



Proyecto Final: Contador Mecánico

Grupo 15

SISTEMAS EMPOTRADOS Y DE TIEMPO REAL
Grado en Ingeniería Informática
Curso 2024/2025

Índice

Índice	1
1. Resumen del Proyecto	2
2. Implementación Técnica	3
3. Pasos Seguidos	5
4. Reparto de Tareas	6
5. Costes de Materiales	7
6. Casos de Uso	8
7. Código	9
8. Problemas Encontrados y Soluciones	11
9. Conclusiones	12

1. Resumen del Proyecto

Este proyecto consiste en el desarrollo de un contador alfanumérico mecánico basado en un display de 7 segmentos operado mediante servomotores, con control centralizado a través de una placa Arduino UNO. A diferencia de los tradicionales displays LED electrónicos que encienden segmentos mediante señales eléctricas, este diseño opta por una representación completamente física y mecánica, donde cada segmento del display es una placa móvil que se activa o desactiva utilizando servos individuales. Esta aproximación no solo resulta visualmente atractiva, sino que también tiene un fuerte componente didáctico, al permitir observar de forma tangible el funcionamiento interno de un display de 7 segmentos.

El sistema realiza una animación de carga que simula tres olas de movimiento consecutivas, una por cada conjunto de segmentos alineados vertical u horizontalmente, generando un efecto visual dinámico que destaca el funcionamiento mecánico de los componentes. Finalizada esta secuencia de inicio, el dispositivo entra en un bucle continuo en el que recorre secuencialmente los caracteres numéricos del 0 al 9 y las letras alfabéticas de la A a la G, representando cada uno de ellos mediante la combinación adecuada de segmentos activados.

Cada segmento está conectado a un servomotor controlado mediante señales PWM enviadas desde la placa Arduino. El software desarrollado gestiona la secuencia de caracteres, las posiciones de los servos y la sincronización de los movimientos, incluyendo los retardos y transiciones necesarias para asegurar un funcionamiento fluido y coordinado. El proyecto no solo pone en práctica conocimientos de electrónica y programación, sino también de diseño mecánico y control de actuadores, siendo especialmente útil para fines educativos y de demostración.

2.Implementación Técnica

Plataforma de Control:

El proyecto está basado en una placa Arduino UNO, que se utiliza como unidad de control central. Esta placa se encarga de gestionar la secuencia de activación de los servomotores y la coordinación de los movimientos del display de 7 segmentos. La comunicación entre el Arduino y los servos se realiza a través de señales, las cuales permiten controlar con precisión las posiciones de los servos.

Componentes Electrónicos:

Servomotores: Se emplean servomotores de rotación limitada para activar cada uno de los segmentos del display de 7 segmentos. Cada servo se conecta a un segmento del display, y al mover el servo a distintas posiciones, se permite que cada segmento se muestre o se oculte, formando así los números y letras deseadas.

Display de 7 segmentos mecánico: En lugar de usar un display LED tradicional, se utilizan placas físicas (segmentos) que son movidas por los servos. Cada placa tiene dos posiciones: activada (visible) o desactivada (oculta). La combinación de estas posiciones forma los números del 0 al 9 y las letras de la A a la G.

Fuente de Alimentación: Una fuente de alimentación externa externa de 12V regulada a 5V que es necesaria para suministrar la energía tanto al Arduino como a los servomotores, dado que estos pueden requerir más corriente que la que puede proporcionar el propio Arduino.

Diseño Mecánico:

El display está compuesto por 7 segmentos individuales, cada uno accionado por un servomotor. Estos segmentos están dispuestos en una estructura que permite que los servos muevan cada placa con precisión.

Los servos están montados sobre una base sólida y conectados a cada segmento mediante mecanismos de transmisión (por ejemplo, engranajes o varillas) que convierten el movimiento rotacional del servo en un desplazamiento lineal o angular del segmento.

El diseño de los segmentos está pensado para asegurar que cada placa pueda moverse de forma fluida y precisa, sin interferir con las demás.

Control y Software:

El software de control está desarrollado en el entorno de programación de Arduino (IDE), utilizando el lenguaje de programación C ++

El código controla la secuencia de movimiento de los servos para representar los números y las letras. Utiliza una matriz predefinida de posiciones para cada carácter, que se almacena en un arreglo en la memoria del Arduino.

La animación de inicio, que simula las tres olas de movimiento, se realiza programando una secuencia de movimientos coordinados de los servos, creando un efecto visual atractivo y dinámico antes de comenzar el conteo alfanumérico.

El bucle principal del contador recorre los números del 0 al 9 y las letras de la A a la G en orden, con un pequeño retraso entre cada transición para permitir una visualización clara de cada carácter.

Sincronización y Retardos:

La sincronización de los movimientos de los servos es crucial para asegurar que no haya interferencias entre los segmentos y que cada carácter se forme correctamente. Se han implementado retardos en el código para controlar el tiempo de exposición de cada carácter.

El tiempo de retraso entre cada movimiento está ajustado para asegurar que el sistema sea lo suficientemente rápido como para no ser monótono, pero lo suficientemente lento como para ser legible.

Interacción y Expansión:

En una posible versión futura del proyecto, se podrían agregar sensores o botones para permitir la interacción del usuario, como cambiar la secuencia de caracteres o modificar la velocidad del contador.

Además, la arquitectura del software y el diseño del hardware son fácilmente escalables, permitiendo la expansión del número de segmentos o la adición de nuevas funciones.

3.Pasos Seguidos

1. Diseño conceptual del contador alfanumérico.
2. Modelado 3D de los segmentos móviles e impresión en PLA.
3. Montaje de los servos y fijación en la base.
4. Conexión del circuito eléctrico en la protoboard: servos, fuente de alimentación, regulador de voltaje y Arduino.
5. Programación del código en el entorno Arduino.
6. Pruebas individuales de cada segmento.
7. Implementación de la animación de arranque.
8. Pruebas completas del ciclo de caracteres.
9. Pintura de piezas y retoques finales.

4.Reparto de Tareas

Integrantes	Tareas Realizadas
Pablo	• Participación en la búsqueda de la idea inicial del proyecto. • Soporte en la parte de hardware: conexiones eléctricas y organización del cableado. • Participación activa ensamblaje físico de las piezas. • Compra de un adaptador de alimentación para solventar un problema de voltaje. • Contribución económica equitativa en la compra de materiales. • Colaboración en la creación de la presentación y participación en la exposición. • Participación en el montaje final del sistema. • Redacción parcial de la memoria y blog del proyecto
Alonso	• Participación en la búsqueda de la idea inicial del proyecto. • Desarrollo completo del código para que los servos funcionen y se realicen las animaciones. • Montaje principal del sistema mecánico y electrónico. • Búsqueda, selección y encargo de los materiales (servos, Arduino, UBEC, etc.). • Contribución económica equitativa en la compra de materiales. • Fase de calibración servomotores para solucionar el problema. • Colaboración en la creación de la presentación y participación en la exposición • Redacción parcial de la memoria y blog del proyecto.
Rodrigo	• Participación en la búsqueda de la idea inicial del proyecto. • Realización del diseño y simulación en Tinkercad, utilizado para probar el montaje y ayudar en parte a la solución del problema. • Participación activa durante todo el desarrollo del proyecto, colaborando en tareas técnicas y prácticas según las necesidades del equipo. • Colaboración en la creación de la presentación y participación en la exposición • Compra de la tabla base del proyecto. • Contribución económica equitativa en la compra de materiales. • Participación en el montaje fina del sistema y en la fase de pegado de piezas. • Redacción parcial de la memoria y blog del proyecto.

5. Costes de Materiales

Material	Cantidad	Precio unitario	Total	Observaciones
Arduino UNO	1	-	0€	Proporcionado por la universidad
Servomotores SG90	7	1,20 €	8,40 €	
Cables y protoboard	—	—	0 €	Proporcionado por la universidad
Regulador de voltaje 5V	1	3,00 €	3,00 €	
Fuente de alimentación 12V	1	4,00 €	4,00 €	
Filamento PLA para impresión 3D	~200 g	15 €/kg	3,00 €	
Pintura en spray	1	5,00 €	5,00 €	
Estructura (madera, cartón)	—	—	5,00 €	
Pegamento	1	3,00 €	3,00 €	
Total estimado	—	—	30,40 €	Aproximadamente

6. Casos de Uso

Aunque el proyecto tiene una función principal muy concreta (mostrar letras o números mediante el movimiento de servomotores), su diseño y funcionamiento permiten imaginar distintas aplicaciones en las que puede resultar útil o interesante.

Presentaciones y ferias tecnológicas

Es un proyecto visualmente atractivo y original, por lo que encaja muy bien en exposiciones, concursos o ferias tecnológicas. El movimiento de los segmentos llama la atención y permite mostrar una forma distinta de representar información.

Introducción al uso de servomotores y estructuras mecánicas

Puede ser útil para aprender a manejar servomotores, coordinar movimientos y montar estructuras físicas. Puede utilizarse como ejemplo en talleres básicos de robótica o Arduino, especialmente para mostrar cómo hacer que varias partes móviles trabajen de forma ordenada.

Decoración tecnológica o mensajes animados

El sistema se puede adaptar para mostrar secuencias de letras o números decorativos, como parte de una instalación o como un elemento curioso en una habitación o un aula. También podría modificarse para mostrar mensajes en tiempo real o funcionar como contador visible de eventos o visitas.

Proyecto base para mejoras

A partir de este diseño se pueden desarrollar versiones más completas, como añadir más dígitos, integrar sensores o conectar el sistema a una app o a Internet para hacerlo más dinámico. Esto lo convierte en una buena base para continuar aprendiendo o ampliando el proyecto en el futuro.

7. Código

```
#include <Servo.h> Importamos la librería Servo.h para poder usar sus funciones Servo servos[7];

const int segmentPins[] = {2, 3, 4, 5, 6, 7, 8};
const int ON_POSITION = 0;
const int OFF_POSITION = 90;

const int segmentPatterns[17][7] = {
  {1, 1, 1, 1, 1, 1, 0}, // 0
  {0, 1, 1, 0, 0, 0, 0}, // 1
  {1, 1, 0, 1, 1, 0, 1}, // 2
  {1, 1, 1, 1, 0, 0, 1}, // 3
  {0, 1, 1, 0, 0, 1, 1}, // 4
  {1, 0, 1, 1, 0, 1, 1}, // 5
  {1, 0, 1, 1, 1, 1, 1}, // 6
  {1, 1, 1, 0, 0, 0, 0}, // 7
  {1, 1, 1, 1, 1, 1, 1}, // 8
  {1, 1, 1, 1, 0, 1, 1}, // 9
  {1, 1, 1, 0, 1, 1, 1}, // A
  {0, 0, 1, 1, 1, 1, 1}, // B
  {1, 0, 0, 1, 1, 1, 0}, // C
  {0, 1, 1, 1, 1, 0, 1}, // D
  {1, 0, 0, 1, 1, 1, 1}, // E
  {1, 0, 0, 0, 1, 1, 1}, // F
  {1, 0, 1, 1, 1, 1, 0}  // G
};

int currentPos[7];

void setup() {
  Serial.begin(9600);

  delay(2000);

  for (int i = 0; i < 7; i++) {
    servos[i].attach(segmentPins[i]);
    servos[i].write(OFF_POSITION);
    currentPos[i] = OFF_POSITION;
  }

  waveEffect();
  delay(1000);
}

void loop() {
  for (int i = 0; i <= 9; i++) {
    Serial.print("Mostrando número: ");
    Serial.println(i);
    displayCharacter(i);
    delay(1000);
  }

  for (int i = 10; i <= 16; i++) {
    char letter = 'A' + (i - 10);
    Serial.print("Mostrando letra: ");
    Serial.println(letter);
    displayCharacter(i);
    delay(1000);
  }
}

void smoothMove(int i, int target) {
  if (target > currentPos[i]) {
    for (int pos = currentPos[i]; pos <= target; pos++) {
      servos[i].write(pos);
      delayMicroseconds(1000);
    }
  } else {
    for (int pos = currentPos[i]; pos >= target; pos--) {
      servos[i].write(pos);
    }
  }
}
```

```

        delayMicroseconds(1000);
    }
}
currentPos[i] = target;
}

void waveEffect() {
    for (int i = 0; i < 3; i++) { //
        for (int j = 0; j < 7; j++) {
            servos[j].write(ON_POSITION);
            delay(100);
        }
        for (int j = 0; j < 7; j++) {
            servos[6 - j].write(OFF_POSITION);
            delay(100);
        }
    }
}

void displayCharacter(int index) {
    for (int i = 0; i < 7; i++) {
        int target = (segmentPatterns[index][i] == 1)
            ? ON_POSITION : OFF_POSITION;
        smoothMove(i, target);
    }
}

```

Primero introducimos la librería Servo.h para poder usar las funciones de los servos correctamente, el número de servos que estamos usando (7) y los pines a los que cada uno de ellos están conectados en la placa de arduino.

Las posiciones ON = 0° y OFF 90° controlan si un segmento está encendido o apagado.

Posteriormente con la matriz se definen por líneas los distintos caracteres que vamos a representar que en nuestro caso son 17.

Setup: Lo primero que hace al conectarlo a la corriente.

1. Espera dos segundos
2. Inicializa los servos en la posición deseada por si en anteriores pruebas algunos de los servos han quedado en cualquier otra posición que no sea la inicial.
3. Animación de carga con un efecto de ola que se repite 3 veces.

Loop: Secuencia de animaciones sin fin.

1. Muestra del 0-9.
2. Muestra de la A-G.

Usa la funcion displayCharacter() para mover los servos con un tiempo de espera de 1 segundo entre caracteres.

Funciones:

1. smoothMove()
Mueve el servo de manera fluida hasta su nueva posición.
2. waveEffect()
Animación tipo ola antes de empezar.
3. displayCharacter()
Activa segmentos según la tabla.

8. Problemas Encontrados y Soluciones

Sincronización de los servomotores

Uno de los principales problemas fue lograr que los siete servomotores se movieran de forma coordinada sin generar interferencias eléctricas ni comportamientos erráticos. Al conectar varios servos directamente a la placa Arduino, algunos motores no respondían correctamente o vibraban sin control.

Solución:

Se incorporó un regulador de voltaje de 5V y una fuente de alimentación externa de 12V, lo cual permitió alimentar los servos de forma estable y aislada del Arduino. Además, se añadieron pequeños retardos (delays) en el código para evitar que todos los servos se activaran exactamente al mismo tiempo, reduciendo picos de consumo y mejorando la fluidez del movimiento.

Limitaciones mecánicas de la estructura

El diseño inicial de la estructura del display, impreso en 3D y montado sobre cartón y madera, presentaba problemas de alineación entre los segmentos móviles y los puntos de sujeción de los servos. Esto provocaba bloqueos o movimientos incompletos en algunos segmentos.

Solución:

Se realizaron ajustes en el diseño y montaje de la estructura, mejorando la fijación de los servomotores y rediseñando algunos soportes con filamento PLA para lograr un movimiento más suave y preciso. También se reforzaron algunas partes con pegamento y se verificó la libertad de movimiento de cada segmento antes del montaje final.

9. Conclusiones

Este proyecto ha sido una experiencia útil tanto a nivel técnico como práctico. A lo largo del desarrollo, se han trabajado distintas áreas como el diseño mecánico, la programación de servomotores y la planificación de una estructura física que funcione de forma coordinada.

Uno de los mayores logros ha sido conseguir que los siete servos trabajen juntos de manera fluida, sin interferencias ni bloqueos, algo que no era tan sencillo como parecía al principio. También se ha aprendido a manejar una fuente de alimentación externa y a optimizar el funcionamiento del sistema para evitar sobrecargas.

En cuanto a la parte de montaje, ha sido necesario diseñar y ajustar la estructura física del display, asegurándonos de que cada segmento tuviera libertad de movimiento y estuviera bien alineado. Esto nos ha llevado a realizar varias pruebas, rediseños y adaptaciones sobre la marcha, lo que ha sido una parte fundamental del aprendizaje práctico del proyecto.

Otro aspecto importante ha sido el trabajo en equipo. Hemos tenido que repartir tareas de forma equilibrada, coordinarnos para el montaje y la programación, y colaborar en la resolución de problemas cuando algo no funcionaba como esperábamos. Este proceso nos ha ayudado a mejorar nuestras habilidades de organización, comunicación y toma de decisiones en grupo.

Además, el proyecto nos ha enseñado a enfrentarnos a problemas reales, no previstos inicialmente, y a buscar soluciones creativas con los recursos disponibles. La capacidad de adaptación y la persistencia han sido claves para superar los obstáculos que surgieron a lo largo del desarrollo.

En definitiva, más allá del resultado final, que ha sido satisfactorio y funcional, este proyecto ha sido una oportunidad para aprender de forma práctica, integrar conocimientos de distintas áreas y comprobar cómo una idea inicial puede convertirse en un sistema real mediante trabajo, pruebas y colaboración. También deja la puerta abierta a posibles mejoras y ampliaciones futuras, ya que se trata de una base sólida sobre la que podrían desarrollarse versiones más avanzadas e interactivas.