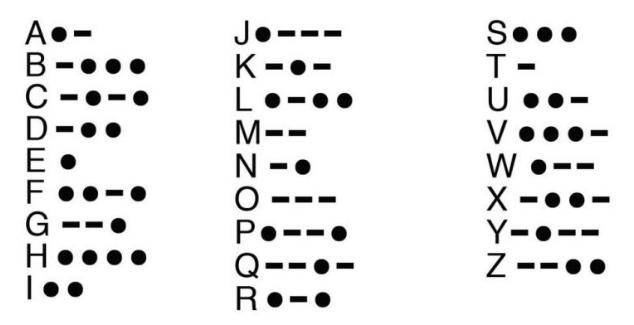
**CÓDIGO MORSE**

El Código morse**,** también conocido como clave morse, es un sistema de representación de letras y números mediante señales emitidas de forma intermitente.

Una raya tiene una duración de aproximadamente tres veces la del punto. Entre las letras de una misma palabra, la ausencia es de aproximadamente tres puntos. Para la separación de palabras transmitidas el tiempo es de aproximadamente tres veces el de la raya.



En el caso de nuestro programa en Tinkercad, cada punto (dot) tiene una duracion de 300 ms, y cada raya (dash) tiene una duracion de 900 ms, la cual es 3 veces la duracion de un punto.

**FUNCIONES EN ARDUINO**

**Arduino Serial Read**

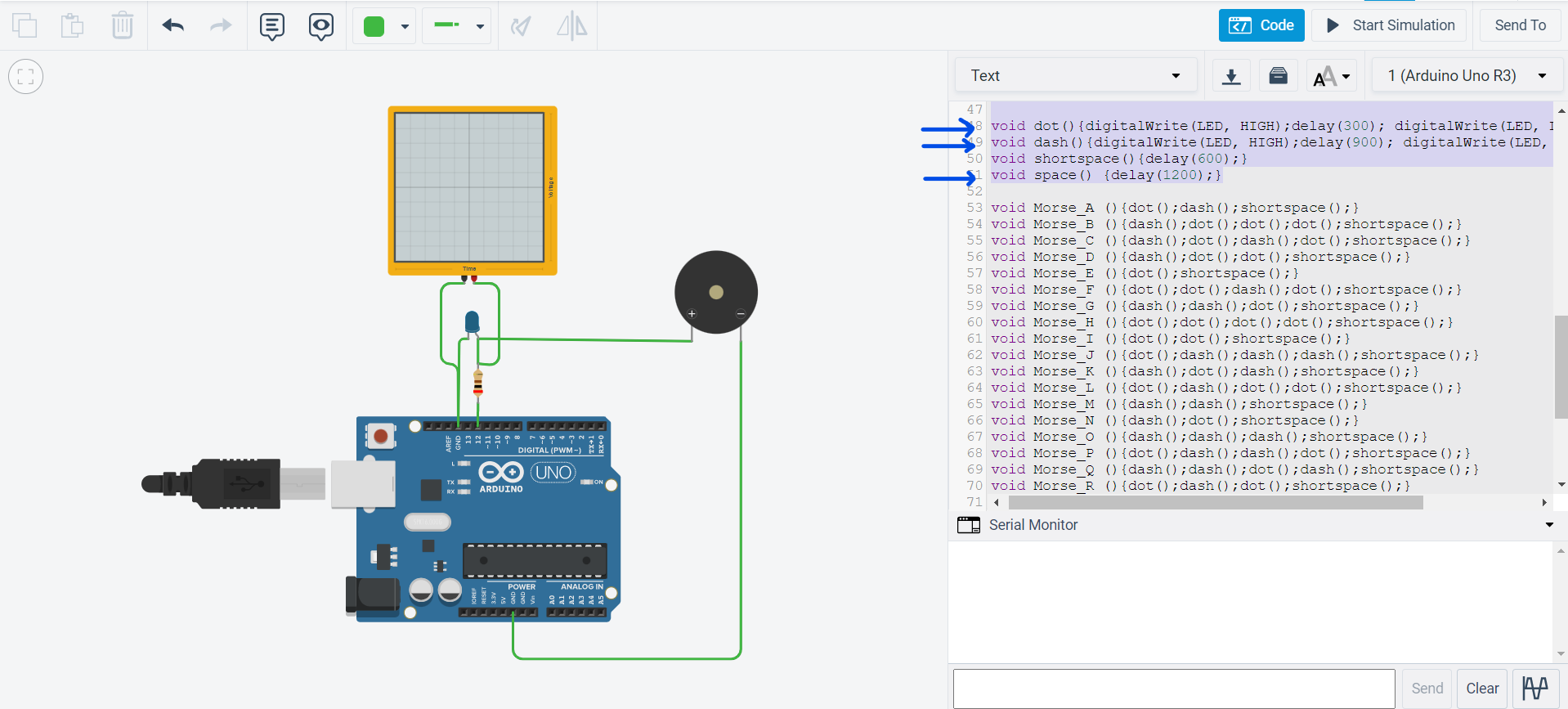
En el lenguaje Arduino Serial Read es una función que permite leer (recibir) bytes mediante un puerto Serial. La función puede funcionar asíncronamente. Una de las características principales del puerto serial de Arduino, es que sólo puede enviar o recibir un byte por transmisión. Entonces para poder usar a la función Serial.read() se requiere indicar, de preferencia, cuantos bytes son los que se quieren leer. Esto es, utilizar a la función Serial.available( ) en conjunto con Arduino Serial Read para hacer la lectura de los bytes seriales.

**Funcion if ( )**

La instrucción **if en arduino**, seguida de una condición, hará posible que nuestro programa coja distintos caminos dependiendo de si se cumplen o no.

**SOBRE EL CODIGO Y CIRCUITO:**

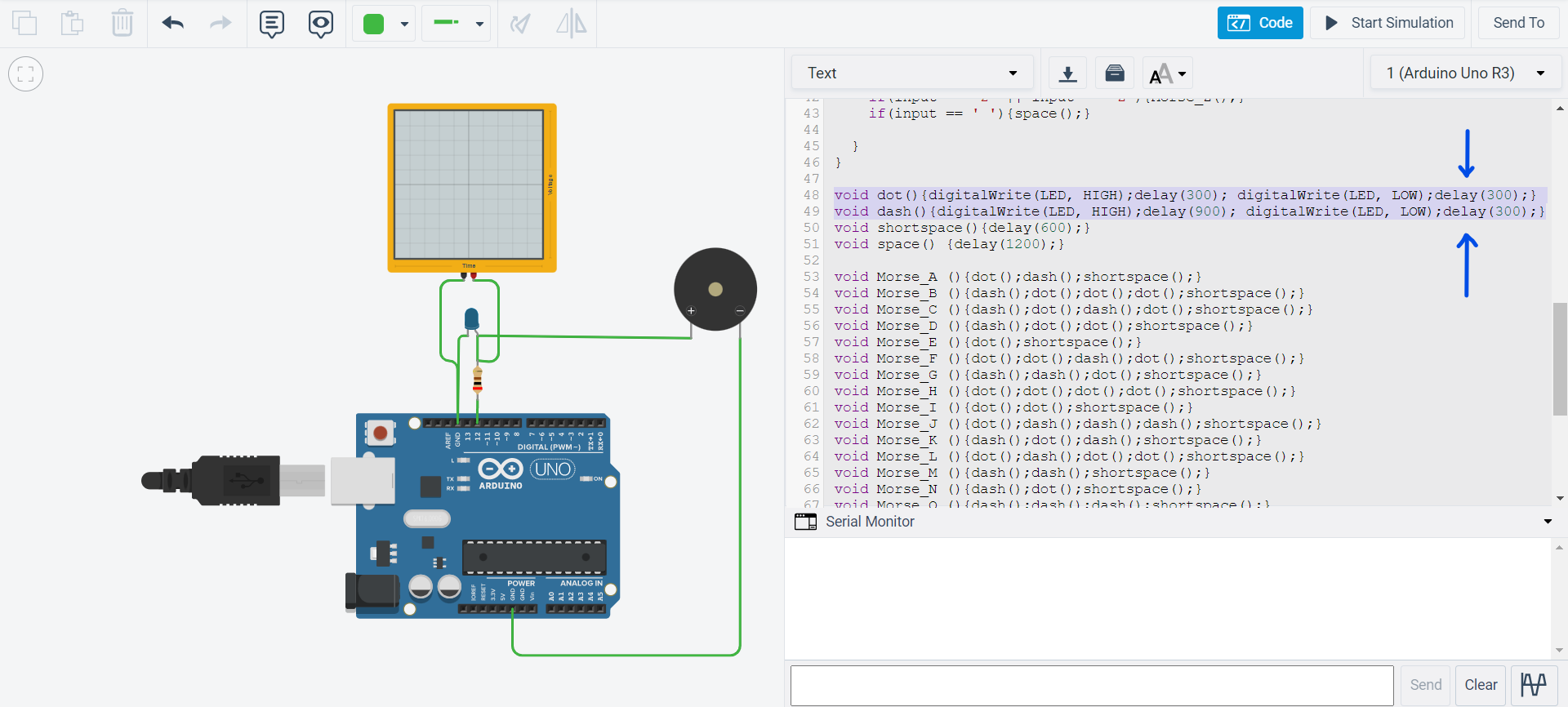
Hay una funcion diferente para los puntos ‘dot ( )’ y para las rayas ‘dash ( )’, ademas hay una funcion para la separacion entre palabras ‘space ( )’.

****

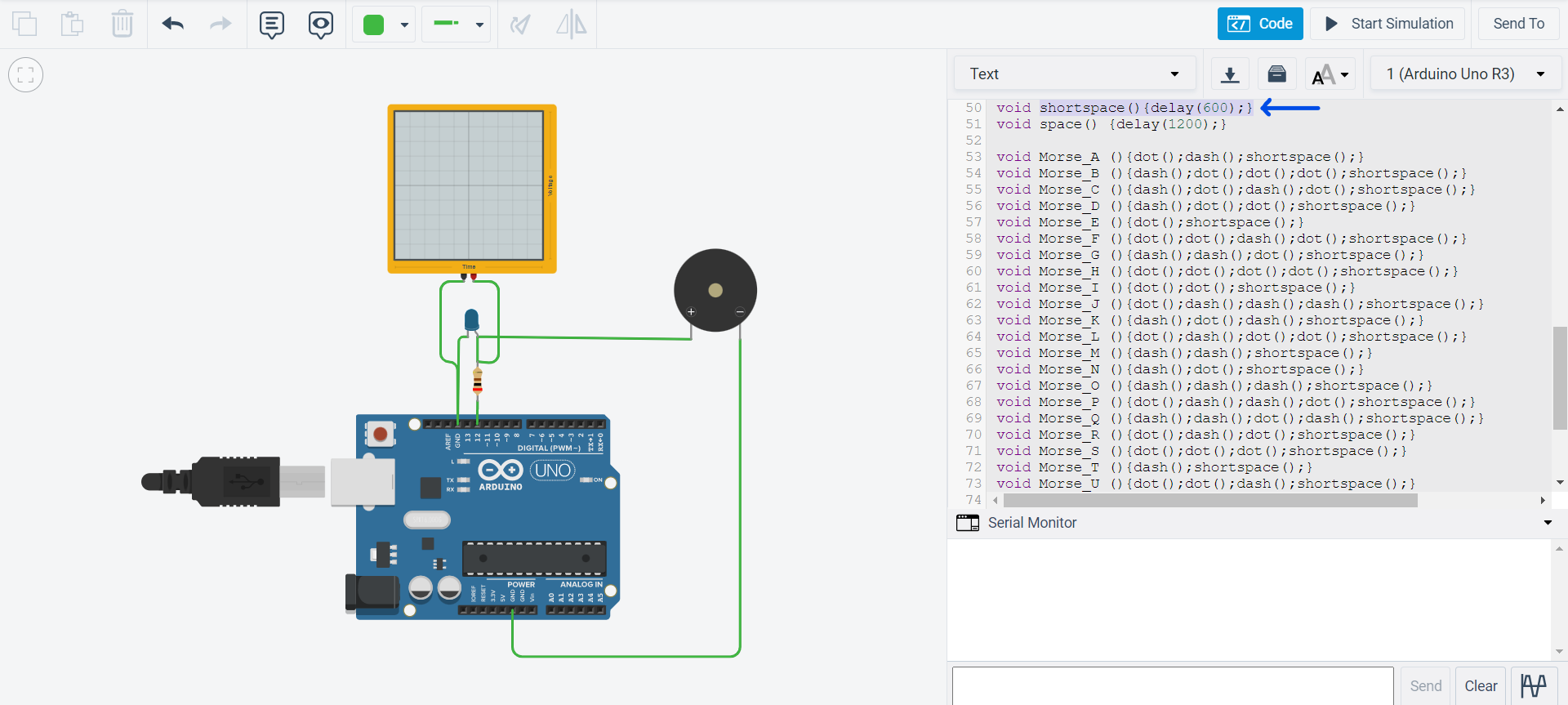
Hay 3 indicadores del codigo morse:

* Diodo.
* Osciloscopio.
* Sumbador (Piezo).

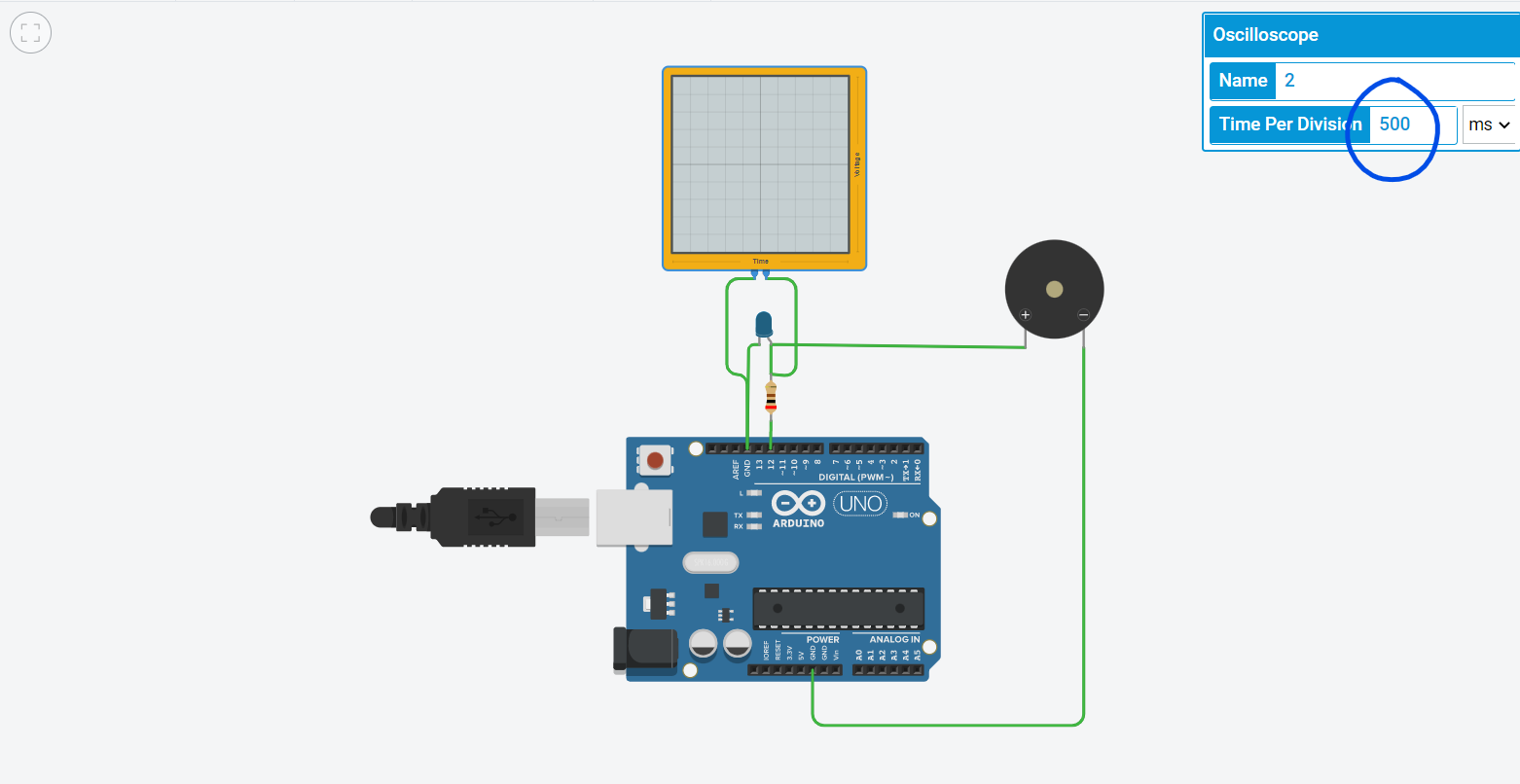
Al momento del desarrollo del codigo tube un problema al darle el ultimo delay a las funciones los puntos y las rayas, no le había colocado un delay al final y la salida de las letras no estaba trabajando bien, se estaban solapando los tiempos. Le coloqué el ultimo delay de 300 ms al final de la funcion ‘dot ( )’ y ‘dash ( )’ y funcionó como se debe.



La separacion entre letras de una misma parabra es de 900 ms, equivalente al intervalo de tiempo de tres puntos. Para lograr este intervalo, al final de las funciones de puntos ‘dot( )’ y rayas ‘dash( )’ se colocó un delay de 300 ms, y al final de cada funcion de las respectivas letras se colocó una función ‘shortspace( )’ que añade un delay de 600 ms, sumados nos da un delay total de 900 ms entre letras de una misma palabra.



El scope tenia un tiempo de 1 segundo en el eje x, y estaba mostrando la grafica de forma muy lenta, le baje el tiempo a 500 ms y mejoró la respuestas.



Para la lectura de dato a introducido por teclado se utilizó la funcion Serial.read ( ) de Arduino, en conjunto con la funcion Serial.available ( ).

El programa ejecutará la funcion de las respectivas letras solo si estas son introducidas a traves del Serial Monitor. Para saber cual debe ejecutarse se utilizaron condicionales if para comparar los caracteres de entradas con caracteres ya colocados en el codigo, los cuales son las letras del abecedario.

La idea de modularizar las funciones la tomé del video de youtube que indico en la bibliografia.

**CÓDIGO**

int LED= 12;

char input;

void setup()

{

pinMode(LED, OUTPUT);

Serial.begin(9600);

}

void loop()

{

if(Serial.available())

{

input = Serial.read();

Serial.println(input);

if(input == 'a' || input == 'A'){Morse\_A();}

if(input == 'b' || input == 'B'){Morse\_B();}

if(input == 'c' || input == 'C'){Morse\_C();}

if(input == 'd' || input == 'D'){Morse\_D();}

if(input == 'e' || input == 'E'){Morse\_E();}

if(input == 'f' || input == 'F'){Morse\_F();}

if(input == 'g' || input == 'G'){Morse\_G();}

if(input == 'h' || input == 'H'){Morse\_H();}

if(input == 'i' || input == 'I'){Morse\_I();}

if(input == 'j' || input == 'J'){Morse\_J();}

if(input == 'k' || input == 'K'){Morse\_K();}

if(input == 'l' || input == 'L'){Morse\_L();}

if(input == 'm' || input == 'M'){Morse\_M();}

if(input == 'n' || input == 'N'){Morse\_N();}

if(input == 'o' || input == 'O'){Morse\_O();}

if(input == 'p' || input == 'P'){Morse\_P();}

if(input == 'q' || input == 'Q'){Morse\_Q();}

if(input == 'r' || input == 'R'){Morse\_R();}

if(input == 's' || input == 'S'){Morse\_S();}

if(input == 't' || input == 'T'){Morse\_T();}

if(input == 'u' || input == 'U'){Morse\_U();}

if(input == 'v' || input == 'V'){Morse\_V();}

if(input == 'w' || input == 'W'){Morse\_W();}

if(input == 'x' || input == 'X'){Morse\_X();}

if(input == 'y' || input == 'Y'){Morse\_Y();}

if(input == 'z' || input == 'Z'){Morse\_Z();}

if(input == ' '){space();}

}

}

void dot(){digitalWrite(LED, HIGH);delay(300); digitalWrite(LED, LOW);delay(300);}

void dash(){digitalWrite(LED, HIGH);delay(900); digitalWrite(LED, LOW);delay(300);}

void shortspace(){delay(600);}

void space() {delay(1200);}

void Morse\_A (){dot();dash();shortspace();}

void Morse\_B (){dash();dot();dot();dot();shortspace();}

void Morse\_C (){dash();dot();dash();dot();shortspace();}

void Morse\_D (){dash();dot();dot();shortspace();}

void Morse\_E (){dot();shortspace();}

void Morse\_F (){dot();dot();dash();dot();shortspace();}

void Morse\_G (){dash();dash();dot();shortspace();}

void Morse\_H (){dot();dot();dot();dot();shortspace();}

void Morse\_I (){dot();dot();shortspace();}

void Morse\_J (){dot();dash();dash();dash();shortspace();}

void Morse\_K (){dash();dot();dash();shortspace();}

void Morse\_L (){dot();dash();dot();dot();shortspace();}

void Morse\_M (){dash();dash();shortspace();}

void Morse\_N (){dash();dot();shortspace();}

void Morse\_O (){dash();dash();dash();shortspace();}

void Morse\_P (){dot();dash();dash();dot();shortspace();}

void Morse\_Q (){dash();dash();dot();dash();shortspace();}

void Morse\_R (){dot();dash();dot();shortspace();}

void Morse\_S (){dot();dot();dot();shortspace();}

void Morse\_T (){dash();shortspace();}

void Morse\_U (){dot();dot();dash();shortspace();}

void Morse\_V (){dot();dot();dot();dash();shortspace();}

void Morse\_W (){dot();dash();dash();shortspace();}

void Morse\_X (){dash();dot();dot();dash();shortspace();}

void Morse\_Y (){dash();dot();dash();dash();shortspace();}

void Morse\_Z (){dash();dash();dot();dot();shortspace();}

**BIBLIOGRAFÍA**

<https://www.uclv.edu.cu/ese-alfabeto-compuesto-por-puntos-y-rayas-llamado-codigo-morse/#:~:text=Ese%20alfabeto%20compuesto%20por%20puntos%20y%20rayas%20llamado%20C%C3%B3digo%20Morse,-Por%20Liset%20Hernandez&text=El%20C%C3%B3digo%20morse%2C%20tambi%C3%A9n%20conocido,tres%20veces%20la%20del%20punto>.

<https://hetpro-store.com/TUTORIALES/arduino-serial-read/>

<https://www.youtube.com/watch?v=osPw81rCbE4&t=87s>