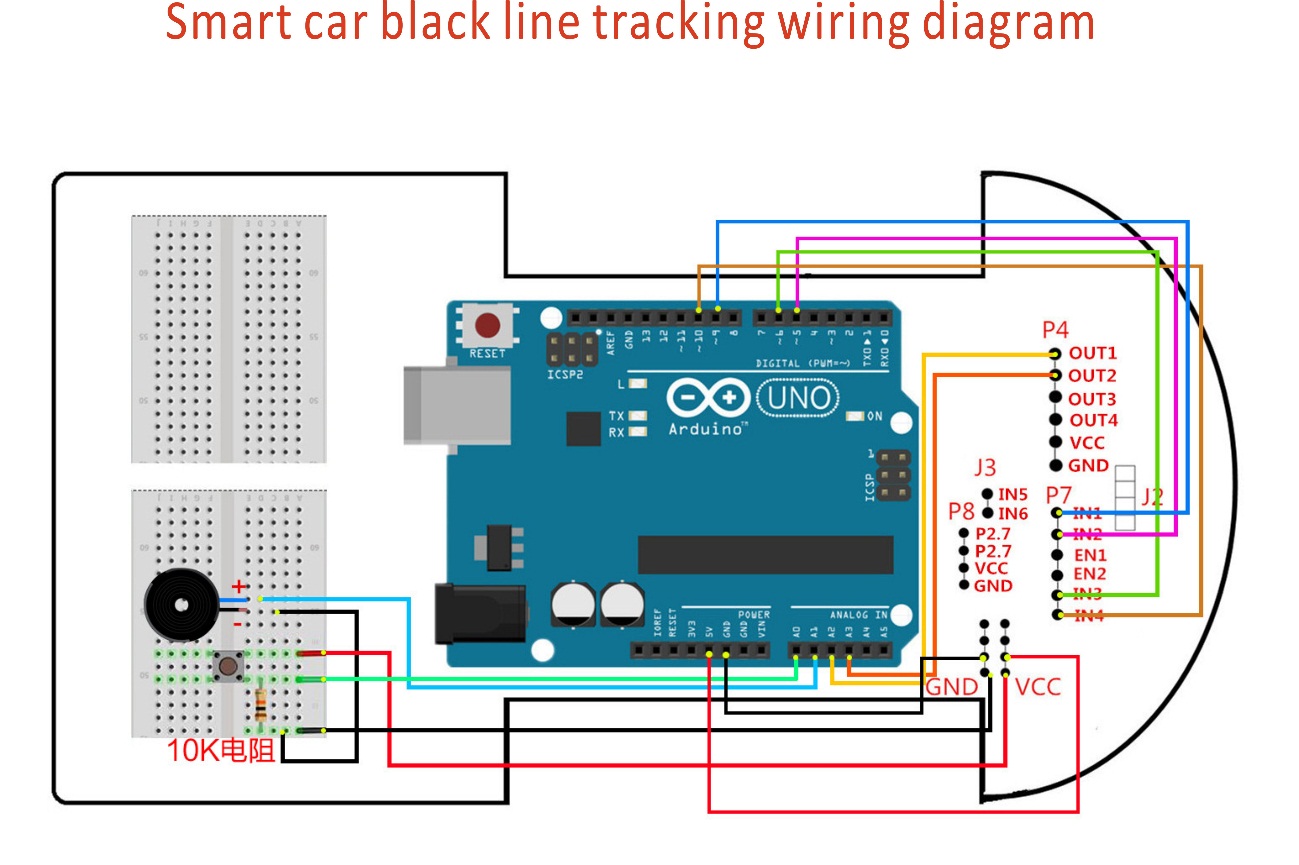
**Carro seguidor de línea**

1. Funcionamiento



1. Código

int Left\_motor\_back=9; //(IN1)

int Left\_motor\_go=5; //(IN2)

int Right\_motor\_go=6; //(IN3)

int Right\_motor\_back=10; //(IN4)

int key=A0;

int beep=A1;

const int SensorRight = A2; //(P3.2 OUT1)

const int SensorLeft = A3; //(P3.3 OUT2)

int SL;

int SR;

void setup()

{

pinMode(Left\_motor\_go,OUTPUT);

pinMode(Left\_motor\_back,OUTPUT);

pinMode(Right\_motor\_go,OUTPUT);

pinMode(Right\_motor\_back,OUTPUT);

pinMode(key,INPUT);

pinMode(beep,OUTPUT);

pinMode(SensorRight, INPUT);

pinMode(SensorLeft, INPUT);

}

void run()

{

digitalWrite(Right\_motor\_go,HIGH);

digitalWrite(Right\_motor\_back,LOW);

analogWrite(Right\_motor\_go,150);

analogWrite(Right\_motor\_back,0);

digitalWrite(Left\_motor\_go,HIGH);

digitalWrite(Left\_motor\_back,LOW);

analogWrite(Left\_motor\_go,150);

analogWrite(Left\_motor\_back,0);

}

void brake()

{

digitalWrite(Right\_motor\_go,LOW);

digitalWrite(Right\_motor\_back,LOW);

digitalWrite(Left\_motor\_go,LOW);

digitalWrite(Left\_motor\_back,LOW);

}

void left()

{

digitalWrite(Right\_motor\_go,HIGH);

digitalWrite(Right\_motor\_back,LOW);

analogWrite(Right\_motor\_go,150);

analogWrite(Right\_motor\_back,0);

digitalWrite(Left\_motor\_go,LOW);

digitalWrite(Left\_motor\_back,LOW);

analogWrite(Left\_motor\_go,0);

analogWrite(Left\_motor\_back,0);

}

void spin\_left()

{

digitalWrite(Right\_motor\_go,HIGH);

digitalWrite(Right\_motor\_back,LOW);

analogWrite(Right\_motor\_go,200);

analogWrite(Right\_motor\_back,0);

digitalWrite(Left\_motor\_go,LOW);

digitalWrite(Left\_motor\_back,HIGH);

analogWrite(Left\_motor\_go,0);

analogWrite(Left\_motor\_back,200);

}

void right()

{

digitalWrite(Right\_motor\_go,LOW);

digitalWrite(Right\_motor\_back,LOW);

analogWrite(Right\_motor\_go,0);

analogWrite(Right\_motor\_back,0);

digitalWrite(Left\_motor\_go,HIGH);

digitalWrite(Left\_motor\_back,LOW);

analogWrite(Left\_motor\_go,150);

analogWrite(Left\_motor\_back,0);

}

void spin\_right(int time)

{

digitalWrite(Right\_motor\_go,LOW);

digitalWrite(Right\_motor\_back,HIGH);

analogWrite(Right\_motor\_go,0);

analogWrite(Right\_motor\_back,200);

digitalWrite(Left\_motor\_go,HIGH);

digitalWrite(Left\_motor\_back,LOW);

analogWrite(Left\_motor\_go,200);

analogWrite(Left\_motor\_back,0);

delay(time \* 100);

}

void back(int time)

{

digitalWrite(Right\_motor\_go,LOW);

digitalWrite(Right\_motor\_back,HIGH);

analogWrite(Right\_motor\_go,0);

analogWrite(Right\_motor\_back,150);

digitalWrite(Left\_motor\_go,LOW);

digitalWrite(Left\_motor\_back,HIGH);

analogWrite(Left\_motor\_go,0);

analogWrite(Left\_motor\_back,150);

delay(time \* 100);

}

void keysacn()

{

int val;

val=digitalRead(key);

while(!digitalRead(key))

{

val=digitalRead(key);

}

while(digitalRead(key))

{

delay(10);

val=digitalRead(key);

if(val==HIGH)

{

digitalWrite(beep,HIGH);

while(!digitalRead(key))

digitalWrite(beep,LOW);

}

else

digitalWrite(beep,LOW);

}

}

void loop()

{

//keysacn();

while(1)

{

SR = digitalRead(SensorRight);

SL = digitalRead(SensorLeft);

if (SL == LOW&&SR==LOW)

run();

else if (SL == HIGH & SR == LOW)

left();

else if (SR == HIGH & SL == LOW)

right();

else

brake();

}

}