

Documentación:

Sensores:

- **Sensor de flama :**

- **Descripción:** El sensor de flama es un módulo receptor de infrarrojos, puede detectar una flama que emite una radiación en una longitud de onda de 760 a 1100 nm, este sensor puede detectar a una distancia máxima de 80 cm.
- **Especificaciones:**
 - Ángulo de detección: 60 grados
 - Voltaje de operación: 3.3 ~ 5 VCD
 - Consumo de corriente: 15 mA
 - Tipo de salida: Digital y analógica
 - Dimensiones: 32 x 14 x mm
- **Usos:**
 - Detección de incendios



- **Ejemplo de uso:**

```
void setup()//declaracion del pin de salida de zumbador
{
    pinMode(9, OUTPUT);
}
void loop()//función de loop-lectura del valor del sensor de llama
{
    int valor = analogRead(A0) ;
    if ( valor > 10){
        beep1(200) ;//función de llamada al zumbador
    }
}
void beep1(unsigned char pausa)//función del zumbador con los intervalos de silencio correspondientes
{
    analogWrite(9, 200);
    delay(pausa);           // Espera
    analogWrite(9, 0);      // Apaga
    delay(pausa);           // Espera
}
```

- **Sensor infrarrojo CNY70:**

- **Descripción :** El dispositivo CNY70 es un sensor óptico infrarrojo, de un rango de corto alcance (menos de 5 cm) que se utiliza para detectar colores de objetos y superficies. Su uso más común es para construir pequeños robots siguelineas. Contiene un emisor de radiación infrarroja -fotodiodo- y un receptor -fototransistor-. El fotodiodo emite un haz de radiación infrarroja, el fototransistor recibe ese haz de luz cuando se refleja sobre alguna superficie u objeto. El sensor CNY70 Puede utilizarse como entrada digital o analógica.

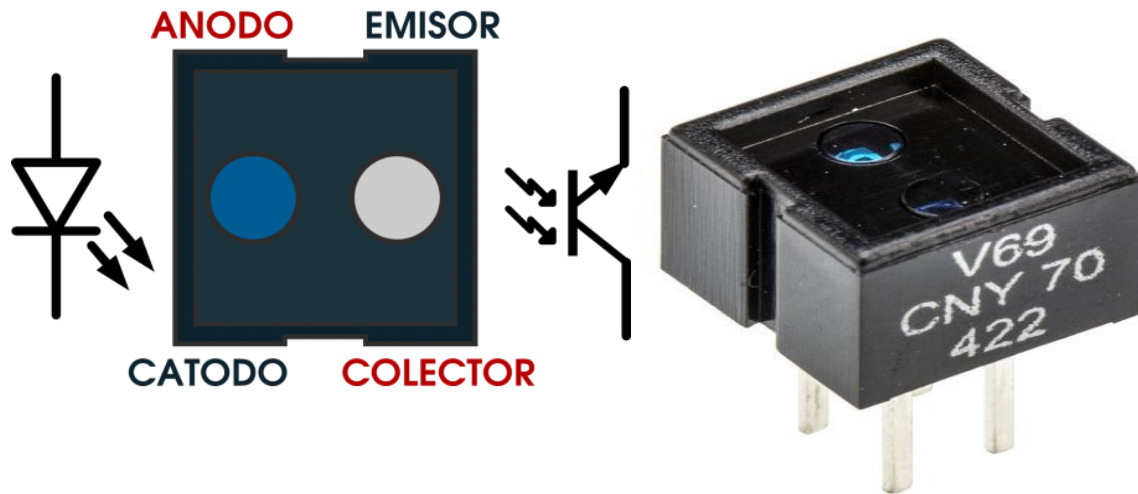
- **Especificaciones:**

- Sensor óptico por reflexión
- Combina un led infrarrojo emisor y un fototransistor receptor en un mismo módulo
- Longitud de onda del emisor: 950 nm
- Rango de operación con cambio >20% de la corriente de colector: 0 mm a 5 mm
- Distancia operativa a corriente de salida pico: <0.5 mm
- Filtro contra la luz del espectro visible
- Corriente del LED en directo max: 50 mA
- Voltaje inverso del LED max: 5 V
- Corriente de colector del fototransistor en directo max: 50 mA
- Voltaje colector-emisor del fototransistor max: 32 V
- Voltaje emisor-colector del fototransistor max: 7 V

- **Usos:**

- Detección de presencia o ausencia de objetos
- Detección de posición
- Seguimiento de línea en robótica
- Medición de revoluciones

- Escaneo de discos codificados
- Encoders



- Ejemplo de uso:

```
int ledPin1 = 8;    // pin de un LED
int ledPin2 = 9;    // pin de otro LED
int infraPin = 4;   // pin del infrarrojos utilizado como entrada digital
int valorInfra = 0; // Valor inicial de la lectura digital del infrarrojos.

void setup() {
  pinMode(ledPin1, OUTPUT);    // Inicializa el pin del LED1 como salida digital
  pinMode(ledPin2, OUTPUT);    // Inicializa el pin del LED2 como salida digital
  pinMode(infraPin, INPUT);    // Inicializa el pin 4 como entrada digital
}

void loop() {
  valorInfra = digitalRead(infraPin); // Lee el valor de la entrada 4, esto es, el valor que lee el infrarrojo
  digitalWrite(ledPin1, valorInfra); /* Escribe en el pin 8 el valor que lee del infrarrojo
  Si el infrarrojo lee 0, entonces, el LED estará apagado
  Si el infrarrojo lee 1, entonces, el LED estará encendido */

  valorInfra = !valorInfra; // Se asigna a valorInfra el valorInfra negado.
  digitalWrite(ledPin2, valorInfra); /* Escribe en el pin 9 el valor negado que lee del infrarrojo
  Si el infrarrojo lee 0, entonces, el LED conectado al pin 9 estará encendido
  Si el infrarrojo lee 1, entonces, el LED conectado al pin 9 estará apagado */
}
```

- Motores:

- **Descripcion:** Un motor de corriente continua convierte la energía eléctrica en mecánica. Se compone de dos partes: el estator y el rotor. El estator es la parte mecánica del motor donde están los polos del imán. El rotor es la parte móvil del motor con devanado y un núcleo, al que llega la corriente a través de las escobillas. Cuando la corriente eléctrica circula por el devanado del rotor, se crea un campo electromagnético. Este interactúa con el

campo magnético del imán del estator. Esto deriva en un rechazo entre los polos del imán del estator y del rotor creando un par de fuerza donde el rotor gira en un sentido de forma permanente.

Si queremos cambiar el sentido de giro del rotor, tenemos que cambiar el sentido de la corriente que le proporcionamos al rotor; basta con invertir la polaridad de la pila o batería.

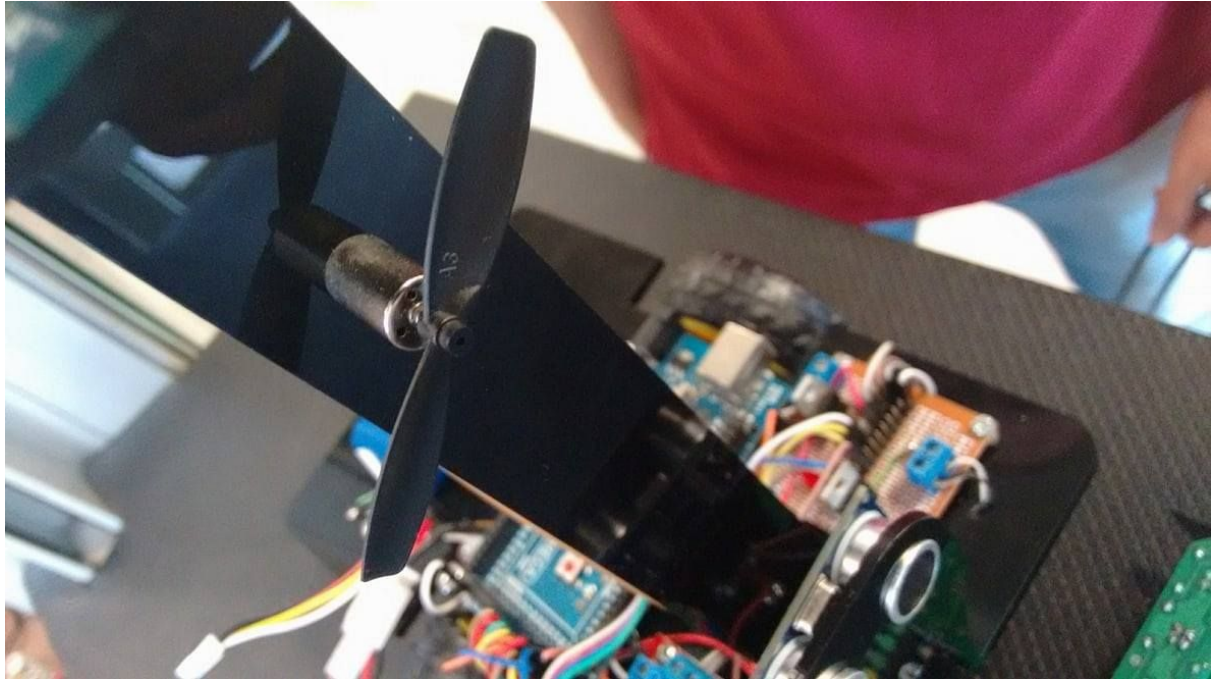
- **Especificaciones:**

- Motor DC de imán permante y escobillas
- Voltaje de operación nominal: 5 V
- Rango de voltaje de operación: 1 V a 7 V
- Velocidad de operación: < 15000 rpm
- Velocidad con carga: 11600 rpm
- Carga recomendada: 0.49 mN·m
- Rango de cargas de operación: 0.1 mN·m a 0.98 mN·m
- Corriente con carga: (@ 5 V, 0.49 mN·m): 270 mA máx.
- Corriente sin carga: (@ 5 V): 120 mA máx.
- Corriente de arranque (@ 5 V, 0.49 mN·m, método de los 2 puntos): 1300 mA (1.3 A) máx.
- Resistencia terminal (@ 20 °C, posición del rotor a 2R/3):
: 4.4 $\Omega \pm 10\%$
- Dirección de rotación en ambos sentidos
- Diámetro: 1.56 cm (0.614")
- Peso aprox: 10 g

- (Motor de las llantas)



(Ventilador)



○ Ejemplo de uso:

```
//Pin de salida del motor
const int control = 9 ;

void setup(){
  //definimos el pin 9 (variable 'control') como salida
  pinMode(control, OUTPUT) ;
}

void loop(){
  // vamos aumentando la velocidad del motor a medida que aumentan las iteraciones del ciclo.
  for ( int n = 0 ; n < 255 ; n++){
    //Se escribe la velocidad en el motor
    analogWrite (control, n) ;
    delay(15) ;
  }
}
```

Sensores en el carrito:

- **Sensores de flama:**

-

Nombre	Pin	Salida	Tipo
s_t1	36	Digital	INPUT
s_t2	34	Digital	INPUT
s_t3	32	Digital	INPUT
s_t4	30	Digital	INPUT
s_t5	28	Digital	INPUT

- **Sensores infrarojo:**

-

Nombre	Pin	Salida	Tipo
lineaCen	2	Analoga	INPUT
linealzq1	4	Analoga	INPUT
linealzq2	3	Analoga	INPUT
lineaDer1	0	Analoga	INPUT
lineaDer2	1	Analoga	INPUT

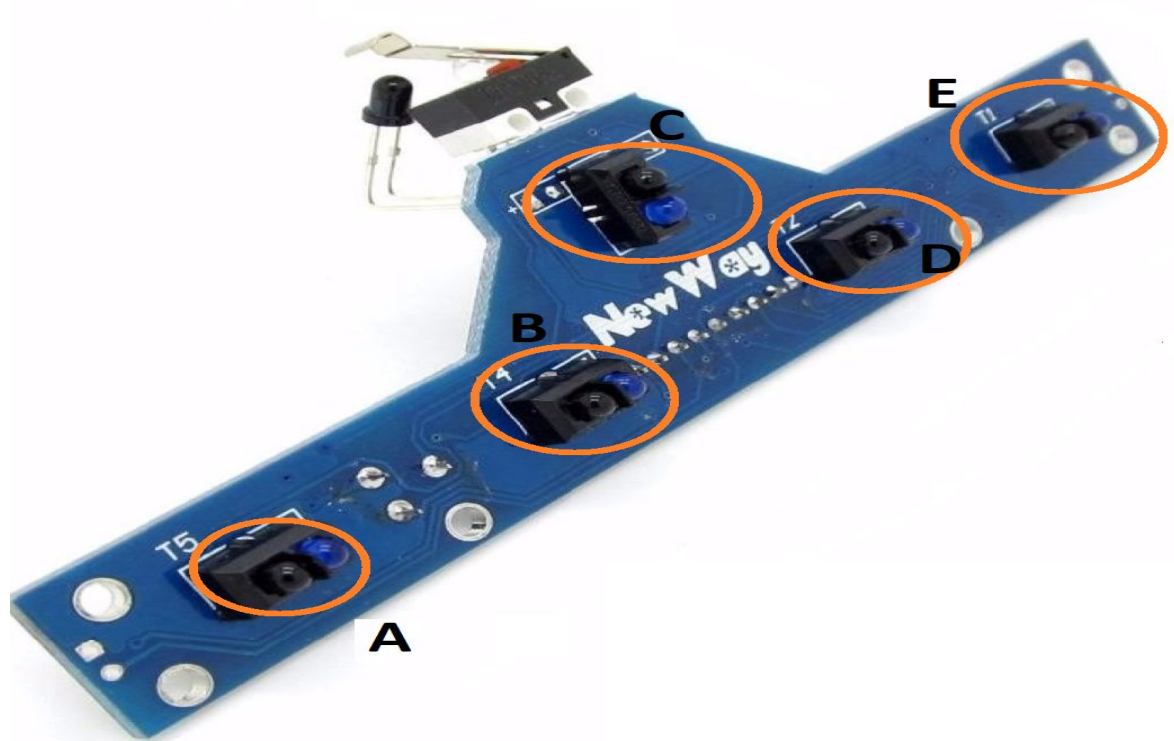
- **Motores:**

-

Nombre	Pin	Salida	Tipo
enableA	9	Analoga	OUTPUT
adelanteA	7	Digital	OUTPUT
atrasA	6	Digital	OUTPUT
adelanteB	5	Digital	OUTPUT
atrasB	4	Digital	OUTPUT
enableB	3	Analoga	OUTPUT
apa(Ventilador)	29	Digital	OUTPUT

○

¿Cómo funciona?



El carro se ubica en la pista, primero detecta el color dónde está y lo establece como “negro” , además a este valor se le agrega una tolerancia de (± 50), si A , B , C , D y E , están detectando el color “negro” el carro seguirá derecho , si A o B o C , detectan un color diferente de negro entonces el carro retrocederá y girará 45 grados a la izquierda , de esta forma ha cambiado su rumbo y vuelve a estar en color “ negro” ahora si D o E, detectan un color diferente de negro entonces el carro retrocederá y girará 45 grados a la derecha, de nuevo ha cambiado su rumbo y vuelve a estar en color “ negro”. El proceso anterior se repite hasta que se detecta calor con alguno de los sensores , si alguno de los dos sensores que están más a la izquierda detecta calor el carro girará hacia la derecha hasta que el sensor de la mitad detecte calor una vez el sensor de la mitad perciba el calor de la vela el carro encenderá el ventilador ahora si alguno de los dos sensores que están más a la derecha detecta calor el carro girará hacia la izquierda hasta que el sensor de la mitad detecte calor una vez el sensor de la mitad perciba el calor de la vela el carro encenderá el ventilador.

Codigo:

```
//sensore  
s de calor  
derecha a  
izquierda
```

```
#define s_t1 36  
#define s_t2 34  
#define s_t3 32  
#define s_t4 30  
#define s_t5 28
```

```
//ultrasonido  
#define ultra_trig_at 24  
#define ultra_echo_at 25  
#define ultra_trig_ad 26  
#define ultra_echo_ad 27
```

```
//sensores de color  
#define lineaCen 2 //verde  
#define lineaIzq1 4  
#define lineaIzq2 3  
#define lineaDer1 0  
#define lineaDer2 1
```



```
//motores
#define enableA 9
#define adelanteA 7
#define atrasA 6
#define adelanteB 5
#define atrasB 4
#define enableB 3


//ventilador
#define apa 29


//Velocidad general del carro , variables globales para
modificar la velocidad rapidamente
int velA = 50;
int velB = 55;


void setup() {
  pinMode(ultra_echo_at, INPUT);
  pinMode(ultra_echo_ad, INPUT);
  pinMode(s_t1, INPUT);
  pinMode(s_t2, INPUT);
  pinMode(s_t3, INPUT);
  pinMode(s_t4, INPUT);
  pinMode(s_t5, INPUT);
  pinMode(ultra_trig_at, OUTPUT);
```

```
pinMode(ultra_trig_ad, OUTPUT);

// pinMode(enableA,OUTPUT);
pinMode(enableB, OUTPUT);
pinMode (adelanteA, OUTPUT);
pinMode (atrasA, OUTPUT);
pinMode (adelanteB, OUTPUT);
pinMode (atrasB, OUTPUT);
pinMode (apa, OUTPUT);

pinMode (lineaCen, INPUT);
pinMode (lineaIzq1 , INPUT);
pinMode (lineaIzq2, INPUT);
pinMode (lineaDer1, INPUT);
pinMode (lineaDer2, INPUT);

Serial.begin(9600);

}

void loop() {
    int negro = analogRead(lineaCen); //Calibra el color
    inicial en este caso negro
    delay(200);
    arranque(); // Arranque
```

```
while (!sensor()) { //Mientras no detecte calor entra a
este ciclo
    if (analogRead(lineaCen) >= (negro - 100) &&
analogRead(lineaCen) <= (negro + 100) ) { //Se
establece una tolerancia para el color
        adelante(); // El carro se mueve hacia adelante
    } else { //Si encuentra blanco , se detiene y retrocede y
gira hacia la derecha
        apagar();
        atras();
        delay(100);
        apagar();
        delay(10);
        arranque();
        derecha();
        delay(200);
        apagar();
        delay(25);
    }
```

```
if ( !(analogRead(lineaIzq1) >= (negro - 100) &&
analogRead(lineaIzq1) <= (negro + 100)) ) { //Si
encuentra blanco a la izquierda , retrocede y se mueve a
la derecha
```

```
    atras();
    delay(200);
    apagar();
    delay(30);
```

```
    arranque();  
    derecha();  
    delay(200);  
    apagar();  
    delay(25);  
}
```

```
    if ( !(analogRead(lineaIzq2) >= (negro - 100) &&  
analogRead(lineaIzq2) <= (negro + 100)) ) { //Si  
encuentra blanco en el sensor interno de la izquierda  
retrocede y se mueve a la derecha
```

```
    atras();  
    delay(200);  
    apagar();  
    delay(30);  
    arranque();  
    derecha();  
    delay(200);  
    apagar();  
    delay(25);  
}
```

```
    if ( !(analogRead(lineaDer1) >= (negro - 100) &&  
analogRead(lineaDer1) <= (negro + 100)) ) { //Si  
encuentra blanco a la derecha retrocede y se mueve a la  
izquierda
```

```
    atras();  
    delay(200);  
    apagar();  
    delay(30);  
    arranque();  
    izquierda();  
    delay(200);  
    apagar();  
    delay(25);  
}
```

```
    if ( !(analogRead(lineaDer2) >= (negro - 100) &&  
analogRead(lineaDer2) <= (negro + 100)) ) { //Si  
encuentra blanco en el sensor interno de la derecha ,  
retrocede y se mueve a la izquierda
```

```
    atras();  
    delay(200);  
    apagar();  
    delay(30);  
    arranque();  
    izquierda();  
    delay(200);  
    apagar();  
    delay(25);  
}  
}
```

```
        if((digitalRead(s_t1)==1||digitalRead(s_t2))&&
digitalRead(s_t3)){ // Si alguno de los sensores de la
derecha detecta calor , gira y enciende el ventilador
    digitalWrite(apa,1);
    izquierda();
    delay(100);
    adelante();

        while((digitalRead(s_t3)==1    &&
digitalRead(s_t4)==1    &&    digitalRead(s_t2)==1)    ||
(digitalRead(s_t3)==1    &&    digitalRead(s_t4)==1)    ||
(digitalRead(s_t4)==1 && digitalRead(s_t2)==1) ){ //
Mientras esta detectando calor en el centro , enciende el
ventilador y lo apaga
    apagar();
    digitalWrite(apa,1);
    delay(300);
    //digitalWrite(apa,0);
    adelante();
    delay(80);
}
    apagar();
    digitalWrite(apa,0);

    //...
}

        if((digitalRead(s_t1)==1||digitalRead(s_t2))&&
!(digitalRead(s_t3) ==1)){//Gira hacia la izquierda
```


mientras el sensor de la mitad detecte calor enciende el ventilador y se mueve hacia adelante

```
digitalWrite(apa,1);  
izquierda();  
delay(100);  
adelante();  
while((digitalRead(s_t3)==1 &&  
digitalRead(s_t4)==1 && digitalRead(s_t2)==1) ||  
(digitalRead(s_t3)==1 && digitalRead(s_t4)==1) ||  
(digitalRead(s_t4)==1 && digitalRead(s_t2)==1)  
){//Mientras esta detectando calor en el centro , enciende  
el ventilador.  
    apagar();  
    digitalWrite(apa,1);  
    delay(300);  
    //digitalWrite(apa,0);  
    adelante();  
    delay(80);  
}  
apagar();  
digitalWrite(apa,0);  
  
//... }  
  
}
```

```
if((digitalRead(s_t4)==1||digitalRead(s_t5))&&  
(digitalRead(s_t3) ==1) ){//Si alguno de los sensores de  
la derecha detecta calor , gira hacia la derecha y va hacia  
adelante
```

```
digitalWrite(apa,1);
derecha();
delay(100);
adelante();

while((digitalRead(s_t3)==1 &&
digitalRead(s_t4)==1 && digitalRead(s_t2)==1) ||
(digitalRead(s_t3)==1 && digitalRead(s_t4)==1) ||
(digitalRead(s_t4)==1 && digitalRead(s_t2)==1) ){
//Mientras que este detectando calor enciende el
ventilador
    apagar();
    digitalWrite(apa,1);
    delay(300);
    //digitalWrite(apa,0);
    adelante();
    delay(80);
}

apagar(); //Se apaga el carro
digitalWrite(apa,0); //Se apaga el ventilador

}

//Si detecta calor en la derecha , aunque el sensor del
medio no lo haga, gira a la derecha , va hacia adelante y
enciende el ventilador hasta apagar la vela
if((digitalRead(s_t4)==1 || digitalRead(s_t5))&&
!(digitalRead(s_t3) ==1) ){
//...
digitalWrite(apa,1);
derecha();
```

```
        delay(100);  
        adelante();  
        while((digitalRead(s_t3)==1  &&  
digitalRead(s_t4)==1  &&  digitalRead(s_t2)==1)  ||  
(digitalRead(s_t3)==1  &&  digitalRead(s_t4)==1)  ||  
(digitalRead(s_t4)==1 && digitalRead(s_t2)==1) ){  
            apagar();  
            digitalWrite(apa,1);  
            delay(300);  
            //digitalWrite(apa,0);  
            adelante();  
            delay(80);  
        }  
        apagar(); //Se apaga el carro  
        digitalWrite(apa,0); // Se apaga el ventilador  
  
    }  
  
}
```

```
//Este metodo determina si alguno de los sensores  
termicos ha detectado calor
```

```
bool sensor() {  
  
    if (digitalRead(s_t1) == 1) {
```

```
        return true;
    }
    if (digitalRead(s_t2) == 1) {
        return true;
    }
    if (digitalRead(s_t3) == 1) {
        return true;
    }
    if (digitalRead(s_t4) == 1) {
        return true;
    }
    if (digitalRead(s_t5) == 1) {
        return true;
    }
    return false;
}
```

// Este metodo hace que el carro vaya derecho

```
void adelante() {
    analogWrite(enableA, velA);
    digitalWrite(adelanteA, 1);
    analogWrite(enableB, velB);
    digitalWrite(adelanteB, 1);
}
```

```
// Este metodo hace que el carro retroceda durante 0.1 segundos
```

```
void atras() {  
    analogWrite(enableA, 70);  
    digitalWrite(atrasA, 1);  
    analogWrite(enableB, 77);  
    digitalWrite(atrasB, 1);  
    delay(100);  
}
```

```
//Este metodo gira hacia la derecha, apagando la rueda de la derecha y usando la velocidad general del carro para la otra rueda
```

```
void derecha() {  
    analogWrite(enableA, velA);  
    digitalWrite(adelanteA, 1);  
    analogWrite(enableB, 0);  
    digitalWrite(adelanteB, 0);  
}
```

```
//Este metodo gira hacia la izquierda apagando la rueda de la izquierda y usando la velocidad general del carro para la otra rueda
```

```
void izquierda() {  
    analogWrite(enableA, 0);
```

```
digitalWrite(adelanteA, 0);  
analogWrite(enableB, velB);  
digitalWrite(adelanteB, 1);  
}
```

//Este metodo apaga los motores del carro

```
void apagar() {  
    analogWrite(enableA, 0);  
    digitalWrite(adelanteA, 0);  
    analogWrite(enableB, 0);  
    digitalWrite(adelanteB, 0);  
    digitalWrite(atrasA, 0);  
    digitalWrite(atrasB, 0);  
}
```

//Este metodo arranca el carro con una velocidad mas alta de la requerida para moverse por el circuito

```
void arranque() {  
    analogWrite(enableA, 70);  
    digitalWrite(adelanteA, 1);  
    analogWrite(enableB, 77);  
    digitalWrite(adelanteB, 1);  
    delay(100);  
}
```


Referencias:

*<https://www.prometec.net/motorcc/>

*<http://www.tecnosefarad.com/2014/03/sensor-de-infrarrojos-cny70-como-entrada-digital/>