



ugr | Universidad
de **Granada**

PDIH

PERIFÉRICOS Y DISPOSITIVOS DE INTERFAZ HUMANA.

Traductor de Mensajes a Código Morse con Arduino.

Autor: Miguel Molinero Martín



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

—
Curso 2023 - 2024

Memoria del Proyecto final: Traductor de Mensajes a Código Morse con Arduino.

Alumno: Miguel Molinero Martín

1. Introducción

La importancia de la comunicación ha sido fundamental a lo largo de la historia de la humanidad, y en la época digital actual, esta necesidad es aún más clara. Desde el inicio de los sistemas de telegrafía hasta la llegada de la Internet, ha habido una evolución continua en los métodos de comunicación que ha permitido nuevas formas de interacción entre las personas.

El proyecto actual busca investigar el uso del código Morse, una forma clásica y efectiva de comunicación en este contexto. Creado por Samuel Morse y Alfred Vail en los años 1830, el código Morse fue empleado extensamente en las telecomunicaciones por más de cien años; aunque ha sido mayormente sustituido por tecnologías más modernas, sigue teniendo relevancia y reconocimiento en la actualidad.

El principal propósito de esta iniciativa es combinar la práctica del código Morse con la tecnología actual, utilizando las ventajas y facilidades de las placas Arduino y componentes electrónicos básicos. Este sistema interactivo que convierte mensajes a código Morse se puede crear con una placa Arduino UNO, una pantalla LCD, un buzzer y un LED.

Al incluir estos componentes, el proyecto pretende ofrecer educación sobre el código Morse y su uso en la era digital, al mismo tiempo que muestra la flexibilidad y la capacidad de innovación de la tecnología de hardware de código abierto como Arduino.

A continuación, se detallará el proceso de diseño hardware y software y evaluación del sistema.

A	--	J	-----	S	---	2	-----
B	----	K	---	T	-	3	-----
C	----	L	----	U	---	4	-----
D	---	M	--	V	----	5	-----
E	.	N	--	W	---	6	-----
F	----	O	---	X	----	7	-----
G	---	P	----	Y	----	8	-----
H	----	Q	----	Z	----	9	-----
I	..	R	---	1	-----	0	-----

2. Objetivos

El principal propósito del proyecto es crear un sistema interactivo que le permita a los usuarios introducir mensajes a través del teclado y convertirlos de manera automática al código Morse.

Se planea emplear componentes electrónicos sencillos y económicos, como una pantalla LCD, un buzzer y un LED, para desarrollar el sistema. Esto no solo hace que el proyecto sea más accesible, sino que también muestra la versatilidad de estos componentes en aplicaciones prácticas y educativas.

Suministrar retroalimentación tanto visual como auditiva: Durante la traducción de mensajes a código Morse, el sistema necesita ofrecer retroalimentación tanto visual como auditiva. Esto se consigue a través de la intermitencia del LED y los sonidos del buzzer, que simbolizan los puntos y las rayas del código Morse, respectivamente. Este feedback tiene como objetivo mejorar la comprensión y la experiencia del usuario al usar el sistema.

Desarrollar una interfaz de usuario de fácil uso: El objetivo es garantizar una experiencia sencilla y clara al introducir mensajes y ver el código Morse convertido. Esto se consigue utilizando una pantalla LCD para exhibir tanto el mensaje introducido como su correspondiente traducción a código Morse, lo que brinda una retroalimentación clara y al instante al usuario.

3. Diseño hardware

El sistema se pone en marcha en una placa Arduino UNO y emplea los elementos que se detallan a continuación:

1. Pantalla LCD para mostrar el mensaje introducido y su traducción a código Morse.
2. Buzzer para emitir los sonidos correspondientes al código Morse.
3. LED para representar visualmente los puntos y rayas del código Morse.

Representación en Tinkercad:

<https://www.tinkercad.com/things/aE4VVnsIFAp-magnificent-bombul-blorr/edit#sharecode=6Rj2OPza0UmDLIRciMK1bc8rvZhniAvMkkEPFIwdGis>

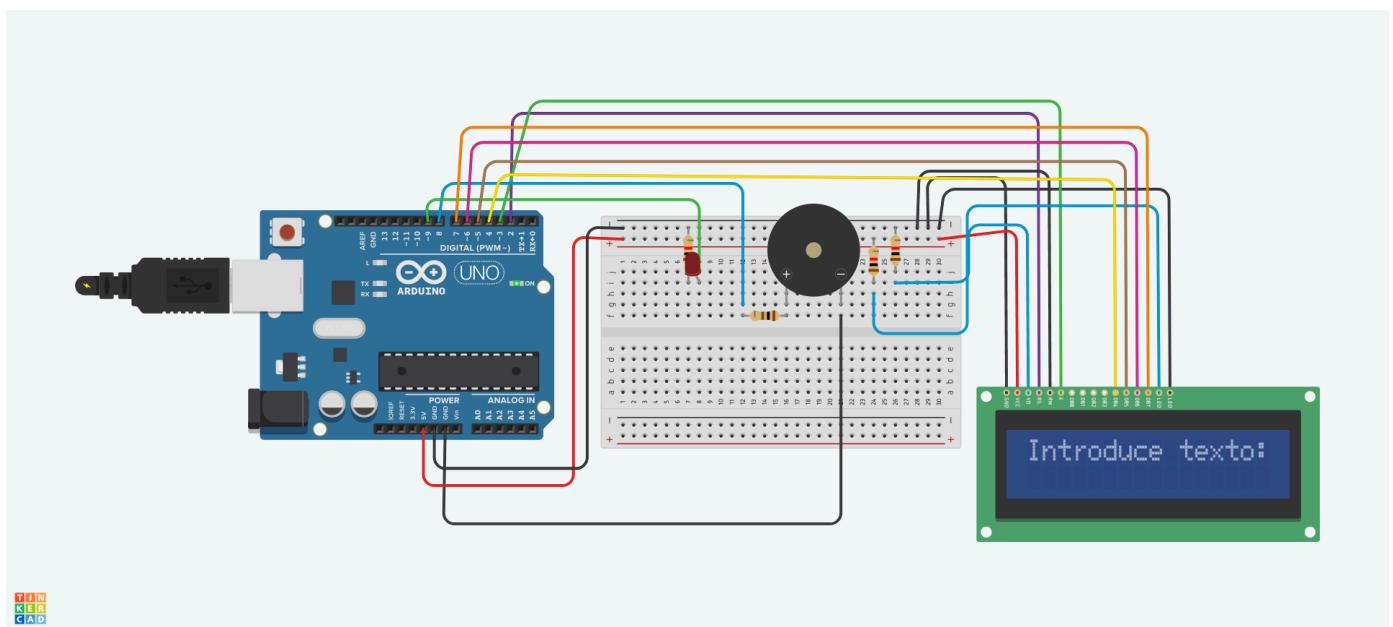


Imagen 1: Representación del diseño en Tinkercad

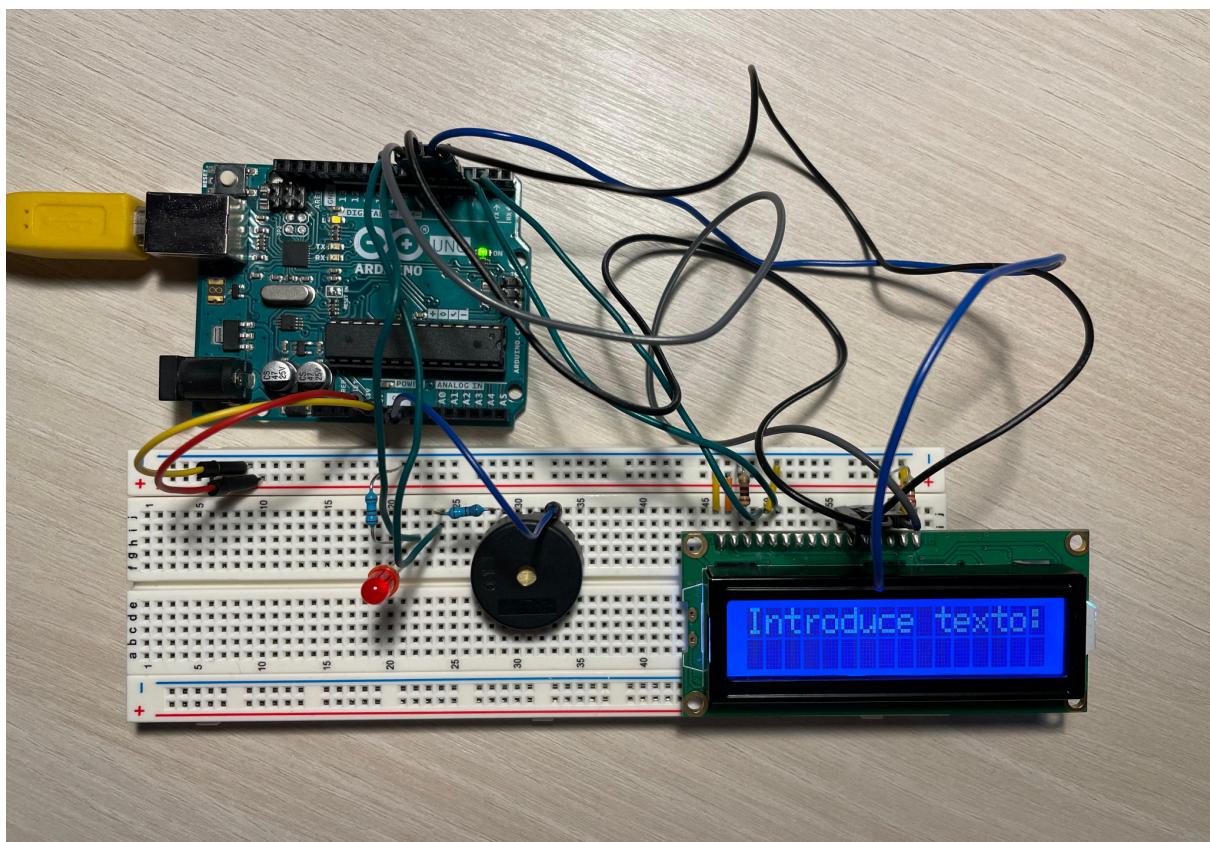


Imagen 2: Diseño real del sistema

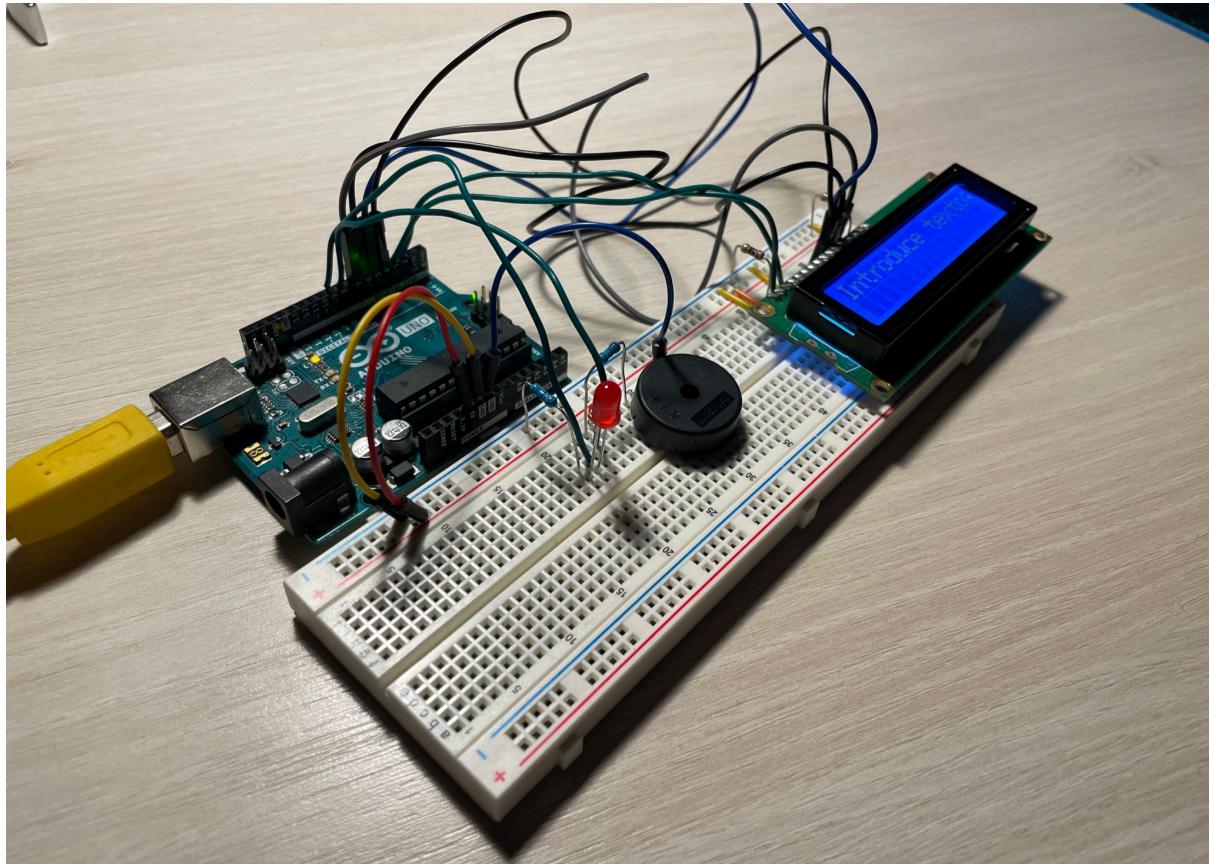


Imagen 3: Diseño real del sistema

4. Código del sistema

Explicación detallada del código Arduino que traduce un mensaje ingresado por el usuario a código Morse y muestra tanto el mensaje original como su equivalente en Morse en una pantalla LCD. Además, emite el código Morse a través de un buzzer y un LED.

LIBRERÍAS

LiquidCrystal.h: Esta librería facilita el control de pantallas LCD basadas en el controlador Hitachi HD44780 o equivalentes. Es utilizada para manejar la pantalla LCD en el pin 2 al 7 de Arduino.

```
#include <LiquidCrystal.h>
```

Definimos los pines del LCD, el buzzer y el LED.

```
LiquidCrystal lcd(2, 3, 4, 5, 6, 7); // inicializamo el LCD

#define buzzerPin 8
#define ledPin 9
```

FUNCIONES

void setup(): Inicia la comunicación serial a una velocidad de 9600 baudios. Inicializamos la pantalla LCD para 16 columnas y 2 filas utilizando la función begin(). Por último, configuramos los pines buzzerPin y ledPin como salidas digitales mediante pinMode().

```
void setup() {
    Serial.begin(9600);
    lcd.begin(16, 2);

    pinMode(buzzerPin, OUTPUT);
    pinMode(ledPin, OUTPUT);
}
```

void punto():

Función de codificación de código morse.

Crea un sonido breve (punto Morse) con el buzzer usando la función tone(). Activamos la luz del LED usando digitalWrite(), espera durante 300 milisegundos empleando la función delay(). Por último, apagamos tanto el buzzer como el LED. Espera otros 300 milisegundos.

```
void punto() {
    tone(8, 1047, 300); // activamos el buffer 300 ms a un tono de 1047Hz
    digitalWrite(ledPin, HIGH); // Enciendemos el LED
    delay(300); // Duración del punto en Morse
    digitalWrite(buzzerPin, LOW); // Apagamos el buzzer
    digitalWrite(ledPin, LOW); // Apagamos el LED
    delay(300);
}
```

void linea():

Genera un sonido largo (línea Morse) mediante el buzzer. Encendemos el LED. Espera 800 milisegundos. Apagamos el buzzer y el LED y esperamos 300 milisegundos adicionales de separación entre puntos y líneas.

```
void linea() {  
    tone(8, 1047, 800); // activamos el buzzer ahora durante 800ms  
    digitalWrite(ledPin, HIGH); // Enciendemos el LED  
    delay(800); // Duración de la raya en Morse  
    digitalWrite(buzzerPin, LOW); // Apagamos el buzzer  
    digitalWrite(ledPin, LOW); // Apagamos el LED  
    delay(300);  
}
```

void loop():

El bucle `loop()` del programa se encarga de manejar la entrada de texto y la traducción a código Morse, así como la visualización y emisión de dicho código.

En cuanto a la entrada de texto, se lleva a cabo de la siguiente manera: primero, se declara un array de caracteres llamado *mensaje* para almacenar el mensaje ingresado. Luego, se muestra un mensaje en la pantalla LCD solicitando al usuario que introduzca el texto -> “Introduce texto: “. El programa espera hasta que haya datos disponibles en el puerto serial. Cuando se detecta la presencia de datos, se leen los caracteres uno por uno y se almacenan en el array *mensaje*.

Finalmente, se termina el mensaje agregando un carácter nulo al final. En lo que respecta a la visualización y emisión en código Morse, el proceso es el siguiente: primero, se muestra el mensaje ingresado por el usuario en la primera fila de la pantalla LCD.

Luego, cada carácter del mensaje se convierte en su equivalente en código Morse y se muestra en la segunda fila de la pantalla LCD. Además, se utilizan las funciones `punto()` y `línea()` para emitir los sonidos correspondientes al código Morse a través del buzzer y el LED.

```

void loop() {
    char mensaje[100];
    int i = 0;

    lcd.clear();
    lcd.setCursor(0, 0); // nos situamos en la primera fila de la pantalla LCD

    lcd.print("Introduce texto: ");
    while (Serial.available() == 0) {

    }

    while (Serial.available() > 0) {
        mensaje[i++] = Serial.read(); // leemos el mensaje introducido por terminal
        lcd.write(mensaje[i - 1]); // mostramos el texto en la LCD
        delay(100);
    }

    mensaje[i] = '\0'; // añadimos el caracter nulo al final de nuestro mensaje


    // codificación en código morse de nuestro mensaje
    lcd.setCursor(0, 1); // nos situamos en la segunda fila de la pantalla LCD
    lcd.print("Morse: ");
    for (int j = 0; mensaje[j] != '\0'; j++) {
        switch (mensaje[j]) {
            case 'a':
            case 'A':
                lcd.print(".");
                punto();
                linea();
                delay(750);
                break;
            case 'b':
            case 'B':
                lcd.print("-... ");
                linea();
                punto();
                punto();
                punto();
                delay(750);
                break;
            case 'c':
            case 'C':
                lcd.print("-.-. ");
                linea();
                punto();
                linea();
        }
    }
}

```

...

```
case '9':
    lcd.print("----.");
    linea();
    linea();
    linea();
    linea();
    punto();
    delay(750);
    break;
case '0':
    lcd.print("-----");
    linea();
    linea();
    linea();
    linea();
    linea();
    delay(750);
    break;

case ' ':
    lcd.print("  "); // espacio entre palabras
    break;
default:
    lcd.print("? "); // para un carácter desconocido
}
}
}
```

CODIFICACIÓN EN CÓDIGO MORSE:

En la codificación en código Morse, un punto se representa con un sonido corto, mientras que una línea se representa con un sonido largo. Cada letra o número se separa con un espacio, y las palabras se separan con dos espacios. Se incluyen conversiones de todas las letras del alfabeto y los números del 0 al 9 al código Morse. Cualquier carácter no reconocido se mostrará como un signo de interrogación (?) en la pantalla LCD.



5. Demostración del funcionamiento del sistema

Enlace al vídeo demostración del funcionamiento del sistema:

<https://youtu.be/kFIXOvGoeeg>