

Práctica 4

Diseño de Algoritmos

David Ruiz Rodríguez, Mohamed Rodrigo El Badry,
Nicolás Recinella Vidán, Denilson Palomino Adán

19 de diciembre de 2024

Índice

1. Introducción	2
2. Ejercicio 1	3
3. Ejercicio 2	3
4. Ejercicio 3	3

1. Introducción

El objetivo de esta tarea es familiarizarse con el microcontrolador Arduino y su programación básica mediante la simulación en el **Simulador Oviedo**. En esta práctica, se programará un coche robot para que siga una línea en un circuito.

¿Qué es un Arduino?

Arduino es un microcontrolador que incluye:

- **CPU:** Procesa las instrucciones.
- **Memoria:** Almacena temporalmente datos.
- **Pines de entrada y salida (I/O):** Permiten la interacción con el entorno.

Tipos de pines

- **Digitales:** Dos estados posibles:
 - **LOW (0):** Voltaje bajo ($-V_{cc}$).
 - **HIGH (1):** Voltaje alto ($+V_{cc}$).
- **Analógicos:** Admiten valores continuos entre $-V_{cc}$ y $+V_{cc}$.

Componentes electrónicos utilizados

Motores

Controlan el movimiento del coche. La velocidad se controla con valores entre 0 y 180:

- 90: Parado.
- 180: Máxima velocidad hacia delante en el motor derecho.
- 0: Máxima velocidad hacia delante en el motor izquierdo.

Sensores

Detectan el color negro en el suelo. Usan pines digitales para leer valores:

- **0 (LOW):** No detecta la línea.
- **1 (HIGH):** Detecta la línea.

2. Ejercicio 1

Para girar el coche a la derecha se utiliza la función `turnRight` la que generara un giro en el sentido de las agujas del reloj y para que gire a la izquierda `turnLeft`. Para activar la función `turnRight` se utiliza la tecla D y la `turnLeft` la tecla A.

Los sensores se leen dentro función `forward` que invoca `digitalRead` y este indica si esta en la linea o no tanto en la izquierda como en la derecha. La función `forward` se activa al presionar a,w,s,d.

3. Ejercicio 2

Se nos pide que hagamos unos cambios en la función `loop` del código para esto llamamos a la función `forward()` cuando no hay ninguna curva lo que significa que queremos que siga recto manteniendo la dirección, en el resto de los casos cuando detectamos una curva a la izquierda o derecha vamos a esa dirección respectivamente

```
void loop(){
  readIRSensor();

  if(irSensorValues[0] == NO_LINEA && irSensorValues[3] == NO_LINEA && (irSensor
    forward();
  }
  else if (irSensorValues[0] == LINEA ){
    turnLeft();
  }
  else if ( irSensorValues[3] == LINEA){
    turnRight();
  }
}
```

Figura 1: Cambios Ejercicio 2

4. Ejercicio 3

Para solucionar el caso de que si está completamente fuera del circuito, usamos el generador de números pseudoaleatorio «Linear Congruential Generator», lo llamamos en la función `loop` en el caso de que este fuera del circuito:

```
45 void loop(){
46   readIRSensor();
47
48   if(irSensorValues[0] == NO_LINEA && irSensorValues[3] == NO_LINEA && (irSensor
49     forward();
50   }
51   else if (irSensorValues[0] == LINEA ){
52     turnLeft();
53   }
54   else if ( irSensorValues[3] == LINEA){
55     turnRight();
56   }
57   else if (irSensorValues[0] == NO_LINEA && irSensorValues[2] == NO_LINEA){
58     searchLine();
59   }
60 }
61
```

Figura 2: Cambios loop, llamada función searchLine

Esta función se encarga de generar un bit b que puede tener valores 0 o 1, en el caso de que este sea 0 se girara a la izquierda y en el caso de que sea 1, girara a la derecha hasta que llegue al circuito donde con su movimiento descrito en el ejercicio anterior

```
7 void searchLine(){
8   x = (a*x + c) % m;
9   int b = (x >> 3) % 2;
10
11   if (b==0){
12     servoIzq.write(180);
13     servoDer.write(180);
14   }
15   else {
16     servoIzq.write(0);
17     servoDer.write(0);
18   }
19   delay(QUARTER_BACK_TIME);
20   forwardMotor(QUARTER_BACK_TIME);
21 }
22
```

Figura 3: Función searchLine