



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Grado Tecnología Digital y Multimedia

## **Sistemas Embebidos**

CURSO ACADÉMICO 2023-2024

E.T.S.I. DE TELECOMUNICACIÓN

MEMORIA TRABAJO FINAL

# **ALARMA DE MOVIMIENTO**

PABLO MORTERA HERRERO & JÚLIA PERIS ALVENTOSA

# ÍNDICE

1. Introducción

2. Objetivos

3. Desarrollo

3.1 Esquema de montaje

3.2 Algoritmo del trabajo

4. Conclusión

## 1. Introducción

---

En esta memoria se ve reflejada el desarrollo y explicación sobre la primera propuesta; ampliación de la funcionalidad de uno de los sistemas planteados en la práctica de programación de sistemas. Más en específico, nuestra elección sobre las propuestas planteadas, ha sido la alarma de movimiento, que hemos desarrollado como proyecto final de la asignatura.

Esta memoria está compuesta por esta breve introducción, por un segundo apartado donde se describen los objetivos a conseguir tras la realización de la misma. Un tercer apartado sobre el desarrollo compuesto por dos subapartados; el esquema de montaje y por el algoritmo del trabajo, y por último encontramos un apartado dedicado a las conclusiones a las que hemos llegado mientras realizábamos la tarea.

## 2. Objetivos

---

El objetivo principal de este proyecto final es desarrollar un programa en Arduino, el cual gestione dos sensores de movimiento y que actúen como alarma, además de un menú gestionado por medio de un LCD, este menú nos proporcionará información sobre el funcionamiento en tiempo real de los sensores.

Este objetivo forma parte de la primera opción de proyectos para este trabajo "Ampliación de la funcionalidad de sistemas anteriores".

### 3. Desarrollo

---

Para el desarrollo del proyecto hemos hecho uso del editor de código *ArduinoIDE* y del simulador *Tinckercad*.

#### 3.1 Esquema de montaje

Como hemos comentado anteriormente, para el montaje del circuito hemos hecho uso del simulador *Tinckercad*, con el que hemos hecho tanto el esquema de montaje completo, y posteriormente hemos probado su funcionamiento, (hemos usado *Tinckercad* en vez de otro simulador como *SimulIDE* con preferencias a la hora de montar y ejecutar la simulación, por lo que nos hemos adaptado a los materiales que este simulador nos ofrece). Además, hemos tratado de realizar un montaje claro, que fuese fácilmente replicable en físico, y empleable en el caso de querer un sistema de alarmas de movimiento para uso personal.

Para crear el circuito de la alarma, hemos usado los siguientes materiales:

- Placa Arduino uno R3
- Protoboard
- Sensor PIR (2)
- Zumbador
- Led rojo
- Panel LCD 16x2

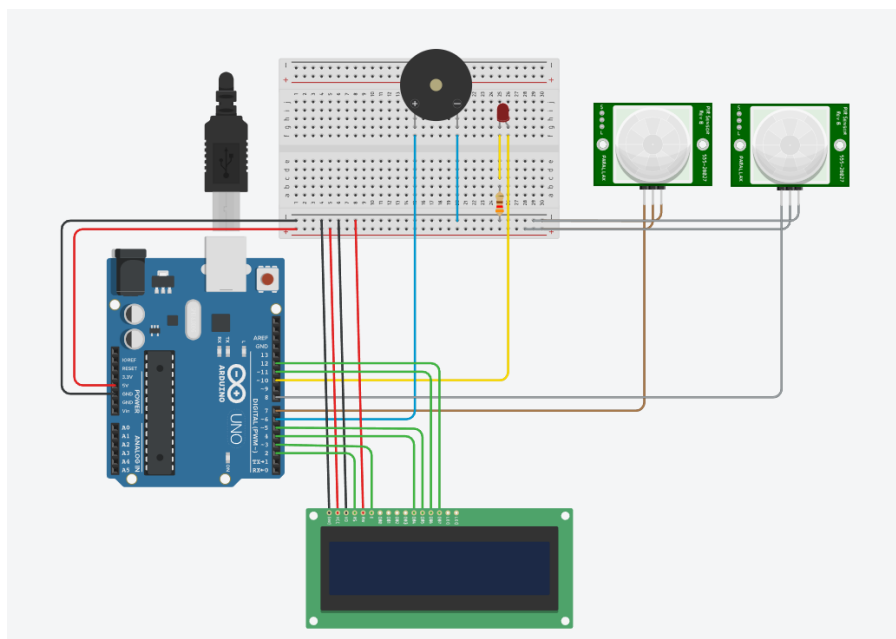
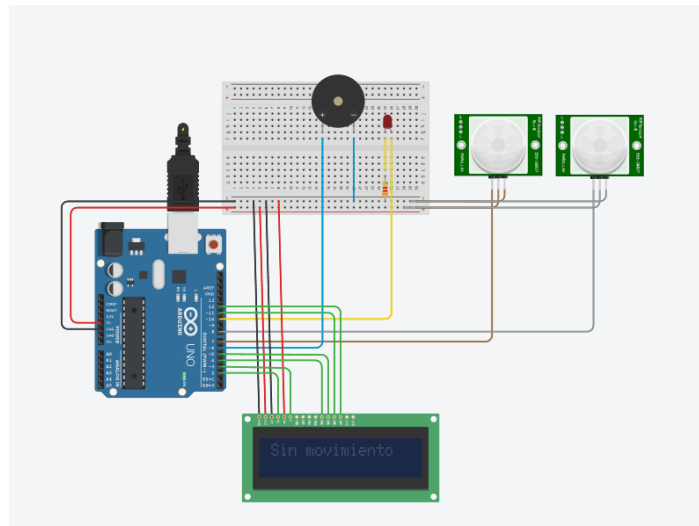


Figura 1: Esquema de montaje

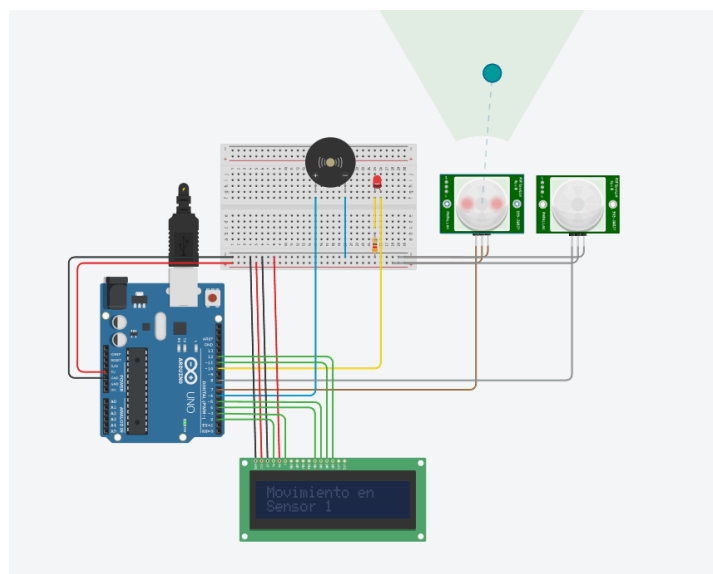
Hemos dispuesto los dos sensores de movimiento PIR uno al lado del otro por comodidad a la hora de simularlo, pero podrían colocarse en cualquier posición.

Cuando uno de los dos sensores capta movimiento se activan simultáneamente tanto el zumbador como el led rojo, estos son los que actúan como alarma. Esta 'alarma' es igual para cualquiera de los dos sensores, lo que podría ser un problema a la hora del funcionamiento. Para lidiar con este inconveniente disponemos de la pantalla LCD que actúa como menú, indicándonos la situación de cada una de los dos sensores, ya que, aunque la alarma sea igual para los dos sensores, el LCD nos indica cuál de los dos sensores es el que está captando el movimiento.

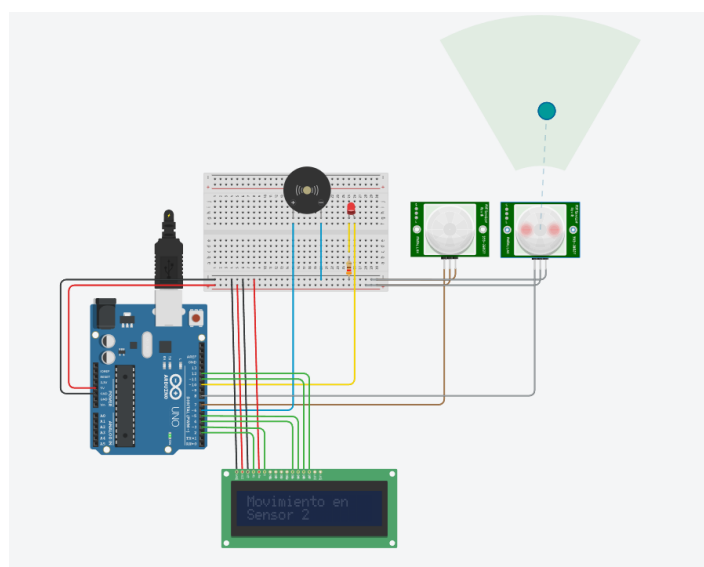
El funcionamiento sería el mostrado en las siguientes imágenes en las cuales mostramos las distintas situaciones; en la primera de las imágenes donde se muestra el sistema cuando no hay movimiento, en la segunda cuando está en el sensor número 1 habiendo movimiento y en la tercera cuando está en el sensor número 2 también en movimiento.



*Figura 2: Simulación sin movimiento*



*Figura 3: Simulación con movimiento en sensor 1*



*Figura 4: Simulación con movimiento en el sensor 2*

## 3.2 Algoritmo del trabajo

Para la edición del código hemos utilizado el editor de código *ArduinoIDE* y para poder comprender el funcionamiento del código vamos a convertirlo a pseudocódigo para poder ver más claramente el funcionamiento de las funciones:

Comenzamos importando la librería `LiquidCrystal.h`

Definimos los pines `rs`, `en`, `d4`, `d5`, `d6` y `d7` para el LCD

Configuramos el LCD con estos pines

Definimos el pin `pir1` para el primer sensor de movimiento (PIR)

Definimos el pin `pir2` para el segundo sensor de movimiento (PIR)

Definimos el pin `zumbador` para el buzzer

Definimos el pin `led` para el LED

Inicio del programa:

- Iniciar la comunicación para enviar mensajes a la computadora
- Configurar el LCD para mostrar texto con 16 columnas y 2 filas
- Configurar los pines `pir1` y `pir2` como entradas (para leer sensores)
- Configurar los pines `zumbador` y `led` como salidas (para controlar el buzzer y el LED)
- Enviar el mensaje "Inicialización completa" a la computadora

Bucle repetitivo (se ejecuta una y otra vez):

- Si el sensor `pir1` detecta movimiento:
  - Borrar el texto en el LCD
  - Mostrar "Movimiento en" en la primera línea del LCD
  - Mostrar "Sensor 1" en la segunda línea del LCD
  - Encender el zumbador
  - Encender el LED
- Si no, pero el sensor `pir2` detecta movimiento:
  - Borrar el texto en el LCD
  - Mostrar "Movimiento en" en la primera línea del LCD
  - Mostrar "Sensor 2" en la segunda línea del LCD
  - Encender el zumbador
  - Encender el LED
- Si ninguno de los dos sensores detecta movimiento:
  - Borrar el texto en el LCD
  - Mostrar "Sin movimiento" en la primera línea del LCD
  - Apagar el zumbador
  - Apagar el LED
- Esperar un momento (medio segundo)
- Enviar a la computadora el estado actual de los sensores (si detectan o no movimiento)

## 4. Conclusión

---

Consideramos que ha sido un trabajo interesante, hemos elegido la ampliación de este a pesar de que en su momento hicimos el de las notas musicales, ya que nos resultaba mas interesante. Hemos intentado enfocar el trabajo de una manera que fuese sencillo de replicar en físico a la hora del montaje, ya que nos parecía un sistema que puede ser muy útil.

Hemos tratado de completar el trabajo lo mejor, el único objetivo que nos habría gustado mejorar es el menú del LCD, haciéndolo mas complejo, pero hemos quedado muy satisfecho con el resultado final del proyecto.

Respecto a los programas utilizados hemos usado arduinoIDE que es el programa que hemos usado durante toda la asignatura, pero en vez de utilizar simulIDE como hemos hecho con el resto de practicas hemos usado tinckercad a pesar de disponer de menos componentes que simulIDE, porque tiene una interfaz mas agradable a la vista, y nos resultaba mas cómodo a la hora de simular el funcionamiento del programa.