

Apellidos del profesor: Călugăru	Nombre: Dumitru
Título: VIAJE A RUMANÍA	Duración: 2 horas
Asignatura: TIC	
Objetivos: Competencia general: 1. Practicar el pensamiento computacional en diferentes contextos de programación de robots reales. Competencia específica: 1.1 Probar algoritmos para manejar dispositivos robóticos de movimiento espacial Competencia específica 1.2 Utilizar algoritmos en diferentes situaciones para codificar robots Objetivo de la actividad: Codificar un robot que, con la ayuda de sensores de color, se desplace por una ruta sencilla y preestablecida.	
Elementos clave de CC: Descomposición; Reconocimiento de Patrones; Abstracción; Diseño de Algoritmos	
Grupo de edad: De 12 a 14 años	
Situaciones de Aprendizaje: aula	Tipo de Actividad: extracurricular
Materiales: <ul style="list-style-type: none"> - Kits de robots (microcontrolador, sensores, motores, servomotores, relés, puertos, actuadores, etc.), - banderas de los países asociados. 	Recursos: <ul style="list-style-type: none"> - libro técnico: especificaciones del kit - esquema funcional - ordenadores portátiles - ruta
Desarrollo del Aprendizaje	
Definición del Problema: El profesor explica a los alumnos que el trabajo consiste en escribir y probar los algoritmos utilizados para controlar los movimientos de un robot en distintas direcciones y comprender los valores leídos por los sensores.	
Introducción: Los alumnos identificarán los algoritmos utilizados para accionar los mecanismos de un robot (por ejemplo, seguir la línea -la trayectoria predeterminada- para completar la tarea «Viaje a Rumanía». Los alumnos trabajarán en equipos. Cada equipo codificará el robot para que viaje desde su país de origen (Turquía, España, Italia) hasta Rumanía.	
Evaluación Previa (opcional)	
1. Descomposición 1. <u>Analizar, explicar y comprender los algoritmos utilizados en el diseño de un robot, como paso previo a su diseño.</u>	

Las tareas requieren elegir el destino, establecer la ruta óptima, seguir la ruta/recorrido, volver a la ruta si hay oscilaciones de movimiento, detener el movimiento cuando se ha alcanzado el destino.

Cambiar las reglas de desplazamiento si es necesario debido a las condiciones a las que se enfrenta.

2. Escribir el algoritmo de programación del robot

Cada equipo decidirá, en función de los componentes proporcionados, la geometría de un robot, la funcionalidad de los componentes, los circuitos electrónicos que deben estar formados por componentes pasivos y activos.

Uniendo los bloques funcionales con un algoritmo que correlacione su uso con el camino para llegar al destino se obtendrá un programa reflejado por el código fuente. El código fuente será ejecutado por la plataforma Arduino que incluye un procesador ESP 32.

3. Escribir el programa, el código fuente

Teniendo en cuenta el número de sensores, el número de motores de propulsión y su control mediante la información de los sensores, resulta el siguiente código fuente:

// Código del Robot Seguidor de Línea Arduino

```
#define enA 5//Enable1 L293 Pin enA
#define in1 6 //Motor1 L293 Pin in1
#define in2 7 //Motor1 L293 Pin in1
#define in3 9 //Motor2 L293 Pin in1
#define in4 10 //Motor2 L293 Pin in1
#define enB 8 //Enable2 L293 Pin enB
#define R_S 4//ir sensor Right
#define L_S 2 //ir sensor Left

void setup(){
pinMode(R_S, INPUT);

pinMode(L_S, INPUT);
pinMode(enA, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
pinMode(enB, OUTPUT);
digitalWrite(enA, HIGH);
digitalWrite(enB, HIGH);
delay(1000);}

void loop(){

if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0)){forward();}
if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 0)){turnRight();}
if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)){turnLeft();}
if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 1)){Stop();}
```

```

}

void forward(){

digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);}
void turnRight(){
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH); }

void turnLeft(){

digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW); }

void Stop(){

digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);

digitalWrite(in4, LOW);
}

```

4. Recorrer la ruta predeterminada

Para seguir la ruta, conecta el robot mediante el cable USB al ordenador, conecta el COM adecuado, conecta el procesador, comprueba el código y luego súbelo a la plataforma Arduino. Después de cargar el código, coloca el robot en la ruta y sigue su progreso.

2. Reconocimiento de patrones.

Se entregan kits a cada grupo, se construyen los robots y se trazan las rutas entre los puntos de partida y el destino fijado.

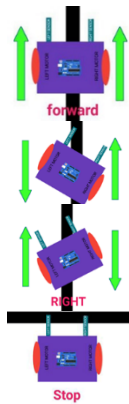
3. Abstracción:

El profesor subraya el significado de cada etapa.

Si los alumnos escriben el algoritmo correcto, el robot seguirá la ruta prevista.

4. Diseño de algoritmos:

- Hacia adelante
- Girar a la izquierda
- Girar a la derecha
- Parar



Evaluación:

Análisis del rendimiento del robot para diferentes entradas.

Prueba de evaluación posterior (opcional):

Feedback basado en la prueba posterior (opcional):

Resultados esperados:

Diseño correcto del robot, su programación, trazado correcto de rutas.

Notas: