



UNIVERSIDAD DE GRANADA

Trabajo PDIH: Código Morse en Arduino

Curso 2024-2025

Francisco Quiles Ramírez

Índice:

¿Qué es el código Morse?.....	3
¿Por qué es interesante hacerlo con Arduino?.....	3
Proyecto.....	3
Materiales.....	4
Diseño.....	4
Código.....	5
Resultados.....	11
Referencias.....	12

¿Qué es el código Morse?

El código morse es un sistema de codificación de caracteres alfabéticos y numéricos que utiliza secuencias de señales, generalmente en forma de pulsos eléctricos, sonidos o luces, para transmitir mensajes a largas distancias. Fue desarrollado por Samuel Morse y Alfred Vail en 1830 y desempeñó un papel crucial en la comunicaciones durante más de un siglo.

Consiste en asignar a cada letra y número una secuencia única de puntos cortos y rayas largas. Siendo la duración de la raya tres veces la del punto y entre dos símbolos de la misma letra habrá una pausa similar a la duración del punto, entre dos letras de una palabra, la duración será como de tres puntos y para separar palabras el intervalo es de tres veces la duración de una raya.

El código Morse fue desarrollado para usar el telégrafo, un dispositivo ideado también por Samuel Morse en 1832. Emplea un dispositivo llamado "transmisor" que convierte los mensajes en señales eléctricas, y un "receptor" que recupera las señales y las traduce nuevamente en mensajes legibles. El primer mensaje se envió el 24 de mayo de 1844, cuando el propio Morse transmitió "Lo que Dios ha creado" ("What hath God wrought", una cita bíblica, Números 23:23) desde la Corte Suprema de los Estados Unidos en Washington, D.C. a su asistente, Alfred Vail, en Baltimore, Maryland. Rápidamente, esta tecnología comenzó a utilizarse en negocios, también revolucionó el periodismo e incluso tuvo un considerable impacto en las relaciones sentimentales.

¿Por qué es interesante hacerlo con Arduino?

Utilizar el código Morse con Arduino es una forma didáctica y atractiva de unir historia, tecnología y programación. Este proyecto permite comprender cómo se codifica y transmite información mediante señales simples, algo esencial en la electrónica y las telecomunicaciones. Con Arduino, es posible recrear el funcionamiento del telégrafo usando LEDs, botones o sonidos, lo que facilita el aprendizaje práctico de conceptos como la temporización, la gestión de entradas/salidas digitales y la interpretación de eventos. Además, Arduino ofrece una plataforma flexible, accesible y ampliamente documentada, lo que permite a estudiantes y aficionados experimentar de forma sencilla y económica con sistemas reales de comunicación digital.

Proyecto

El montaje realizado con Arduino permite interactuar con el código Morse de forma bidireccional, ofreciendo dos modos de funcionamiento que enriquecen la experiencia didáctica. En el primer modo, el usuario puede escribir letras pulsando un botón: una pulsación corta se interpreta como punto y una larga como raya. El sistema mide la duración de cada pulsación, enciende un LED como retroalimentación visual y muestra en el monitor

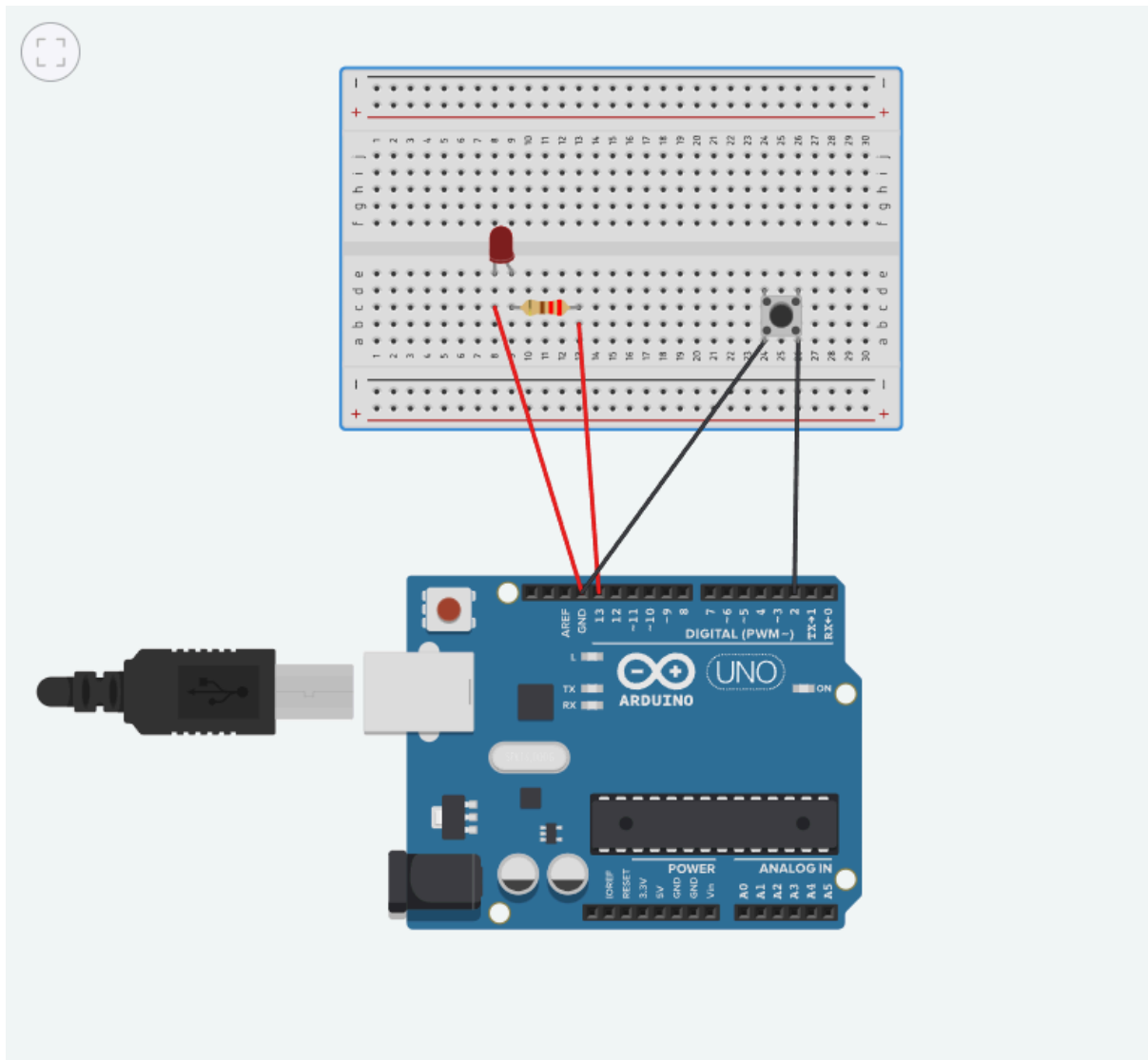
serie la secuencia introducida, interpretando letras, palabras y frases completas. En el segundo modo, el usuario escribe una palabra desde el teclado del ordenador a través del monitor serie, y Arduino traduce automáticamente cada letra a código Morse. A continuación, reproduce la palabra mediante el parpadeo del LED, diferenciando claramente los puntos y rayas con distintas duraciones, así como las pausas entre letras y palabras. Este diseño no solo simula el funcionamiento real del código Morse, sino que también permite comprender cómo Arduino puede recibir, procesar y emitir información codificada a través de entradas físicas y comandos digitales, todo en un circuito compacto y fácil de montar

Materiales

- 1 Arduino UNO
- 1 Pulsador
- 1 LED
- 1 resistencia de 220Ω
- Cables
- 1 placa

Diseño

El diseño del circuito se ha basado en la sencillez y funcionalidad, utilizando componentes básicos para implementar un sistema de detección de pulsaciones en Arduino. Se ha empleado un pulsador conectado al pin digital 2 del Arduino, configurado con la resistencia interna de pull-up activada (INPUT_PULLUP), lo que permite prescindir de resistencias externas y simplificar el montaje. Esto significa que, en reposo, el pin está en estado alto (HIGH) y al presionar el botón, se conecta directamente a tierra (GND), generando un estado bajo (LOW) que el programa detecta como pulsación. Además, se ha conectado un LED al pin digital 13 (que ya tiene una resistencia integrada en muchas placas Arduino) o alternativamente a otro pin con una resistencia externa de 220Ω para limitar la corriente y proteger el componente. El LED se enciende cada vez que el botón es pulsado, proporcionando una retroalimentación visual clara. Gracias al uso de la función `millis()`, el programa mide con precisión cuánto tiempo se mantiene pulsado el botón y, según esa duración, clasifica la señal como punto o raya. Esta información se muestra en el monitor serie, lo que permite al usuario comprobar la correcta interpretación del código Morse. Todo el montaje se ha realizado sobre una protoboard, lo que facilita la conexión de los componentes sin necesidad de soldaduras y permite realizar ajustes o mejoras de manera ágil. Además, el sistema incluye un segundo modo en el que el usuario puede introducir palabras directamente desde el teclado del ordenador, las cuales se traducen automáticamente a código Morse y se reproducen visualmente mediante el parpadeo del LED, ampliando así las posibilidades interactivas y educativas del proyecto.



Código

El código está estructurado para ofrecer dos modos de funcionamiento distintos, que el usuario puede seleccionar desde el monitor serie al iniciar el programa. Una vez encendido, Arduino muestra un menú donde pregunta si se desea escribir con el botón (modo 1) o introducir una palabra desde el teclado del ordenador (modo 2). En el modo 1, el sistema detecta las pulsaciones del botón mediante la función `digitalRead()` y calcula su duración utilizando `millis()`, que devuelve el tiempo actual en milisegundos desde que Arduino se inició. Cuando se detecta una pulsación, se almacena el momento en que comenzó; al soltar el botón, se calcula la duración de esa pulsación. Si fue breve (menos de 200 ms), se interpreta como un punto (.), y si fue más larga (entre 300 y 1000 ms), se interpreta como una raya (-). Cada punto o raya se añade a una cadena llamada `morseBuffer`. Cuando pasan 2 segundos sin nuevas pulsaciones, el programa asume que se ha completado una letra y llama a la función `morseToChar()` para traducir la secuencia almacenada a su letra correspondiente. Estas letras se acumulan formando palabras, y si pasan 5 segundos sin actividad, se considera que la palabra ha terminado. En ese momento, Arduino solicita

confirmación al usuario: si pulsa de nuevo brevemente, continúa escribiendo otra palabra; si la pulsación es larga, finaliza la frase y la muestra completa en el monitor serie. Todo este flujo se controla mediante estructuras if que evalúan tanto el tiempo transcurrido como el estado actual del botón, utilizando variables booleanas y de texto que indican si se está introduciendo una letra, una palabra o esperando confirmación.

En el modo 2, el usuario introduce directamente una palabra a través del teclado del ordenador, y Arduino la procesa carácter por carácter. Cada letra se convierte en su equivalente en código Morse mediante la función `charToMorse()` y se reproduce visualmente encendiendo un LED: pulsos breves para puntos y pulsos largos para rayas, con pausas adecuadas entre cada símbolo y letra para que el mensaje sea claramente legible. Así, el sistema combina entrada digital y entrada por teclado, mostrando cómo Arduino puede recibir, interpretar y emitir información en código Morse tanto desde un botón físico como desde el ordenador, integrando entradas y salidas de manera precisa y pedagógica.

Siendo el código el siguiente:

```
#define PINBT 2
#define PINLED 13

#define TSHORT 200
#define TLONG 300
#define TLONGE 1000

#define LETTER_TIMEOUT 2000
#define WORD_TIMEOUT 5000

bool btPressed = false;
unsigned long lastPressed = 0;
unsigned long lastInputTime = 0;

String morseBuffer = "";
String palabraActual = "";
String fraseCompleta = "";

bool esperandoConfirmacion = false;
bool modoSeleccionado = false;
int modo = 0; // 1 = botón, 2 = texto

void setup() {
  pinMode(PINBT, INPUT_PULLUP);
  pinMode(PINLED, OUTPUT);
  Serial.begin(9600);
  delay(1000);
  mostrarMenu();
}
```

```

void loop() {
  if (!modoSeleccionado) {
    if (Serial.available()) {
      String entrada = Serial.readStringUntil('\n');
      entrada.trim();
      if (entrada == "1") {
        modo = 1;
        modoSeleccionado = true;
        Serial.println("Modo 1: escribir con el boton.");
      } else if (entrada == "2") {
        modo = 2;
        modoSeleccionado = true;
        Serial.println("Modo 2: escribe una palabra y la mostrare en Morse con el LED.");
        Serial.println("Escribe la palabra:");
      } else {
        Serial.println("Opción no valida. Escribe 1 o 2.");
      }
    }
  }
  return;
}

```

// MODO 1: escribir pulsando

```

if (modo == 1) {
  modoBoton();
}

```

// MODO 2: escribir palabra por teclado

```

if (modo == 2) {
  if (Serial.available()) {
    String palabra = Serial.readStringUntil('\n');
    palabra.trim();
    Serial.print("Mostrando en Morse: ");
    Serial.println(palabra);
    reproducirMorse(palabra);
    Serial.println("Finalizado.");
    mostrarMenu(); // volver al menú
    modoSeleccionado = false;
  }
}
}

```

// Función que gestiona el MODO BOTÓN

```

void modoBoton() {
  unsigned long now = millis();
  btPressed = !digitalRead(PINBT);

  if (btPressed && lastPressed == 0) {
    lastPressed = now;
  }
}

```

```

    digitalWrite(PINLED, HIGH);
}

if (!btPressed && lastPressed != 0) {
    unsigned long duration = now - lastPressed;
    digitalWrite(PINLED, LOW);
    lastPressed = 0;

    if (esperandoConfirmacion) {
        if (duration <= TSHORT) {
            Serial.println("CONTINUAR");
            esperandoConfirmacion = false;
        } else if (duration >= TLONG && duration <= TLONGE) {
            Serial.println("TERMINAR");
            Serial.print("Frase completa: ");
            Serial.println(fraseCompleta);
            Serial.println("-----");
            fraseCompleta = "";
            esperandoConfirmacion = false;
        }
        return;
    }

    if (duration <= TSHORT) {
        morseBuffer += ".";
        Serial.print(".");
    } else if (duration >= TLONG && duration <= TLONGE) {
        morseBuffer += "-";
        Serial.print("-");
    } else {
        Serial.print("(Ignorado)");
    }

    lastInputTime = now;
}

if (morseBuffer.length() > 0 && now - lastInputTime > LETTER_TIMEOUT) {
    char letra = morseToChar(morseBuffer);
    Serial.print(" Letra: ");
    Serial.println(letra);
    palabraActual += letra;
    morseBuffer = "";
}

if (palabraActual.length() > 0 && now - lastInputTime > WORD_TIMEOUT) {
    Serial.print("Palabra completa: ");
    Serial.println(palabraActual);
    fraseCompleta += palabraActual + " ";
}

```



```

    palabraActual = "";
    esperandoConfirmacion = true;
    Serial.println("¿Seguir escribiendo?");
    Serial.println("Pulsa punto para CONTINUAR, raya para TERMINAR.");
}
}

```

// Función que reproduce palabra en Morse con el LED

```

void reproducirMorse(String palabra) {
    palabra.toUpperCase();
    for (int i = 0; i < palabra.length(); i++) {
        char letra = palabra[i];
        if (letra == ' ') {
            delay(1500);
            continue;
        }

```

```

        String morse = charToMorse(letra);
        for (int j = 0; j < morse.length(); j++) {
            if (morse[j] == '.') {
                digitalWrite(PINLED, HIGH);
                delay(400);
                digitalWrite(PINLED, LOW);
            } else if (morse[j] == '-') {
                digitalWrite(PINLED, HIGH);
                delay(1000);
                digitalWrite(PINLED, LOW);
            }
            delay(400);
        }
        delay(800);
    }
}

```

// Mostrar menú al inicio

```

void mostrarMenu() {
    Serial.println("Selecciona modo:");
    Serial.println("1 - Escribir con el boton");
    Serial.println("2 - Escribir palabra desde teclado");
    Serial.print("Escribe tu eleccion: ");
}

```

// Traducción de código morse a letra

```

char morseToChar(String code) {
    if (code == ".-") return 'A'; if (code == "-...") return 'B';
    if (code == "-.-.") return 'C'; if (code == "-..") return 'D';
    if (code == ".") return 'E'; if (code == "..-") return 'F';
    if (code == "--.") return 'G'; if (code == "....") return 'H';

```

```

if (code == ".") return 'I'; if (code == ".---") return 'J';
if (code == "-.-") return 'K'; if (code == "-..") return 'L';
if (code == "--") return 'M'; if (code == "-.") return 'N';
if (code == "---") return 'O'; if (code == "---") return 'P';
if (code == "--.") return 'Q'; if (code == "-.") return 'R';
if (code == "...") return 'S'; if (code == "-") return 'T';
if (code == "-.-") return 'U'; if (code == "...") return 'V';
if (code == "-.-") return 'W'; if (code == "-..") return 'X';
if (code == "-.-") return 'Y'; if (code == "-..") return 'Z';
return '?';
}

```

// Traducción de letra a morse

```

String charToMorse(char c) {
    switch (c) {
        case 'A': return ".-"; case 'B': return "-...";
        case 'C': return "-.-."; case 'D': return "-..";
        case 'E': return "."; case 'F': return ".-.";
        case 'G': return "--."; case 'H': return "....";
        case 'I': return ".."; case 'J': return ".---";
        case 'K': return "-.-"; case 'L': return "-.-.";
        case 'M': return "--"; case 'N': return "-.";
        case 'O': return "---"; case 'P': return "-.-.";
        case 'Q': return "--.-"; case 'R': return "-.-";
        case 'S': return "..."; case 'T': return "-";
        case 'U': return "-.-"; case 'V': return "...-";
        case 'W': return "-.-"; case 'X': return "-.-.";
        case 'Y': return "-.-"; case 'Z': return "--..";
        default: return "";
    }
}

```

Resultados

Monitor en serie

```
Selecciona modo:
1 - Escribir con el boton
2 - Escribir palabra desde teclado
Escribe tu eleccion: Modo 1: escribir con el boton.
.... Letra: H
--- Letra: O
... Letra: L
.- Letra: A
Palabra completa: HOLA
¿Seguir escribiendo?
Pulsa punto para CONTINUAR, raya para TERMINAR.
CONTINUAR
-- Letra: M
.(Ignorado).- Letra: U
-. Letra: N
-.. Letra: D
--- Letra: O
Palabra completa: MUNDO
¿Seguir escribiendo?
Pulsa punto para CONTINUAR, raya para TERMINAR.
TERMINAR
Frase completa: HOLA MUNDO
-----
```



Opción 1: Escribir con el botón

https://drive.google.com/file/d/1vCIBa4JxI8vElkV5V-jo3kNpniLKKBWw/view?usp=drive_link

Opción 2: Escribir desde teclado

https://drive.google.com/file/d/1hw_aY1cSC_ZFIHHRbmpsztYP_2LR2aoo/view?usp=drive_link

Referencias

- https://www.nationalgeographic.com.es/ciencia/codigo-morse-que-es-como-funciona-que-tiene-que-ver-titanic_19830
- <https://www.hackster.io/ppenguin/simple-morse-code-transceiver-5732f1>
- <https://www.youtube.com/watch?v=hTC7e5rX1tk>
- <https://app.kumumarket.com/producto/6365>
- <https://blogs.etsii.urjc.es/dseytr/traductor-bidireccional-espanol-morse/>