



Tecnológico de Monterrey

Campus Guadalajara

Profesores: María José Gutiérrez Martínez de Castro y Sofía Gudiño Arroyo

Proyecto de Introducción a la Ingeniería Mecatrónica

Robot Seguidor de línea



Alumnos:

Fabrizzio Octavio Cortez Pacheco A01633220

Marco Alonso Barajas Contreras A01229230

Leonardo Moisés Espinosa Cervera A01351259

Jorge Enrique Esqueda López A01633959

Aitor Cifuentes Quijas A01634435

Introducción

La finalidad de este proyecto es diseñar, construir, programar e implementar un robot capaz de seguir una línea negra sobre una superficie blanca al integrar la electrónica, la mecánica, la programación y el control.

En esta era tecnológica donde la necesidad de trabajos simultáneos son casi indispensables se tiene una alternativa al transporte dentro de una empresa, una aplicación real para un seguidor de línea podría ser en cualquier industria donde se necesitan plataformas o robots de transporte autónomos que sean capaces de seguir una línea y transportar piezas necesarias para ciertos procesos a lo largo de toda la fábrica sin necesidad de un humano que las controle.

Marco Teórico

En la figura 1 se muestra el diseño del sistema de control de sentido de giro y velocidad en diagrama a bloques que es esencial para el correcto funcionamiento del robot seguidor de línea:

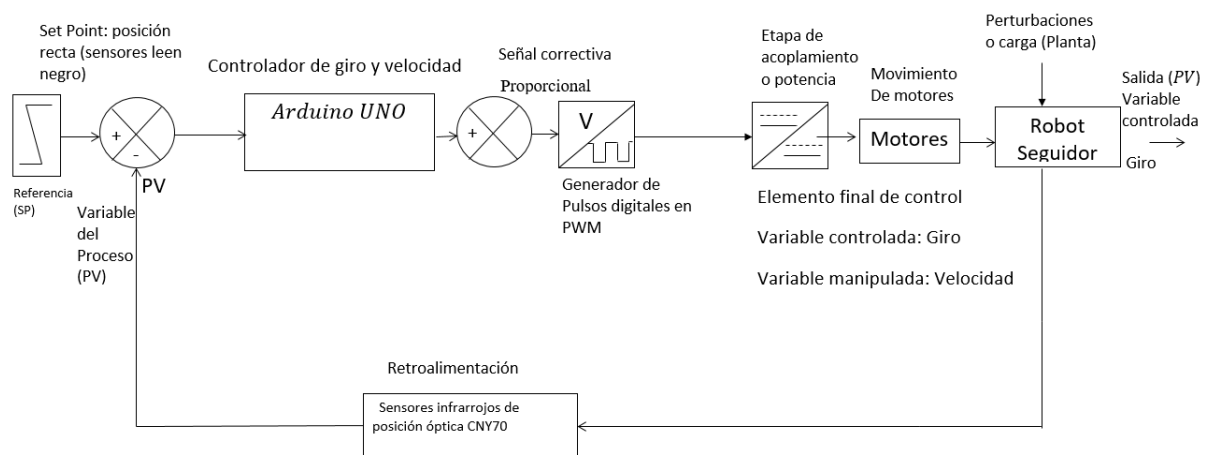


Figura 1. Diagrama a bloques del Sistema de Control de Sentido de Giro y Velocidad.

Al inicio del diagrama podemos ver la asignación del set point o del punto de referencia de la forma de movimiento de nuestro robot que va ser un movimiento rectilíneo uniforme. Lo que va a significar que los 2 sensores ópticos CNY70 leen negro y si en la vida real existe un error en esas 2 condiciones, el error se envía al controlador Arduino UNO el cual corrige ese error, de manera proporcional, al mandar una señal de modulación de ancho de pulso (PWM) hacia una etapa de potencia que modifica el giro y la velocidad de los 2 motores que van a afectar la planta o el robot seguidor.

Podemos conocer hacia donde se va el robot seguidor gracias a la retroalimentación de los sensores, si estos 2 bajan su voltaje es porque los sensores están leyendo negro porque la configuración de la PCB está diseñado para que los sensores lean voltajes menores cuando ven negro y voltajes mayores cuando ven blanco. Por lo tanto, esos valores de los sensores se envían al controlador Arduino que los interpreta como señales analógicas que serán

convertidas en señales digitales por los ADC internos del microcontrolador con un rango de (0-1023) y que al leer negro va leer un valor bajo de aproximadamente 100 y 300 para el blanco. Por lo tanto, podemos conocer que sensor está activado y por lo tanto darle una tarea al microcontrolador Arduino que corrija la orientación y velocidad (Por la modulación de ancho de pulso PWM) del seguidor de línea al mover solo uno de los motores mientras que el otro está apagado.

Desarrollo:

Primero se diseñó el chasis del seguidor de línea con ayuda de un software CAD (Computer Aided Design) donde se basaron todas las medidas relevantes a considerar tales como el ancho de la cinta negra a seguir, las dimensiones máximas posibles, el tamaño de los motores, llantas, arduino, circuito impreso, fuente de baterías, rueda loca y la altura de los sensores al suelo.

Después se generó esa base del carro con ayuda del AutoCAD 2017 y se guardó en formato .DXF para luego cortarla a laser. La base se puede observar en la figura 2:

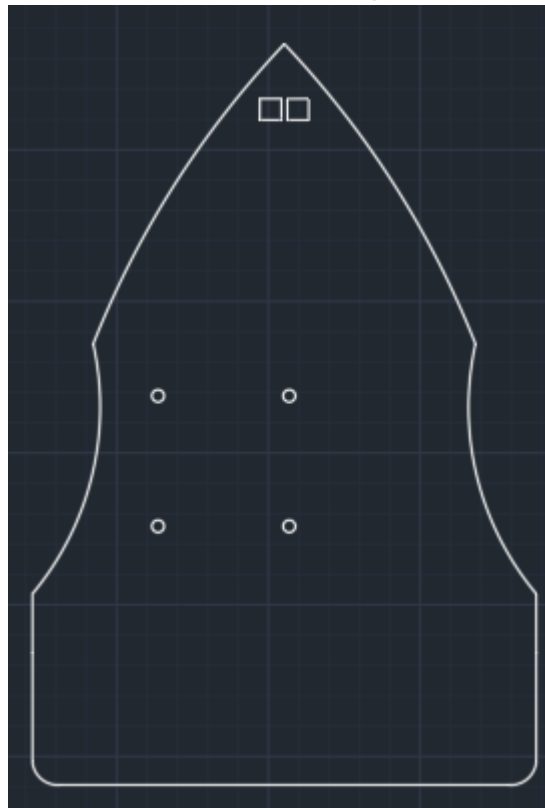


Figura 2. Diseño del seguidor de línea

Después, se diseñó la PCB para que los sensores detectaran más voltaje cuando leen blanco y menos cuando leen negro debido a la configuración de emisor común de los fototransistores de los CNY70. Además, se utilizó la configuración de 2 transistores TIP41C como interruptores, para controlar el encendido y apagado de los motores a la velocidad de recepción de la señal PWM que enviará la tarjeta para disminuir la potencia de un motor.

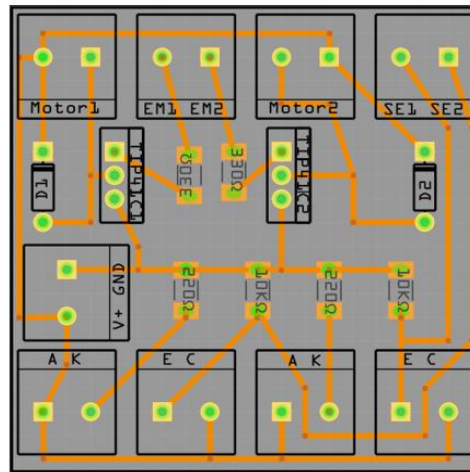


Figura 3. Diseño de la PCB del seguidor de línea

En esta figura 3 se observa que la PCB (Printed Circuit Board) o circuito impreso tiene primero las terminales de voltaje V+ y tierra GND o el negativo de nuestra fuente que es donde se van a conectar las 4 pilas de voltaje. Después, la fuente alimenta a los 2 sensores infrarrojos y a los 2 colectores de los 2 transistores que están en paralelo con cada uno de los diodos de protección y que están esperando el momento de activarse con la ayuda de la base de cada transistor. Después, la información de cada sensor infrarrojo compuesto de ánodo, cátodo, emisor y colector donde el emisor de cada sensor se van a las terminales o clemas SE1 y SE2 que son la salida de la información de los 2 sensores pero esos van a ser la entrada del microcontrolador del Arduino. Después, el Arduino procesa la información de esos 2 sensores y con su programa manda otras terminales de salida hacia las clemas EM1 y EM2. Después, las señales de EM1 Y EM2 se van a las resistencias que activan a las bases de cada transistor a través de un voltaje inyectado. Por lo tanto, la corriente ya puede fluir entre colector y emisor que hacen que cada motor gire y luego se vaya a tierra y pasan por los diodos que sirven de protección.

Luego se realizó el proceso de elaboración del circuito impreso por medio de la técnica del planchado que tiene el paso de planchado del circuito, utilización de cloruro férrico, perforación de la PCB y soldado de componentes en el cual en la figura 4 se ve el segundo paso del método que es antes y después de haber utilizado con precaución el cloruro férrico.

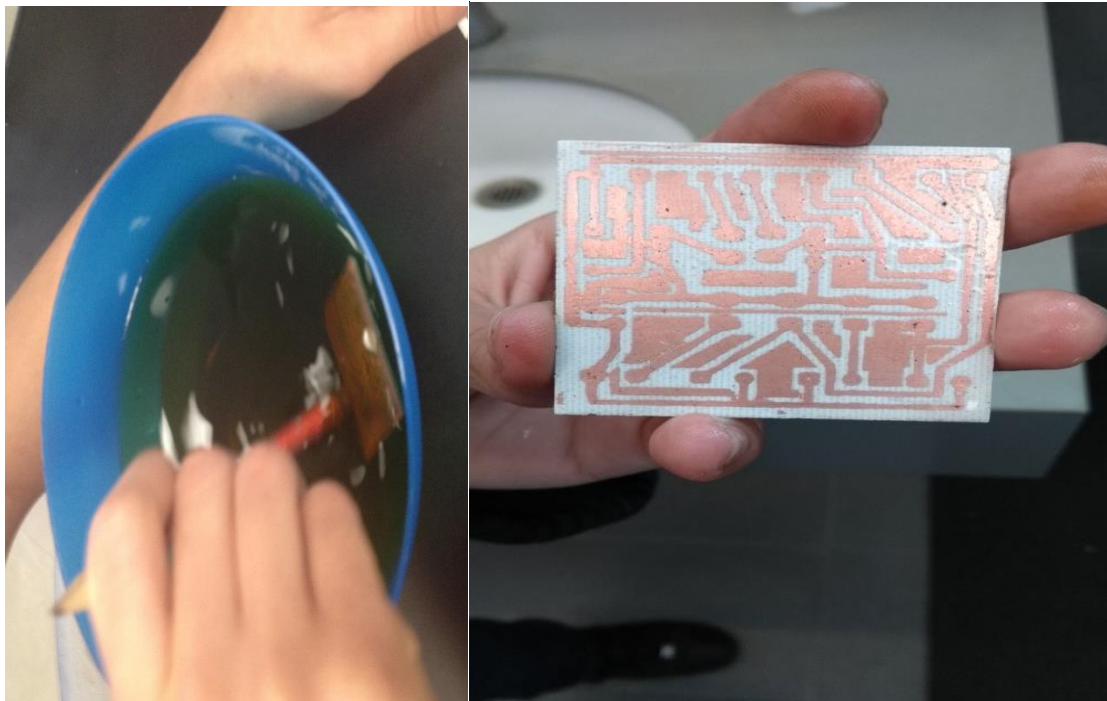


Figura 4. Elaboración con cloruro férrico de la PCB del seguidor de línea

Como parte del desarrollo del proyecto tuvimos prácticas de programación cuyo objetivo fue familiarizarnos con la programación de la plataforma Arduino.

Lectura de sensores CNY70:

El programa es básicamente declarar 2 variables donde uno funcione para configurar el sensor como entrada analógica y que va conectado al microcontrolador y la otra variable para ser una variable auxiliar que ayude a leer las diferentes lecturas del sensor analógico e infrarrojo CNY70. Después de obtener el valor del sensor se guarda ese valor en la variable auxiliar y se imprime ese valor en el puerto serial del Arduino.

```

Sensor_read Arduino 1.6.7
Archivo Editar Programa Herramientas Ayuda

Sensor_read

int sensorderechopin = A0;
int sensorizquierdopin = A1;
float value = 0;
float value2=0;
float voltaje_derecho;
float voltaje_izquierdo;

void setup() {
  Serial.begin (9600);
  pinMode (sensorderechopin,INPUT);
  pinMode (sensorizquierdopin,INPUT);
}

void loop() {
  value = analogRead(sensorderechopin);
  value2=analogRead(sensorizquierdopin);
  voltaje_derecho=(value*(5))/1023;
  voltaje_izquierdo=(value2*(5))/1023;
  Serial.print("PRRO El sensor derecho dice: ");
  Serial.println (voltaje_derecho,DEC);
  Serial.print("PRRO El sensor izquierdo dice: ");
  Serial.println (voltaje_izquierdo,DEC);
  delay (1000);
}
  
```

Figura 5. Código de lectura de los sensores CNY70

La figura 6 muestra los sensores infrarrojos encendidos que envían la información analógica hacia el Arduino.

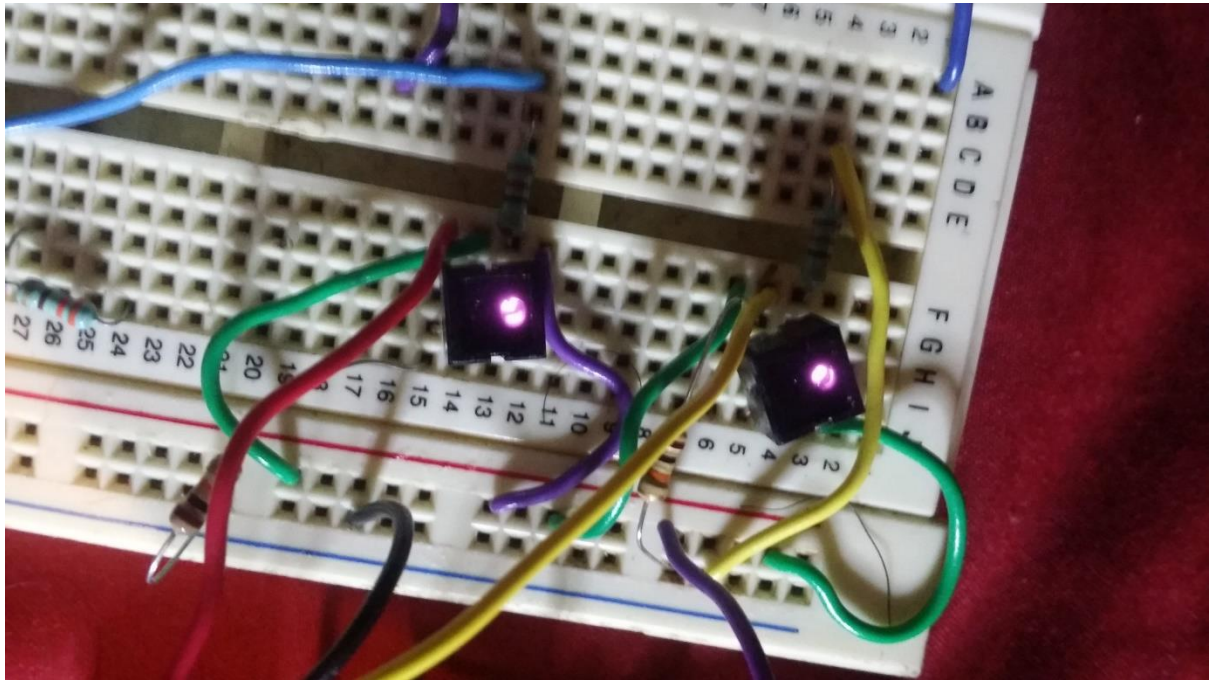
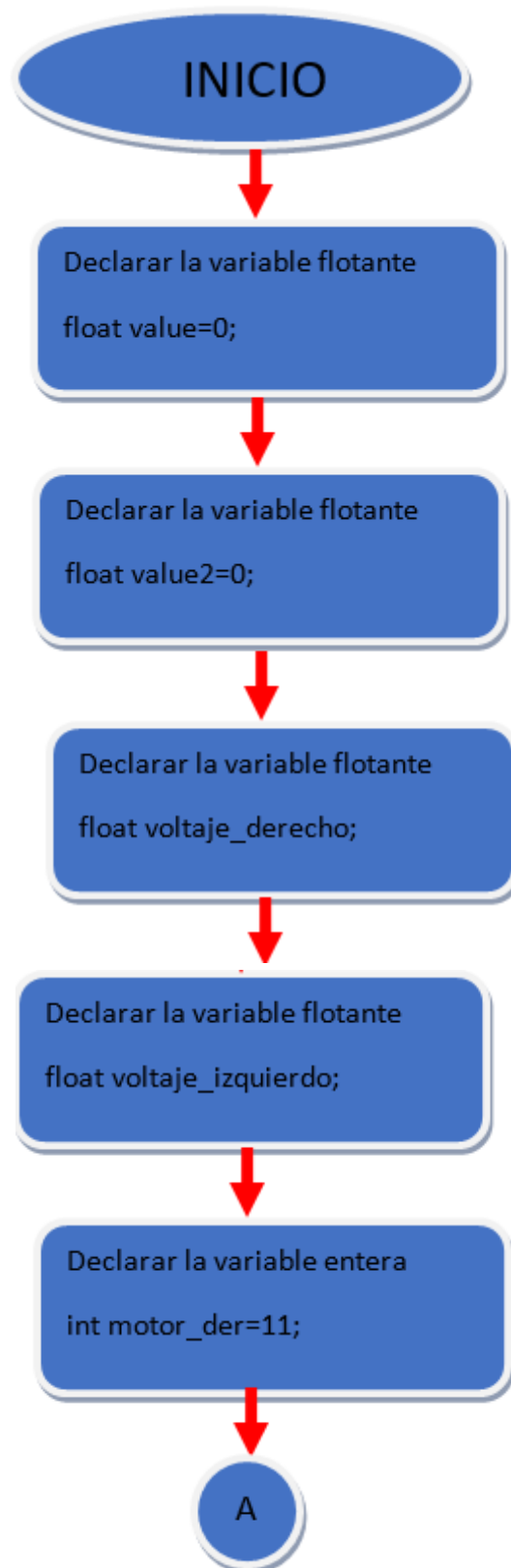


Figura 6. Funcionamiento de los sensores infrarrojos

EQUIPO 1 DIAGRAMA DE FLUJO Y PWM



A

Declarar la variable entera
`int motor_izq=10;`

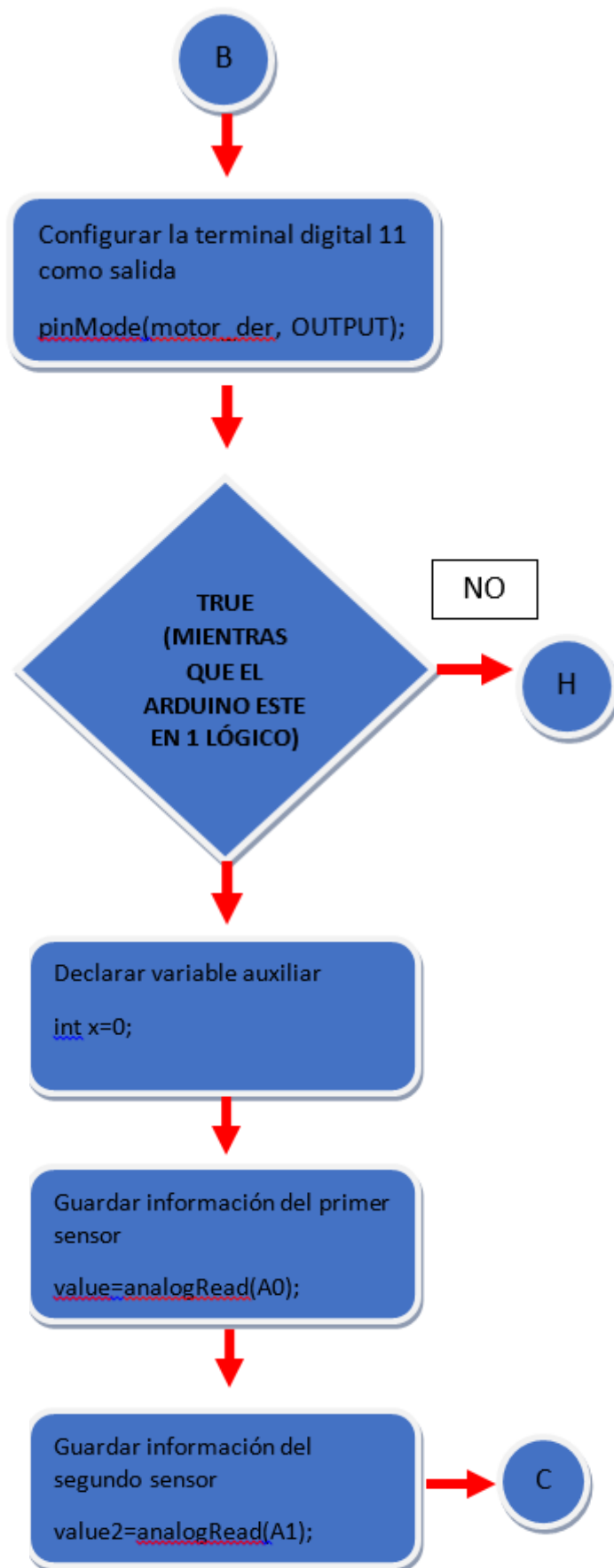
Iniciar comunicación serial a
9600 bits/segundo
`Serial.begin(9600);`

Configurar la terminal A0 como
entrada
`pinMode(A0, INPUT);`

Configurar la terminal A1 como
entrada
`pinMode(A1, INPUT);`

Configurar la terminal digital 10
como salida
`pinMode(motor_izq, OUTPUT);`

B



C

Proceso de conversión de valores
a voltaje del sensor derecho
 $\text{voltaje_derecho} = (\text{value} * 5) / 1023;$

Proceso de conversión de valores a
voltaje del sensor izquierdo
 $\text{voltaje_izquierdo} = (\text{value2} * 5) / 1023;$

Imprimir texto en el monitor
serial

```
Serial.print("PRRO El sensor  
derecho dice:");
```

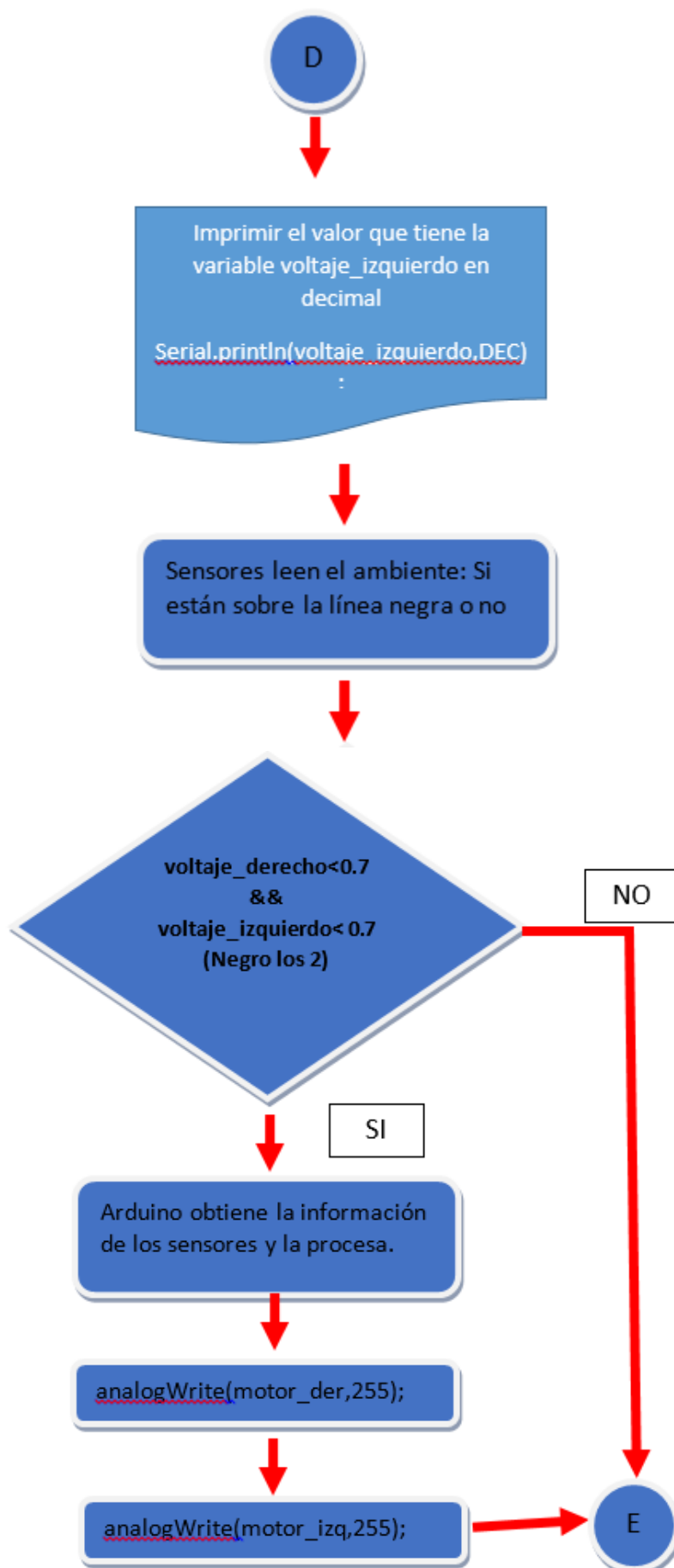
Imprimir el valor que tiene la
variable `voltaje_derecho` en decimal

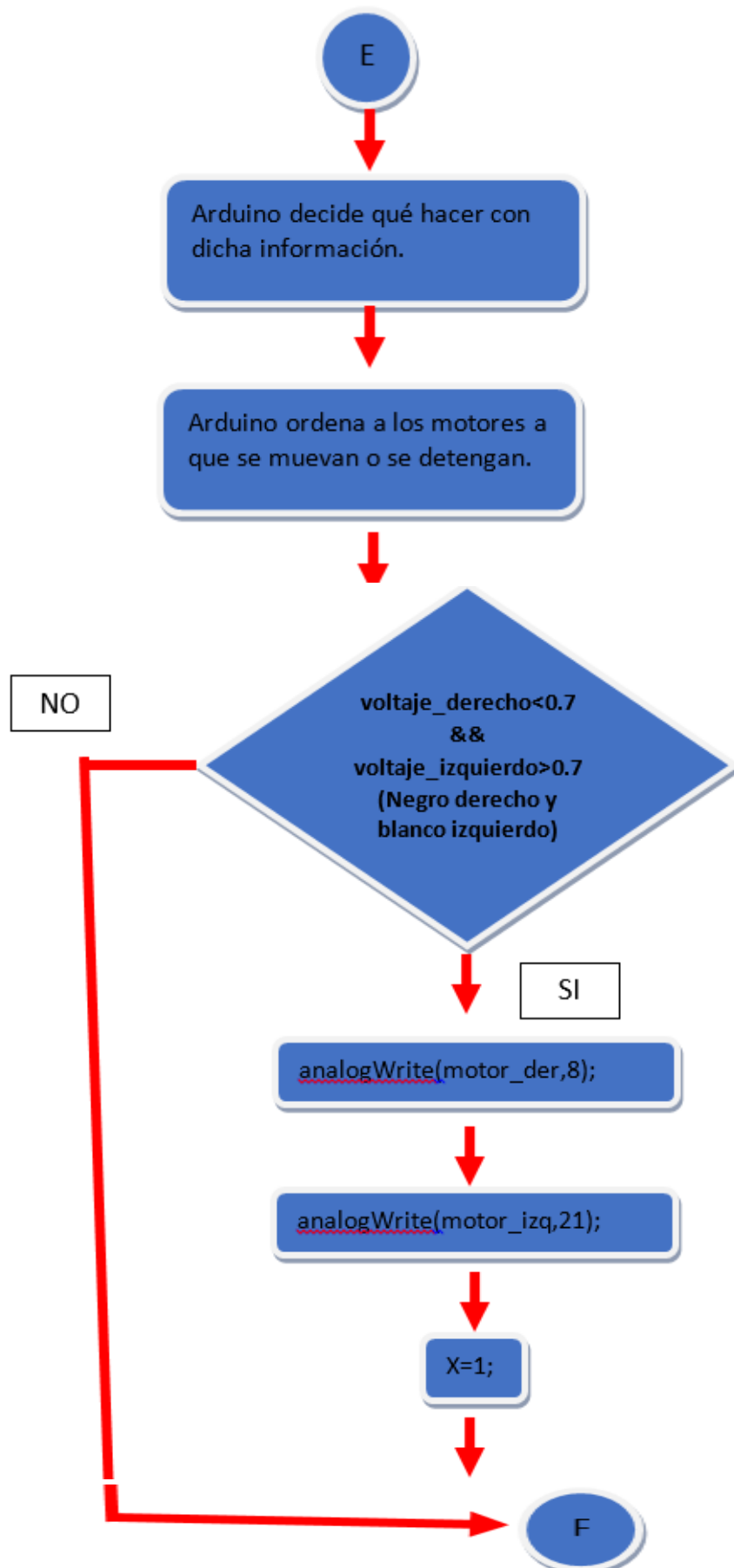
```
Serial.println(voltaje_derecho, DEC);
```

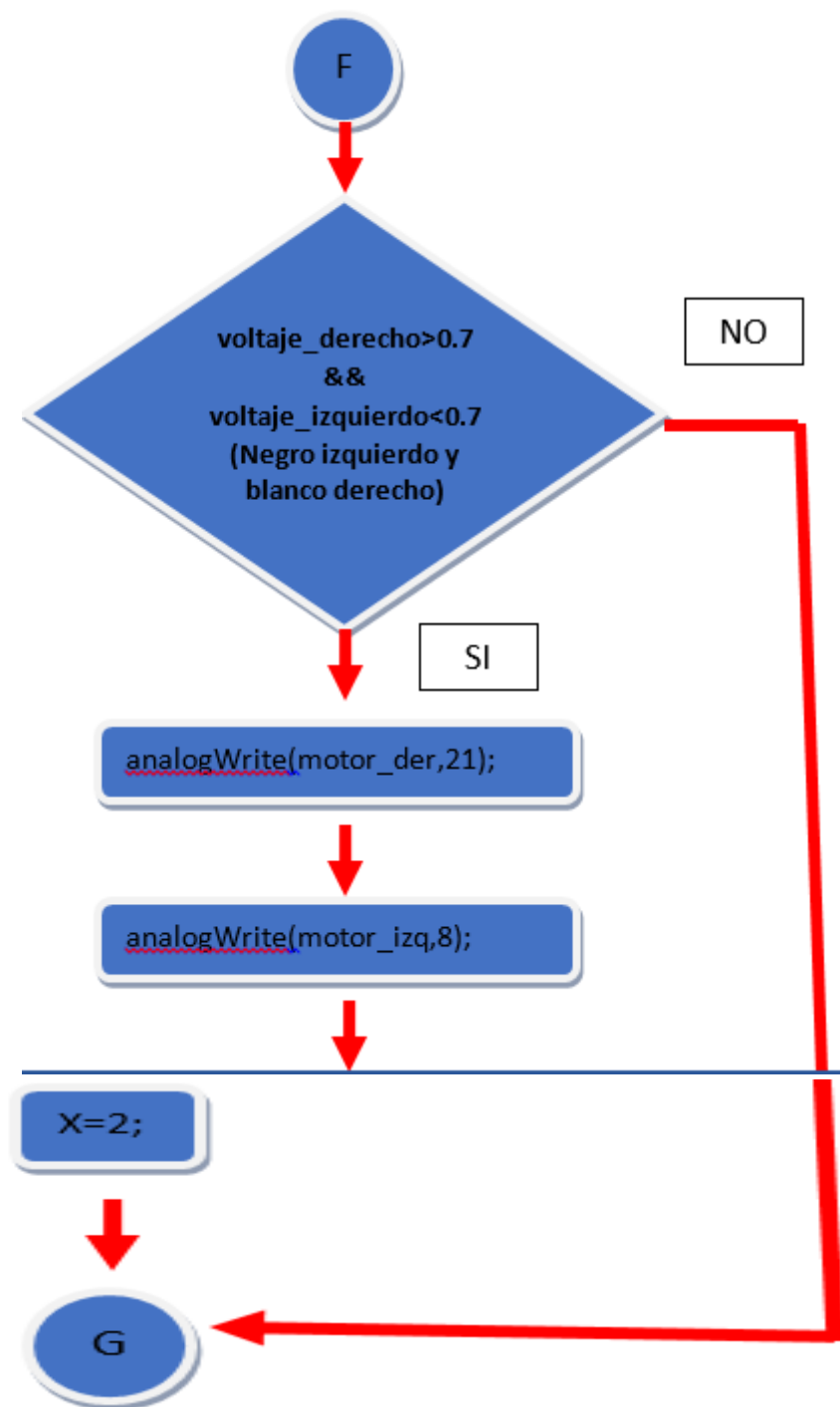
Imprimir texto en el monitor
serial

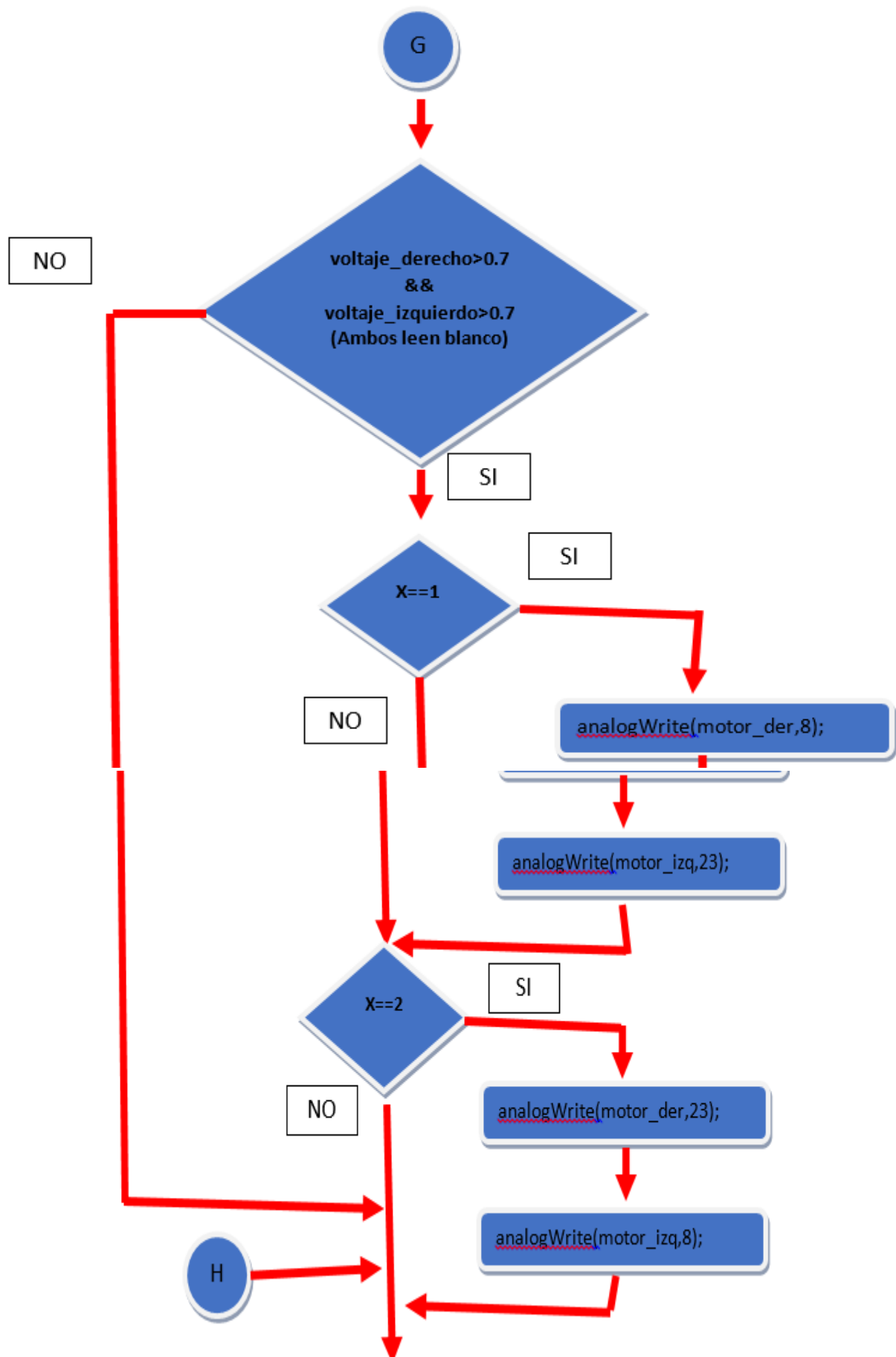
```
Serial.print("PRRO El sensor  
izquierdo dice:");
```

D









Código del motor en función PWM

```
PROYECTO_SEGUIDOR_DE_LINEA Arduino 1.6.7
Archivo Editar Programa Herramientas Ayuda

float value = 0; //ESTE COMANDO SIRVE PARA DECLARAR UNA
//VARIABLE FLOTANTE value QUE EMPIEZA EN 0 Y VA A SERVIR PARA
//LEER EL PRIMER SENSOR CNY70 DE LA TERMINAL A0 DEL ARDUINO

float value2=0; //ESTE COMANDO SIRVE PARA DECLARAR UNA
//VARIABLE FLOTANTE value2 QUE EMPIEZA EN 0 Y VA A SERVIR PARA
//LEER EL SEGUNDO SENSOR CNY70 DE LA TERMINAL A1 DEL ARDUINO

float voltaje_derecho; //ESTE COMANDO SIRVE PARA DECLARAR UNA
//VARIABLE FLOTANTE voltaje_derecho QUE NO EMPIEZA CON UN
//VALOR PORQUE VA A SERVIR PARA CONVERTIR LOS VALORES DE
//LA VARIABLE value (0-1023) A VALORES DE VOLTAJE DE 0-5V
//CON EL FIN DE MANEJAR LOS VOLTAJES EN TODAS LAS CONDICIONES
//DEL SEGUIDOR DE LINEA

float voltaje_izquierdo; //ESTE COMANDO SIRVE PARA DECLARAR UNA
//VARIABLE FLOTANTE voltaje_derecho QUE NO EMPIEZA CON UN
//VALOR PORQUE VA A SERVIR PARA CONVERTIR LOS VALORES DE
//LA VARIABLE value2 (0-1023) A VALORES DE VOLTAJE DE 0-5V
//CON EL FIN DE MANEJAR LOS VOLTAJES EN TODAS LAS CONDICIONES
//DEL SEGUIDOR DE LINEA

int motor_der=11; //ESTE COMANDO SIRVE PARA DECLARAR
//UNA VARIABLE ENTERA motor_der ASIGNANDOLE EL VALOR DECIMAL 11
//PORQUE LUEGO SE VA A USAR ESE VALOR DECIMAL PARA CONVERTIRLO
//EN EL PIN 11 DEL PUERTO DIGITAL DEL ARDUINO.

int motor_izq=10; //ESTE COMANDO SIRVE PARA DECLARAR
//UNA VARIABLE ENTERA motor_izq ASIGNANDOLE EL VALOR DECIMAL 10
//PORQUE LUEGO SE VA A USAR ESE VALOR DECIMAL PARA CONVERTIRLO
//EN EL PIN 10 DEL PUERTO DIGITAL DEL ARDUINO.

void setup()//ESTE ES EL INICIO DE CONFIGURACIÓN DE PUERTOS Y
//MODULOS DE COMUNICACIÓN QUE SOLO SE HACE 1 VEZ

{
  Serial.begin(9600); //ESTE COMANDO SIRVE PARA INICIAR LA
  //COMUNICACIÓN SERIAL CON LA COMPUTADORA A UNA VELOCIDAD DE
  //9600 BITS/SEGUNDO

  pinMode(A0,INPUT); //ESTE COMANDO SIRVE PARA CONFIGURAR
  //LA TERMINAL ANALÓGICA A0 DEL ARDUINO COMO ENTRADA PORQUE
  //AHÍ ES DONDE VA CONECTADO EL PRIMER SENSOR

  pinMode(A1,INPUT); //ESTE COMANDO SIRVE PARA CONFIGURAR
  //LA TERMINAL ANALÓGICA A1 DEL ARDUINO COMO ENTRADA PORQUE
  //AHÍ ES DONDE VA CONECTADO EL SEGUNDO SENSOR

  pinMode(motor_der,OUTPUT); //ESTE COMANDO SIRVE PARA
  //CONFIGURAR LA TERMINAL DIGITAL 11 DEL ARDUINO COMO SALIDA
  //PORQUE AHÍ ES DONDE VA CONECTADO EL PRIMER MOTOR. ADEMÁS,
  //EL CÓDIGO SABE QUE ES EL 11 PORQUE EL VALOR DE
  //motor_der era 11

  pinMode(motor_izq,OUTPUT); //ESTE COMANDO SIRVE PARA CONFIGURAR
```



```

PROYECTO_SEGUIDOR_DE_LINEA

pinMode (motor_isq,OUTPUT); //ESTE COMANDO SIRVE PARA CONFIGURAR
//LA TERMINAL DIGITAL 10 DEL ARDUINO COMO SALIDA PORQUE
//AHI ES DONDE VA CONECTADO EL SEGUNDO MOTOR. ADEMÁS, EL CÓDIGO
//SABE QUE ES EL 10 PORQUE EL VALOR DE motor_isq era 10
}

void loop() //ESTE ES EL CICLO LOOP QUE DICE QUE
//MIENTRAS EL MICROCONTROLADOR SE ENCUENTRA ENERGIZADO,
//SE VAN A EJECUTAR LAS INSTRUCCIONES QUE DICE ESTE COMANDO
//POR TIEMPO INDEFINIDO HASTA QUE YA NO SE CUMPLA LA CONDICIÓN
{
  int x=0; //CON ESTE COMANDO SIRVE PARA DECLARAR UNA VARIABLE
  // ENTERA X Y DARLE EL VALOR INICIAL DE 0 PORQUE ESA VARIABLE
  //ES UNA VARIABLE AUXILIAR QUE ME VA A SERVIR CUANDO TENGA LA
  //CONDICIÓN DE QUE LOS 2 SENSORES LEEN BLANCO

  value = analogRead(A0); //CON ESTE COMANDO SIRVE PARA GUARDAR
  //LA INFORMACIÓN DEL PRIMER SENSOR A LA VARIABLE value QUE
  //ESTA CONECTADO A LA TERMINAL A0 DEL ARDUINO CON UN RANGO
  //DE VALORES DE 0--1023.

  value2 = analogRead(A1); //CON ESTE COMANDO SIRVE PARA GUARDAR
  //LA INFORMACIÓN DEL SEGUNDO SENSOR A LA VARIABLE value2 QUE
  //ESTA CONECTADO A LA TERMINAL A1 DEL ARDUINO CON UN RANGO
  //DE VALORES DE 0--1023.

  voltaje_derecho=(value*(5))/1023; //CON ESTE COMANDO SIRVE
  //PARA HACER UN PROCESO DE CONVERSIÓN DE LOS VALORES DE 0-1023

```

```

PROYECTO_SEGUIDOR_DE_LINEA

//A LOS VALORES DE VOLTAJE DEL SENSOR DERECHO A 0-5VOLTS Y
//GUARDAR ESOS VALORES EN LA VARIABLE voltaje_derecho

voltaje_izquierdo=(value2*(5))/1023; //CON ESTE COMANDO SIRVE
//PARA HACER UN PROCESO DE CONVERSIÓN DE LOS VALORES DE 0-1023
//A LOS VALORES DE VOLTAJE DEL SENSOR IZQUIERDO A 0-5VOLTS Y
//GUARDAR ESOS VALORES EN LA VARIABLE voltaje_izquierdo

Serial.print("PRRO El sensor derecho dice: "); //CON ESTE
//COMANDO SIRVE PARA IMPRIMIR TEXTO EN EL MONITOR SERIAL
//PARA QUE LA COMPUTADORA ME AYUDE Y ME DIGA SI EL ARDUINO
//ESTA LEYENDO EL SENSOR DERECHO

Serial.println (voltaje_derecho,DEC); //CON ESTE
//COMANDO SIRVE PARA IMPRIMIR TEXTO EN EL MONITOR SERIAL
//PARA QUE LA COMPUTADORA ME AYUDE Y ME DIGA SI EL ARDUINO
//ESTA LEYENDO LOS SENSORES Y TAMBIÉN ME IMPRIME EL VALOR
//QUE TIENE LA VARIABLE voltaje_derecho en formato decimal.

Serial.print("PRRO El sensor izquierdo dice: "); //CON ESTE
//COMANDO SIRVE PARA IMPRIMIR TEXTO EN EL MONITOR SERIAL
//PARA QUE LA COMPUTADORA ME AYUDE Y ME DIGA SI EL ARDUINO
//ESTA LEYENDO EL SENSOR IZQUIERDO

Serial.println (voltaje_izquierdo,DEC); //CON ESTE
//COMANDO SIRVE PARA IMPRIMIR TEXTO EN EL MONITOR SERIAL
//PARA QUE LA COMPUTADORA ME AYUDE Y ME DIGA SI EL ARDUINO
//ESTA LEYENDO LOS SENSORES Y TAMBIÉN ME IMPRIME EL VALOR

```

```

PROYECTO_SEGUIDOR_DE_LINEA

//QUE TIENE LA VARIABLE voltaje_izquierdo en formato decimal.

if(voltaje_derecho<0.7 && voltaje_izquierdo<0.7) //CON ESTE
//COMANDO SIRVE PARA HACER LA CONDICIÓN IF EN EL CUAL
//CONSISTE SI LA CONDICION ENCERRADA SE CUMPLE SE VAN A
//CUMPLIR TODAS LAS SENTENCIAS DE CONTROL QUE MARCAN ESTA
//CONDICIONANTE. LA CONDICIÓN ES SI cuando lee negro en
//ambos sensores, avanza recto
{
    analogWrite(motor_der,63); //CON ESTE COMANDO SIRVE
    //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
    //ES DARLE EL VALOR ANALOGICO DE 255 EN FORMA DE VOLTAJE
    //EN 5 VOLTS EN LA TERMINAL 11 DEL ARDUINO

    analogWrite(motor_izq,63); //CON ESTE COMANDO SIRVE
    //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
    //ES DARLE EL VALOR ANALOGICO DE 255 EN FORMA DE VOLTAJE
    //EN 5 VOLTS EN LA TERMINAL 10 DEL ARDUINO
}

if(voltaje_derecho<0.7 && voltaje_izquierdo>0.7) //CON ESTE
//COMANDO SIRVE PARA HACER LA CONDICIÓN IF EN EL CUAL
//CONSISTE SI LA CONDICION ENCERRADA SE CUMPLE SE VAN A
//CUMPLIR TODAS LAS SENTENCIAS DE CONTROL QUE MARCAN ESTA
//CONDICIONANTE. LA CONDICIÓN ES SI cuando lee negro en el
//sensor derecho y blanco en el izquierdo
{

```

```

PROYECTO_SEGUIDOR_DE_LINEA

    analogWrite(motor_der,63); //CON ESTE COMANDO SIRVE
    //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
    //ES DARLE EL VALOR ANALOGICO DE 21 EN FORMA DE VOLTAJE
    //EN LA TERMINAL 11 DEL ARDUINO. Se mueve a la
    //izquierda al mover el motor derecho

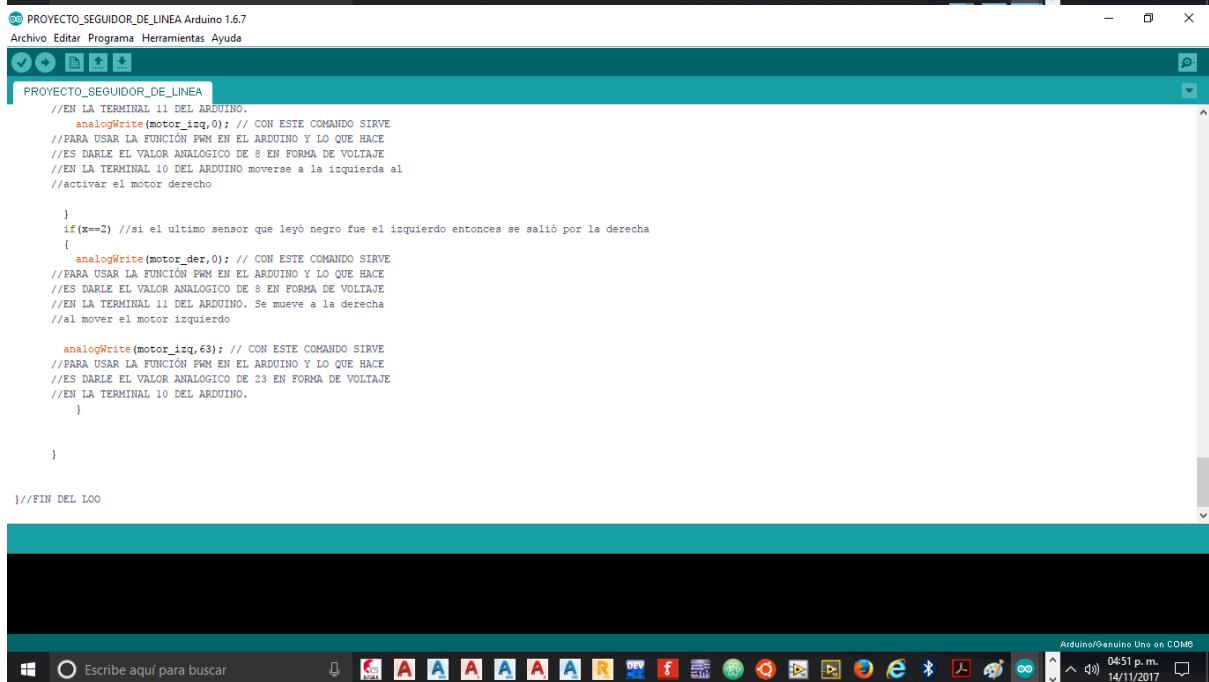
    analogWrite(motor_izq,0); // CON ESTE COMANDO SIRVE
    //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
    //ES DARLE EL VALOR ANALOGICO DE 8 EN FORMA DE VOLTAJE
    //EN LA TERMINAL 10 DEL ARDUINO.

    x=2; //ultimo sensor que leyó negro fue izquierdo
}

if(voltaje_derecho>0.7 && voltaje_izquierdo<0.7) ///CON ESTE
//COMANDO SIRVE PARA HACER LA CONDICIÓN IF EN EL CUAL
//CONSISTE SI LA CONDICION ENCERRADA SE CUMPLE SE VAN A
//CUMPLIR TODAS LAS SENTENCIAS DE CONTROL QUE MARCAN ESTA
//CONDICIONANTE. LA CONDICIÓN ES SI cuando lee negro en el
//izquierdo y blanco en derecho
{

    analogWrite(motor_der,0); // CON ESTE COMANDO SIRVE
    //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
    //ES DARLE EL VALOR ANALOGICO DE 8 EN FORMA DE VOLTAJE
    //EN LA TERMINAL 11 DEL ARDUINO. Se mueve a la derecha
    //al mover el motor izquierdo

```



Por último la programación final del carrito:

```
seguidor Arduino 1.6.7
Archivo Editar Programa Herramientas Ayuda

//Declaración de variables
// - const = constante
// - int = entero

// Entradas y salidas de Arduino
const int sensor1=A0; //Pin del sensor 1 (entrada)
const int sensor2=A1; //Pin del sensor 2 (entrada)
const int motor1=11; //Pin del motor 1 (salida)
const int motor2=10; //Pin del motor 2 (salida)

// Variables del programa
int lecturaSensor1=0; //Valor que leerá el sensor 1
int lecturaSensor2=0; //Valor que leerá el sensor 2

// Constantes del programa: MODIFICARLAS COMO SEA NECESARIO para configurar el seguidor
// Velocidad: puede ir de 0 a 255
const int velocidadMaxima=255;
const int velocidadMedia=0;
const int velocidadMinima=0;
// Sensor: limites de lectura de sensor, de 0 a 1023. Configurar el minimo valor constante que lee sobre superficie negra y el máximo valor constante sobre superficie blanca
const int sensorNegro=300;
const int sensorBlanco=100;

void setup() {
  //Set-up del Arduino
  // Entradas: pines de los sensores
  pinMode (sensor1,INPUT);
  pinMode (sensor2,INPUT);
}

// Salidas: pines de los motores
pinMode (motor1,OUTPUT);
pinMode (motor2,OUTPUT);
Serial.begin(9600);
}

void loop() {
  int x=0;
  //Código que se repetirá constantemente

  //Asignación del valor que lee cada sensor
  lecturaSensor1 = analogRead(sensor1);
  lecturaSensor2 = analogRead(sensor2);

  //Código para leer sensores
  /*Serial.println(lecturaSensor1);
  delay(1500);
  Serial.println(lecturaSensor2);
  delay(1500);*/

  //Casos de lectura de los sensores:
  // Cuando ambos sensores están sobre superficie negra: los dos motores pueden ir a su velocidad máxima (por ejemplo: sobre una recta)
  if(lecturaSensor1<sensorNegro && lecturaSensor2<sensorNegro){
    analogWrite(motor1,velocidadMaxima);
    analogWrite(motor2,velocidadMaxima);
  }
  // Cuando el sensor 1 esta sobre una superficie negra y el sensor 2 esta sobre una superficie blanca: el motor 2 debe ir mas rápido que el motor 1
  else if(lecturaSensor1<sensorNegro && lecturaSensor2>sensorBlanco){
  }
}
```

seguidor Arduino 1.6.7

Archivo Editar Programa Herramientas Ayuda

```
seguidor
// Cuando el sensor 1 esta sobre una superficie negra y el sensor 2 esta sobre una superficie blanca: el motor 2 debe ir mas rápido que el motor 1
else if(lecturaSensor1<sensorNegro&&lecturaSensor2>sensorBlanco){
  analogWrite(motor1,velocidadMaxima);
  analogWrite(motor2,velocidadMedia);
  x=1;
}
// Cuando el sensor 2 esta sobre una superficie negra y el sensor 1 esta sobre una superficie blanca: el motor 1 debe ir mas rápido que el motor 2
else if(lecturaSensor1>sensorBlanco&&lecturaSensor2<sensorNegro){
  analogWrite(motor1,velocidadMedia);
  analogWrite(motor2,velocidadMaxima);
  x=2;
}
// Cuando los dos sensores se encuentran sobre una superficie blanca: ambos motores se detienen porque se salió de la pista el seguidor
else{
  if(x==1) //si el ultimo sensor que leyó negro fue el derecho entonces se salió por la izquierda.
  {
    analogWrite(motor2,63); //CON ESTE COMANDO SIRVE
    //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
    //ES DARLE EL VALOR ANALOGICO DE 21 EN FORMA DE VOLTAJE
    //EN LA TERMINAL 11 DEL ARDUINO.
    analogWrite(motor1,0); // CON ESTE COMANDO SIRVE
    //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
    //ES DARLE EL VALOR ANALOGICO DE 8 EN FORMA DE VOLTAJE
    //EN LA TERMINAL 10 DEL ARDUINO moverse a la izquierda al
    //activar el motor derecho
  }
}
```

Windows taskbar: Escribe aquí para buscar, Arduino/Genuino Uno on COM6, 04:54 p. m. 14/11/2017

seguidor Arduino 1.6.7

Archivo Editar Programa Herramientas Ayuda

```
seguidor
if(x==1) //si el ultimo sensor que leyó negro fue el derecho entonces se salió por la izquierda.
{
  analogWrite(motor2,63); //CON ESTE COMANDO SIRVE
  //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
  //ES DARLE EL VALOR ANALOGICO DE 21 EN FORMA DE VOLTAJE
  //EN LA TERMINAL 11 DEL ARDUINO.
  analogWrite(motor1,0); // CON ESTE COMANDO SIRVE
  //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
  //ES DARLE EL VALOR ANALOGICO DE 8 EN FORMA DE VOLTAJE
  //EN LA TERMINAL 10 DEL ARDUINO moverse a la izquierda al
  //activar el motor derecho
}
if(x==2) //si el ultimo sensor que leyó negro fue el izquierdo entonces se salió por la derecha
{
  analogWrite(motor2,0); // CON ESTE COMANDO SIRVE
  //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
  //ES DARLE EL VALOR ANALOGICO DE 8 EN FORMA DE VOLTAJE
  //EN LA TERMINAL 11 DEL ARDUINO. Se mueve a la derecha
  //al mover el motor izquierdo
  analogWrite(motor1,63); // CON ESTE COMANDO SIRVE
  //PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
  //ES DARLE EL VALOR ANALOGICO DE 23 EN FORMA DE VOLTAJE
  //EN LA TERMINAL 10 DEL ARDUINO.
}
```

Windows taskbar: Escribe aquí para buscar, Arduino/Genuino Uno on COM6, 04:54 p. m. 14/11/2017

```
seguidor Arduino 1.6.7
Archivo Editar Programa Herramientas Ayuda

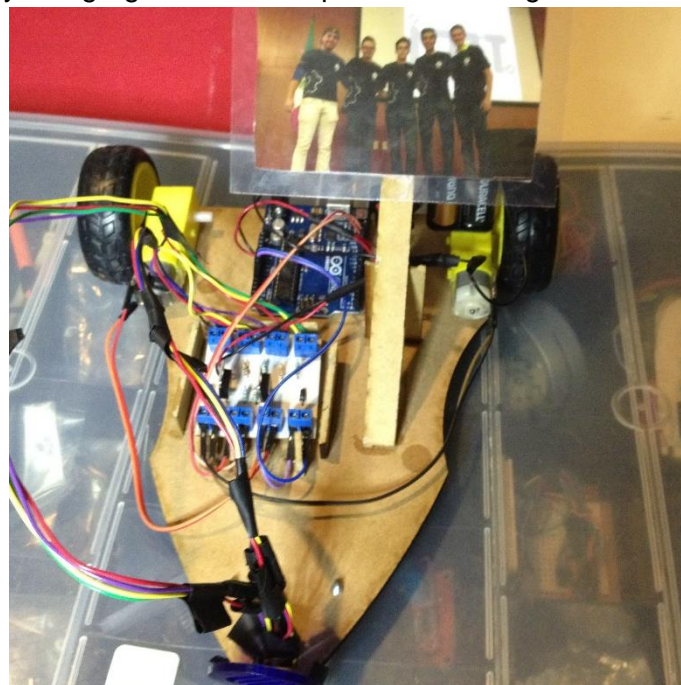
seguidor

analogWrite(motor2,63); //CON ESTE COMANDO SIRVE
//PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
//ES DARLE EL VALOR ANALOGICO DE 21 EN FORMA DE VOLTAJE
//EN LA TERMINAL 11 DEL ARDUINO.
analogWrite(motor1,0); // CON ESTE COMANDO SIRVE
//PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
//ES DARLE EL VALOR ANALOGICO DE 8 EN FORMA DE VOLTAJE
//EN LA TERMINAL 10 DEL ARDUINO moverse a la izquierda al
//activar el motor derecho

}
if(x==2) //si el ultimo sensor que leyó negro fue el izquierdo entonces se salió por la derecha
{
  analogWrite(motor2,0); // CON ESTE COMANDO SIRVE
//PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
//ES DARLE EL VALOR ANALOGICO DE 8 EN FORMA DE VOLTAJE
//EN LA TERMINAL 11 DEL ARDUINO. Se mueve a la derecha
//al mover el motor izquierdo

  analogWrite(motor1,63); // CON ESTE COMANDO SIRVE
//PARA USAR LA FUNCIÓN PWM EN EL ARDUINO Y LO QUE HACE
//ES DARLE EL VALOR ANALOGICO DE 23 EN FORMA DE VOLTAJE
//EN LA TERMINAL 10 DEL ARDUINO.
}
}
}
```

Al finalizar con el diseño computarizado y cortarlo se agregaron los componentes del kit con kola loca y tornillos y se agregaron adornos para hacerlo original. Resultado final del carrito:



Link del carrito seguidor de línea funcional:

<https://www.youtube.com/watch?v=nJbMykDC1e8>

Conclusiones finales:

Nos dimos cuenta en este proyecto que la PCB del seguidor de línea estaba diseñada para que la configuración de los sensores tenga más voltaje cuando leen blanco y menos voltaje cuando leen negro. Por lo tanto, tuvimos que cambiar las condiciones al revés en el código del programa para que el robot seguidor de línea tuviera la posibilidad de seguir la línea negra. Además, vimos como juntar la electrónica, la mecánica, la programación y el control para crear un robot autónomo que siguiera la línea, desde diseñarlo con el dibujo computarizado y hacer el circuito impreso hasta juntar todo el chasis del carrito y calibrar los sensores junto con sus condiciones en la plataforma del Arduino en lenguaje de programación en C. Además, se aplicaron las teorías de los modos de control como es el modo PWM para cambiar la velocidad de los motores cuando el robot tenía que cambiar su dirección para dar vuelta.