

Alumno: David Espinosa

Código morse

¿Qué es el código morse?

El código morse es el sistema de comunicación basado en letras y números utilizando señales que se emiten de forma que se interrumpe y prosigue cada cierto tiempo.

A ● -	J ● - - -	S ● ● ●
B - ● ● ●	K - ● -	T -
C - ● - ●	L ● - ● ●	U ● ● -
D - ● ●	M - -	V ● ● ● -
E ●	N - ●	W ● - -
F ● ● - ●	O - - -	X - ● ● -
G - - ●	P ● - - ●	Y - ● - -
H ● ● ● ●	Q - - ● -	Z - - ● ●
I ● ●	R ● - ●	

Introduciendo el texto el programa lo transcribirá a código morse, y se verá cómo se enciende el led y suena el zumbador en tiempos cortos (puntos) y largos (rayas), así Arduino deberá reconocer los caracteres y saber a qué código morse equivale.

Luego la operación inversa, donde se introducirá por un “switch” (pulsador), el código en morse y el Arduino deberá ser capaz de reconocer a que letra se corresponde.

Descripción de la solución:

Para la primera parte, a través del monitor serie del ordenador se introducirá a Arduino la letra que se quiere poner, Arduino buscará este carácter, y localizará su correspondiente secuencia en morse, compuesta de pulsos cortos y largos correspondientes a las rayas y puntos. Una vez identificada comenzara a ejecutarla, mediante el apagado y encendido del LED rojo y los pitidos del piezo, además mostrara por pantalla la letra que está reproduciendo.

Se podrán introducir palabras o secuencias de caracteres completos y la ira transmitiendo en morse secuencialmente.

Arduino reconocerá los 26 caracteres del abecedario (no incluye la Ñ, ya que no tiene codificación morse), se deben introducir en mayúscula, si no reconoce el carácter, se indicará en el monitor de serie que "se pruebe de nuevo que no se ha reconocido ese carácter".

En la segunda parte, Arduino deberá contar el tiempo de la pulsación de switch1, y en función del tiempo clasificado como un punto o una raya, así consecutivamente hasta que pulsando el switch2 se le indica que ya a terminado de introducirse la letra y la buscará en sus datos, hasta encontrar su correspondencia con el carácter correspondiente, mostrando por pantalla que letra a identificado.

Es importante que el punto sea pulsación corta, y no debe exceder los 0.5 segundos, porque por encima de este valor Arduino empezara a reconocerlo como raya.

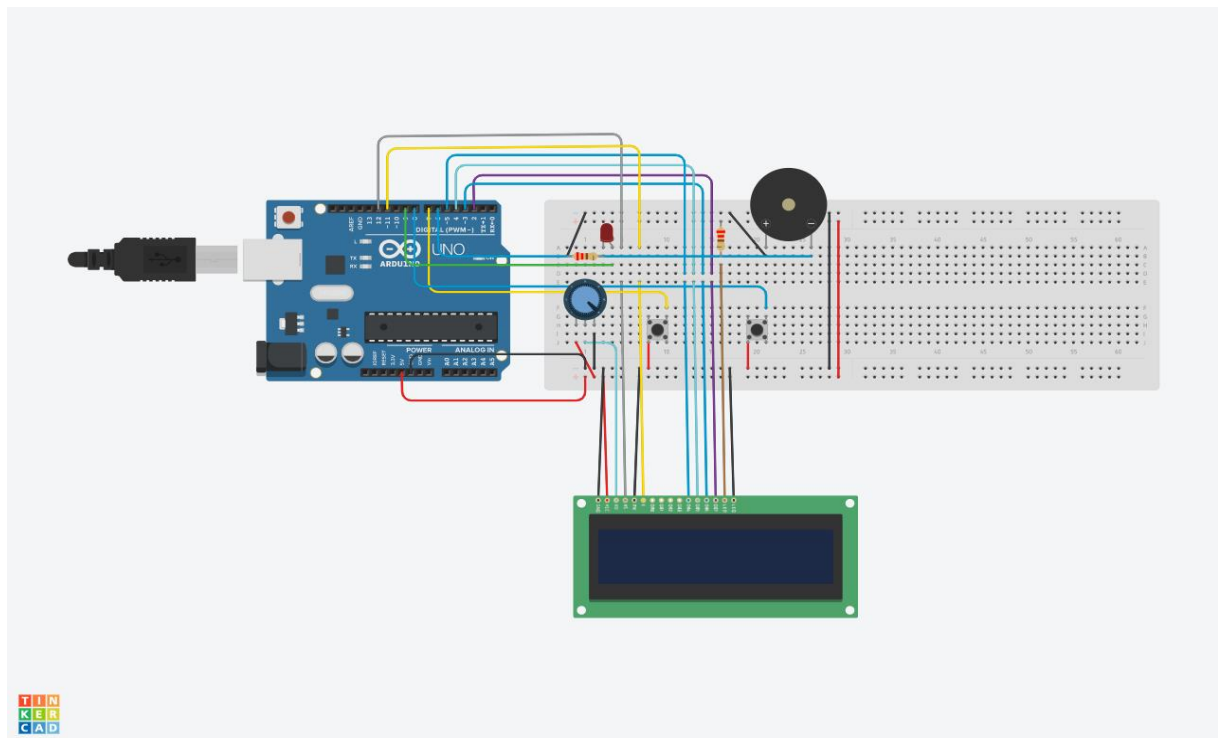
Para guardarlo en memoria, decidí solucionarlo mediante la creación de un vector de 4 dígitos que se iniciara en 0, las señales cortas (puntos) se identificaran y guardaran como 1, mientras que las señales largas (rayas) como 2, de tal manera que la estructura del vector se ira recorriendo y al final estará compuesta de 0, 1 y 2, y Arduino comparara con el vector o secuencia que corresponde a cada letra hasta encontrar la adecuada.

LISTA DE MATERIALES:

PARA ESTE PROYECTO HEMOS NECESITADO:

- Placa Arduino Uno
- Cable USB
- Protoboard
- LED rojo
- Resistencia de 330 ohm
- Piezo
- Dos pulsadores
- Cables

Esquema del Circuito:



Declaración de variables:

Librería LiquidCrystal para poder trabajar con la pantalla lcd

Neat Snicket

Se han guardado todos los cambios.

Código Iniciar simulación Enviar a

1 (Arduino Uno R3)

Texto

```

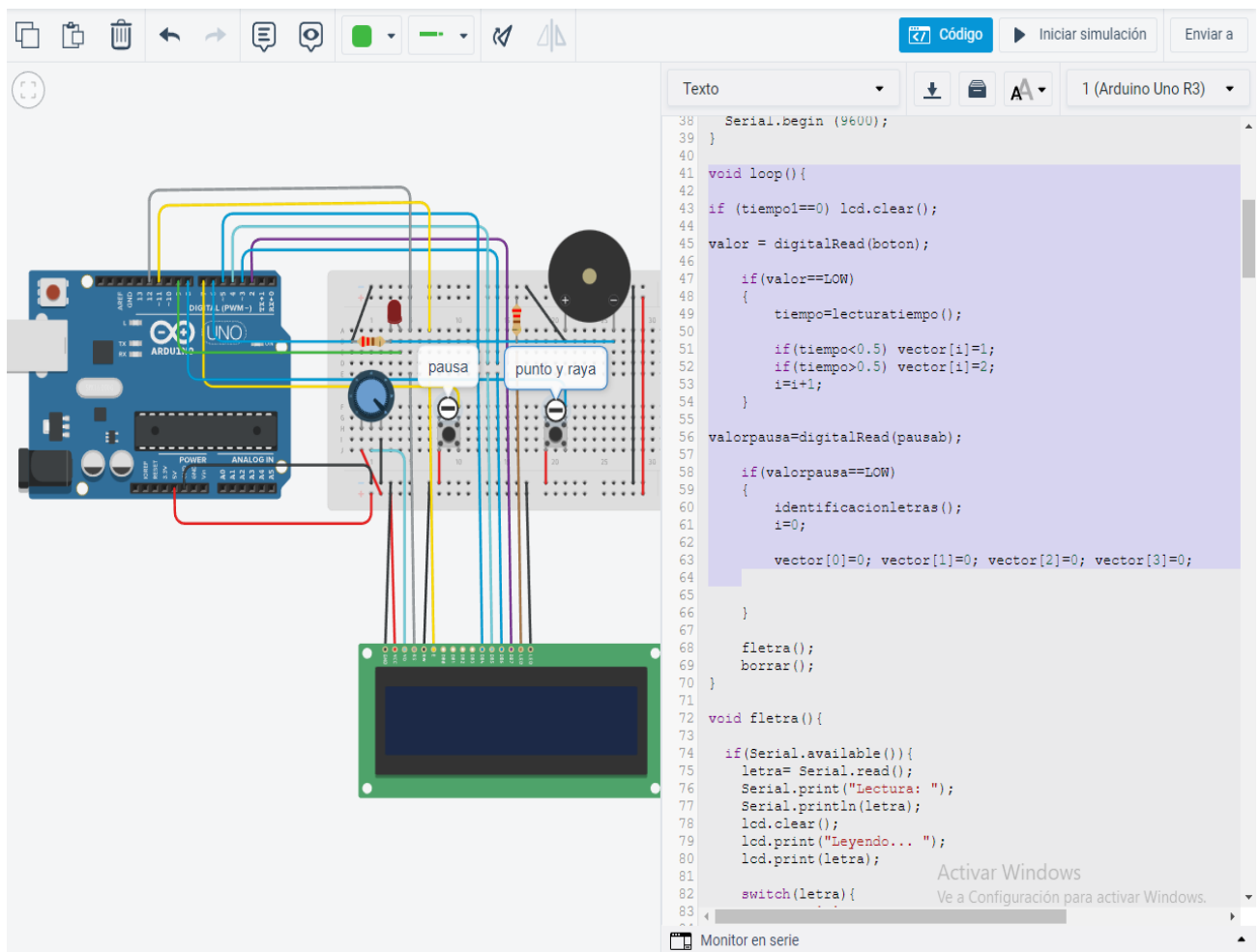
1 #include <LiquidCrystal.h>
2
3 int ledpin=9;
4 int buzzerpin=6;
5 int boton=8;
6 int pausa=7;
7
8 int valor=0;
9 int valorpausa=0;
10 float tiempo=0;
11 long tiempo1=0;
12 long tiempo2=0;
13 float tiempo3=0;
14 int vector[4]={0,0,0,0};
15 int i=0;
16
17 int switchvalue=0;
18 int punto=300;
19 int raya=500;
20 int pausa=400;
21 char letra;
22
23 int valorb=0;
24 long borr=0;
25 long tiempoob=0;
26
27 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
28
29 void setup(){
30   lcd.begin(16, 2);
31   // Imprimir un mensaje en la pantalla LCD.
32   lcd.print("Traductor Morse");
33
34   pinMode(ledpin, OUTPUT);
35   pinMode(buzzerpin, OUTPUT);
36   pinMode(boton, INPUT_PULLUP);
37   pinMode(pausab, INPUT_PULLUP);
38   Serial.begin(9600);
39 }
40
41 void loop(){
42
43   if (tiempo1==0) lcd.clear();
44   valor = digitalRead(boton);
45
46

```

Monitor en serie

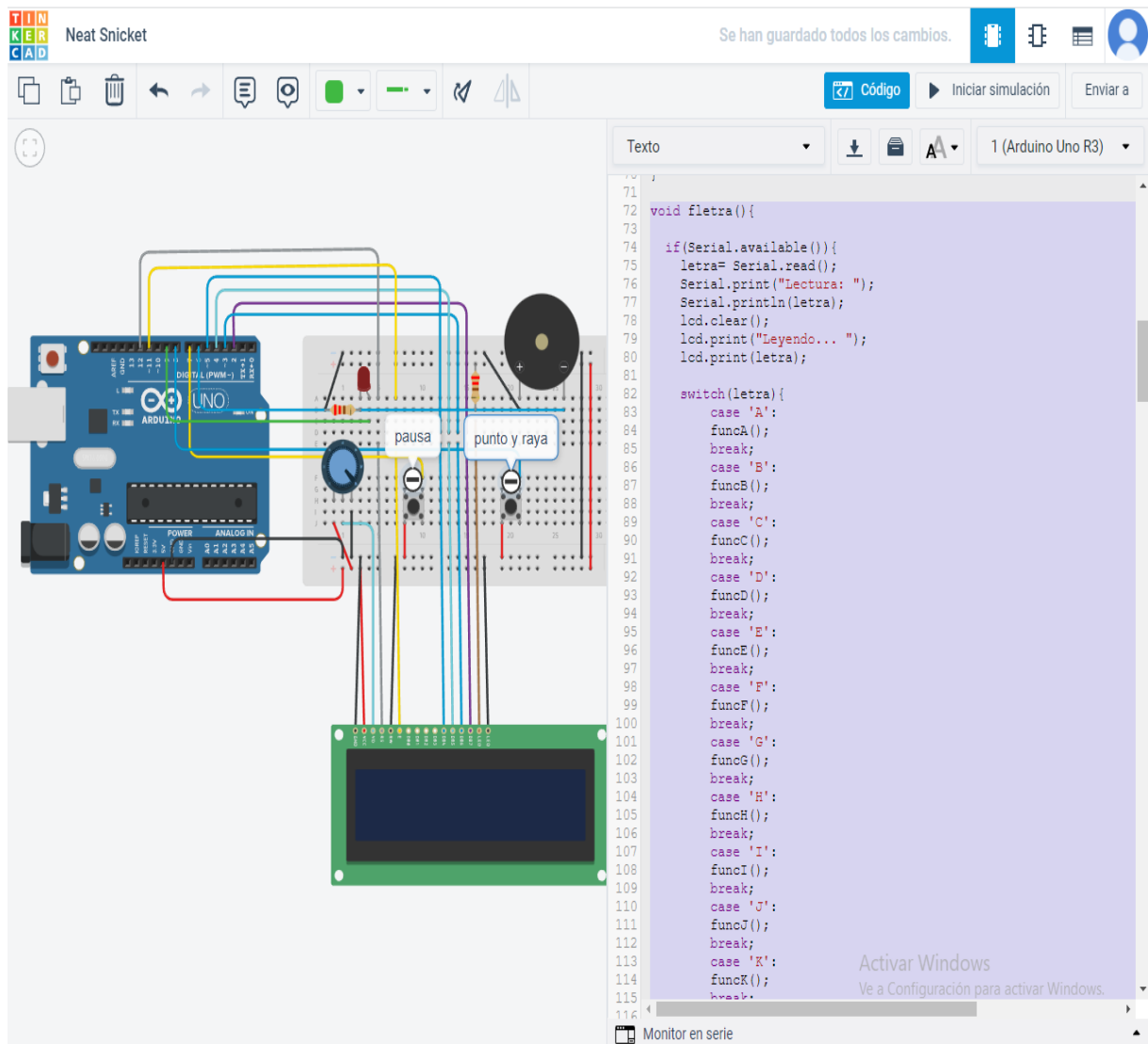
La función principal se iniciará cuando reconozca una pulsación, si este valor es menor de medio segundo almacenara un 1(correspondiente a un punto), si el valor es superior lo reconocerá como una raya y almacenara un 2, tras eso recorre el vector un paso.

La variable “pausa” será el segundo pulsador que dará a entender que a terminado de introducir la letra, no todas las letras tienen 4 dígitos en morse), tras eso comienza la búsqueda de alguna coincidencia, y luego reiniciara el vector



Arduino solicitará por monitor serie una lectura, que almacenará en el char letra, esta se imprimirá por pantalla con el comando `lcd print` propio de la librería.

A continuación, buscara coincidencia del carácter introducido con alguna de los 26 programados, y llamara a la función correspondiente para cada letra



The screenshot displays the Arduino IDE interface. On the left, a circuit diagram shows an Arduino Uno connected to a breadboard. The breadboard contains a potentiometer, two push buttons labeled 'pausa' and 'punto y raya', and an LCD screen. Wires connect the Arduino's digital pins to the buttons and the LCD's data bus. The right side of the IDE shows the C++ code for the project.

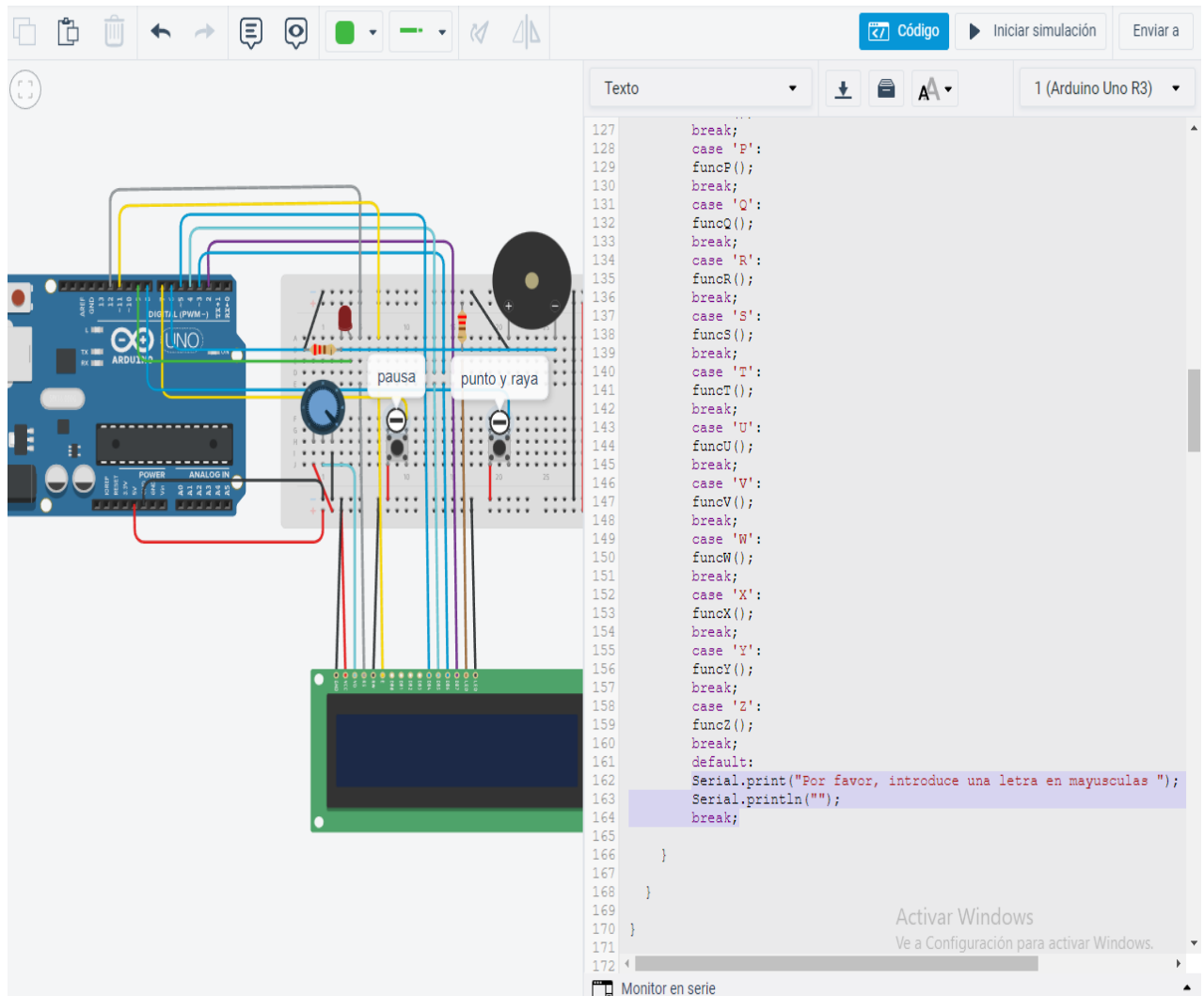
```
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116
```

```
void flettra(){  
  
    if(Serial.available()){  
        letra= Serial.read();  
        Serial.print("Lectura: ");  
        Serial.println(letra);  
        lcd.clear();  
        lcd.print("Leyendo... ");  
        lcd.print(letra);  
  
        switch(letra){  
            case 'A':  
                funcA();  
                break;  
            case 'B':  
                funcB();  
                break;  
            case 'C':  
                funcC();  
                break;  
            case 'D':  
                funcD();  
                break;  
            case 'E':  
                funcE();  
                break;  
            case 'F':  
                funcF();  
                break;  
            case 'G':  
                funcG();  
                break;  
            case 'H':  
                funcH();  
                break;  
            case 'I':  
                funcI();  
                break;  
            case 'J':  
                funcJ();  
                break;  
            case 'K':  
                funcK();  
                break;  
        }  
    }  
}
```

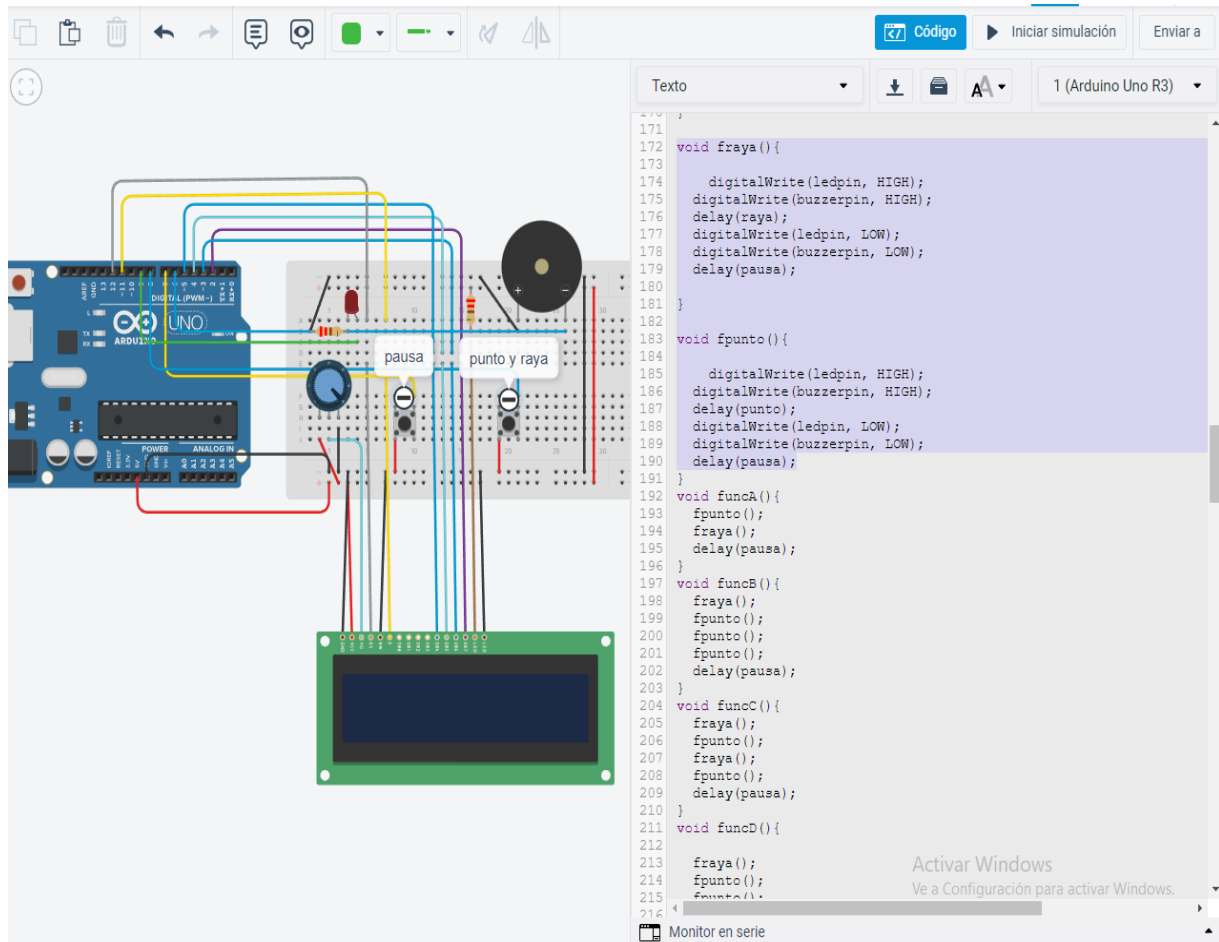
Activar Windows
Ve a Configuración para activar Windows.

Monitor en serie

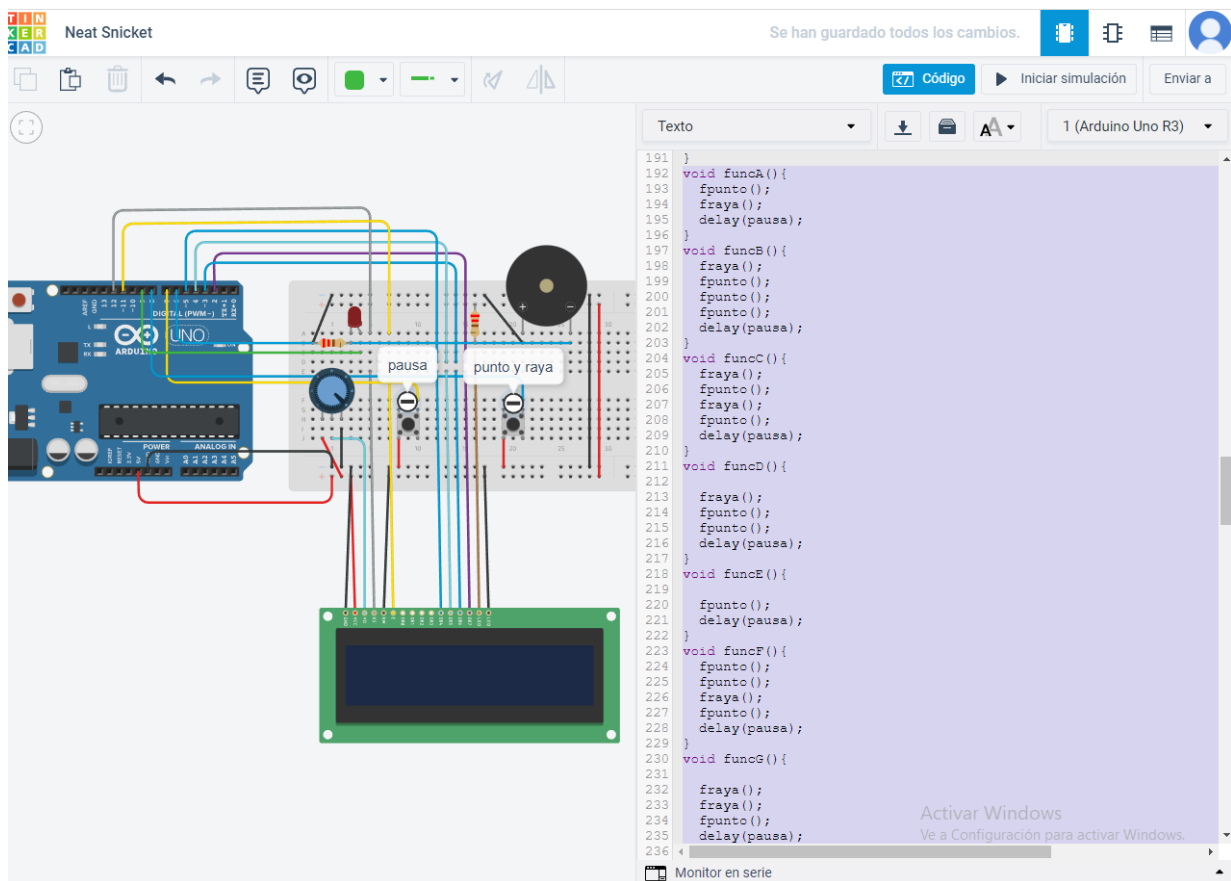
En caso de no encontrar coincidencia("default"), solicitara que se introduzca un carácter adecuado.



Las funciones raya y punto, generan el encendido de buzzer y led durante el tiempo adecuado, de tal manera que para emitir “un punto o raya” solo haga falta llamar una de estas funciones.

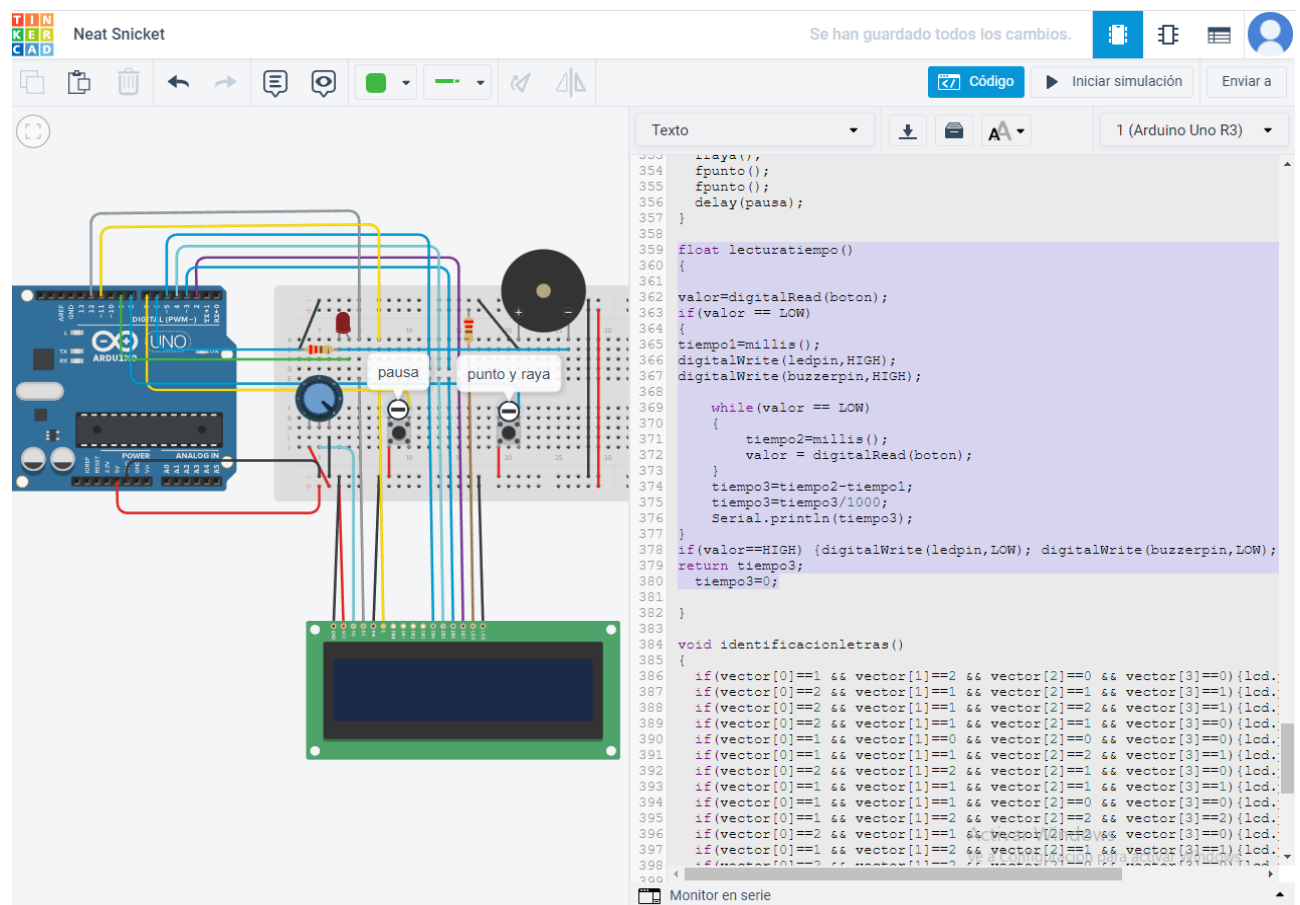


Como ya se tiene declaradas las funciones que generen rayas y puntos, solo se debe crear una función para cada una de las letras, e introducir la secuencia correspondiente en código morse, así cuando se llame a una de estas funciones de letra, pueda reproducir automáticamente la secuencia adecuada en código morse.



Función float para tomar una lectura de tiempo, la lectura comienza cuando tocamos el pulsador y acaba cuando lo soltamos, la diferencia será el tiempo que hemos pulsado el pulsador, la división por 1000 es para trabajar en segundos.

Para controlar mejor el tiempo mientras se pulsa se encenderá el led y el buzzer, la función devuelve el intervalo de tiempo en segundos.



The screenshot displays the Tinkercad software interface. On the left, a 3D model of an Arduino Uno R3 is connected to a breadboard circuit. The circuit includes a push button labeled 'punto y raya', a buzzer, and an LED. Wires connect the button to digital pins, the buzzer to a digital pin, and the LED to a digital pin. A potentiometer is also present on the breadboard. On the right, the code editor shows the following C++ code:

```
353 delay();
354 fpunto();
355 fpunto();
356 delay(pausa);
357 }
358
359 float lecturatiempo()
360 {
361     valor=digitalRead(boton);
362     if(valor == LOW)
363     {
364         tiempo1=millis();
365         digitalWrite(ledpin,HIGH);
366         digitalWrite(buzzerpin,HIGH);
367
368         while(valor == LOW)
369         {
370             tiempo2=millis();
371             valor = digitalRead(boton);
372         }
373         tiempo3=tiempo2-tiempo1;
374         tiempo3=tiempo3/1000;
375         Serial.println(tiempo3);
376     }
377
378     if(valor==HIGH) {digitalWrite(ledpin,LOW); digitalWrite(buzzerpin,LOW);
379     return tiempo3;
380     tiempo3=0;
381 }
382 }
383
384 void identificacionletras()
385 {
386     if(vector[0]==1 && vector[1]==2 && vector[2]==0 && vector[3]==0){lcd.
387     if(vector[0]==2 && vector[1]==1 && vector[2]==1 && vector[3]==1){lcd.
388     if(vector[0]==2 && vector[1]==1 && vector[2]==2 && vector[3]==1){lcd.
389     if(vector[0]==2 && vector[1]==1 && vector[2]==1 && vector[3]==0){lcd.
390     if(vector[0]==1 && vector[1]==0 && vector[2]==0 && vector[3]==0){lcd.
391     if(vector[0]==1 && vector[1]==1 && vector[2]==2 && vector[3]==1){lcd.
392     if(vector[0]==2 && vector[1]==2 && vector[2]==1 && vector[3]==0){lcd.
393     if(vector[0]==1 && vector[1]==1 && vector[2]==1 && vector[3]==1){lcd.
394     if(vector[0]==1 && vector[1]==2 && vector[2]==0 && vector[3]==0){lcd.
395     if(vector[0]==1 && vector[1]==2 && vector[2]==2 && vector[3]==2){lcd.
396     if(vector[0]==2 && vector[1]==1 && vector[2]==2 && vector[3]==0){lcd.
397     if(vector[0]==1 && vector[1]==2 && vector[2]==1 && vector[3]==1){lcd.
398     if(vector[0]==2 && vector[1]==2 && vector[2]==2 && vector[3]==1){lcd.
399     if(vector[0]==2 && vector[1]==2 && vector[2]==2 && vector[3]==2){lcd.
400 }
```

The interface also shows a 'Monitor en serie' window at the bottom right.

La función de identificación compara los valores del vector y si encuentra una coincidencia mostrara por pantalla la letra que corresponde.

TINER Neat Snicket

Se han guardado todos los cambios.

Código Iniciar simulación Enviar a

1 (Arduino Uno R3)

Texto

```
369 while(valor == LOW)
370 {
371     tiempo2=millis();
372     valor = digitalRead(boton);
373 }
374 tiempo3=tiempo2-tiempo1;
375 tiempo3=tiempo3/1000;
376 Serial.println(tiempo3);
377 }
378 if(valor==HIGH) {digitalWrite(ledpin,LOW); digitalWrite(buzzerpin,LOW);}
379 return tiempo3;
380 tiempo3=0;
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
```

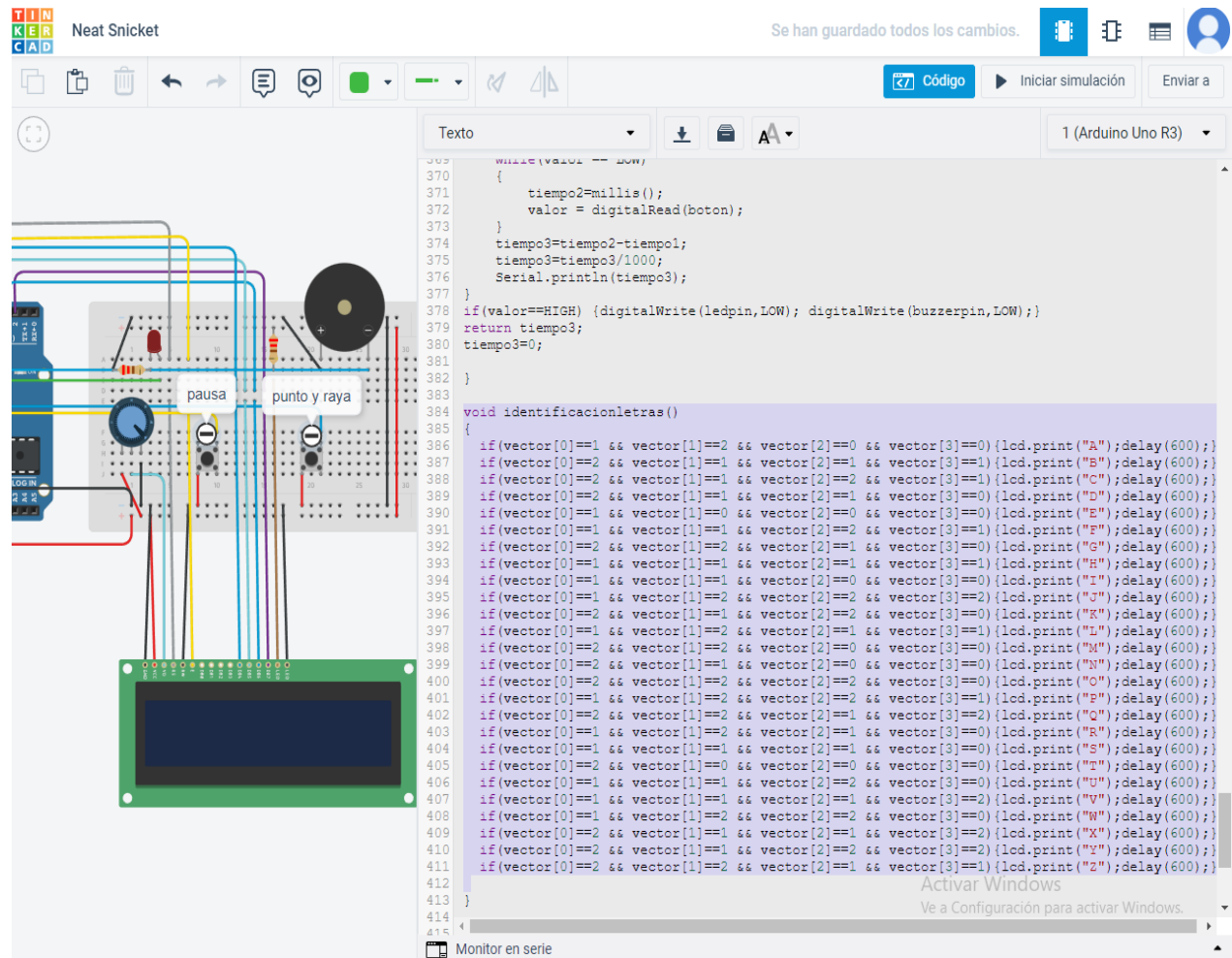
void identificacionletras()

```
{
386 if(vector[0]==1 && vector[1]==2 && vector[2]==0 && vector[3]==0){lcd.print("A");delay(600);}
387 if(vector[0]==2 && vector[1]==1 && vector[2]==1 && vector[3]==1){lcd.print("B");delay(600);}
388 if(vector[0]==2 && vector[1]==1 && vector[2]==2 && vector[3]==1){lcd.print("C");delay(600);}
389 if(vector[0]==2 && vector[1]==1 && vector[2]==1 && vector[3]==0){lcd.print("D");delay(600);}
390 if(vector[0]==1 && vector[1]==0 && vector[2]==0 && vector[3]==0){lcd.print("E");delay(600);}
391 if(vector[0]==1 && vector[1]==1 && vector[2]==2 && vector[3]==1){lcd.print("F");delay(600);}
392 if(vector[0]==2 && vector[1]==2 && vector[2]==1 && vector[3]==0){lcd.print("G");delay(600);}
393 if(vector[0]==1 && vector[1]==1 && vector[2]==1 && vector[3]==1){lcd.print("H");delay(600);}
394 if(vector[0]==1 && vector[1]==1 && vector[2]==0 && vector[3]==0){lcd.print("I");delay(600);}
395 if(vector[0]==1 && vector[1]==2 && vector[2]==2 && vector[3]==2){lcd.print("J");delay(600);}
396 if(vector[0]==2 && vector[1]==1 && vector[2]==2 && vector[3]==0){lcd.print("K");delay(600);}
397 if(vector[0]==1 && vector[1]==2 && vector[2]==1 && vector[3]==1){lcd.print("L");delay(600);}
398 if(vector[0]==2 && vector[1]==2 && vector[2]==0 && vector[3]==0){lcd.print("M");delay(600);}
399 if(vector[0]==2 && vector[1]==1 && vector[2]==0 && vector[3]==0){lcd.print("N");delay(600);}
400 if(vector[0]==2 && vector[1]==2 && vector[2]==2 && vector[3]==0){lcd.print("O");delay(600);}
401 if(vector[0]==1 && vector[1]==2 && vector[2]==2 && vector[3]==1){lcd.print("P");delay(600);}
402 if(vector[0]==2 && vector[1]==2 && vector[2]==1 && vector[3]==2){lcd.print("Q");delay(600);}
403 if(vector[0]==1 && vector[1]==2 && vector[2]==1 && vector[3]==0){lcd.print("R");delay(600);}
404 if(vector[0]==1 && vector[1]==1 && vector[2]==1 && vector[3]==0){lcd.print("S");delay(600);}
405 if(vector[0]==2 && vector[1]==0 && vector[2]==0 && vector[3]==0){lcd.print("T");delay(600);}
406 if(vector[0]==1 && vector[1]==1 && vector[2]==2 && vector[3]==0){lcd.print("U");delay(600);}
407 if(vector[0]==1 && vector[1]==1 && vector[2]==1 && vector[3]==2){lcd.print("V");delay(600);}
408 if(vector[0]==1 && vector[1]==2 && vector[2]==2 && vector[3]==0){lcd.print("W");delay(600);}
409 if(vector[0]==2 && vector[1]==1 && vector[2]==1 && vector[3]==2){lcd.print("X");delay(600);}
410 if(vector[0]==2 && vector[1]==1 && vector[2]==2 && vector[3]==2){lcd.print("Y");delay(600);}
411 if(vector[0]==2 && vector[1]==2 && vector[2]==1 && vector[3]==1){lcd.print("Z");delay(600);}
412 }
413 }
414 }
415 }
```

Activar Windows

Ve a Configuración para activar Windows.

Monitor en serie



Codigo del circuito:

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

```
int ledpin=9;
```

```
int buzzerpin=6;
```

```
int boton=8;
```

```
int pausab=7;
```

```
int valor=0;
```

```
int valorpausa=0;
```

```
float tiempo=0;
```

```
long tiempo1=0;
```

```
long tiempo2=0;
```

```
float tiempo3=0;
```

```
int vector[4]={0,0,0,0};
```

```
int i=0;
```

```
int switchvalue=0;
```

```
int punto=300;
```

```
int raya=500;
```

```
int pausa=400;
```

```
char letra;
```

```
int valorb=0;
```

```
long borr=0;
```

```
long tiempob=0;
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
void setup(){
```

```
  lcd.begin(16, 2);
```

```
  // Inprimir un mensaje en la pantalla LCD.
```

```
  lcd.print("Traductor Morse");
```

```
  pinMode(ledpin,OUTPUT);
```

```
  pinMode(buzzerpin,OUTPUT);
```

```
  pinMode(boton,INPUT_PULLUP);
```

```
  pinMode(pausab,INPUT_PULLUP);
```

```
  Serial.begin (9600);
```

```
}
```

```
void loop(){
```

```
  if (tiempo1==0) lcd.clear();
```

```
  valor = digitalRead(boton);
```

```
  if(valor==LOW)
```

```
  {
```

```
    tiempo=lecturatiempo();
```

```
if(tiempo<0.5) vector[i]=1;
```

```
if(tiempo>0.5) vector[i]=2;
```

```
i=i+1;
```

```
}
```

```
valorpausa=digitalRead(pausab);
```

```
if(valorpausa==LOW)
```

```
{
```

```
identificacionletras();
```

```
i=0;
```

```
vector[0]=0; vector[1]=0; vector[2]=0; vector[3]=0;
```

```
}
```

```
fletra();
```

```
    borrar();
```

```
}
```

```
void fletra(){
```

```
    if(Serial.available()){
```

```
        letra= Serial.read();
```

```
        Serial.print("Lectura: ");
```

```
Serial.println(letra);
```

```
lcd.clear();
```

```
lcd.print("Leyendo... ");
```

```
lcd.print(letra);
```

```
switch(letra){
```

```
    case 'A':
```

```
        funcA();
```

```
        break;
```

```
    case 'B':
```

```
        funcB();
```

```
        break;
```

```
    case 'C':
```

```
        funcC();
```

```
        break;
```

```
    case 'D':
```

```
        funcD();
```

```
        break;
```

```
    case 'E':
```

```
        funcE();
```

```
        break;
```

```
    case 'F':
```

```
        funcF();
```

```
        break;
```

```
    case 'G':
```

```
        funcG();
```

```
break;  
case 'H':  
    funcH();  
    break;  
case 'I':  
    funcI();  
    break;  
case 'J':  
    funcJ();  
    break;  
case 'K':  
    funcK();  
    break;  
case 'L':  
    funcL();  
    break;  
case 'M':  
    funcM();  
    break;  
case 'N':  
    funcN();  
    break;  
case 'O':  
    funcO();  
    break;  
case 'P':
```



```
funcP();  
break;  
case 'Q':  
funcQ();  
break;  
case 'R':  
funcR();  
break;  
case 'S':  
funcS();  
break;  
case 'T':  
funcT();  
break;  
case 'U':  
funcU();  
break;  
case 'V':  
funcV();  
break;  
case 'W':  
funcW();  
break;  
case 'X':  
funcX();  
break;
```

```
        case 'Y':
            funcY();
            break;
        case 'Z':
            funcZ();
            break;
    default:
        Serial.print("Por favor, introduce una letra en mayusculas ");
        Serial.println("");
        break;

    }

}

}

}

void fraya(){

    digitalWrite(ledpin, HIGH);
    digitalWrite(buzzerpin, HIGH);
    delay(raya);
    digitalWrite(ledpin, LOW);
    digitalWrite(buzzerpin, LOW);
    delay(pausa);
```

```
}
```

```
void fpunto(){
```

```
    digitalWrite(ledpin, HIGH);
```

```
    digitalWrite(buzzerpin, HIGH);
```

```
    delay(punto);
```

```
    digitalWrite(ledpin, LOW);
```

```
    digitalWrite(buzzerpin, LOW);
```

```
    delay(pausa);
```

```
}
```

```
void funcA(){
```

```
    fpunto();
```

```
    fraya();
```

```
    delay(pausa);
```

```
}
```

```
void funcB(){
```

```
    fraya();
```

```
    fpunto();
```

```
    fpunto();
```

```
    fpunto();
```

```
    delay(pausa);
```

```
}
```

```
void funcC(){
```

```
    fraya();
```

```
    fpunto();
```

```
    fraya();  
    fpunto();  
    delay(pausa);  
}  
void funcD(){
```

```
    fraya();  
    fpunto();  
    fpunto();  
    delay(pausa);  
}  
void funcE(){
```

```
    fpunto();  
    delay(pausa);  
}
```

```
void funcF(){  
    fpunto();  
    fpunto();  
    fraya();  
    fpunto();  
    delay(pausa);  
}
```

```
void funcG(){
```

```
    fraya();
```

```
    fraya();  
    fpunto();  
    delay(pausa);  
}  
void funcH(){
```

```
    fpunto();  
    fpunto();  
    fpunto();  
    fpunto();  
    delay(pausa);  
}
```

```
void funcI(){
```

```
    fpunto();  
    fpunto();  
    delay(pausa);  
}
```

```
void funcJ(){
```

```
    fpunto();  
    fraya();  
    fraya();  
    fraya();  
    delay(pausa);  
}
```

```
void funcK(){  
    fraya();  
    fpunto();  
    fraya();  
    delay(pausa);  
}
```

```
void funcL(){  
    fpunto();  
    fraya();  
    fpunto();  
    fpunto();  
    delay(pausa);  
}
```

```
void funcM(){  
    fraya();  
    fraya();  
    delay(pausa);  
}
```

```
void funcN(){  
    fraya();  
    fpunto();  
    delay(pausa);  
}
```

```
void funcO(){  
    fraya();  
    fraya();
```

```
    fraya();  
    delay(pausa);  
}  
void funcP(){  
    fpunto();  
    fraya();  
    fraya();  
    fpunto();  
    delay(pausa);  
}  
void funcQ(){  
    fraya();  
    fraya();  
    fpunto();  
    fraya();  
    delay(pausa);  
}  
void funcR(){  
    fpunto();  
    fraya();  
    fpunto();  
    delay(pausa);  
}  
void funcS(){  
    fpunto();  
    fpunto();
```

```
    fpunto();  
    delay(pausa);  
}  
void funcT(){  
    fraya();  
    delay(pausa);  
}  
void funcU(){  
    fpunto();  
    fpunto();  
    fraya();  
    delay(pausa);  
}  
void funcV(){  
    fpunto();  
    fpunto();  
    fpunto();  
    fraya();  
    delay(pausa);  
}  
void funcW(){  
    fpunto();  
    fraya();  
    fraya();  
    delay(pausa);  
}
```



```
void funcX(){  
    fraya();  
    fpunto();  
    fpunto();  
    fraya();  
    delay(pausa);  
}
```

```
void funcY(){  
    fraya();  
    fpunto();  
    fraya();  
    fraya();  
    delay(pausa);  
}
```

```
void funcZ(){  
    fraya();  
    fraya();  
    fpunto();  
    fpunto();  
    delay(pausa);  
}
```

```
float lecturatiempo()  
{
```

```
    valor=digitalRead(boton);
```

```

if(valor == LOW)
{
    tiempo1=millis();
    digitalWrite(ledpin,HIGH);
    digitalWrite(buzzerpin,HIGH);

    while(valor == LOW)
    {
        tiempo2=millis();
        valor = digitalRead(boton);
    }
    tiempo3=tiempo2-tiempo1;
    tiempo3=tiempo3/1000;
    Serial.println(tiempo3);
}

if(valor==HIGH) {digitalWrite(ledpin,LOW); digitalWrite(buzzerpin,LOW);}
return tiempo3;
tiempo3=0;

}

void identificacionletras()
{
    if(vector[0]==1 && vector[1]==2 && vector[2]==0 &&
vector[3]==0){lcd.print("A");delay(600);}

    if(vector[0]==2 && vector[1]==1 && vector[2]==1 &&
vector[3]==1){lcd.print("B");delay(600);}

```

```
if(vector[0]==2 && vector[1]==1 && vector[2]==2 &&
vector[3]==1){lcd.print("C");delay(600);}
```

```
if(vector[0]==2 && vector[1]==1 && vector[2]==1 &&
vector[3]==0){lcd.print("D");delay(600);}
```

```
if(vector[0]==1 && vector[1]==0 && vector[2]==0 &&
vector[3]==0){lcd.print("E");delay(600);}
```

```
if(vector[0]==1 && vector[1]==1 && vector[2]==2 &&
vector[3]==1){lcd.print("F");delay(600);}
```

```
if(vector[0]==2 && vector[1]==2 && vector[2]==1 &&
vector[3]==0){lcd.print("G");delay(600);}
```

```
if(vector[0]==1 && vector[1]==1 && vector[2]==1 &&
vector[3]==1){lcd.print("H");delay(600);}
```

```
if(vector[0]==1 && vector[1]==1 && vector[2]==0 &&
vector[3]==0){lcd.print("I");delay(600);}
```

```
if(vector[0]==1 && vector[1]==2 && vector[2]==2 &&
vector[3]==2){lcd.print("J");delay(600);}
```

```
if(vector[0]==2 && vector[1]==1 && vector[2]==2 &&
vector[3]==0){lcd.print("K");delay(600);}
```

```
if(vector[0]==1 && vector[1]==2 && vector[2]==1 &&
vector[3]==1){lcd.print("L");delay(600);}
```

```
if(vector[0]==2 && vector[1]==2 && vector[2]==0 &&
vector[3]==0){lcd.print("M");delay(600);}
```

```
if(vector[0]==2 && vector[1]==1 && vector[2]==0 &&
vector[3]==0){lcd.print("N");delay(600);}
```

```
if(vector[0]==2 && vector[1]==2 && vector[2]==2 &&
vector[3]==0){lcd.print("O");delay(600);}
```

```
if(vector[0]==1 && vector[1]==2 && vector[2]==2 &&
vector[3]==1){lcd.print("P");delay(600);}
```

```
if(vector[0]==2 && vector[1]==2 && vector[2]==1 &&
vector[3]==2){lcd.print("Q");delay(600);}
```

```
    if(vector[0]==1 && vector[1]==2 && vector[2]==1 &&
vector[3]==0){lcd.print("R");delay(600);}
```

```
    if(vector[0]==1 && vector[1]==1 && vector[2]==1 &&
vector[3]==0){lcd.print("S");delay(600);}
```

```
    if(vector[0]==2 && vector[1]==0 && vector[2]==0 &&
vector[3]==0){lcd.print("T");delay(600);}
```

```
    if(vector[0]==1 && vector[1]==1 && vector[2]==2 &&
vector[3]==0){lcd.print("U");delay(600);}
```

```
    if(vector[0]==1 && vector[1]==1 && vector[2]==1 &&
vector[3]==2){lcd.print("V");delay(600);}
```

```
    if(vector[0]==1 && vector[1]==2 && vector[2]==2 &&
vector[3]==0){lcd.print("W");delay(600);}
```

```
    if(vector[0]==2 && vector[1]==1 && vector[2]==1 &&
vector[3]==2){lcd.print("X");delay(600);}
```

```
    if(vector[0]==2 && vector[1]==1 && vector[2]==2 &&
vector[3]==2){lcd.print("Y");delay(600);}
```

```
    if(vector[0]==2 && vector[1]==2 && vector[2]==1 &&
vector[3]==1){lcd.print("Z");delay(600);}
```

```
}
```

```
void borrar()
```

```
{
```

```
    valorb=digitalRead(pausab);
```

```
    while(valorb == LOW)
```

```
    {
```

```
        tiempob=millis();
```

```
valorb = digitalRead(pausab);
```

```
}
```

```
borr=tiempob;
```

```
borr=borr/1000;
```

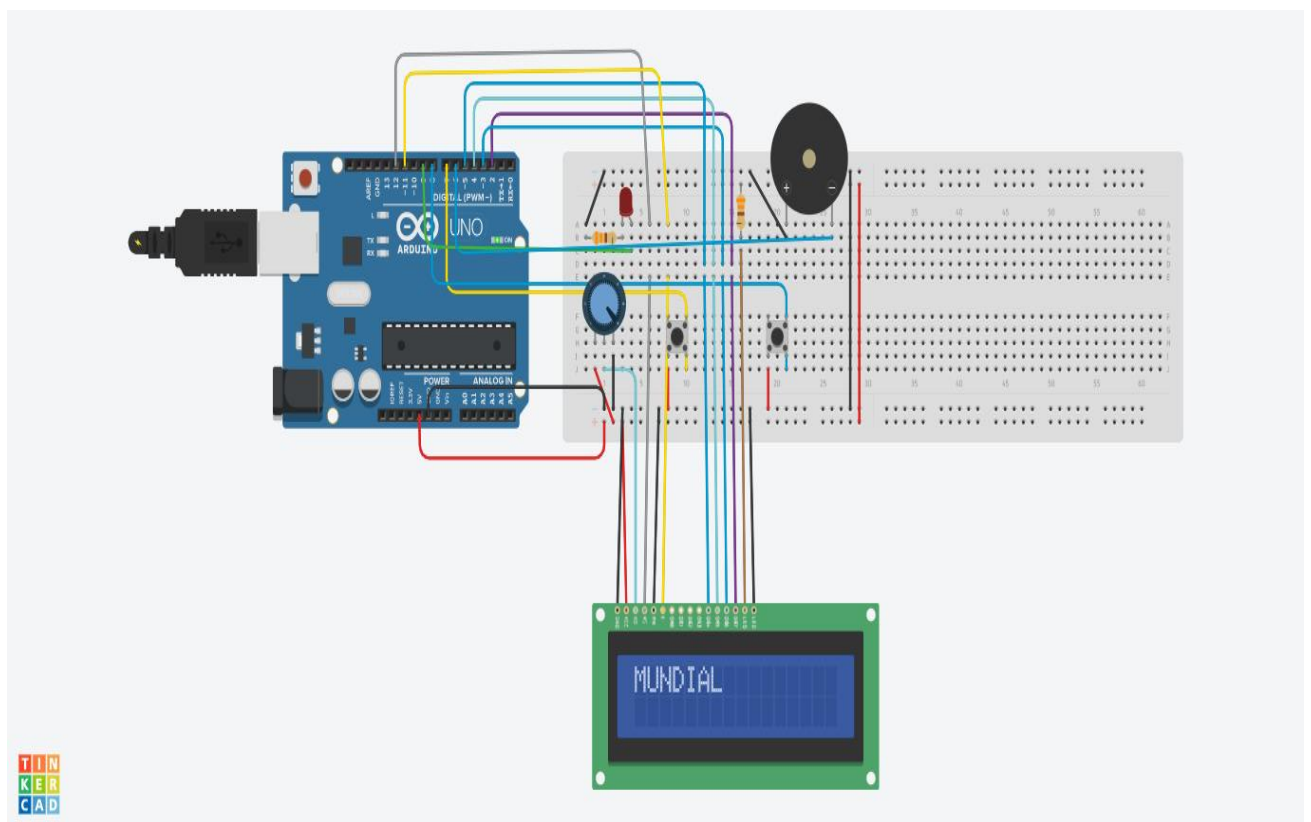
```
if(borr>1) lcd.clear();
```

```
borr=0;
```

```
tiempob=0;
```

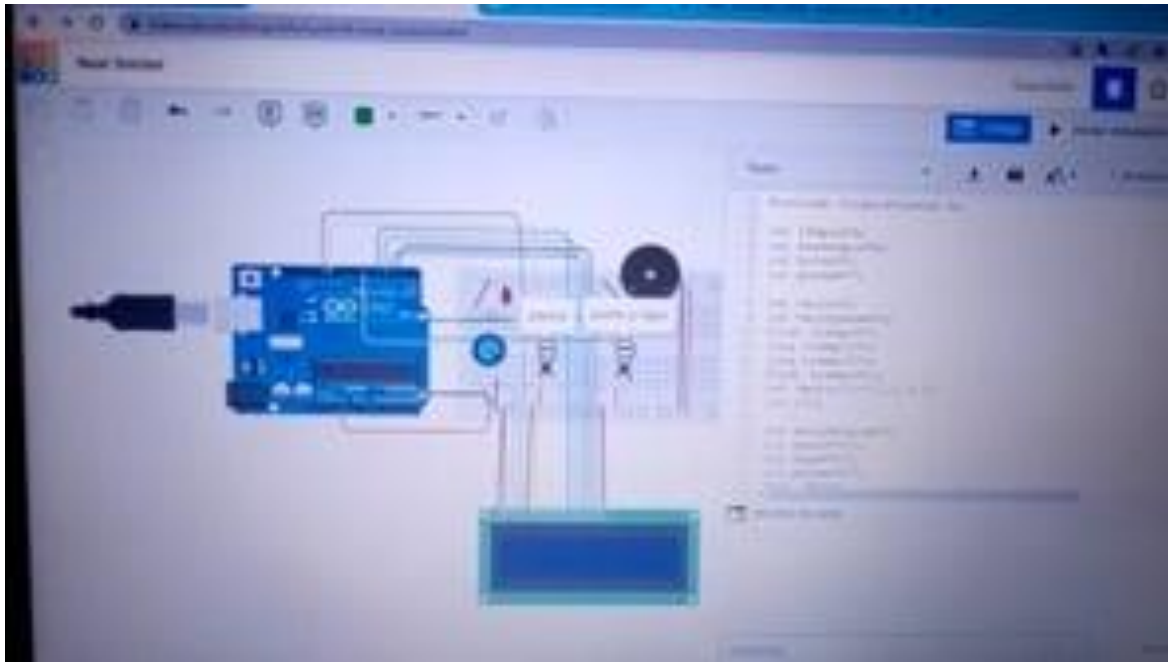
```
}
```

Demostración del circuito:

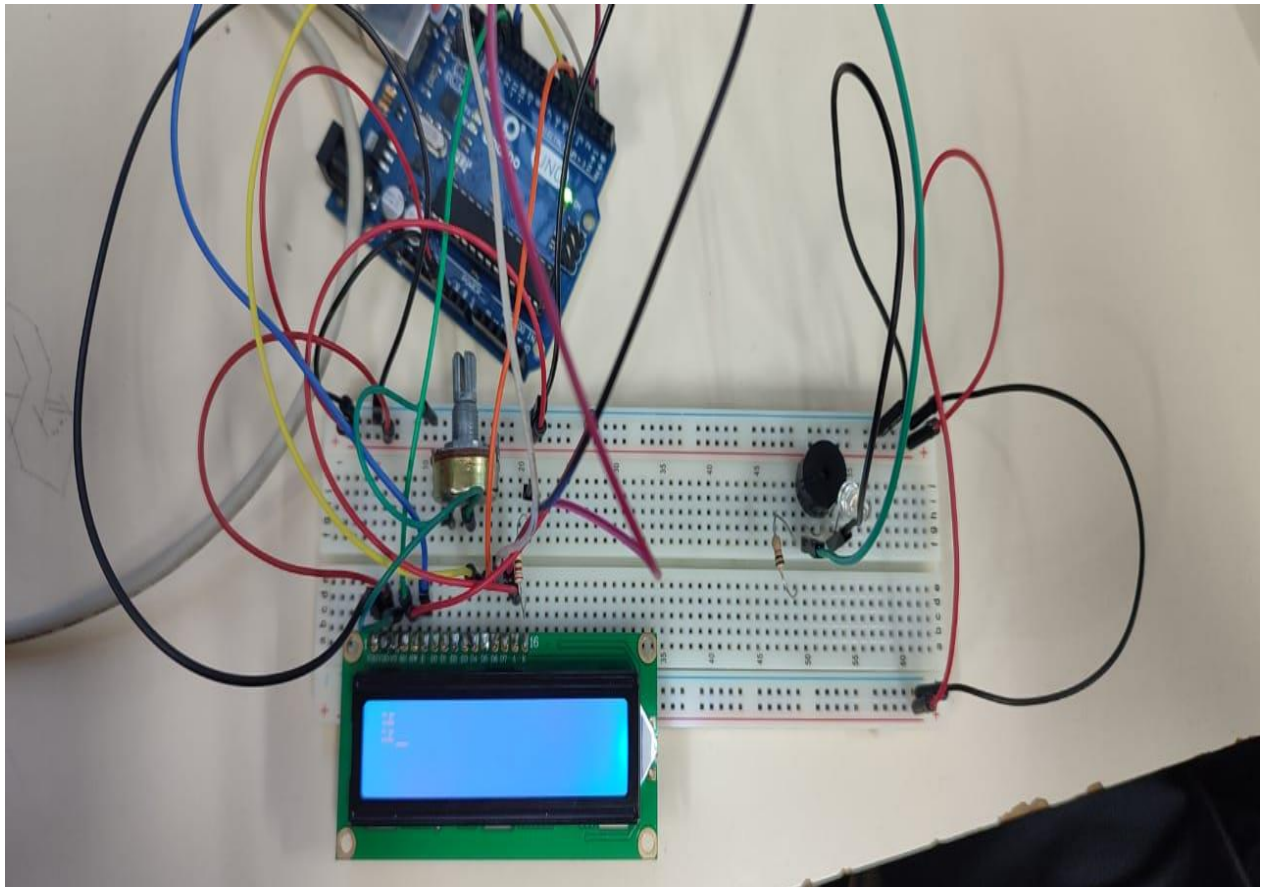


Video del circuito leyendo las letras por el serial:

[proyecto morse.mp4](#)



Demostración física del proyecto:



Biografía:

Enlaces de Youtube:

<https://www.youtube.com/watch?v=hTC7e5rX1tk>

<https://www.youtube.com/watch?v=lunnM6IPkyo>

<https://www.youtube.com/watch?v=osPw81rCbE4>

<https://www.youtube.com/watch?v=acZXhwxqUY>

<https://www.youtube.com/watch?v=plu52oniqTY>

<https://www.youtube.com/watch?v=dKFR9dqu4c>

Enlaces de páginas web:

<https://programarfacil.com/tutoriales/fragmentos/sos-en-arduino-codigo-morse/>

<https://www.instructables.com/Proyecto-C%C3%B3digo-Morse-con-Arduino-y-Android/>

<https://makinandovelez.wordpress.com/2019/02/26/practica-5-arduino-senal-sos-codigo-morse/>

<https://blog.bricogeek.com/noticias/arduino/como-hacer-un-receptor-de-codigo-morse-con-arduino/>

<https://programacionfacil.org/blog/mensaje-sos-con-un-led-en-arduino/?amp=1>

Principales Problemas:

- Al inicio se tuvo problemas de que al correr el programa d C++ no hacia la comunicación con el puerto serial al que estaba conectado al Arduino, la manera en que resolvimos esto fue juntar los códigos, tanto el de C++ como el de Arduino en un mismo fichero.

- Esto resolvió el problema y así ya se logró traducir los caracteres que introducimos mediante teclado a código morse.

Conclusiones:

- Con el desarrollo de este proyecto concluimos en que fue un excelente ejercicio para comenzar a conocer los componentes de los arduinos, así como el vasto número de aplicaciones que estos pueden llegar a tener y las soluciones que pueden llegar a tener y las soluciones que pueden llegar a brindar, desde algo muy sencillo como prender una luz hasta algo muy complejo como mover un motor.