



PDIH

PERIFÉRICOS Y DISPOSITIVOS DE INTERFAZ HUMANA.

Trabajo Teoría: Código Morse usando Arduino.

Autor: Arturo Alonso Carbonero
Correo: alonsoarturo@correo.ugr.es
Autora: Cristina María Crespo Arco
Correo: cmcrespo@correo.ugr.es
Autor: Andrés Piqueras Brück
Correo: andrespquieras@correo.ugr.es

Profesor: Pedro A. Castillo Valdivieso



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Curso 2021 - 2022

Índice

1. De lenguaje natural a código morse.	2
1.1. Componentes eléctricos.	2
1.2. Código.	3
1.3. Esquema de conexiones eléctricas.	6
1.3.1. Vista de Circuito.	6
1.3.2. Vista esquemática.	6
1.4. Funcionamiento del proyecto: En el simulador Tinkercad.	7
1.5. Funcionamiento del proyecto.	7
2. De morse a lenguaje natural.	8
2.1. Componentes eléctricos.	8
2.2. Código.	9
2.3. Esquema de conexiones eléctricas.	14
2.3.1. Vista de Circuito.	14
2.3.2. Vista esquemática.	14
2.4. Funcionamiento del proyecto: En el simulador Tinkercad.	15
2.5. Funcionamiento del proyecto.	15
3. Posible ampliación del programa.	16
3.1. Mejorar el proyecto para que necesite un sólo botón.	16
3.1.1. Componentes eléctricos.	16
3.1.2. Esquema de conexiones eléctricas.	16
3.2. Unir los dos proyectos en uno.	18
3.2.1. Componentes eléctricos.	18
3.2.2. Esquema de conexiones eléctricas.	18

1. De lenguaje natural a código morse.

El programa proporciona una salida visual a través de un LED que se corresponde con el equivalente en código Morse a la palabra introducida por el usuario. El programa recorre la cadena de caracteres recibida letra por letra y traduce cada una para indicarle posteriormente al LED cuándo iluminarse y cuándo no.

Para cada letra, el LED se iluminará durante 500 milisegundos para los puntos y 3 segundos para las rayas con pausas de medio segundo entre cada uno y una espera de 2 segundos al terminar. Tras mostrar una palabra, para los espacios, la espera será de 7 segundos.

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

Figura 1: Código morse.

1.1. Componentes eléctricos.

- 1 Arduino Uno R3
- 1 Resistencias 220 Ω
- 1 LED Rojo
- 3 Cables macho-macho

1.2. Código.

```
// C++ code
String cadena = ""; // Declarar la variable cadena
int numLed = 13; // Declarar e inicializar el puerto de
// salida donde está conectado el LED

// Función que le indica al LED cuánto tiempo debe estar encendida para
// indicar que es un punto (0,5 segundos).
void Punto(){
    digitalWrite(numLed, HIGH); // Encender LED
    delay(500); // Esperar durante 0,5 segundos
    digitalWrite(numLed, LOW); // Apagar LED
    delay(500); // Esperar durante 0,5 segundos
}

// Función que le indica al LED cuánto tiempo debe estar encendida para
// indicar que es una raya (3 segundos).
void Raya(){
    digitalWrite(numLed, HIGH); // Encender LED
    delay(3000); // Esperar durante 3 segundos
    digitalWrite(numLed, LOW); // Apagar LED
    delay(500); // Esperar durante 0,5 segundos
}

// Función que apaga el LED durante 7 segundos para indicar que se ha
// cambiado a una palabra nueva.
void Espacio(){
    digitalWrite(numLed, LOW); // Apagar LED
    delay(7000); // Esperar durante 7 segundos
}

// Función para mostrar la letra pasada como parámetro de entrada
/* La función consta de un condicional switch donde la variable que se
va a comparar es la letra que se pasa como parámetro de entrada.
Respecto a los bloques de instrucciones del switch, hemos implementado
uno por cada letra del abecedario (cada bloque está constituido por
dos casos posibles que puede cumplir el condicional, la letra en
mayúscula y la letra en minúscula). Otro por cada número del 0 al 9
y, por último, tres bloques de instrucciones dos para los signos de
puntuación '.' y ',' y el otro para el espacio.

Dentro de cada bloque de instrucciones llama a las funciones raya() o
punto() de acuerdo al código morse. O a la función Espacio() cuando
se encuentre con un espacio en la cadena ' ' */
```

```

void MostrarLetra(char letra){

    switch (letra) {
        case 'A': case 'a': Punto(); break;
        case 'B': case 'b': Raya(); break;
        case 'C': case 'c': Punto(); break;
        case 'D': case 'd': Raya(); break;
        case 'E': case 'e': Punto(); break;
        case 'F': case 'f': Raya(); break;
        case 'G': case 'g': Punto(); break;
        case 'H': case 'h': Punto(); break;
        case 'I': case 'i': Punto(); break;
        case 'J': case 'j': Punto(); break;
        case 'K': case 'k': Raya(); break;
        case 'L': case 'l': Punto(); break;
        case 'M': case 'm': Raya(); break;
        case 'N': case 'n': Punto(); break;
        case 'O': case 'o': Raya(); break;
        case 'P': case 'p': Punto(); break;
        case 'Q': case 'q': Raya(); break;
        case 'R': case 'r': Punto(); break;
        case 'S': case 's': Raya(); break;
        case 'T': case 't': Punto(); break;
        case 'U': case 'u': Raya(); break;
        case 'V': case 'v': Punto(); break;
        case 'W': case 'w': Raya(); break;
        case 'X': case 'x': Punto(); break;
        case 'Y': case 'y': Raya(); break;
        case 'Z': case 'z': Punto(); break;
    }
}

```

```

        case '1':           case '4':           case '7':           case '.':
            Punto();         Punto();         Raya();          Punto();
            Raya();          Punto();         Raya();          Raya();
            Raya();          Punto();         Punto();         Punto();
            Raya();          Punto();         Punto();         Raya();
            Raya();          Raya();          Punto();         Punto();
            break;           break;           break;           Raya();
        case '2':           case '5':           case '8':           break;
            Punto();         Punto();         Raya();          case ',':
            Punto();         Punto();         Raya();          Raya();
            Raya();          Punto();         Raya();          Raya();
            Raya();          Punto();         Punto();         Punto();
            Raya();          Punto();         Punto();         Punto();
            break;           break;           break;           Raya();
        case '3':           case '6':           case '9':           Raya();
            Punto();         Raya();          Raya();          break;
            Punto();         Punto();         Raya();          case '':
            Punto();         Punto();         Raya();          Espacio();
            Raya();          Punto();         Raya();          break;
            Raya();          Punto();         Punto();         break;
            break;           break;           break;           break;

}

// put your setup code here, to run once:
void setup() {
    pinMode(numLed, OUTPUT);           // Pin de salida del LED
    cadena = "PDIH, prueba P3";       // Cadena que se quiere traducir
                                       // de lenguaje natural a morse
    int tam = cadena.length();         // Calcular el tamaño de la cadena

    delay(500);
    int i;
    for (i = 0; i < tam; i += 1){      // Recorrer la cadena letra por letra
        MostrarLetra(cadena[i]);       // Llama a la función
                                       // MostrarLetra(char letra)
                                       // pasandole como parámetro
                                       // cada letra de la cadena.
        delay(2000);                  // Se realiza una espera de 2 segundos
                                       // para marcar el fin de cada letra
    }
}

void loop() { }


```

1.3. Esquema de conexiones eléctricas.

1.3.1. Vista de Circuito.

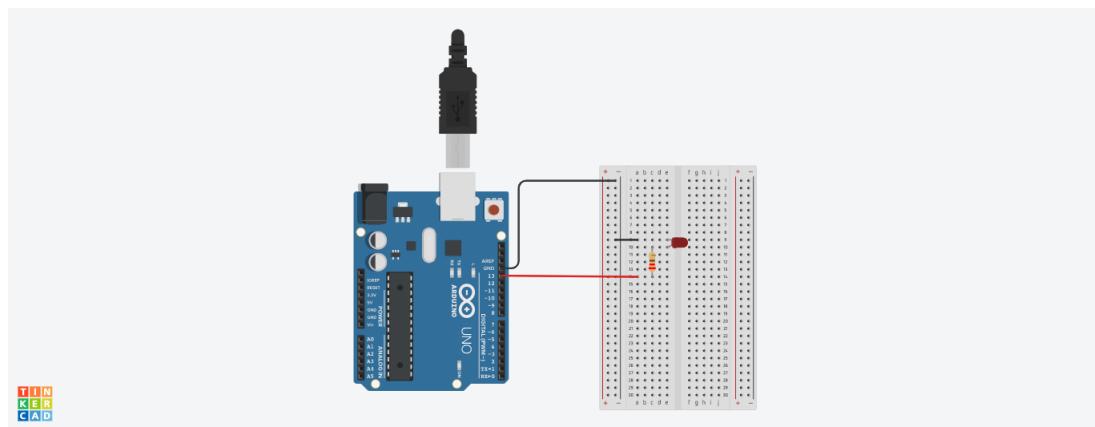


Figura 2: Circuito.

1.3.2. Vista esquemática.

Figura 3: Esquema.

1.4. Funcionamiento del proyecto: En el simulador Tinkercad.

Enlace al proyecto en Tinkercad:

<https://www.tinkercad.com/things/hsqepJnIw14-morse-parte1/editel?sharecode=g-1q6DxwTX6RDucjq-vwovethlQmX6Cz7aAk0bP1jRY>

Enlace al vídeo con una demostración del funcionamiento del proyecto en el simulador:

<https://drive.google.com/file/d/1N2MLooBZP3cgtd-GNj7I1iIaTIiD07Sq/view?usp=sharing>

1.5. Funcionamiento del proyecto.

- Circuito.

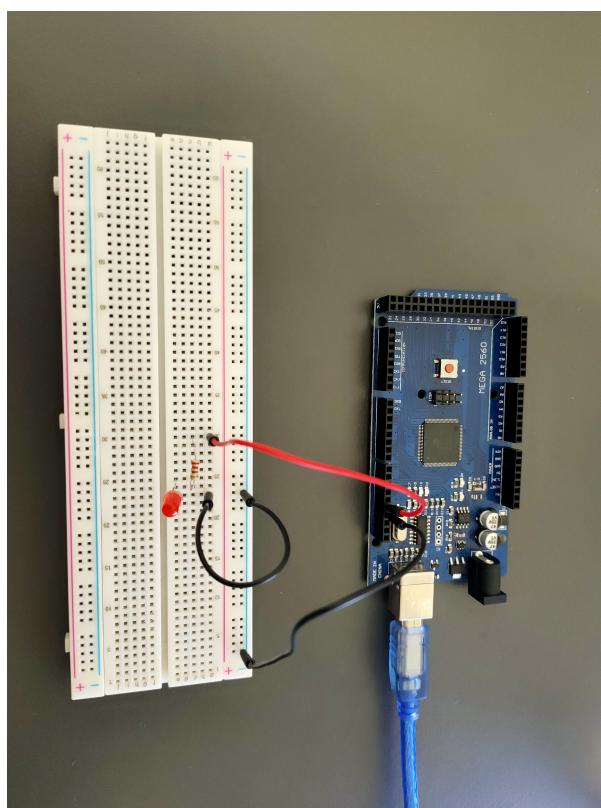


Figura 4: Arduino.

- Enlace al vídeo con una demostración del funcionamiento del proyecto:

https://drive.google.com/file/d/18YfjpTt48uW9ddsY7mlRUB778FinI_C9/view?usp=sharing

<https://drive.google.com/file/d/1LJ6D4mVsi12tLgZUHJAvv4ubYiZlGeA9/view?usp=sharing>

2. De morse a lenguaje natural.

El programa dispone de tres pulsadores que reciben la entrada en código Morse. Un pulsador para los puntos, otro para las rayas y uno último para indicar el final de la letra o símbolo introducido, de forma que obtenemos por pantalla el resultado en lenguaje natural. Entre cada pulsación debe ocurrir una pausa de un segundo. El sistema almacena cada símbolo introducido por el usuario en un array de elementos que, posteriormente, recorre para comparar cada posición con su equivalente en lenguaje natural. La separación entre diferentes palabras se indica mediante un punto o una coma.

Según el tamaño del array de entrada, el resultado pertenecerá a un conjunto de caracteres en lenguaje natural u otro, tal y como se muestra en la siguiente imagen, siendo 1 el tamaño mínimo y 6 el máximo.

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

Figura 5: Código morse.

2.1. Componentes eléctricos.

- 1 Arduino Uno R3
- 3 Resistencias 220 Ω
- 3 Pulsador
- 11 Cables macho-macho

2.2. Código.

```
// C++ code
int numPulsadorPunto = 7; // Declarar e inicializar el puerto de entrada
                          // donde está conectado el pulsador para el punto
int numPulsadorRaya = 6; // Declarar e inicializar el puerto de entrada
                          // donde está conectado el pulsador para la raya
int numPulsador = 5;    // Declarar e inicializar el puerto de entrada
                        // donde está conectado el fin de letra

int const TAM = 10;      // Declarar e inicializar la variable TAM con el
                        // tamaño máximo del vector (10)
int morse_array[TAM];   // Declarar el array donde se van a almacenar la
                        // letra en morse
int tamano;             // Declarar el tamaño que tendrá el array con la
                        // letra

// Función que devuelve la letra correspondiente al código morse que se ha
// introducido como parámetro de entrada
/* Si observamos el código morse podemos clasificar el código según la suma
de puntos y rayas de cada letra en seis grupos ya que es el número máximo
de puntos/rayas que hay. Si el tamaño está entre 1 y 4 se va a tratar de
una letra, si es 5 va a ser un número y por último si es 6 se va a tratar
de un signo de puntuación.
```

Partiendo de esto hemos implementado un condicional if en el cual cada bloque de instrucciones se realiza si se cumple que el tamaño pasado como parámetro corresponde al valor 1, 2, 3, 4, 5 o 6. Dentro del bloque de instrucciones hemos implementado otro condicional if donde comprobamos si los valores del array corresponden con el código morse de cada letra teniendo en cuenta que el 1 corresponde al punto y el 2 a la raya.

```
/*
char buscarletra(int tam_array, int *morse){
    char letra = ' ';

    if(tam_array == 1){
        if(morse[0] == 1){
            letra = 'E';
        } else if(morse[0] == 2){
            letra = 'T';
        }
    } else if (tam_array == 2){
        if(morse[0] == 1 && morse[1] == 1){
            letra = 'I';
        } else if(morse[0] == 1 && morse[1] == 2){
            letra = 'A';
        } else if(morse[0] == 2 && morse[1] == 1){
            letra = 'N';
        }
    }
}
```

```

} else if(morse[0] == 2 && morse[1] == 2){
    letra = 'M';
}
} else if (tam_array == 3){
    if(morse[0] == 1 && morse[1] == 1 && morse[2] == 1){
        letra = 'S';
    } else if(morse[0] == 1 && morse[1] == 1 && morse[2] == 2){
        letra = 'U';
    } else if(morse[0] == 1 && morse[1] == 2 && morse[2] == 1){
        letra = 'R';
    } else if(morse[0] == 1 && morse[1] == 2 && morse[2] == 2){
        letra = 'W';
    } else if(morse[0] == 2 && morse[1] == 1 && morse[2] == 1){
        letra = 'D';
    } else if(morse[0] == 2 && morse[1] == 1 && morse[2] == 2){
        letra = 'K';
    } else if(morse[0] == 2 && morse[1] == 2 && morse[2] == 1){
        letra = 'G';
    } else if(morse[0] == 2 && morse[1] == 2 && morse[2] == 2){
        letra = 'O';
    }
} else if (tam_array == 4){
    if(morse[0] == 1 && morse[1] == 1 && morse[2] == 1 && morse[3] == 1){
        letra = 'H';
    } else if(morse[0] == 1 && morse[1] == 1 && morse[2] == 1 &&
              morse[3] == 2){
        letra = 'V';
    } else if(morse[0] == 1 && morse[1] == 1 && morse[2] == 2 &&
              morse[3] == 1){
        letra = 'F';
    } else if(morse[0] == 1 && morse[1] == 2 && morse[2] == 1 &&
              morse[3] == 1){
        letra = 'L';
    } else if(morse[0] == 1 && morse[1] == 2 && morse[2] == 2 &&
              morse[3] == 1){
        letra = 'P';
    } else if(morse[0] == 1 && morse[1] == 2 && morse[2] == 2 &&
              morse[3] == 2){
        letra = 'J';
    } else if(morse[0] == 2 && morse[1] == 1 && morse[2] == 1 &&
              morse[3] == 1){
        letra = 'B';
    } else if(morse[0] == 2 && morse[1] == 1 && morse[2] == 1 &&
              morse[3] == 2){
        letra = 'X';
    } else if(morse[0] == 2 && morse[1] == 1 && morse[2] == 2 &&
              morse[3] == 1){
        letra = 'C';
    }
}

```

```

} else if(morse[0] == 2 && morse[1] == 1 && morse[2] == 2 &&
          morse[3] == 2){
    letra = 'Y';
} else if(morse[0] == 2 && morse[1] == 2 && morse[2] == 1 &&
          morse[3] == 1){
    letra = 'Z';
} else if(morse[0] == 2 && morse[1] == 2 && morse[2] == 1 &&
          morse[3] == 2){
    letra = 'Q';
}
} else if (tam_array == 5){
    if(morse[0] == 1 && morse[1] == 2 && morse[2] == 2 && morse[3] == 2
       && morse[4] == 2){
        letra = '1';
    } else if(morse[0] == 1 && morse[1] == 1 && morse[2] == 2 &&
              morse[3] == 2 && morse[4] == 2){
        letra = '2';
    } else if(morse[0] == 1 && morse[1] == 1 && morse[2] == 1 &&
              morse[3] == 2 && morse[4] == 2){
        letra = '3';
    } else if(morse[0] == 1 && morse[1] == 1 && morse[2] == 1 &&
              morse[3] == 1 && morse[4] == 2){
        letra = '4';
    } else if(morse[0] == 1 && morse[1] == 1 && morse[2] == 1 &&
              morse[3] == 1 && morse[4] == 1){
        letra = '5';
    } else if(morse[0] == 2 && morse[1] == 1 && morse[2] == 1 &&
              morse[3] == 1 && morse[4] == 1){
        letra = '6';
    } else if(morse[0] == 2 && morse[1] == 2 && morse[2] == 1 &&
              morse[3] == 1 && morse[4] == 1){
        letra = '7';
    } else if(morse[0] == 2 && morse[1] == 2 && morse[2] == 2 &&
              morse[3] == 1 && morse[4] == 1){
        letra = '8';
    } else if(morse[0] == 2 && morse[1] == 2 && morse[2] == 2 &&
              morse[3] == 2 && morse[4] == 1){
        letra = '9';
    } else if(morse[0] == 2 && morse[1] == 2 && morse[2] == 2 &&
              morse[3] == 2 && morse[4] == 2){
        letra = '0';
    }
} else if (tam_array == 6){
    if(morse[0] == 1 && morse[1] == 2 && morse[2] == 1 && morse[3] == 2
       && morse[4] == 1 && morse[5] == 2){
        letra = '..';
    } else if(morse[0] == 2 && morse[1] == 2 && morse[2] == 1 &&
              morse[3] == 1 && morse[4] == 2 && morse[5] == 2){

```

```

        letra = ',';
    }
}

return letra;
}

void setup() {
// put your setup code here, to run once:
pinMode(numPulsador, INPUT);           // Pin de entrada del pulsador de fin
                                         // de letra
pinMode(numPulsadorPunto, INPUT);      // Pin de salida del pulsador del
                                         // punto
pinMode(numPulsadorRaya, INPUT);       // Pin de salida del pulsador de la
                                         // raya
Serial.begin(9600);

tamano = 0;                           // Inicializar el tamaño que tendrá
                                         // el vector con la letra
for(int i = 0; i < TAM; i++){         // Inicializar el array donde se van
                                         // a almacenar la letra en morse
    morse_array[i] = 0;
}
}

void loop() {
// put your main code here, to run repeatedly:
// Comprueba si se pulsa el botón del punto
if ((digitalRead(numPulsadorPunto) == true)) {
    tamano += 1;                      // Aumenta el tamaño que tendrá el array
                                         // donde se almacena la letra
    morse_array[tamano - 1] = 1;        // Almacena el valor 1 en la última
                                         // posición del array indicando que es
                                         // un punto
    delay(500);                       // Se realiza una espera de 0,5 segundo.
                                         // Para asegurarnos de que se añade solo
                                         // una vez el valor 1 en el array
    //Serial.print(".");
                                         // Muestra por pantalla un '.' Para
                                         // comprobar que entra en el condicional
                                         // cuando se ha pulsado el botón del
                                         // punto
}

// Comprueba si se pulsa el botón de la raya
if ((digitalRead(numPulsadorRaya) == true)) {
    tamano += 1;                      // Aumenta el tamaño que tendrá el array
                                         // donde se almacena la letra
}
}

```

```

morse_array[tamano - 1] = 2;           // Almacena el valor 2 en la última
                                         // posición del array indicando que es
                                         // una raya
delay(500);                          // Se realiza una espera de 0,5 segundo.
                                         // Para asegurarnos de que se añade solo
                                         // una vez el valor 2 en el array
//Serial.print("-");                  // Muestra por pantalla un '-' Para
                                         // comprobar que entra en el condicional
                                         // cuando se ha pulsado el botón de la
                                         // raya
}

// Comprueba si se pulsa el botón de fin de letra
if ((digitalRead(numPulsador) == true)) {
    // Declarar la variable letra e asignarle el valor del resultado de la
    // función buscarletra(int tam_array, int *morse_array) pasándole como
    // parámetros el tamaño del array y el array donde se almacena la letra
    char letra = buscarletra(tamano, morse_array);
    // Comprueba que el valor de la letra no sea ' ' ya que esto se debe a
    // que no ha entrado en ningún condicional
    if (letra == ' '){
        Serial.println(" El valor introducido es incorrecto");
    } else {
        //Serial.print(" --> "); // Muestra por pantalla " --> "
        Serial.print(letra);      // Muestra por pantalla la letra obtenida
    }
}

delay(500);                          // Se realiza una espera de 0,5 segundo.
                                         // Para asegurarnos de que solo se realiza
                                         // una vez el condicional al pulsar el
                                         // botón fin de letra
tamano = 0;                          // Se vuelve a asignar el valor 0 al
                                         // tamaño del array para poder escribir de
                                         // nuevo otra letra
for(int i = 0; i < TAM; i++){ // Se vuelve a asignar el valor 0 al array
                                         // para poder escribir de nuevo otra letra
    morse_array[i] = 0;
}
}
}

```

2.3. Esquema de conexiones eléctricas.

2.3.1. Vista de Circuito.

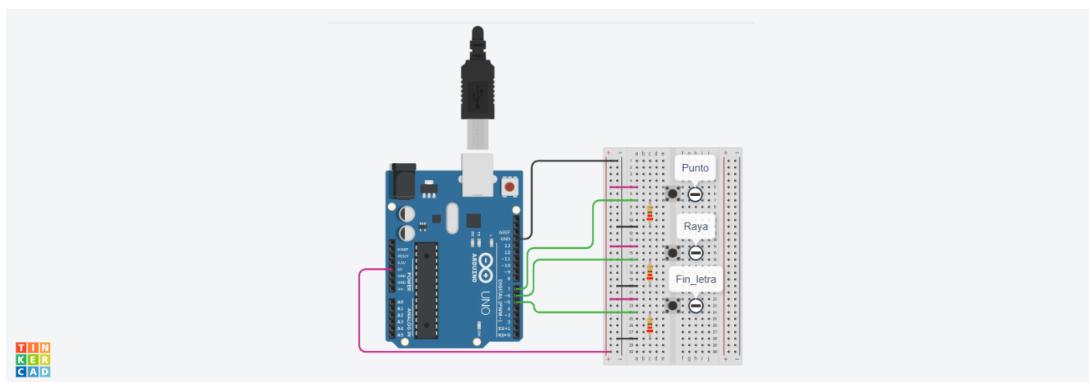


Figura 6: Circuito.

2.3.2. Vista esquemática.

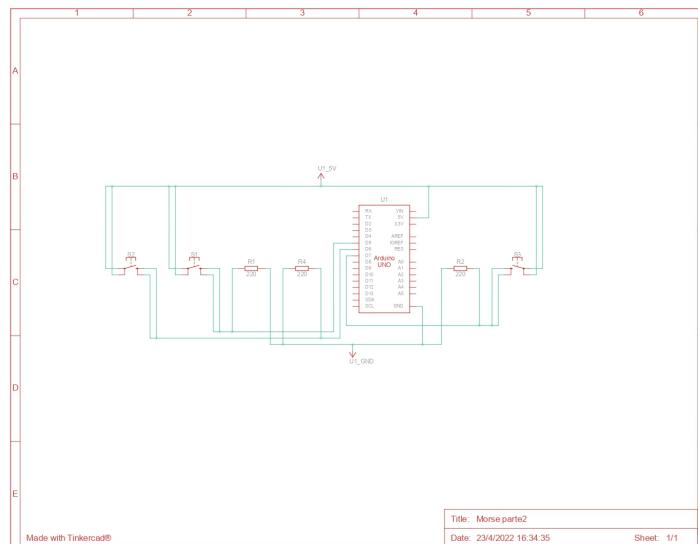


Figura 7: Esquema.

2.4. Funcionamiento del proyecto: En el simulador Tinkercad.

Enlace al proyecto en Tinkercad:

https://www.tinkercad.com/things/cUpPDibYRBs-morse-parte2/editel?sharecode=Y0jI27Pq85KCzAshAQWYUd7Q4sLd7KTqXsSkZt157_U

Enlace al vídeo con una demostración del funcionamiento del proyecto en el simulador:

<https://drive.google.com/file/d/13gymyNSeAp84T4LZNfp0KpTvVj-EhnIT/view?usp=sharing>

2.5. Funcionamiento del proyecto.

- Circuito.

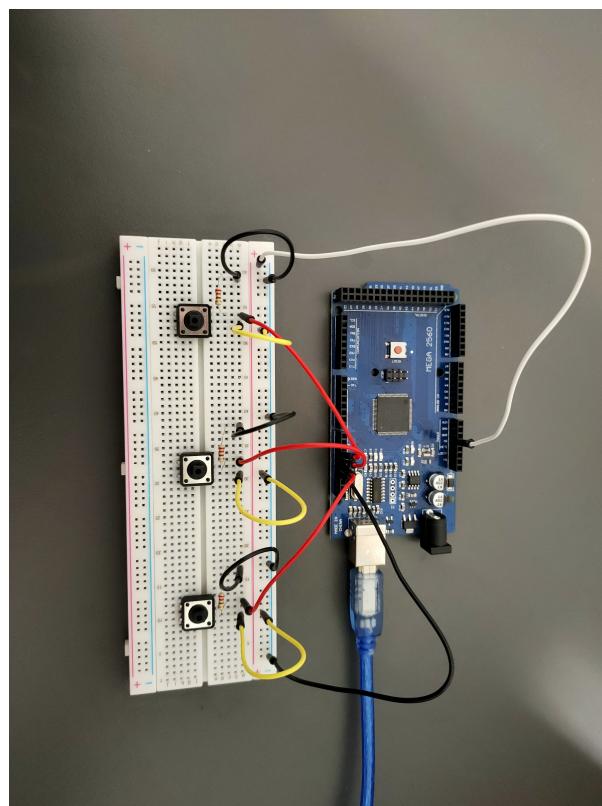


Figura 8: Arduino.

- Enlace al vídeo con una demostración del funcionamiento del proyecto:

<https://drive.google.com/file/d/1Lup17nMI1H7XE18R4lAT4tCZIvGGqYtd/view?usp=sharing>

<https://drive.google.com/file/d/12AZBK1Pe1drastKuo-sFLRnXWAPWCgBR/view?usp=sharing>

3. Posible ampliación del programa.

3.1. Mejorar el proyecto para que necesite un sólo botón.

Una posible ampliación para el segundo programa (pasar de código morse a lenguaje natural) sería dejando un botón para introducir los puntos y rayas y otro botón para el final de la letra. Esto se podría hacer calculando el tiempo que se mantiene pulsado el botón. Por ejemplo, si se pulsa menos de 2 segundos se considera un punto y si es durante un periodo mayor a 2s se considera una raya, la palabra finaliza al pulsar el segundo botón (el de fin de letra).

Además, podríamos reducir aun más el número de pulsadores empleados a uno si tenemos en cuenta el tiempo que el botón está sin pulsar. Por ejemplo, cuando pasan más de 10 segundos sin que se pulse el botón eso significaría que se ha terminado la letra que se estaba introduciendo y en este caso mostraría dicha letra.

Un ejemplo de los circuitos que podrían usarse para este proyecto serían los siguientes:

3.1.1. Componentes eléctricos.

Caso 1

- 1 Arduino Uno R3
- 2 Resistencias 220 Ω
- 2 Pulsador
- 8 Cables macho-macho

Caso 2

- 1 Arduino Uno R3
- 1 Resistencia 220 Ω
- 1 Pulsador
- 5 Cables macho-macho

3.1.2. Esquema de conexiones eléctricas.

- Vista de Circuito.



Figura 9: Circuito.

■ Vista esquemática.

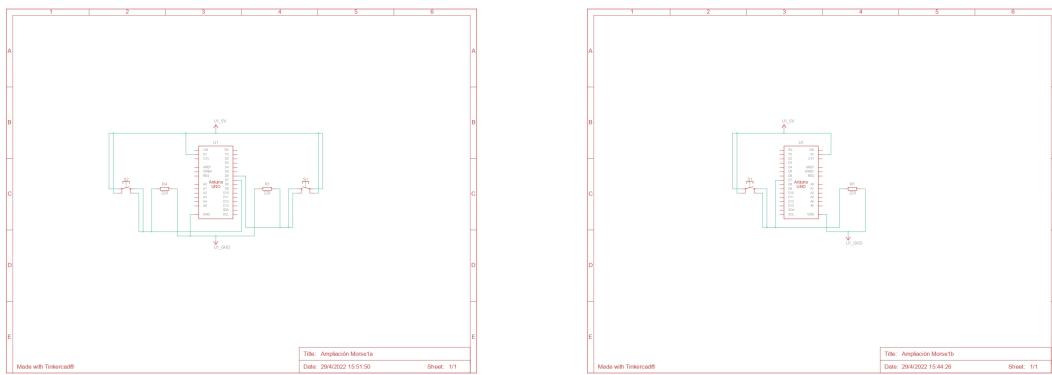


Figura 10: Esquema.

3.2. Unir los dos proyectos en uno.

Otra posible ampliación para este programa sería unir ambos proyectos en un solo proyecto que permitiese traducir de lenguaje natural a código morse y viceversa.

Para desarrollar este proyecto se podría implementar un código donde al inicio pre-guntase si se quiere pasar de lenguaje natural a código morse o, por el contrario, se quiere pasar de código morse a lenguaje natural. Esto se podría hacer introduciendo por teclado una respuesta, por ejemplo, mediante un listado donde las opciones tienen asociados un número y tienes que introducir el número correspondiente a la acción que quieras realizar.

Se implementarían dos funciones principales cada una con uno de los proyectos explicados anteriormente. Y al elegir una opción llama a la función correspondiente.

Un ejemplo del circuito que podría tener este proyecto sería el siguiente:

3.2.1. Componentes eléctricos.

- 1 Arduino Uno R3
- 4 Resistencias 220 Ω
- 1 LED Rojo
- 3 Pulsador
- 13 Cables macho-macho

3.2.2. Esquema de conexiones eléctricas.

- Vista de Circuito.

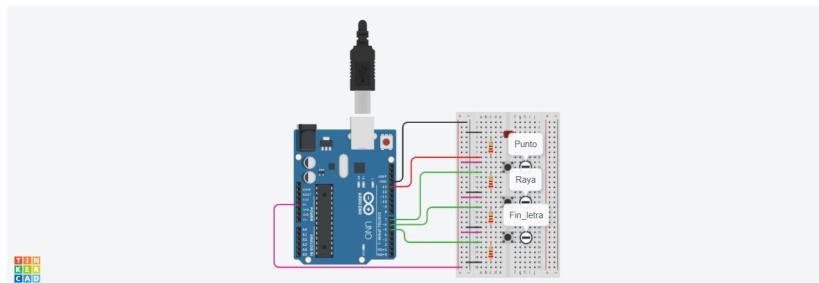


Figura 11: Circuito.

■ Vista esquemática.

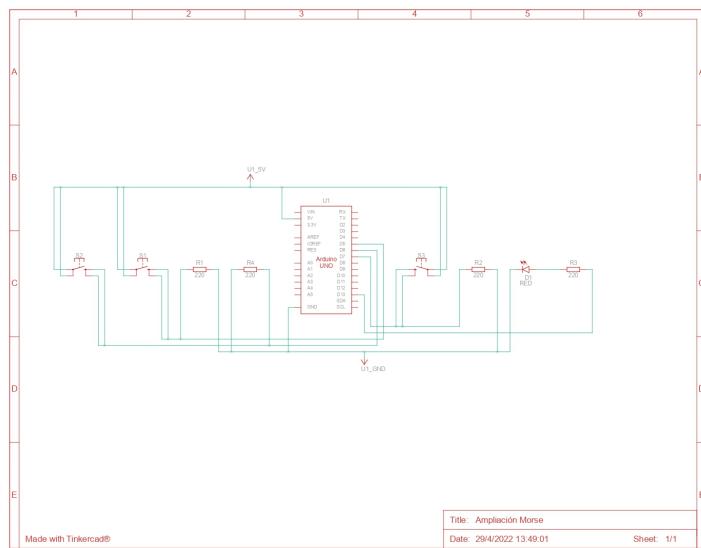


Figura 12: Esquema.