



INSTITUTO TECNOLÓGICO DE MORELIA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

SISTEMAS PROGRAMABLES

PRÁCTICA 0: CONTADOR ASCENDENTE Y DESCENDENTE

PRESENTA:

OMAR Yael GALVAN MATA (21121517)

SANTIAGO GONZALEZ LARA (22121360)

DOCENTE:

LUIS ULISES CHÁVEZ CAMPOS

ÍNDICE

Objetivo.....	3
Marco Teórico / Introducción Teórica.....	3
Procedimiento Experimental / Desarrollo.....	4
Materiales y Equipo / Material Utilizado.....	4
Análisis Teórico.....	4
Simulación y Código.....	5
Resultados y Análisis.....	7
Discusión.....	7
Interpretación de los resultados.....	7
Comparación con referencias previas o literatura.....	8
Análisis de posibles errores.....	8
Limitaciones del experimento.....	8
Propuesta de mejoras.....	8
Implicaciones del resultado.....	8
6. Conclusiones.....	9
Síntesis de los resultados más relevantes.....	9
Cumplimiento de los objetivos.....	9
Implicaciones del experimento.....	9
Limitaciones y mejoras futuras.....	9

Objetivo

El objetivo de esta práctica es familiarizarse con el uso del microcontrolador Arduino como herramienta fundamental para el desarrollo de sistemas programables. A través de la implementación de circuitos básicos y la programación de la placa Arduino, los estudiantes aprenderán a:

- Configurar y programar entradas y salidas digitales.
- Comprender el funcionamiento básico de componentes como LEDs y botones pulsadores.
- Aplicar conceptos básicos de programación en el entorno de desarrollo integrado (IDE) de Arduino.
- Desarrollar habilidades para el diseño y la implementación de sistemas programables simples, sentando las bases para proyectos más avanzados en el futuro.

Esta práctica proporcionará una introducción práctica y accesible al mundo de la electrónica programable, facilitando el aprendizaje y la aplicación de conceptos clave en la automatización y el control de procesos.

Marco Teórico / Introducción Teórica

Un sistema programable es un dispositivo electrónico diseñado para controlar y automatizar procesos mediante software. En este caso, se emplea la plataforma **Arduino**, ampliamente utilizada por su facilidad de uso y versatilidad.

- **Arduino:** plataforma de hardware y software libre que permite programar microcontroladores para interactuar con el entorno mediante sensores, actuadores y periféricos. Su programación se basa en dos funciones principales: `setup()` (inicialización) y `loop()` (ejecución continua).
- **LED:** diodo emisor de luz que indica visualmente el estado de un sistema. Se conecta con su ánodo al voltaje y su cátodo a tierra mediante una resistencia limitadora.
- **Push Button:** interruptor que permite al usuario interactuar con el sistema al cerrar o abrir un circuito. En esta práctica se usa uno para incrementar el contador y otro para disminuirlo.

El contador binario de 4 bits permite representar valores del **0 al 15 en binario**, utilizando los 4 LEDs como salidas digitales que muestran cada bit.

Procedimiento Experimental / Desarrollo

Materiales y Equipo / Material Utilizado

- 1 Placa **Arduino UNO**
- 4 LEDs
- 4 Resistencias (220 Ω – 1k Ω)
- 2 Botones pulsadores
- Protoboard
- Cables de conexión

Análisis Teórico

El proyecto consiste en implementar un **contador binario de 4 bits** utilizando una placa Arduino.

- **Funcionamiento general:**
 - Los 4 LEDs representan los bits de un número binario (del 0 al 15 en decimal).
 - Cada pulsación del **botón ascendente** incrementa el valor binario mostrado.
 - Cada pulsación del **botón descendente** decrementa el valor binario mostrado.
 - Si el contador supera el límite (mayor a 15 o menor a 0), se reinicia en el valor contrario (0 \rightarrow 15 y 15 \rightarrow 0).
- **Lógica de control:**

Se utilizan estructuras condicionales (**if-else**) para verificar el estado de los botones y actualizar el contador. Los LEDs muestran los bits mediante operaciones de desplazamiento (**>>**) y enmascaramiento binario (**&**).
- **Asignación de pines:**
 - LEDs: **13, 12, 11, 10** (del bit más significativo al menos significativo).
 - Botones: **7** (ascendente) y **6** (descendente), configurados con **resistencia pull-up interna**.

Simulación y Código

El circuito puede simularse en plataformas como **Tinkercad** o montarse físicamente en protoboard.

El siguiente código implementa el contador ascendente y descendente:

```
// Pines de LEDs (de más significativo a menos significativo)
const int ledPins[4] = {13, 12, 11, 10};
```

```
// Pines de botones
const int btnUp = 7; // Botón para contar ascendente
const int btnDown = 6; // Botón para contar descendente
```

```
int counter = 0; // Contador de 0 a 15
```

```
// Variables para antirrebote
int lastStateUp = HIGH;
int lastStateDown = HIGH;
```

```
void setup() {
  // Configurar LEDs como salida
  for (int i = 0; i < 4; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
}
```

```
// Configurar botones como entrada con resistencia pull-up
pinMode(btnUp, INPUT_PULLUP);
pinMode(btnDown, INPUT_PULLUP);
```

```
// Mostrar el valor inicial
mostrarNumero(counter);
}
```

```
void loop() {
  // Leer estado de los botones
  int readingUp = digitalRead(btnUp);
  int readingDown = digitalRead(btnDown);
```

```
// Botón ascendente (activo en LOW por pull-up)
if (readingUp == LOW && lastStateUp == HIGH) {
  counter++;
  if (counter > 15) counter = 0; // Reinicia si pasa de 15
  mostrarNumero(counter);
  delay(200); // Delay para antirrebote
}
```

```
// Botón descendente
```

```
if (readingDown == LOW && lastStateDown == HIGH) {  
    counter--;  
    if (counter < 0) counter = 15; // Regresa a 15 si baja de 0  
    mostrarNumero(counter);  
    delay(200); // Delay para antirrebote  
}  
  
// Guardar estado previo  
lastStateUp = readingUp;  
lastStateDown = readingDown;  
}  
  
// Función para mostrar número en binario con LEDs  
void mostrarNumero(int num) {  
    for (int i = 0; i < 4; i++) {  
        int bitValue = (num >> i) & 1; // Extraer cada bit  
        digitalWrite(ledPins[i], bitValue);  
    }  
}
```

Con este programa, cada LED encendido representa un bit en 1 y apagado un bit en 0, mostrando así el valor binario correspondiente a cada número entre 0 y 15.

Resultados y Análisis

Al ejecutar el programa en Arduino:

- Los LEDs muestran correctamente los números binarios de 0 a 15.
- Cada pulsación del botón ascendente incrementa el número, y cada pulsación del descendente lo disminuye.
- Se logra comprobar el funcionamiento del microcontrolador como sistema programable sencillo.

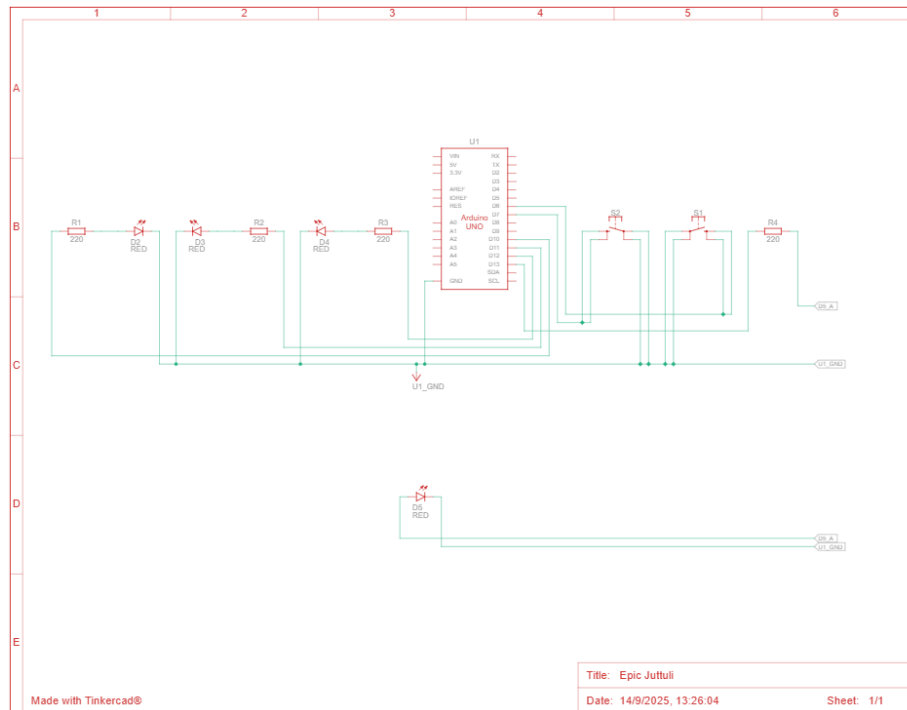
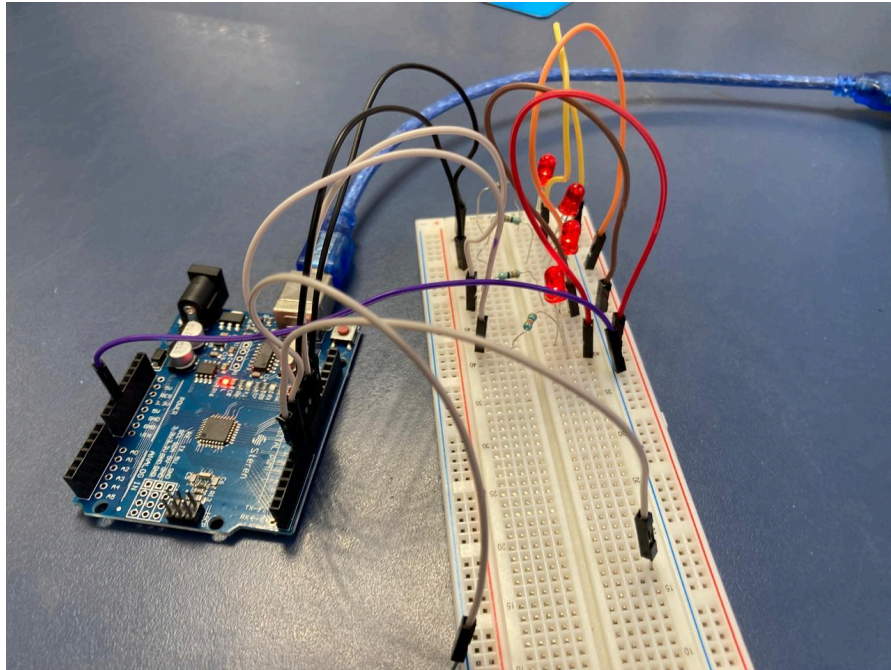


Diagrama de conexiones



c circuito funcionando

Discusión

Interpretación de los resultados

Los resultados obtenidos muestran que el **contador binario de 4 bits** implementado en Arduino funciona correctamente, ya que los LEDs representan de manera fiel los números binarios del 0 al 15. Cada pulsación del botón ascendente incrementó el valor mostrado y cada pulsación del botón descendente lo redujo, cumpliendo así con el objetivo planteado. Esto confirma que la hipótesis de que es posible diseñar un sistema programable sencillo utilizando entradas digitales (botones) y salidas digitales (LEDs) es válida.

Comparación con referencias previas o literatura

La práctica coincide con lo reportado en manuales de electrónica básica y guías de Arduino, donde se utilizan LEDs y botones para ilustrar los principios de los sistemas programables y la representación binaria. En comparación con ejemplos previos, los resultados fueron consistentes: el sistema responde adecuadamente a la interacción del usuario y refleja la lógica binaria de manera predecible. No se observaron discrepancias significativas con la teoría o la literatura de referencia.

Análisis de posibles errores

Durante la implementación podrían haber ocurrido errores relacionados con el **rebote de los botones**, ya que un pulsador mecánico no siempre genera una transición limpia de estado, lo que puede producir incrementos o decrementos no deseados en el contador. Este problema se mitigó con la inclusión de retardos en el código (debounce simple), aunque no se implementó una técnica más robusta. Otro posible error se encuentra en las conexiones físicas: un mal contacto en la protoboard podría haber alterado temporalmente el funcionamiento del circuito.

Limitaciones del experimento

El experimento estuvo limitado por la simplicidad de los componentes: se utilizó un sistema manual con botones que no garantiza exactitud en cada pulsación. Tampoco se evaluó el desempeño con diferentes condiciones de carga en los pines de salida ni se probaron alternativas de antirrebote más precisas. Además, el uso de solo 4 bits restringe el rango del contador a 16 valores, lo que limita la exploración de números binarios de mayor magnitud.

Propuesta de mejoras

En una próxima iteración, sería recomendable implementar un **sistema de antirrebote por software más avanzado**, como el uso de temporizadores o librerías especializadas, o incluso añadir **circuitos externos de acondicionamiento** (filtros RC o flip-flops). También se podría ampliar el proyecto a más bits utilizando registros de desplazamiento o expandir la salida con displays de 7 segmentos para una representación más clara. Otra mejora sería registrar las pulsaciones con interrupciones en lugar de leer continuamente el estado de los botones, lo que optimizaría el uso del microcontrolador.

Implicaciones del resultado

Los resultados obtenidos tienen un valor didáctico importante, ya que permiten comprender los principios de la lógica digital, el conteo binario y la interacción entre hardware y software en sistemas programables. Aunque se trata de un proyecto básico, constituye una base sólida para escalar hacia aplicaciones más complejas, como temporizadores, contadores de eventos o interfaces hombre-máquina. Además, este ejercicio introduce a los estudiantes en la importancia de diseñar sistemas confiables considerando problemas prácticos como el rebote de los botones.

6. Conclusiones

Síntesis de los resultados más relevantes

La práctica permitió implementar con éxito un **contador binario de 4 bits** utilizando un microcontrolador Arduino, LEDs y botones pulsadores. Se comprobó que el sistema responde de forma correcta a las pulsaciones, mostrando en los LEDs los números binarios del 0 al 15 de manera ascendente o descendente según el botón presionado.

Cumplimiento de los objetivos

Se cumplieron los objetivos planteados: se configuraron y programaron entradas y salidas digitales, se comprendió el funcionamiento de componentes básicos (LEDs y botones) y se aplicaron conceptos de programación en el entorno Arduino. Esto permitió afianzar las bases para el diseño de sistemas programables más complejos.

Implicaciones del experimento

Los resultados obtenidos refuerzan el aprendizaje práctico de la lógica digital y su implementación en sistemas programables. Además, evidencian la relevancia de Arduino como plataforma educativa para introducir a los estudiantes en conceptos de control, automatización y representación binaria, los cuales son fundamentales en el campo de la electrónica y la computación.

Limitaciones y mejoras futuras

El sistema estuvo limitado por el **rebote de los botones** y el rango de conteo restringido a 4 bits. Para mejorar, se recomienda implementar técnicas más robustas de antirrebote, utilizar interrupciones en lugar de lecturas continuas y ampliar el rango del contador con más bits o con salidas a displays. Estas mejoras abrirían la posibilidad de aplicaciones más complejas y cercanas a sistemas de uso industrial.