Trabajo Práctico final

Laboratorio de computación 1

2do cuatrimestre 2025

Profesores:

MATIAS JOSE GAGLIARDO

PEDRO FACUNDO IRISO

Grupo:

Nehuen Mattivi – DNI: 47700343

Nicolas Olivares – DNI: 46987189

Tomas cereceda – DNI: 47635691

Leandro Farias – DNI: 37810979

1-Resumen del proyecto:

Este trabajo muestra el desarrollo y la puesta en práctica de un modelo de dispensador automático de alimento para animales de compañía, que emplea un microcontrolador Arduino Uno como núcleo de procesamiento. La meta fundamental es automatizar el suministro de comida, utilizando un sistema que detecta la proximidad, imitando la presencia de una mascota o un plato que no contiene alimento.

El sistema se integra por tres componentes fundamentales:

1. Detección (Entrada): Se emplea un sensor ultrasónico HC-SR04 para calcular distancias. El firmware está programado para realizar una medición actualizada cada 200 milisegundos.

2. Lógica (Procesamiento): La activación principal de la lógica se produce al detectar un objeto dentro de un rango establecido, que varía entre 8 y 16 centímetros.

3. Acción (Salida): Un servomotor (conectado al pin 10) que simula el funcionamiento de una compuerta, acompañado de un LED indicador (pin 12) que ofrece una señal visual de retroalimentación.

La característica más destacada del firmware es su arquitectura no bloqueante. Para evitar que el microcontrolador se congele durante las

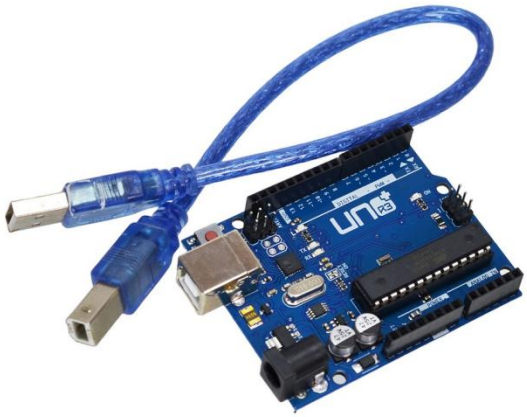
pausas, el sistema evita por completo el uso de delay() en su bucle principal. En su lugar, toda la temporización se gestiona mediante la función millis().

Esto permite al sistema realizar dos tareas en aparente simultaneidad:

- Continuar midiendo la distancia del sensor (cada 200 ms).
- Ejecutar la secuencia de dispensación del motor (que dura varios segundos).

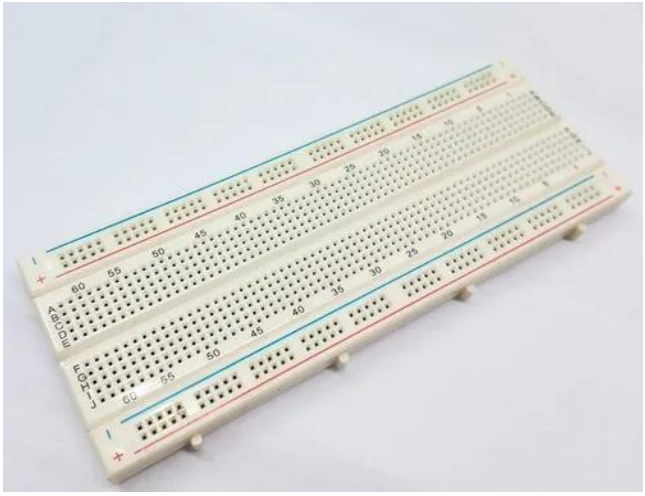
2-Descripción técnica del hardware

- **Arduino uno:**



El Arduino Uno es un placa de desarrollo que se fundamenta en el microcontrolador ATmega328P, que forma parte de la serie AVR, producido por Microchip. Su propósito es simplificar el proceso de aprendizaje y la realización de proyectos en el ámbito de la electrónica y la programación integrada. Opera con un voltaje de 5V y cuenta con 14 pines digitales que pueden utilizarse para entrada o salida, además de 6 pines analógicos para la entrada, lo que permite la conexión de diferentes sensores, actuadores y módulos. También incluye un conector USB tipo B que sirve tanto para la comunicación como para la alimentación.

- **Protoboard**



El protoboard es una herramienta que se utiliza para hacer conexiones temporales, facilitando el ensamblaje y la prueba de circuitos electrónicos sin tener que soldar. Cuando se coloca un componente o un cable en uno de los agujeros, se conecta eléctricamente con los otros agujeros que están en la misma fila o columna interna. Dentro de su diseño hay pistas metálicas que conectan eléctricamente las filas de agujeros. Además, tiene bordes laterales que están marcados en rojo (+) y azul o negro (–) para gestionar el voltaje y tierra (GND).

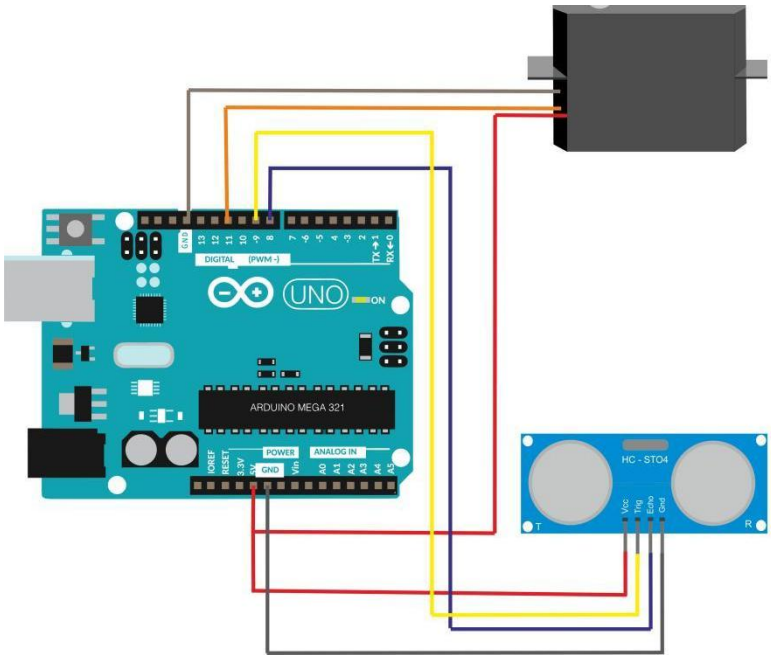
- **Micro Servomotor MG90S:**

El Micro Servomotor MG90S es un pequeño motor controlado electrónicamente que puede girar hasta unos 180° con gran precisión, muy usado en robótica y proyectos con Arduino. Funciona enviándole señales que indican el ángulo exacto al que debe posicionarse, por eso es ideal para mover brazos robóticos, sensores o mecanismos pequeños. Está construido con engranajes metálicos, lo que le da más fuerza y durabilidad que otros servos de plástico. Además, consume poca energía y es liviano, por lo que se adapta muy bien a proyectos educativos y maquetas. Su tamaño compacto permite integrarlo en espacios reducidos sin perder potencia. En resumen, es un componente versátil, preciso y confiable para cualquier proyecto de control de movimiento.

- **Sensor de Ultrasonido HC-SR04**

El Sensor de Ultrasonido HC-SR04 es un dispositivo que permite medir distancias utilizando ondas de sonido de alta frecuencia, imperceptibles para el oído humano. Funciona emitiendo un pulso ultrasónico y midiendo cuánto tarda en rebotar contra un objeto y regresar, lo que permite calcular la distancia con muy buena precisión. Este sensor es ampliamente usado en robótica, alarmas y sistemas de detección porque es barato, fácil de usar y muy confiable. Cuenta con dos partes visibles: el emisor (que envía el sonido) y el receptor (que lo recibe). Además, se conecta fácilmente a placas como Arduino mediante cuatro pines y puede medir distancias que van aproximadamente desde 2 cm hasta 4 metros. Es ideal para proyectos que necesitan detectar obstáculos o medir espacios de forma rápida y precisa.

3-Diagrama de conexiones



4- Descripción general del software:

El software fue desarrollado en el IDE de Arduino (lenguaje C/C++) para controlar un sistema automático de dispensación de alimento para mascotas. El programa utiliza un sensor ultrasónico HC-SR04 para detectar la proximidad (simulando un plato) y un servomotor para accionar el mecanismo de dispensación.

El programa está organizado en torno a una lógica no bloqueante, utilizando la función millis() para gestionar todas las temporizaciones. Esto permite que el sistema maneje dos procesos principales de forma concurrente:

1. La medición periódica de distancia del sensor.
2. La ejecución de una secuencia temporizada (una máquina de estados finitos) para el motor, una vez que se detecta un objeto.

• Estructura general del código

A- Declaración de librerías, pines y variables:

Esta parte del codigo importa la libreria Servo.h que sirve para manejara al ServoMotor, tambien se define los pines del sensor y del motor, crea variables para guardar la distancia del sensor y controla el movimiendo del motor

Ejemplo

```
#include <Servo.h>
const int sensor1 = 8;
const int sensor2 = 9;
```

```
const int motor1 = 12;
const int motor2 = 10;
    Servo servoMotor;
unsigned int distancia;
bool leerSensor = true;
```

B- Función setup:

Esta funcion configura los pines del sensor y del motor, conecta el motor al pin definido y lo deja en posicion 0 y se asegura que el motor es apagado al inicio

Eejmplo

```
void setup() {
    Serial.begin(9600);
    pinMode(sensor1, INPUT);
    pinMode(sensor2, OUTPUT);
    pinMode(motor1, OUTPUT);
    servoMotor.attach(motor2);
    servoMotor.write(0);
```

C- Función loop:

Lo que hace es que cada 200ms mide la distancia y controla la estapas del motor (abrir-cerrar)

Ejemplo

```
void loop() {
  unsigned long tiempoActual = millis();
        if (tiempoActual - tiempoAnterior >= intervaloMedicion) {
              tiempoAnterior = tiempoActual;
              medirSensor();
        } if (movimientoActivo) {
                  controlarMotor(tiempoActual);}
```

D- Función medirSensor:

Genera el pulso del sensor trigger, tambien mide cuanto tarda en volver el pulso del sensor echo y convierte ese tiempo en distancia

Ejemplo

```
void medirSensor() {
        digitalWrite(sensor2, LOW);
        delayMicroseconds(2);
         digitalWrite(sensor2, HIGH);
        delayMicroseconds(10);
         digitalWrite(sensor2, LOW);
        unsigned long duracion = pulseIn(sensor1, HIGH, 30000);
  if (duracion == 0) return;
  distancia = duracion / 58;
```

Luego, si la distancia esta entre 8 y 16 cm, se activa y el motor lo mueve 180°, tambien apaga la lectura del sensor para no reactivarse mientras este en funcionamiento

Ejemplo

```
if (distancia <= 16 && distancia >= 8 && leerSensor && !movimientoActivo) {
        leerSensor = false;
        movimientoActivo = true;
        tiempoMotor = millis();
        estadoMotor = 0;
        servoMotor.write(180);
```

E* Función controlarMotor

Esta funcion controla tres etapas;

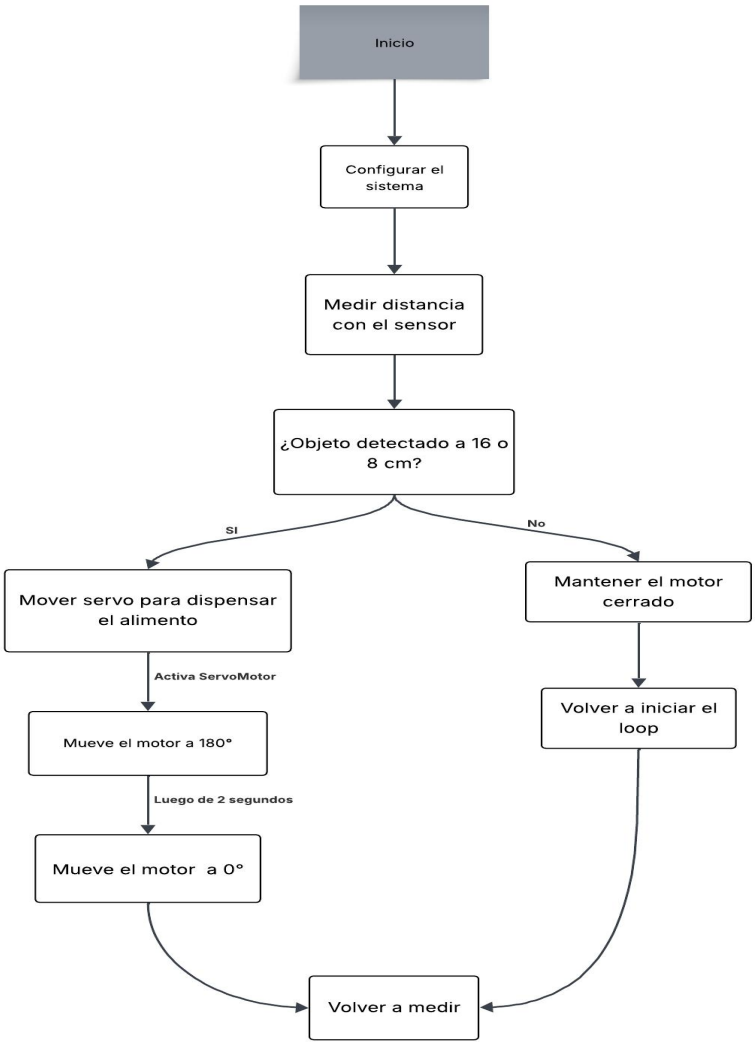
La primera espera 2 segundos, lleva al motor a 0°, la segunda espera otros 2 segundos, lleva al motor a 180°, y la tercera espera 2 segundos y lo vuelve a 0°

Ejemplo

```
void controlarMotor(unsigned long tiempoActual) {
  switch (estadoMotor) {
    case 0:
      if (tiempoActual - tiempoMotor >= 2000) {
        servoMotor.write(0);
        tiempoMotor = tiempoActual;
        estadoMotor = 1;
      } break;
case 1:
      if (tiempoActual - tiempoMotor >= 2000) {
```

```
      servoMotor.write(180);
      tiempoMotor = tiempoActual;
      estadoMotor = 2;
    } break;
case 2:
    if (tiempoActual - tiempoMotor >= 2000) {
      servoMotor.write(0);
      movimientoActivo = false;
      leerSensor = true;
      digitalWrite(motor1, LOW);
    } break;
```

5- Diagrama de máquina de estado



6- Descripción de contador de flancos y control de tiempo

A- Flancos:

El código implementa un mecanismo que detecta cuando la distancia medida entra en un rango específico. Esto se logra usando las variables leerSensor y movimientoActivo, que permiten generar un flanco de

activación, evitando que el motor se active repetidamente mientras el objeto esté frente al sensor:

```
if (distancia <= 16 && distancia >= 8 && leerSensor && !movimientoActivo) {
    leerSensor = false;
    movimientoActivo = true;
    tiempoMotor = millis();
    estadoMotor = 0;
    servoMotor.write(180);
    digitalWrite(motor1, HIGH);
}
```

Esto hace que solo se detecte cuando se cumpla la distancia requerida, leerSensor esta en true solo esa vez, y movimientoActivo en false, luego se invierten, es decir que leerSensor pasa a false y movimientoActivo pasa a true.

Despues aunque el objeto siga adelante del sensor ya no se volveria a activar el motor, ya que:

```
leerSensor == false
```

Esta linea de codigo evita que el motor arranque repetidamente

B- Tiempo

El movimiento del motor se divide en 3 etapas, cada una separada por 2 segundos. Esto se controla midiendo el tiempo transcurrido con millis en lugar de usar delay, permitiendo que el programa siga ejecutándose mientras espera:

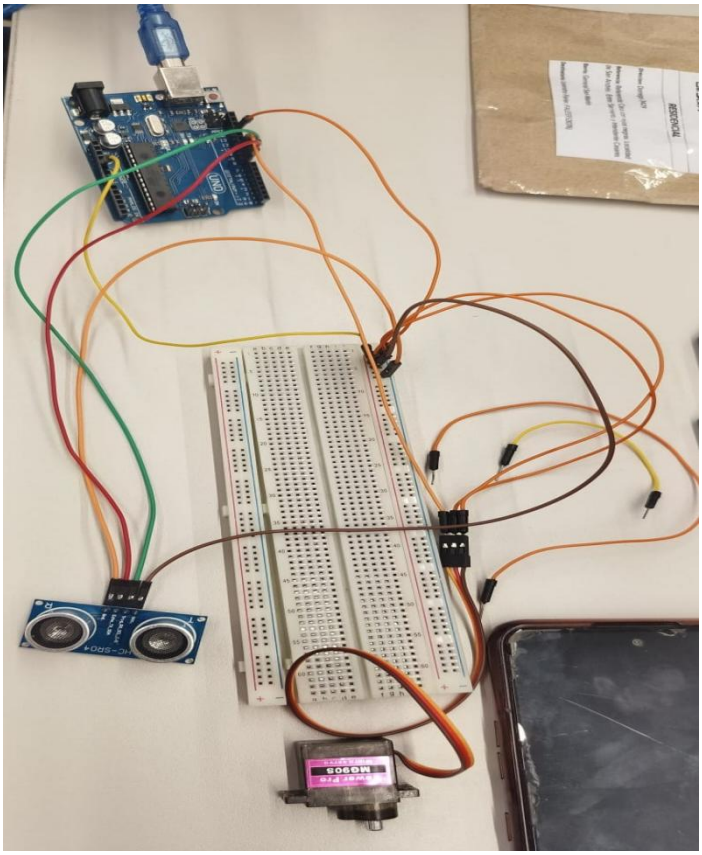
```
case 0:
```

```
if (tiempoActual - tiempoMotor >= 2000) {
    servoMotor.write(0);
    tiempoMotor = tiempoActual;
    estadoMotor = 1;
}
break;
```

Cada etapa se maneja con una máquina de estadoMotor, avanzando secuencialmente hasta completar el ciclo y restablecer la lectura del sensor.

7- Capturas del sistema en funcionamiento

- **Sistema conectado:**



- **Sistema andando:**

HC-SR04
TENEMOS TODO
0810-220-2345

- **Video:** https://github.com/maxi-meri/Trabajo-Arduino-2025/blob/main/media/Demostracion.mp4

8- Conclusión Grupal

A lo largo del desarrollo de este proyecto, logramos aplicar de manera práctica los fundamentos de los sistemas embebidos mediante el diseño y la construcción de un dispensador automático de alimento. Se implementó con éxito un sistema de control que utiliza un sensor ultrasónico HC-SR04 para la detección de proximidad y un servomotor SG90 como actuador principal. El equipo está satisfecho con haber alcanzado el objetivo de automatizar la dispensación de alimento.

El uso de la Máquina de Estados Finitos (FSM) para controlar el servomotor fue crucial, ya que nos permitió manejar la secuencia de apertura y cierre de la compuerta de forma temporizada y completamente no bloqueante. La gestión del tiempo basada en millis() evitó el uso de delay(), asegurando que el sensor pudiese monitorear continuamente el plato. Esto constituye un pilar fundamental en la programación de microcontroladores que requieren alta reactividad.

Como grupo, el proyecto reforzó nuestra comprensión de la integración hardware-software, la gestión de actuadores de precisión y el desarrollo de lógica de control avanzada. Una mejora futura que consideramos esencial es la integración de conectividad Wi-Fi para notificar dispensaciones o la adición de un sensor de bajo nivel para alertar cuando el depósito de alimento esté vacío. El resultado final demuestra que el Arduino UNO es una herramienta muy versátil para soluciones domóticas.