Trabajo PDIH
Codigo Morse usando Arduino

UNIVERSIDAD DE GRANADA

Curso: 2024/2025

William Patrick Quinn Cortes

Laura Riesco Martin

ÍNDICE

1. Introducción

En este proyecto, nuestro objetivo era investigar la aplicación del código Morse como vía de comunicación mediante un sistema electrónico utilizando *Arduino*. El código Morse, creado en el siglo XIX por **Samuel Morse** y **Alfred Vail**, se utilizó extensamente en telegrafía y, a pesar de que se ha suplantado por tecnologías más actuales, sigue siendo usado en circunstancias particulares como urgencias, navegación por mar y accesibilidad para individuos con limitaciones visuales o auditivas.

Nuestra meta es mostrar cómo se puede aplicar de manera práctica el código Morse a través de la programación y el manejo del hardware. Empleamos un LED vinculado a una placa *Arduino UNO* para ilustrar los caracteres del alfabeto en formato Morse, provocando que el LED parpadee con puntos y rayas. Adicionalmente, facilitamos que el usuario envíe un mensaje a través del monitor serie de Arduino, que posteriormente se traduce y se visualiza en tiempo real a través de luces brillantes.

Este proyecto nos brindó la oportunidad de utilizar no solo conceptos de programación y electrónica, sino también de vincular la teoría aprendida en clase con una aplicación palpable y operativa.

2. Marco Teórico

2.1 Código Morse

El código Morse es un sistema de codificación que emplea mezclas de señales breves (puntos) y extensas (rayas) para ilustrar letras, números y signos de puntuación. En nuestro trabajo, empleamos un punto para un breve resplandor de luz *(200 ms)* y una raya para un largo resplandor *(600 ms)*. A continuación, mostramos la tabla de equivalencias utilizada en el proyecto para las letras del alfabeto:

Letra	Código Morse
A	.-
B	-...
C	-.-.
D	-..
…	...
Z	--..

```
//Codigo morse para letras A-Z
String abecedario[] = {
  ".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..",
  ".---", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-."
  "...", "-", "..-", "...-", ".--", "-..-", "-.--", "--.."
};
```

2.2 Arduino y Periféricos

Para realizar la representación del código Morse, utilizamos un *Arduino UNO* y un LED en su papel de dispositivo de salida. Este LED se vincula a un pin digital (el pin 10 en nuestra situación) y cambia de color dependiendo del carácter que el usuario introduce. El sistema analiza la información del monitor serie, traduce el texto y gestiona la condición del LED con las temporizaciones adecuadas.

Para el desarrollo de este esquema se han tenido en cuenta la información correspondiente a los contenidos de la asignatura:

- **Tema 5 (Dispositivos de salida):** Hemos empleado un LED como canal de salida visual, empleando tiempos exactos para ilustrar las señales Morse.
- **Tema 3 (Interfaces hardware):** Implementamos un enlace físico entre el LED, una resistencia y el pin del Arduino, además de activar la interfaz serial para la introducción de datos.
- **Tema 2 (Gestión de periféricos a bajo nivel):** A través de la programación directa, administramos el estado del pin digital, decidiendo cuándo encender o apagar el LED de acuerdo al carácter que se ha interpretado.

3. Materiales y Circuito

Para realizar el montaje del sistema de transmisión en código Morse utilizamos los siguientes componentes:

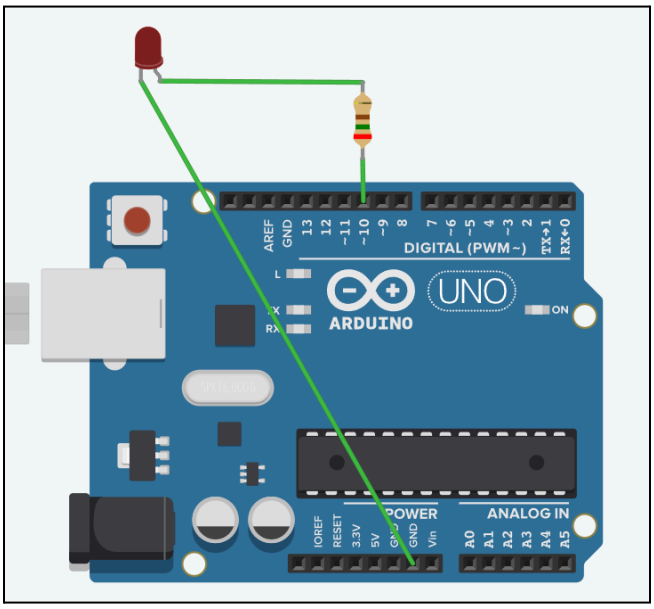
- Placa Arduino UNO
- Un LED rojo
- Una resistencia de 220 Ω
- Cables de conexión

Se llevó a cabo la conexión de la siguiente manera:

1. Unimos el ánodo del LED (con una pierna larga) a una resistencia de 220 Ω, y el otro extremo de la resistencia al pin digital 10 del Arduino.
2. El cátodo del LED de corta longitud se puso en contacto con GND (tierra).

Este diseño garantiza el correcto encendido del LED y previene que se deteriore debido a un exceso de corriente. La resistencia desempeña un papel crucial. Esta restringe la corriente que fluye por el LED, previniendo de esta manera su deterioro.

A continuación, mostramos el esquema del circuito desarrollado:



4. Implementación del Código

El programa elaborado incluye diversas partes fundamentales. Primero, establecemos los periodos que simbolizan cada símbolo Morse:

- **Puntotiempo** = 200 ms
- **Rayatiempo** = 600 ms
- **Pausas entre símbolos** = 300 ms
- **Entre letras** = 800 ms
- **Entre palabras** = 1600 ms

Empleamos un array conocido como ***abecedario[]***, el cual alberga las secuencias Morse relacionadas con cada letra del alfabeto, desde la A hasta la Z. Este array facilita que cada carácter introducido desde el monitor serial se traduzca directamente a su representación en puntos y rayas.

```
  // Validamos que sea una letra válida
  if (c < 'A' || c > 'Z') return;

  int indice = c - 'A'; // Restamos la primera letra p

  // Obtenemos que letra es
  String letra = abecedario[indice];

  for (int i=0; i<letra.length(); i++) {
    if (letra[i] == '.') {
      digitalWrite(ledPin, HIGH);
      delay(puntotiempo);
    }
    else if (letra[i] == '-') {
      digitalWrite(ledPin, HIGH);
      delay(rayatiempo);
    }
    digitalWrite(ledPin, LOW);
    delay(300);  // Pausa entre cada elemento ( es dec
  }
  delay(800);  // Pausa entre cada letra
}
```

El sistema tiene una función principal llamada ***letratomorse(char letra)***,que recibe un carácter, lo transforma Para que opere de manera adecuada con entradas en minúscula, utilizamos *toupper()* para transformar todo a mayúsculas.

```
// Función para convertir una letra en código Morse
void letratomorse(char c) {

  //Pausa larga entre cada palabra
  if (c == ' ') {
    delay(1600);
    return;
  }

   c = toupper(c); // Pasamos a mayuscula toda la inf
```

La función *setup()* establece la comunicación en serie a 9600 baudios y establece el pin digital 10 como entrada. En el *loop()* , el sistema aguarda un mensaje del usuario del monitor serie, lo examina de manera consecutiva y lo traduce mediante *letratomorse()*.

```
void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {
    String mensaje = Serial.readString();
    for (int i = 0; i < mensaje.length(); i++) {
      letratomorse(mensaje[i]);
    }
  }
}
```

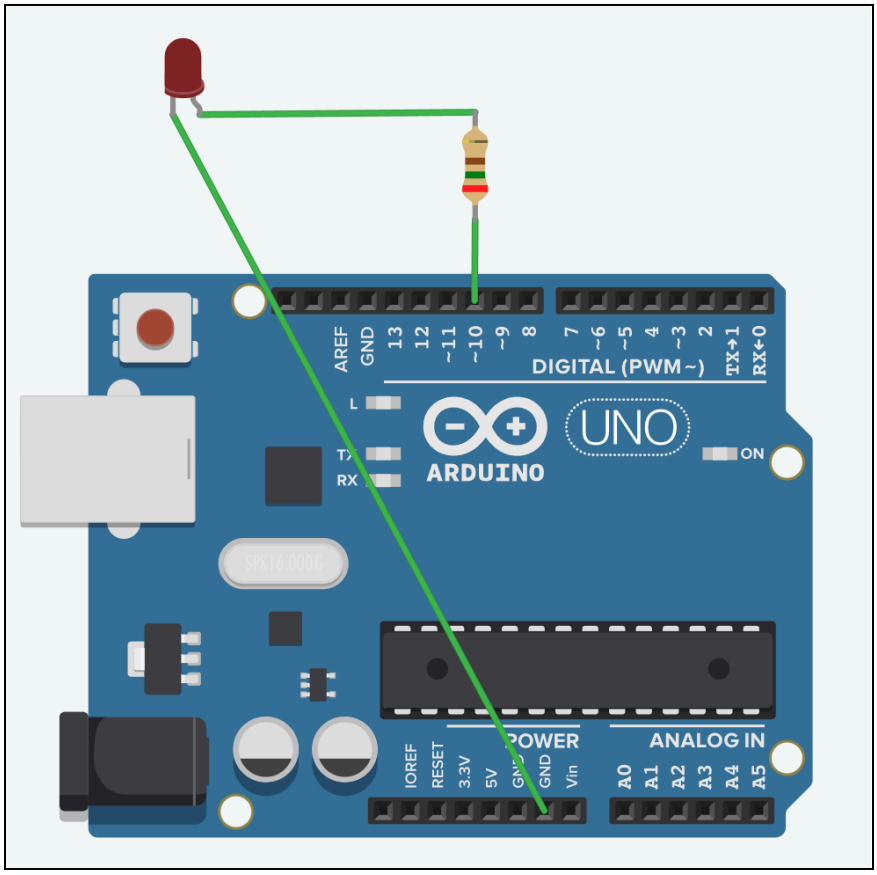
Además, establecimos condiciones para eliminar los caracteres inválidos y hacer las pausas requeridas dependiendo de si se refiere a un espacio entre símbolos, letras o frases.

5. Resultados y Demostración

Cuando evaluamos el sistema, emitimos diversos mensajes como "SOS" para confirmar que el LED se prende con las secuencias anticipadas. Para "SOS", el LED emite tres luces breves (· · ·), tres largas (— — —), y nuevamente tres breves (· · ·), respetando adecuadamente los tiempos de cada símbolo.

Este comportamiento evidencia que la traducción y gestión del hardware se lleva a cabo de manera adecuada, corroborando de esta manera el rendimiento del proyecto.

Como demostración se muestra un enlace a un video con la prueba para mensaje SOS:



Se deja como añadido el enlace al programa desarrollado en TinkerCad.

Enlace: *https://www.tinkercad.com/things/f6sXNR1H5e3-trabajopdih?sharecode=7Bu9LmwR2emzHfnc98HpwCPxQi9hABQ0ieeXOGnkOi4*

6. Dificultades y Soluciones

Durante el desarrollo nos enfrentamos a algunos desafíos:

- Temporización incorrecta: Inicialmente, los valores de delay() no generaban una interpretación precisa del parpadeo. Estimamos los periodos hasta que el patrón se hizo perceptible y comprensible a primera vista.
- Caracteres no válidos: El software presentaba fallos al introducir símbolos o cifras. Esto se soluciona filtrando los caracteres bajo una condición que solo admite letras del alfabeto *(if (letra >= 'A' && letra <= 'Z'))*.

Como posibles mejoras, proponemos:

- Incorporar un buzzer que produce sonido en conjunto con el LED para incrementar la realidad.

7. Conclusiones

Este proyecto nos brindó la oportunidad de poner en práctica varias nociones aprendidas en clase, tales como la administración de periféricos, la utilización de interfaces de hardware y la programación a nivel básico. Al crear un sistema que pueda convertir texto a código Morse a través del control de un LED, potenciamos nuestras competencias en electrónica y programación.

Adicionalmente, entendimos el valor de la correcta temporización de los dispositivos de salida y aprendimos a interactuar con el ambiente físico a través del ambiente de desarrollo de Arduino.

Bibliografía

Documentación de Arduino: https://docs.arduino.cc/

Apuntes de la asignatura: Temas 2, 3 y 5 proporcionados.

Estándar del código Morse: https://morsecw.com/alfabeto.html