

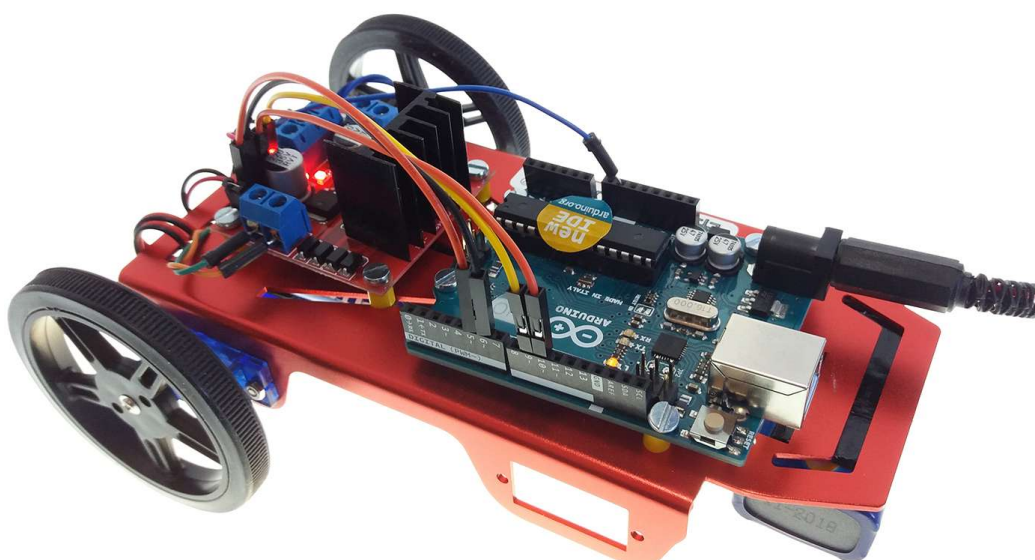
# COCHE PROGRAMADO

MICRO-LOG<sup>®</sup>  
LOGKIT  
4207A

## 1. OBJETIVOS

Construir un coche robótico, el cual se programa para seguir unos movimientos determinados.

## 2. FOTOGRAFÍA



## 3. FUNCIONAMIENTO

El coche se programa con "Ardublock" o "Processing", para que accione los motores y siga una ruta preestablecida.

## 4. LISTA DE MATERIALES

- |                                 |   |
|---------------------------------|---|
| 1 Arduino UNO REV-3 LOG 4031    | 1 Controlador de motores L298N LOG 4044 |
| 1 Cable USB A-B LOG 4009        | 1 Conector 9V para Arduino LOG 7734     |
| 1 Plataforma móvil LOG 4080P    | 1 Latiguillo macho-macho S 9519         |
| 8 Tornillos M3 x 16 mm LOG 464M | 4 Latiguillos macho-hembra S9518        |
| 8 Tuercas M3 LOG 480            | 8 Bulones de plástico S 220P            |
| 1 Portapilas 4xR6 LOG 534       | 1 Hoja Técnica H4207A                   |

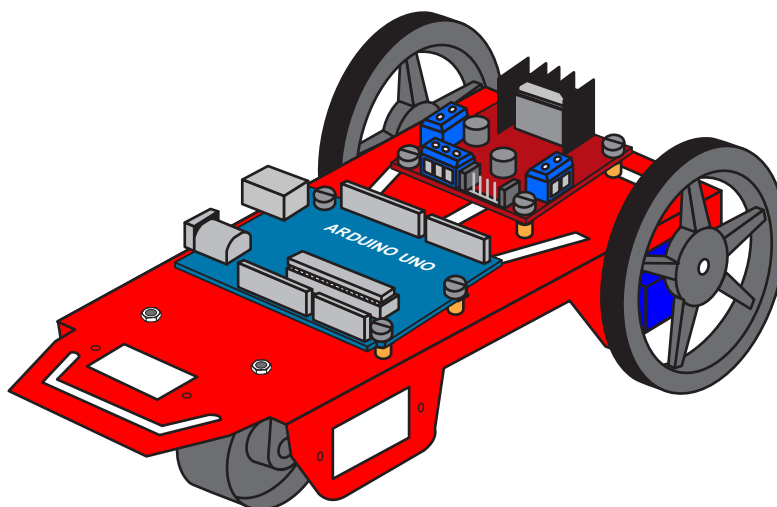
Leer todas las instrucciones y comprobar el listado de materiales antes de empezar el proyecto.

## COCHE PROGRAMADO

MICRO-LOG<sup>®</sup>  
LOGKIT  
4207A

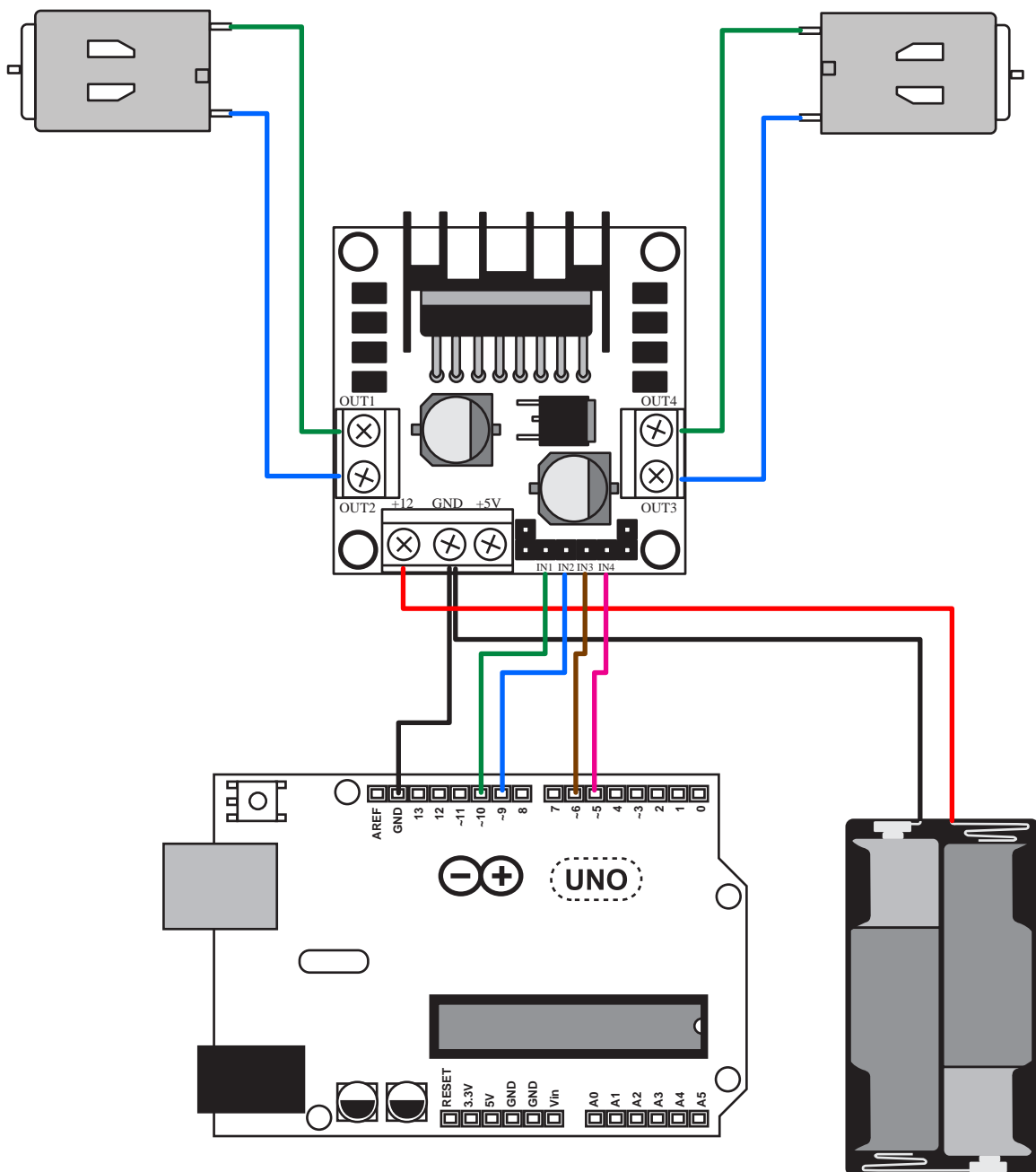
### 5. CONSTRUCCIÓN

- a) Montar la plataforma móvil según sus instrucciones.
- b) Cortar y pelar los cables de la reductora (es aconsejable dar una punta de estaño para fijarlos bien).
- c) Atornillar el Arduino UNO en la plataforma superior, utilizando las torretas pequeñas que incorpora la plataforma.
- d) Atornillar el controlador de motores con los tornillos LOG 464, los bulones de plástico y la tuerca LOG 480.



## 6. CONEXIÓN DEL CONTROL DE MOTORES

- El esquema incluye un controlador de motores L298N puente H. Un puente H consiste en 4 interruptores que nos permiten controlar la polaridad de la corriente que llega a los dos motores conectados al controlador de motores. Así en función de la polaridad podemos controlar el sentido de giro de los motores. Además funcionan como regulador de corriente, pudiendo determinar una velocidad de giro de 0 a 255.
- Para poder controlar la velocidad de giro de los motores, se han conectado a las salidas 5, 6, 9 y 10 de Arduino que permiten la modularidad (PWM).



## 8. PROGRAMACIÓN CON ARDUBLOCK

Ardublock es una herramienta que podemos instalar en Arduino IDE para facilitar los programas. Consiste en un sistema de programación basado en bloques similar a scratch. La principal diferencia radica en que los programas elaborados con ardublock pueden ejecutarse con la tarjeta controladora en modo autónomo, es decir, desconectada del PC

Para instalar Ardublock en el software Arduino IDE, seguimos estos pasos:

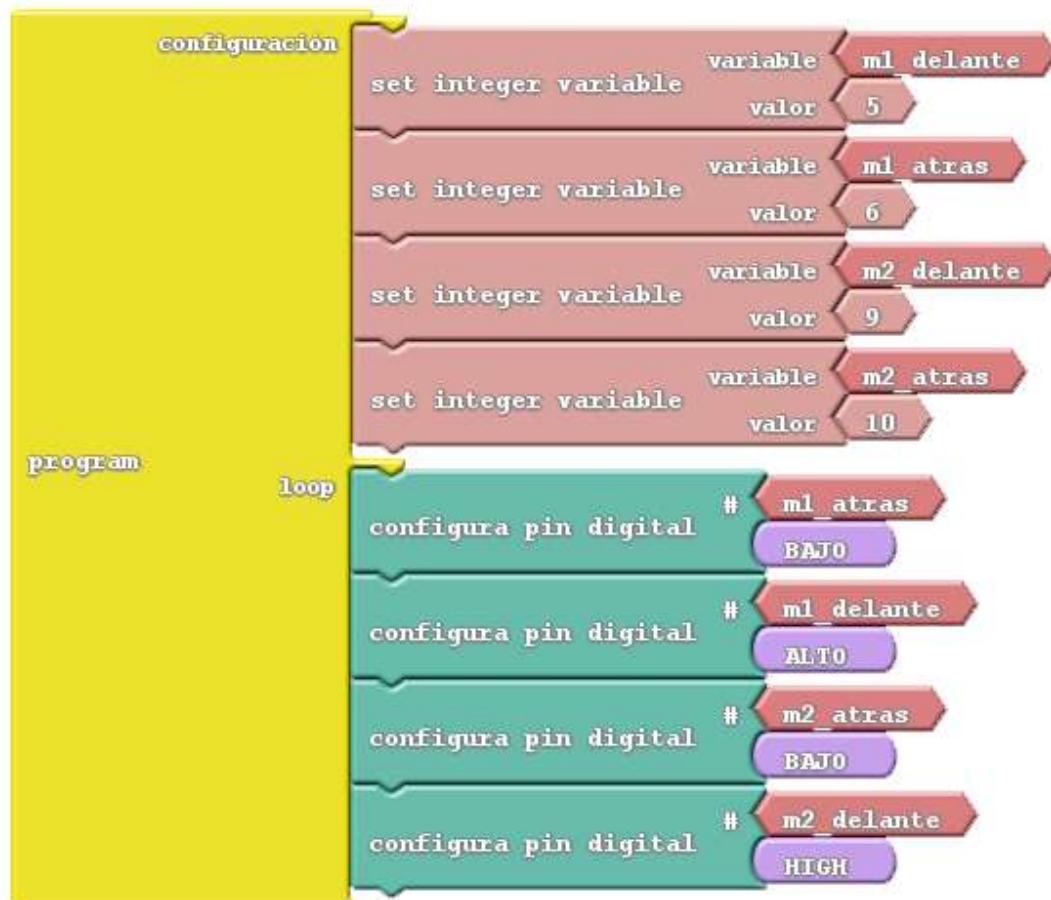
1. Descargamos la carpeta "ArduBlockTool" que podrá localizarse dentro de las características del producto "LOG 4207A" en la tienda virtual.
2. Buscamos en nuestro ordenador la carpeta donde se instaló Arduino (normalmente en la carpeta Arduino dentro de archivos de programa).
3. Dentro de la carpeta Arduino localizamos la carpeta tools.
4. Copiamos la carpeta "ArduBlockTool " dentro de la carpeta tools.

Si ejecutamos Arduino IDE tenemos que localizar dentro del menú herramientas una nueva entrada llama Ardublock que nos da acceso a la nueva plataforma instalada.

### INSTRUCCIONES:

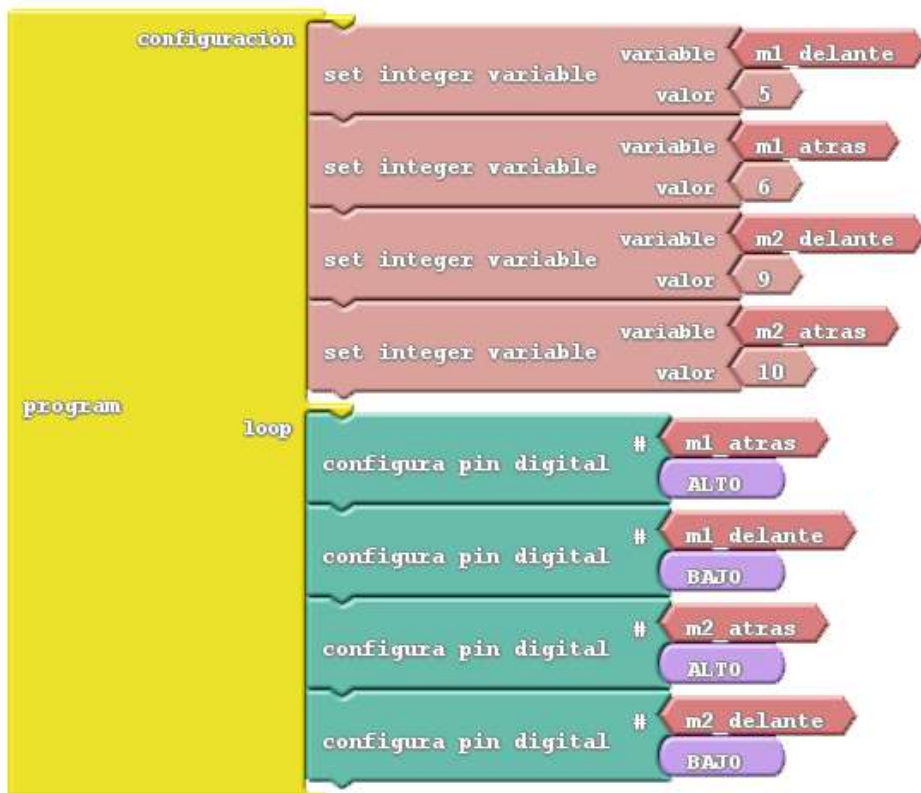
- Combinar las siguientes acciones intercalando "esperas o delays", combinarlo con bucles y condicionales, para que el coche siga una ruta deseada.

### ADELANTE

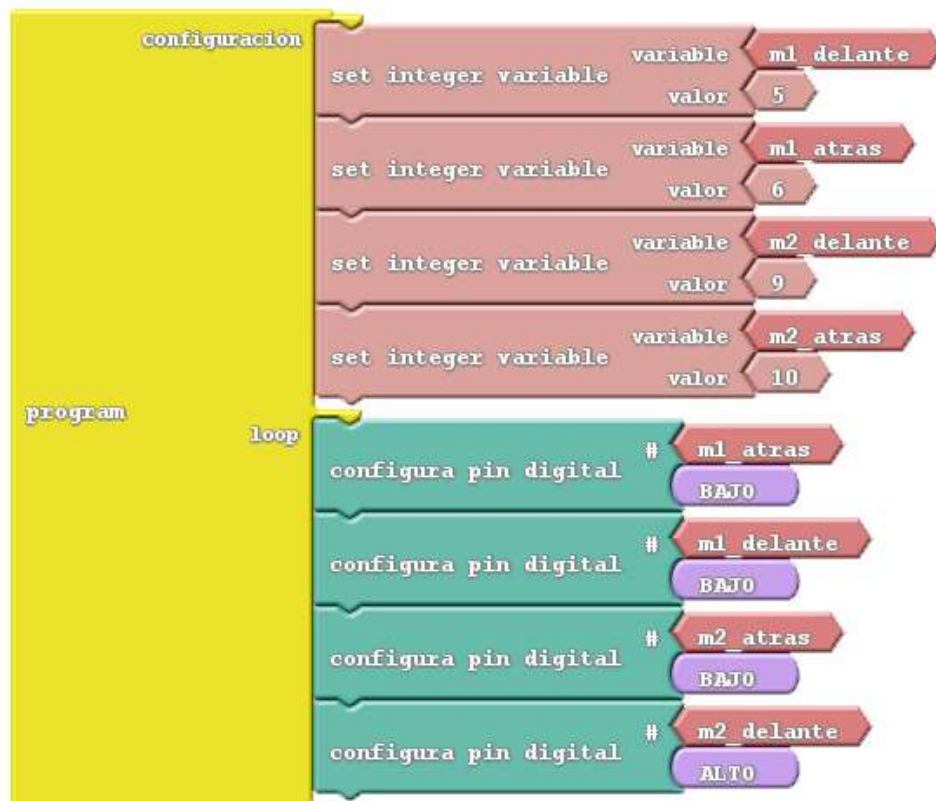


# COCHE PROGRAMADO

## ATRÁS



## IZQUIERDA



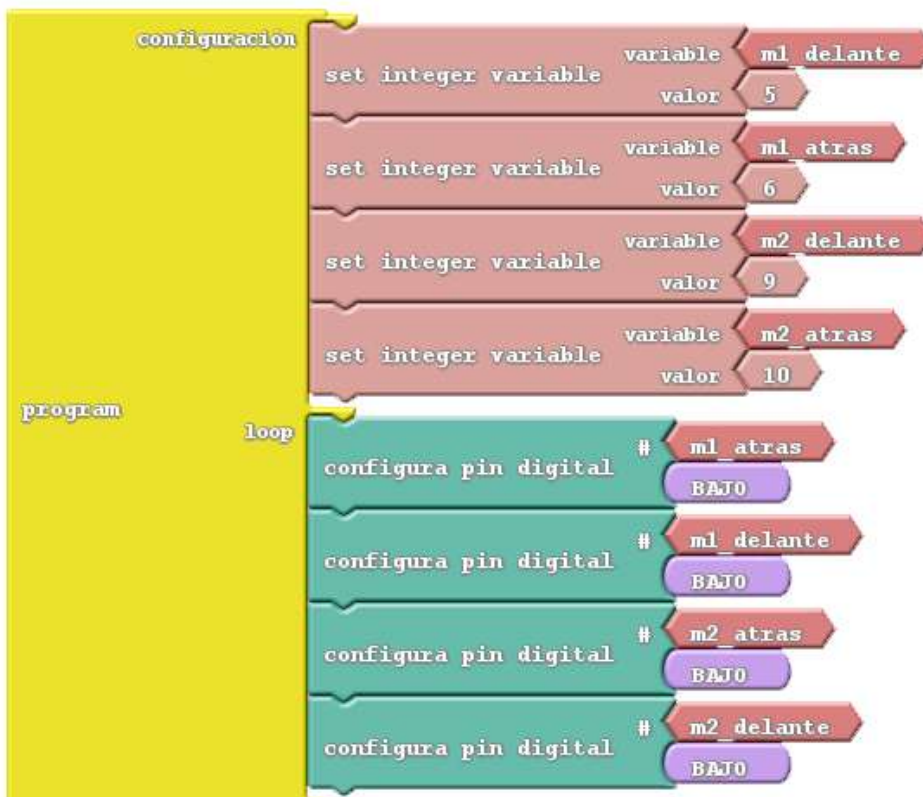


# COCHE PROGRAMADO

## DERECHA

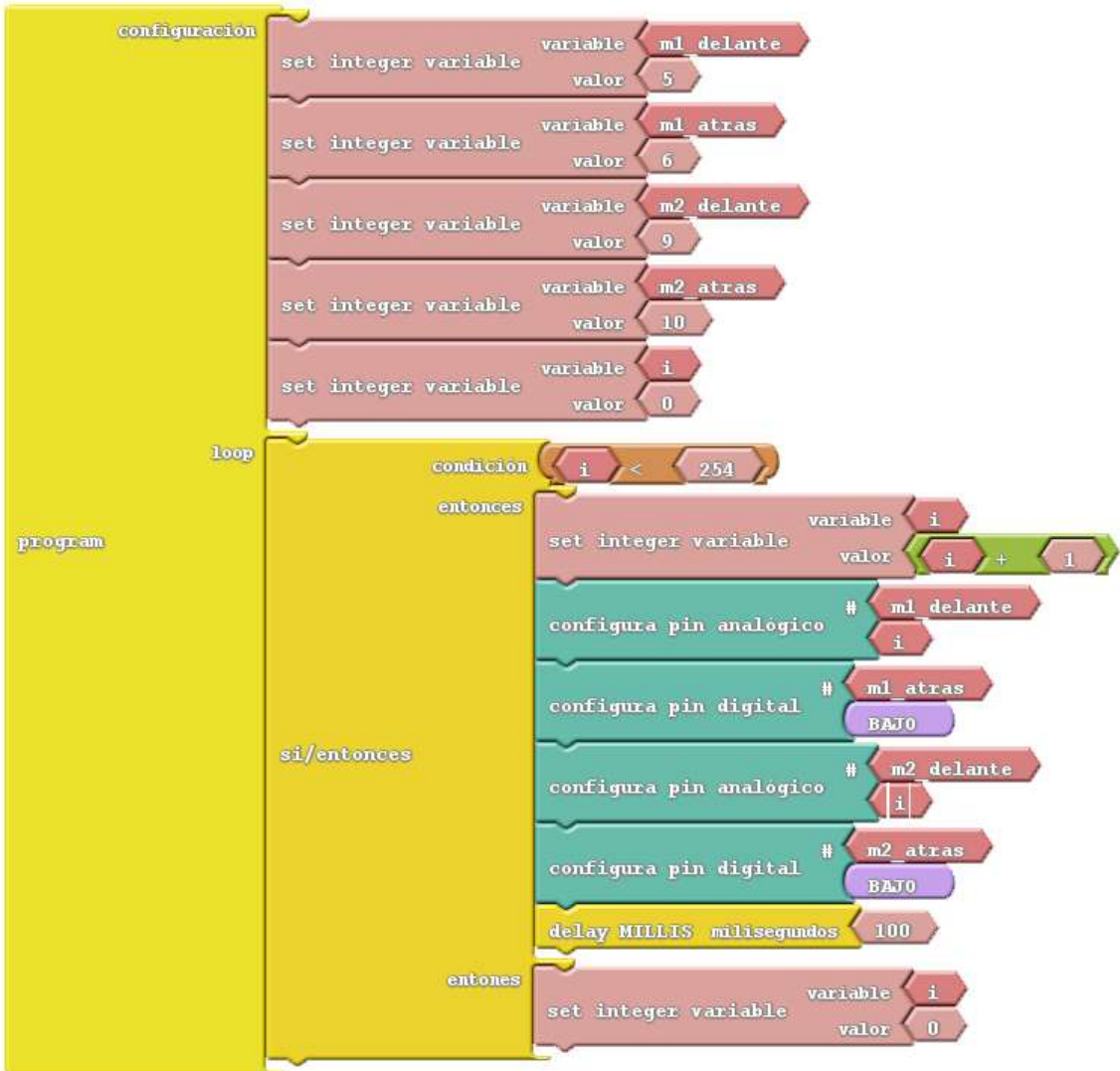


## PARAR



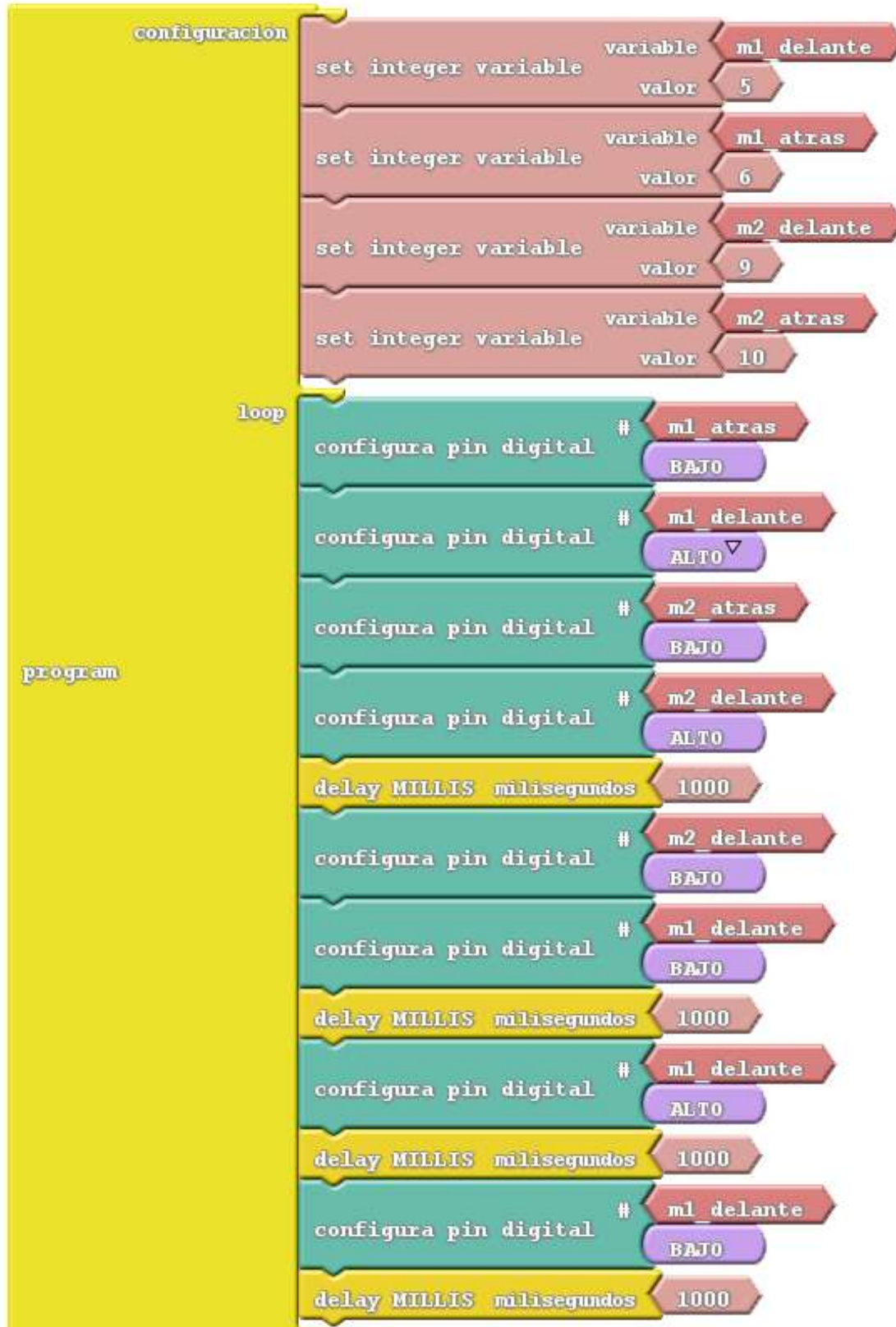
# COCHE PROGRAMADO

## ACELERAR



# COCHE PROGRAMADO

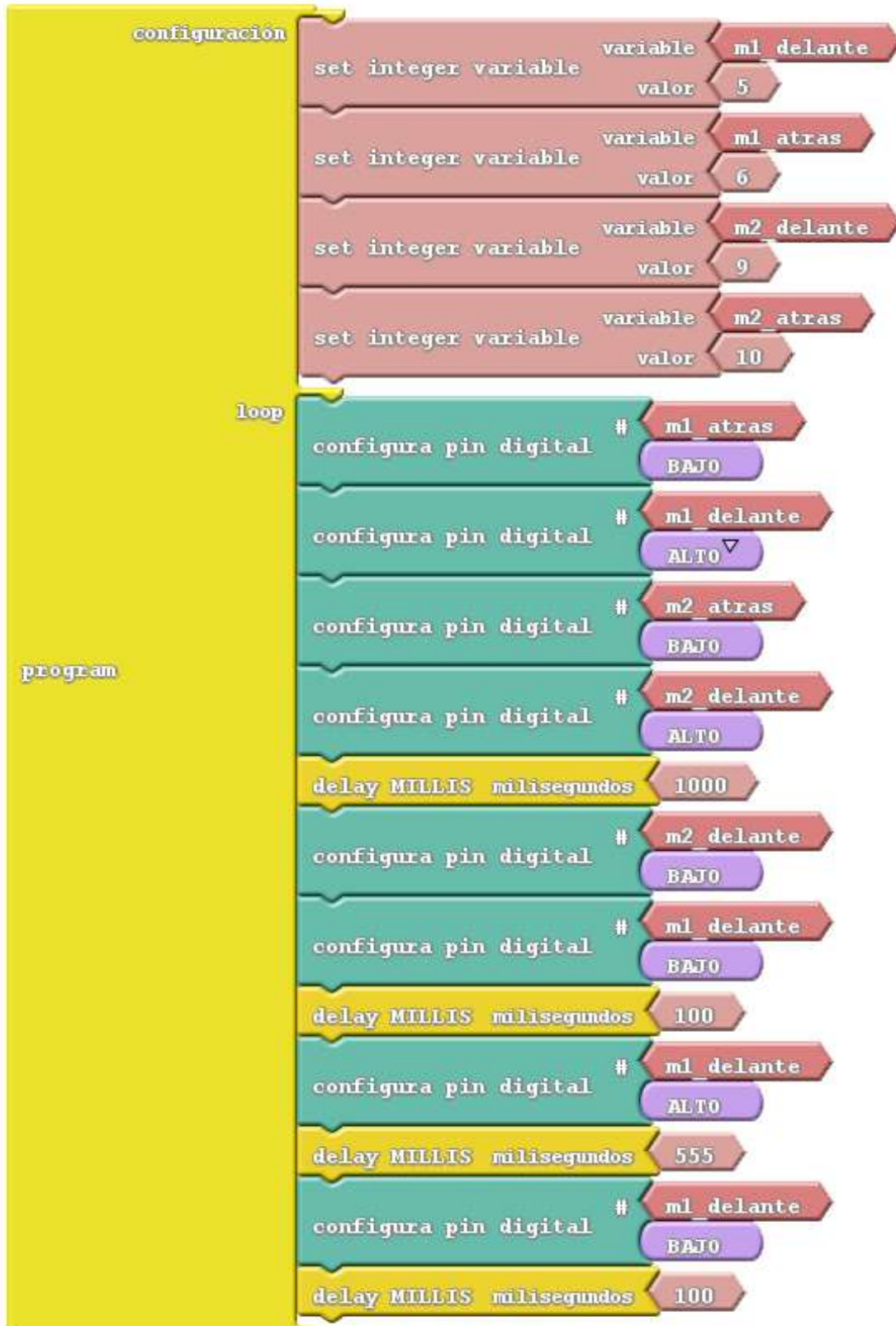
## EJEMPLO DE RUTA CIRCULAR





# COCHE PROGRAMADO

## EJEMPLO DE RUTA CUADRADA



## 10. PROGRAMACIÓN EN PROCESSING

```
/* INSTRUCCIONES BÁSICAS PARA DESPLAZAR EL COCHE */

int m1_delante=5;
int m1_atras=6;
int m2_delante=9;
int m2_atras=10;

void setup() {

  pinMode(m1_delante, OUTPUT);
  pinMode(m1_atras, OUTPUT);
  pinMode(m2_delante, OUTPUT);
  pinMode(m2_atras, OUTPUT);

}

void loop() {

//ADELANTE

  digitalWrite(m1_delante,HIGH);
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,HIGH);
  digitalWrite(m2_atras, LOW);

//ATRAS

  digitalWrite(m1_delante,LOW);
  digitalWrite(m1_atras, HIGH);
  digitalWrite(m2_delante,LOW);
  digitalWrite(m2_atras, HIGH);

//IZQUIERDA

  digitalWrite(m1_delante, LOW);
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante, HIGH);
  digitalWrite(m2_atras, LOW);

//DERECHA

  digitalWrite(m1_delante,HIGH);
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,LOW);
  digitalWrite(m2_atras, LOW);

//ACELERAR

  if (i<254) {
    i=i+1;
    analogWrite(m1_delante,i);
    digitalWrite(m1_atras, LOW);
    analogWrite(m2_delante,i);
    digitalWrite(m2_atras, LOW);
    delay(100);
  }
  else{
    i=0;
  }

//PARAR

  digitalWrite(m1_delante,LOW);
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,LOW);
  digitalWrite(m2_atras, LOW);

}
```

## COCHE PROGRAMADO

```
/* EJEMPLO DE RUTA CIRCULAR */

int m1_delante=5;
int m1_atras=6;
int m2_delante=9;
int m2_atras=10;

void setup() {

  pinMode(m1_delante, OUTPUT);
  pinMode(m1_atras, OUTPUT);
  pinMode(m2_delante, OUTPUT);
  pinMode(m2_atras, OUTPUT);

}

void loop() {

  digitalWrite(m1_delante,HIGH); //recto
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,HIGH);
  digitalWrite(m2_atras, LOW);
  delay(100);

  digitalWrite(m1_delante,LOW); //paro
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,LOW);
  digitalWrite(m2_atras, LOW);
  delay(100);

  digitalWrite(m1_delante,HIGH); //giro
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,LOW);
  digitalWrite(m2_atras, LOW);
  delay(100);

  digitalWrite(m1_delante,LOW); //paro
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,LOW);
  digitalWrite(m2_atras, LOW);
  delay(100);

}
```

## COCHE PROGRAMADO

```
/* EJEMPLO DE RUTA CUADRADA */

int m1_delante=5;
int m1_atras=6;
int m2_delante=9;
int m2_atras=10;

void setup() {

  pinMode(m1_delante, OUTPUT);
  pinMode(m1_atras, OUTPUT);
  pinMode(m2_delante, OUTPUT);
  pinMode(m2_atras, OUTPUT);

}

void loop() {

  digitalWrite(m1_delante,HIGH); //recto
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,HIGH);
  digitalWrite(m2_atras, LOW);
  delay(1000);

  digitalWrite(m1_delante,LOW); //paro
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,LOW);
  digitalWrite(m2_atras, LOW);
  delay(100);

  digitalWrite(m1_delante,HIGH); //giro
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,LOW);
  digitalWrite(m2_atras, LOW);
  delay(555);

  digitalWrite(m1_delante,LOW); //paro
  digitalWrite(m1_atras, LOW);
  digitalWrite(m2_delante,LOW);
  digitalWrite(m2_atras, LOW);
  delay(100);

}
```

## COCHE PROGRAMADO

 MICRO-LOG<sup>®</sup>  
**LOGKIT  
4207A**

### 11. DETALLES DE TIPO PRÁCTICO

- Desconectar la alimentación del controlador de motores L298N cuando no se este utilizando para no gastar las pilas.
- Necesita 4 pilas R6 de 1,5 V y 1 pila 6F22 de 9 V.
- Tiempo de construcción: 6 H.
- Nivel: Difícil

### 12. PRUEBAS

- Cambiar la programación para que siga una ruta programada.