

Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ciencias y Sistemas  
Arquitectura de Computadoras y Ensambladores I  
Sección A  
Segundo Semestre 2020  
Catedráticos: Ing. Otto Rene Escobar Leiva  
Tutores académicos: Herberth Argueta



201800726 María Reneé Juárez Albizures  
201800535 Jessica Elizabeth Botón Pérez  
201801266 Didier Alfredo Domínguez Urías  
201800491 Aracely Jacqueline Méndez González  
201801182 Stefany Samantha Abigail Coromac Huevo

## ----- PRÁCTICA NO. 2 -----

### ||| → DESCRIPCIÓN ← |||

El circuito elaborado consiste en un carro seguidor de línea. El carro es capaz de permanecer en la línea negra sin salirse de esta, tiene la habilidad de esquivar obstáculos que se encuentren en su camino. Identifica objetos sobre la pista y actúa como una barredora de dicho objeto.

### ||| → REQUERIMIENTOS TÉCNICOS ← |||

Para la realización de la práctica, se utilizaron los siguientes materiales.

- Arduino Mega
- Virtual Turtle
- 1 matriz LED
- 1 circuito integrado MAX7219
- 2 motores Stepper
- 1 servomotor
- 1 circuito integrado CNY70

Programas:

- Arduino IDE
- Proteus

# CÓDIGO

» Librerías Utilizadas

Se utilizo la librería ledControl para poder utilizar y manipular la matriz de LED, la librería Servo para controlar la barredora de objetos, y la librería Stepper para el funcionamiento y simulación de las llantas del carro.

```
#include "LedControl.h"
#include <Servo.h>
#include <Stepper.h>
```

## »» Definición de Pines

```

/*-----*-----PINES-----*/
//SENSORES DE CAMINO
int PS1P = 13;
int PS2P = 12;
int PS3P = 11;
int ps1, ps2, ps3;
//PMW
int PWML = 10;
int PWMR = 9;
//SENSORES OBSTACULOS
int OS1 = 8;
int OS2 = 7;
int OS3 = 6;
//LLANTAS
int LWF = 5; //Izquierda al frente
int LWB = 4; //Izquierda atras
int RWF = 3; //Derecha al frente
int RWB = 2; //Derecha atras

// SERVO MOTOR
int PSVM = 35;
Servo motor;
int posicion = 0;
int objeto = 31;//Led Amarilla

```

## »» Variables

[illegible]

```

int posX = 2;
int posY = 6;
boolean ziczacVH = true; // Vertical = true, Abajo = false;
boolean ladoID = true; // Izquierda = true, Derecha = false;
boolean ladoAB = true; // Arriba = true, Abajo = false;
int conteoCamino = 0;

/*-----VARIABLES-----*/
boolean camino = false;
int giros = 0;

int velocidad = 100;
int Motor1 = 50; //IZQ
int Motor2 = 52; //DER
int Phase1 = 51;
int Phase2 = 53;

```

## »» Método setup ()

Muestra un mensaje de bienvenida y luego define los pines de entrada y salida.

```

void setup() {
  Serial.begin(9600);
  Serial.println(">> UNIVERSIDAD DE SAN CARLOS DE GUATEMALA");
  Serial.println(">> I N G E N I E R I A");
  Serial.println(">> A R Q U I 1");
  Serial.println(">> G R U P O 3");
  Serial.println("\n");
  //Para el sensor Ultrasonico
  pinMode(OS2, INPUT); //Trigger
  pinMode(OS1, OUTPUT); //Echo
  digitalWrite(OS1, LOW);

  pinMode(OS3, OUTPUT);
  tortuga.attach(OS3);
  tortuga.write(90);

  //Sensores
  pinMode(PS1P, INPUT);
  pinMode(PS2P, INPUT);
  pinMode(PS3P, INPUT);
  //Llantas
  pinMode(LWF, OUTPUT);
  pinMode(LWB, OUTPUT);
  pinMode(RWF, OUTPUT);
  pinMode(RWB, OUTPUT);
  //PWM
  pinMode(PWML, OUTPUT);
  pinMode(PWMR, OUTPUT);
}

```

## »» Método loop ()

Se verifica si el carro ha encontrado un objeto u obstáculos, de lo contrario sigue su camino siguiendo la línea negra.

```

void loop() {
    long t;
    long sus;
    long auxT;
    long d;
    int contar = 0;
    boolean seguir = true;
    boolean obstaculo = true;
    detener();
    digitalWrite(OS1, HIGH);
    delayMicroseconds(10);
    digitalWrite(OS1, LOW);

    t = pulseIn(OS2, HIGH);
    d = t / 59;
    /*Serial.println("Original");
    Serial.println(t);*/
    if (d > 0) {

        while (t > 60) {
            detener();
            digitalWrite(OS1, HIGH);
            delayMicroseconds(10);
            digitalWrite(OS1, LOW);
            auxT = pulseIn(OS2, HIGH);
            if (auxT == 0) {
                contar++;
            }
            if (auxT != 0) {
                t = auxT;
            }
        }
    }
}

```

## » Métodos de Avance y Movimiento

### » adelante ()

El carro se mueve hacia delante y seguido de esto muestra su camino en la matriz.

```

void adelante() {
    analogWrite(PWML, velocidad);
    analogWrite(PWMR, velocidad);

    digitalWrite(LWF, HIGH);
    digitalWrite(LWB, LOW);

    digitalWrite(RWF, HIGH);
    digitalWrite(RWB, LOW);

    conteoCamino++;
    if (conteoCamino == 8) {
        if (ziczacVH) {
            if (ladoAB) {
                posY--;
            } else {
                posY++;
            }
        } else {
            if (ladoID) {
                posX--;
            } else {
                posX++;
            }
        }
        encender(posX, posY);
        conteoCamino = 0;
    }
}

```

➤ atrás ()

Se mueve hacia atrás.

```
void atras() {
  analogWrite(PWML, velocidad);
  analogWrite(PWMR, velocidad);

  digitalWrite(LWF, LOW);
  digitalWrite(LWB, HIGH);

  digitalWrite(RWF, LOW);
  digitalWrite(RWB, HIGH);
}
```

➤ derecha ()

Se mueve hacia la derecha

```
void derecha(int retraso) {
  analogWrite(PWML, velocidad / 2);
  analogWrite(PWMR, velocidad / 2);

  digitalWrite(LWF, HIGH);
  digitalWrite(LWB, LOW);

  digitalWrite(RWF, LOW);
  digitalWrite(RWB, HIGH);
  delay(retraso);
}
```

➤ esquivar ()

Si el carro encuentra un obstáculo este lo esquiva utilizando este método.

➤ izquierda ()

```
void esquivar() {
  Serial.println("E S Q U I V A N D O   O B J E T O");
  derecha(2100);
  detener();
  adelante();
  delay(2000);
  izquierda(2100);
  detener();
  adelante();
  delay(3500);
  izquierda(2100);
  detener();
  adelante();
  delay(2000);
  derecha(2100);
  detener();
  Serial.println("L I S T O!");
  Serial.println("\n");
}
```

Se mueve hacia la izquierda.

```
void izquierda(int retraso) {
  analogWrite(PWML, velocidad / 2);
  analogWrite(PWMR, velocidad / 2);

  digitalWrite(LWF, LOW);
  digitalWrite(LWB, HIGH);

  digitalWrite(RWF, HIGH);
  digitalWrite(RWB, LOW);
  delay(retraso);
}
```

➤ detener ()

Detiene el carro por un momento.

```
void detener() {
  digitalWrite(LWF, LOW);
  digitalWrite(LWB, LOW);

  digitalWrite(RWF, LOW);
  digitalWrite(RWB, LOW);
}
```

## »» Matriz

Estos métodos realizan la tarea de ir trazando la matriz.

```
void inicializarMatrizControlador() {
    ledControl.shutdown(0, false);
    ledControl.setIntensity(0, 15);
    ledControl.clearDisplay(0);
}

void encender(int posAX, int posAY) {
    if (posAX > 7 || posAY > 7 || posAX < 0 || posAY < 0) {
        posX = 3;
        posY = 3;
        posAX = posX;
        posAY = posY;

        for (int i = 0; i < 8; i++) {
            matriz[i] = "00000000";
        }
        ledControl.clearDisplay(0);
        delay(100);
    }

    String cadena = matriz[posY];
    String partel = "";
    String parte2 = "";

    for (int i = 0; i < posAX; i++) {
        partel += matriz[posAY].charAt(i);
    }

    for (int i = posX + 1; i < 8; i++) {
        parte2 += matriz[posAY].charAt(i);
    }

    matriz[posAY] = partel + "1" + parte2;
}

void objetoMatriz(int posAX, int posAY) {
    String cadena = matriz[posY];
    String partel = "";
    String parte2 = "";

    for (int i = 0; i < posAX; i++) {
        partel += matriz[posAY].charAt(i);
    }

    for (int i = posX + 1; i < 8; i++) {
        parte2 += matriz[posAY].charAt(i);
    }

    matriz[posAY] = partel + "2" + parte2;
}
```

## » Barredora

Da la orden para que el servomotor de 3 vuelta y media cuando el carro encuentra un objeto.

```
/*----- BARREDORA -----*/
void barredora3vueltas() {
  Serial.println("A B A R R E R!");
  Serial.println("- - - -");
  motor.attach(PSVM);
  for (posicion = 1; posicion <= 977; posicion++) {
    motor.write(posicion);
    delay(15);
  }
  //motor.attach(PSVM);
  Serial.println("L I S T O!");
  Serial.println("\n");
}
```

## Circuito

