

Carpeta de campo

Materia: proyecto informático

Alumno: Sebastián Alejandro de León

Grupo: Augusto Plazibat, Ariel Zacarias, Juan

Chambi Proyecto: traductor de código morse

proyecto beta:

El primer proyecto que se nos ideó fue el de una caja musical con una lista de canciones. Lo que hacía era elegir entre un listado de canciones y con una manivela hacías que suene la canción y dejar de girarla se detenía y si le volvías a dar continuaba donde quedo

investigación:

Al descubrir que era muy básico el proyecto original decidimos cambiarlo. Entonces el profesor nos mostró una lista de sugerencias y elegimos código morse en general. investigamos y decidimos hacer un arduino que se comuniquen con otro en código morse

Función del proyecto:

La función de este proyecto es que a partir de un botón el código reciba código morse y con ayuda de vectores lo traduzca a español y lo muestre en la pantalla lcd. Y si nos da el tiempo hacer que se comuniquen con otro arduino para poderse mandar mensajes en código morse entre ellos y traducirlos desde otro arduino.

Martes 19/9:

Abrí un tinkercad y en el código le defini el botón

```
#define botón 6
```

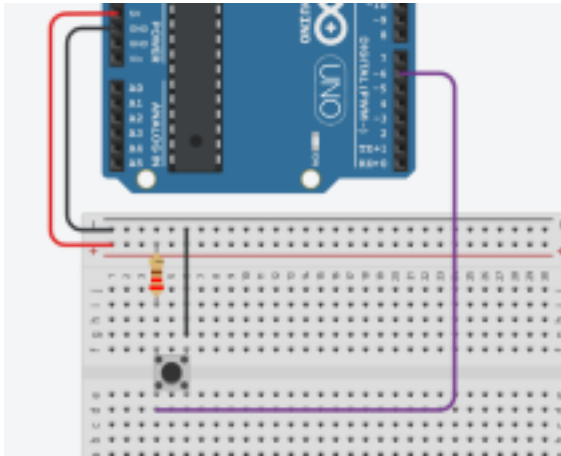
```
void setup()
```

```
{
```

```
  pinMode(boton, INPUT);
```

}

En el circuito conecte el botón



Cree el vector de letras:

```
char letras[]={0, "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K",  
"L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"}
```

el vector de los números:

```
char numeros[]={0, "1", "2", "3", "4", "5", "6", "7", "8", "9", "0"}
```

Googleamos la traduccion del codigo morse

Signo - Código :

A · —

B — · · ·

C — · — ·

D — · ·

E ·

F · · — ·

G — — ·

H · · · ·

I · ·

J · — — —

K — · —

L · — · ·

M — —

N — .
Ñ — — . — —
O — — —
P . — — .
Q — — . —
R . — .
S . . .
T —
U . . —
V . . . —
W . — — . . — — . —
X — . . — , — — . . — —
Y — . — — . .
Z — — . .

y con esa imagen cree el vector de las letras en morse:

```
char letras_morse[]={0,  
".-","...","-.","..",".",".-","_-","...","..","_--",".-","...","_--",".-","_--","_--",  
".-","_--",".-","..",".",".-","_--","_--","_--","_--","_--","_--","_--","_--","_--","_--"};
```

y el vector de los números en morse:

```
char numeros_morse={0,  
".----", "..---", "...--", "....-", ".....", "-....", "--...", "-.-.-", "-----"};
```

Creamos un documento de google para dejar ahi el codigo y poder que cada integrante pueda editarlo desde distintas computadoras

Cree la función “descubrimiento” y “traductor”

```
char descubrimiento(int valor)
```

(lo que hace es recibir las señales del botón y traducirlas a código morse)

```
char traductor(palabra morse)
```

(y este lo que hace es recibir el mensaje de código morse que

genera la función descubrimiento y lo traduce)

dentro de “descubrimiento” creamos los vectores “letra_morse” y “palabra_morse”:

letra_morse[]=

(es donde se va a guardar los puntos y guiones que generan una letra)

palabra_morse[]=

(es donde se va a guardar las letras guardadas en “letra_morse” para generar la palabra)

varias veces entre a los tps anteriores hechos en clase con el profesor para chequear como hacíamos algunas cosas

decidimos agregarle una pantalla lcd

para que muestre la traducción en el mismo arduino

Arduino físico hecho hoy:

componentes:

-cables (obviamente)

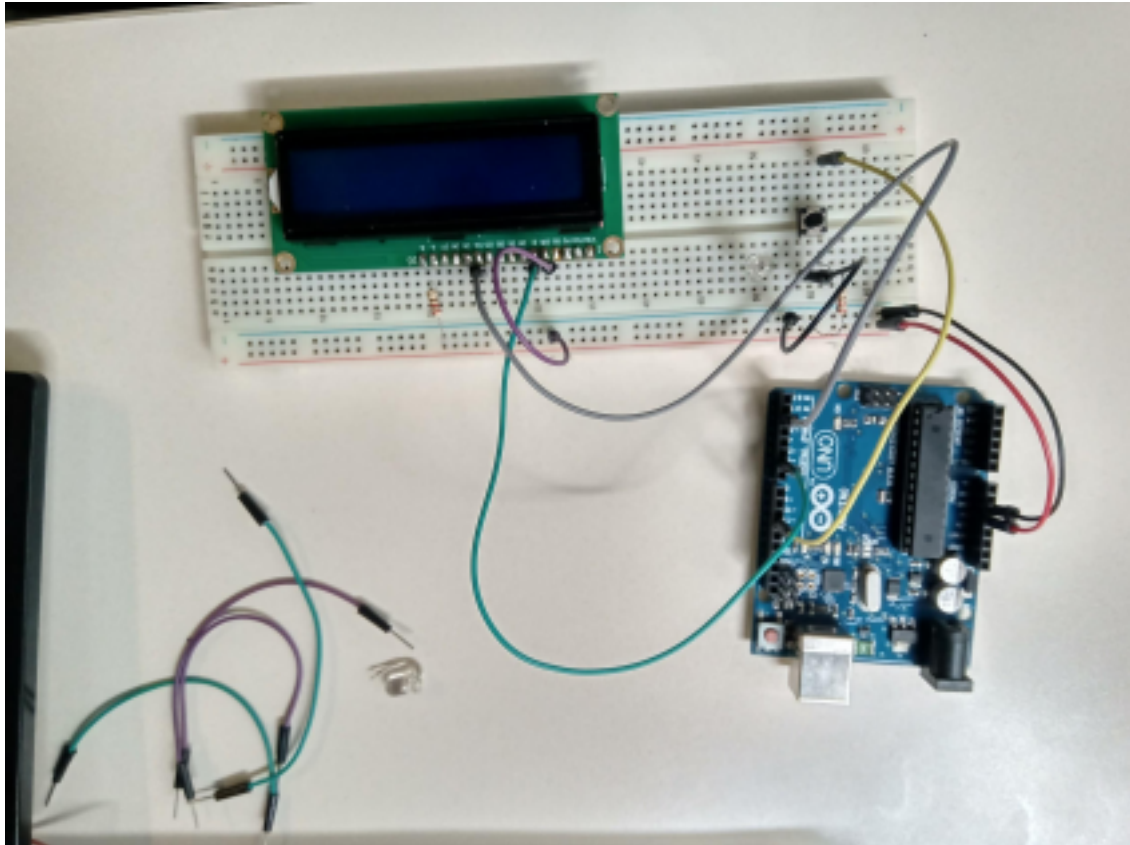
-arduino (obviamente)

-placa (obviamente)

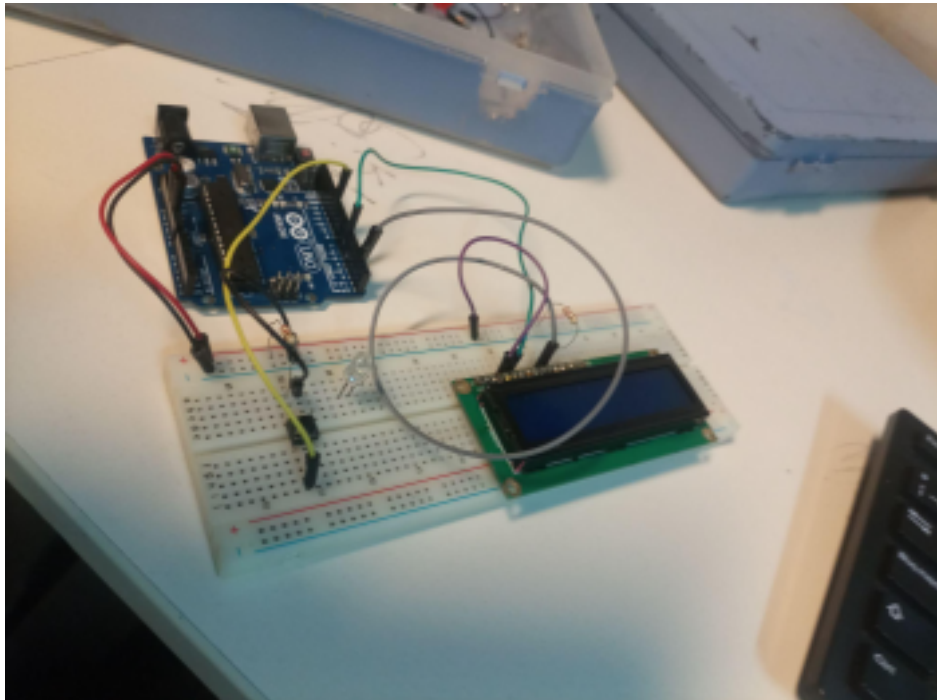
-botón

-pantalla lcd

(lo empezamos a armar pero nos dimos cuenta que no nos daba el tiempo y que los del resto de turnos lo iban a desarmar para hacer sus proyectos y en la próxima clase tendríamos que rearmarlo (osea que fue re al pedo))



Martes 26/9:



al final no lo desarmaron ❖❖

conecte el lcd en el simulador:

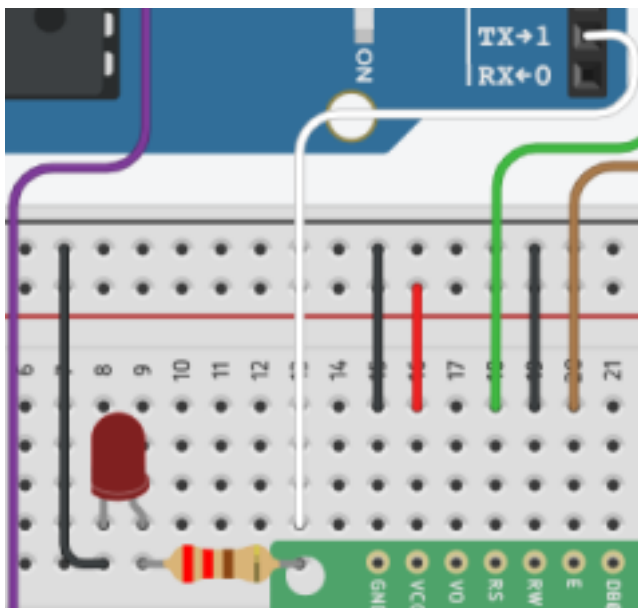
y lo defini

```
#include <LiquidCrystal.h>
void setup()
{
  lcd.createChar(6, si);
  lcd.begin(16 , 1);
}
```

respire

decidimos poner un led para demostrar cuando se toca el botón

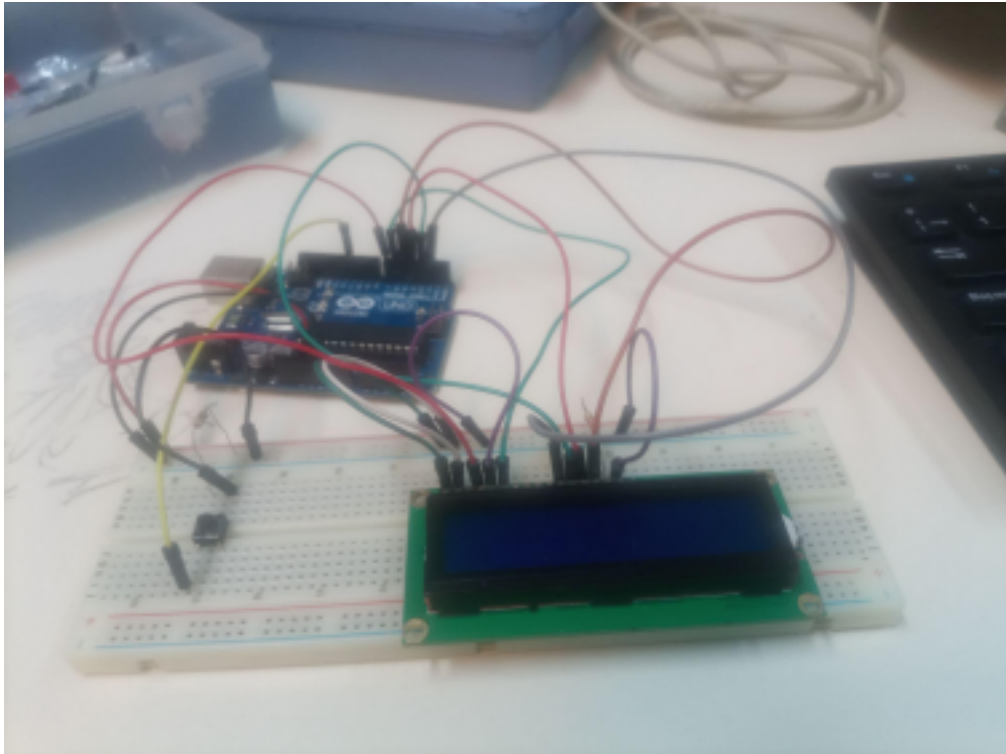
lo puse en el simulador



y lo defini en el código:

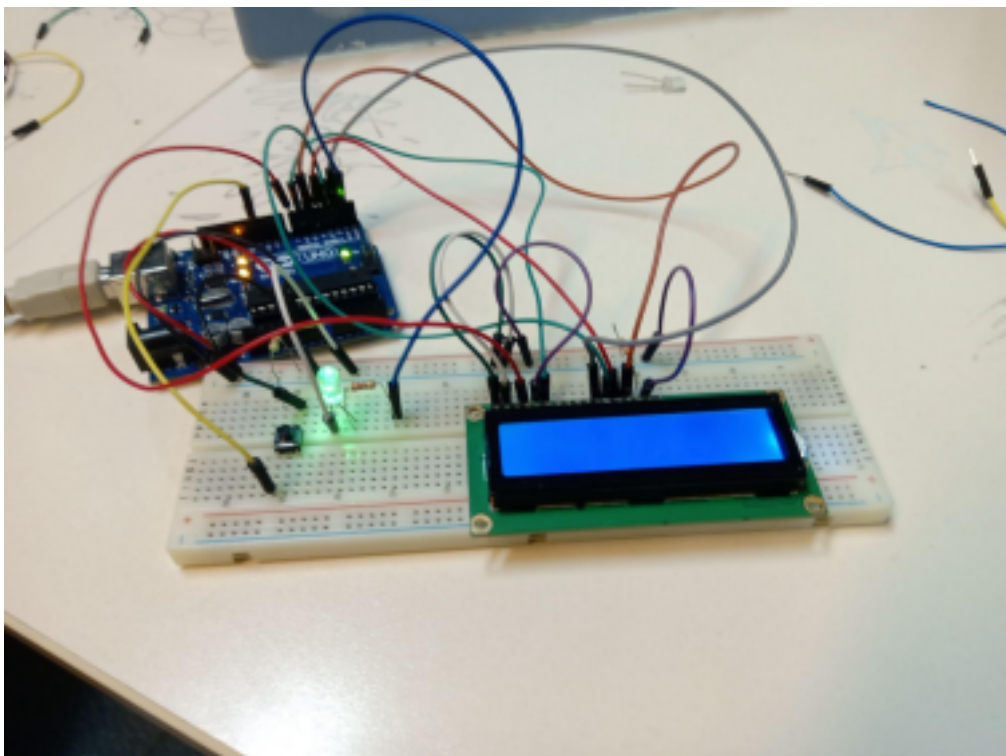
```
#define led 1
void setup()
{
  pinMode(led,OUTPUT);
}
```

termine de armar el arduino fisico:



me falto el led

le conecte el led y lo conectamos para ver si funciona



Investigación

busque en youtube videos de gente con el mismo proyecto para inspirarme y entender cómo continuar

TRADUCTOR A CODIGO MORSE REALIZADO EN ARDUIN...

Traductor Código Morse Arduino

aunque no nos sirvió porque nuestro objetivo es el opuesto al de los proyectos de ellos

le pregunté al profe y dijo que usemos "pulseIn"

investigamos y encontramos este video que explica como funciona: Funcion Arduino pulseIn()

puse en el código el pulseIn

float duracion

void loop:

```
{  
    duracion = pulseIn (11, HIGH, 3000000)  
    duracion_s=duracion/1000000  
    Serial.println(duracion_s)  
}
```

el código no funciona

el código demuestra distintas fallas y no entiendo nada

errores:

vectores

codigo lcd

arreglando errores:

vectores:

puse punto y coma después de todos los vectores

y meter el 0 dentro de comillas

```
String letras[]= {"0", "A ", " B ", " C ", " D ", "E", " F ", "G ", " H ", "I ", " J ",  
" K ", " L ", "M ", "N ", " O ", " P ", " Q ", " R ", " S ", "T", " U ", " V ", " W ", " X  
", " Y ", " Z "};
```

```
String letras_morse[]={"0",
```


codigo lcd:

Estoy haciendo la función descubrir:

mejore el pulseIn:

```
duracion = pulseIn (11, HIGH, 3000000)
```

```
duracion s=duracion/1000000
```

```
Serial.println(duracion_s)
```

y se lo puse a la funcion:

```
char descubrir(float duracion s)
```

Martes 3/10:

el otro grupo que usa nuestra caja desarmo nuestro proyecto ❖❖

intente correr el código y tiro errores:

arregle algunos errores:

descubrí que a varias variables le faltaba el “int” y “float”
también cambié el void loop para ver si ahora funciona
void loop()

```
{  
  duracion = pulseIn (11, HIGH, 3000000);  
  duracion_s = duracion/1000000;  
  Serial.println(duracion_s);  
  descubrir(duracion, posicion);  
  int pos = pos+1;  
  lcd.print(palabra);  
  lcd.setCursor( 3 , 1 );  
}
```

e hice funcionar la funcion descubrir:

Investigacion



tuvimos que cambiar la pantalla led por una con otro tipo de

conexión por recomendación del profe:

Martes 10/10

arreglamos errores del código

Ayude a el alumno álvaro ya que tenia errores y el profe estaba ocupado con otras personas

cambiamos todo el código  :

```
int boton = 11;
```

```
String letras[]={ "0",  
"A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R",  
,"S","T","U","V","W","X","Y","Z"};  
String letras_morse[]={ "0",  
".-","-...","-.-.", "-..", ".","..-","--","....","..",".---","-.-","-...","--","-.", "---","-.-.",  
", "-.-.", "-.-.", "...", "-","..-","...-",".-.", "-.-.", "-.-.", "-.-.", "-.-."};  
String numeros[]={ "0", " 1 ", " 2 ", " 3 ", " 4 ", " 5 ", " 6 ", " 7 ", " 8 ", " 9",  
", " 0 "};  
Sting numeros_morse={ "0",  
".----", "-.---", "-...-", "....-", ".....", "-....", "--...", "----.", "-----"};
```

```
int cant;
```

```
float micro;  
float seg;  
bool presion = true;
```

```
char codigo[5];
```

```
void setup()  
{  
  pinMode(boton, INPUT);  
  Serial.begin(9600);  
}
```

```
void loop()  
{  
  presion = true;  
  
  comienzo();  
  //Serial.println(presion);  
  if (!presion){  
    convertir_caracteres();  
  
  }  
  if (cant >= 1){  
    Serial.print("Código Morse recibido: ");  
    for (int i = 0; i < cant; i++) {  
      Serial.print(codigo[i]);  
    }  
    Serial.println();  
  }  
  for (int i = 0; i < 5; i++){  
    codigo[i]='0';  
  }  
  cant = 0;
```

```
}
```

```
void comienzo(){  
  micro = pulseIn(boton, LOW, 3000000);  
  seg = micro / 1000000;  
  if (seg > 0.03 && seg < 1.00){  
    codigo += '.';  
    presion = !presion;  
    cant ++;  
    Serial.println(codigo[0]);  
  
  }else if(seg > 0.300 && seg < 1.00){  
    codigo += '-';  
    presion = !presion;  
    cant ++;  
    Serial.println(codigo[0]);  
  
  }  
}
```

```
void convertir_caracteres(){  
  for (int i=1;i<5;i++){  
    micro = pulseIn(boton, LOW,  
    3000000); seg = micro / 1000000;  
    if (seg > 0.03 && seg < 0.300){  
      codigo[i]='.';  
      cant ++;  
      Serial.println(codigo[i]);  
      delay(50);  
    }else if(seg > 0.300 && seg < 1.00){  
      codigo[i]='-';  
      cant++;  
      Serial.println(codigo[i]);  
      delay(50);  
    }  
  }  
}
```

```

    }else {
        i=6;
    }

    presion = !presion;
}
}

```

arreglamos errores del nuevo código

```

String numeros_morse={"0",
".----", ".---", "...--", "....-", ".....", "-....", "--...", "---..", "----.", "-----"}
; esta mal escrito: se escribe String
codigo[i]='.';
codigo[i]='-';
esta mal: tiene que ir codigo[0]

```

Sacamos los vectores con números ya que no eran realmente necesarios al escribir una palabra normalmente.

Ayude nuevamente a Álvaro

agregue el código para la pantalla lcd (otra vez)

```

#include <LiquidCrystal.h>

```

```

//////////( rs , e , db4 , db5 , db6 , db7 )
LiquidCrystal lcd( 7 , 6 , 2 , 3 , 4 , 5 );
void setup()
{
    lcd.begin(16 , 2);
}

```

cree la función traducción:

```

void traduccion(){
    for(int i=0; i < L ;i++)
    {
        if (palabra_morse=letras_morse[i]){
            letra=letras[i]
            palabra+=letra
        }
    }
}

```

Investigacion

volvimos a investigar código morse en otra fuente y encontramos un poco mas de informacion y:
sumamos la Ñ al vector de letras

Ñ — — · — —

Martes 18/10

Este martes no pude laburar ya que mi vida no es fácil y no tuve la oportunidad de hacer aparición de mi presencia pretenciosa en este día de escuela/secundaria/lugaralquevoy/meobligan/escuela=prisión/etc. Y no estuve en mi casa para poder ayudar desde ahí (tampoco avanzaron mucho ellos). pero te juro que escuche una canción

esta es la letra:

*We're no strangers to love
 You know the rules and so do I (do I)
 A full commitment's what I'm thinking of
 You wouldn't get this from any other guy
 I just wanna tell you how I'm feeling*

*Gotta make you understand
Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you
We've known each other for so long
Your heart's been aching, but you're too shy to say it (say
it) Inside, we both know what's been going on (going on)
We know the game and we're gonna play it
And if you ask me how I'm feeling
Don't tell me you're too blind to see
Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you
Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you
We've known each other for so long
Your heart's been aching, but you're too shy to say it (to say
it) Inside, we both know what's been going on (going on)
We know the game and we're gonna play it
I just wanna tell you how I'm feeling
Gotta make you understand
Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you
Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye*

*Never gonna tell a lie and hurt you
Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you*

Martes 24/10

en este dia intentamos continuar pero no pudimos ya que hubo un error que no pudimos arreglar y el profe no estuvo para poder ayudarnos

así que como ultimo recurso le preguntamos a varias IAs y nos dieron muchas soluciones y cambios del código. Los probamos para poder avanzar pero ninguno funcionó:

```
1.
#include <LiquidCrystal.h>
#include <vector>

int boton = 11;
char letras[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U',
'V', 'W', 'X', 'Y', 'Z'};
String letras_morse[] = {".-", "-...", "-.-.", "-..", ".", "-.-.", "--.", "...", "..", ".---", "-.-", "-..", "--",
".", "---", ".--.", "--.", "-.", "...", "-", "-.-", "...-", "--", "-.-", "-.-", "-.."};
LiquidCrystal lcd(9, 8, 5, 4, 3, 2);
String letra = "";
int cant;
float micro;
float seg;
bool presion = true;
char codigo[5];
std::vector<char> lettersVector;
void setup()
{
    pinMode(boton, INPUT);
    Serial.begin(9600);
    lcd.begin(16, 2);
}

void loop()
```

```

{
    presion = true;
    comienzo();
    if (!presion)
    {
        convertir_caracteres();
    }
    if (cant >= 1)
    {
        for (int i = 0; i < cant; i++)
        {
            letra = letra + codigo[i];
        }
        for (int i = 0; i < 26; i++)
        {
            if (letra == letras_morse[i])
            {
                lettersVector.push_back(letras[i])
                ; }
            }
        }
        letra = "";
        for (int i = 0; i < 5; i++)
        {
            codigo[i] = '0';
        }
        cant = 0;

        lcd.setCursor(0, 0);
        lcd.print("Letters: ");
        lcd.setCursor(0, 1);
        for (int i = 0; i < lettersVector.size(); i++)
        {
            lcd.print(lettersVector[i]);
        }
    }
}

```

```

void comienzo()
{
    micro = pulseIn(boton, LOW, 3000000);
    seg = micro / 1000000;
    if (seg > 0.00 && seg < 0.30)
    {
        codigo[0] = '.';
        presion = !presion;
        cant++;
    }
}

```

```

    Serial.println(codigo[0]);
}
else if (seg > 0.30 && seg <
2.00) {
    codigo[0] = '-';
    presion = !presion;
    cant++;
    Serial.println(codigo[0]);
}
}

```

```

void convertir_caracteres()
{
    for (int i = 1; i < 5; i++)
    {
        micro = pulseIn(boton, LOW, 3000000);
        seg = micro / 1000000;
        if (seg > 0.00 && seg < 0.300)
        {
            codigo[i] = '-';
            cant++;
            Serial.println(codigo[i]);
        }
        else if (seg > 0.300 && seg <
2.00) {
            codigo[i] = '-';
            cant++;
            Serial.println(codigo[i]);
        }
        else
        {
            i = 6;
        }
    }

    presion = !presion;
}
}

```

2.

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

```

int boton = 11;
char letras[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U',
'V', 'W', 'X', 'Y', 'Z'};
String letras_morse[] = {".-", "-...", "-.-.", "-..", ".", "-.-.", "--.", "...", "..", "---", "-.-", "-..", "--",
"-.", "---", "-.-.", "-.-", "-..", "...", "-", "-.-", "-.-.", "-.-", "-.-."};
LiquidCrystal lcd(9, 8, 5, 4, 3, 2);

```

```

String letra = "";
int cant;
float micro;
float seg;
bool presion = true;
char codigo[5];
char lettersArray[26];
int lettersCount = 0;

void setup()
{
    pinMode(boton, INPUT);
    Serial.begin(9600);
    lcd.begin(16, 2);
}

void loop()
{
    presion = true;
    comienzo();
    if (!presion)
    {
        convertir_caracteres();
    }
    if (cant >= 1)
    {
        for (int i = 0; i < cant; i++)
        {
            letra = letra + codigo[i];
        }
        for (int i = 0; i < 26; i++)
        {
            if (letra == letras_morse[i])
            {
                lettersArray[lettersCount] = letras[i];
                lettersCount++;
            }
        }
    }
    letra = "";
    for (int i = 0; i < 5; i++)
    {
        codigo[i] = '0';
    }
    cant = 0;
}

```

```

lcd.setCursor(0, 0);
lcd.print("Letters: ");
lcd.setCursor(0, 1);
for (int i = 0; i < lettersCount; i++)
{
    lcd.print(lettersArray[i]);
}
}

```

```

void comienzo()
{
    micro = pulseIn(boton, LOW, 3000000);
    seg = micro / 1000000;
    if (seg > 0.00 && seg < 0.30)
    {
        codigo[0] = '.';
        presion = !presion;
        cant++;
        Serial.println(codigo[0]);
    }
    else if (seg > 0.30 && seg <
2.00) {
        codigo[0] = '-';
        presion = !presion;
        cant++;
        Serial.println(codigo[0]);
    }
}

```

```

void convertir_caracteres()
{
    for (int i = 1; i < 5; i++)
    {
        micro = pulseIn(boton, LOW, 3000000);
        seg = micro / 1000000;
        if (seg > 0.00 && seg < 0.300)
        {
            codigo[i] = '.';
            cant++;
            Serial.println(codigo[i]);
        }
        else if (seg > 0.300 && seg <
2.00) {
            codigo[i] = '-';
            cant++;
        }
    }
}

```

```

        Serial.println(codigo[i]);
    }
    else
    {
        i = 6;
    }

    presion = !presion;
}
}

```

3.

```

#include <LiquidCrystal.h>
#include <vector>

```

```

int boton = 11;
char letras[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U',
'V', 'W', 'X', 'Y', 'Z'};
String letras_morse[] = {".-", "...", "-.-", "-..", ".", "-.-", "--.", "...", "..", ".---", "-.-", "-..", "--",
"-.", "---", ".--.", "--.-", ".-.", "...", "-", ".-.", "...-", "--", "-.-", "-..", "-.-", "-.."};
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Ajusta los pines según tu conexión
String letra = "";
int cant;
float micro;
float seg;
bool presion = true;
char codigo[5];
std::vector<char> lettersVector;

```

```

void setup()
{
    pinMode(boton, INPUT);
    Serial.begin(9600);
    lcd.begin(16, 2); // Ajusta el número de columnas y filas según tu pantalla LCD
}

```

```

void loop()
{
    presion = true;
    comienzo();
    if (!presion)
    {
        convertir_caracteres();
    }
    if (cant >= 1)
    {

```

```

for (int i = 0; i < cant; i++)
{
    letra = letra + codigo[i];
}
for (int i = 0; i < 26; i++)
{
    if (letra == letras_morse[i])
    {
        lettersVector.push_back(letras[i]);
    }
}
}
letra = "";
for (int i = 0; i < 5; i++)
{
    codigo[i] = '0';
}
cant = 0;

lcd.setCursor(0, 0

```

4.

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

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

```
char letras[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U',
'V', 'W', 'X', 'Y', 'Z'};
```

```
String letras_morse[] = {".-", "-...", "-.-.", "-..", ".", "-.-.", "--.", "...", "..", ".---", "-.-", "-..", "--",
".", "---", "-.-", "--.", "-.", "...", "-", "-.-", "...", "--", "-.-", "-.-", "-.."};
```

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

```
void loop() {
    String codigo_morse = obtenerCodigoMorse(); // Obtener el código Morse ingresado
    String mensaje = traducirCodigoMorse(codigo_morse); // Traducir el código Morse a
    español
    mostrarMensaje(mensaje); // Mostrar el mensaje en el LCD
    delay(2000); // Esperar 2 segundos antes de volver a traducir
}
```

```
String obtenerCodigoMorse() {
    String codigo = "";
```

// Aquí puedes implementar la lógica para obtener el código Morse, ya sea a través de un botón o cualquier otro método de entrada.

// Por simplicidad, asumiremos que el código Morse se ingresa como una cadena de caracteres.

// Puedes modificar esta función según tus necesidades.

```
codigo = "-.-. --- -- . / -- --- .- ... . / .- / -.-. --- -.. / -- --- .- ... .";  
return codigo;  
}
```

```
String traducirCodigoMorse(String codigo_morse) {
```

```
    String mensaje = "";
```

```
    int longitud = codigo_morse.length();
```

```
    String letra = "";
```

```
    for (int i = 0; i < longitud; i++) {  
        char caracter = codigo_morse.charAt(i);
```

```
        if (caracter == ' ' || i == longitud - 1) {
```

// Si encontramos un espacio o llegamos al final de la cadena, traducimos la letra Morse a español

```
        for (int j = 0; j < 26; j++) {  
            if (letra == letras_morse[j]) {  
                mensaje += letras[j];  
                break;  
            }  
        }
```

```
    }
```

```
    letra = ""; // Reiniciamos la variable letra para la siguiente letra Morse
```

```
    } else {  
        letra += caracter;  
    }  
}
```

```
return mensaje;
```

```
}
```

```
void mostrarMensaje(String mensaje) {
```

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

```
    lcd.setCursor(0, 0);
```

```
    lcd.print("Mensaje:");
```

```
    lcd.setCursor(0, 1);
```

```
    lcd.print(mensaje);
```

```
}
```

5.

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



```

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int boton = 6; // Pin del botón

char letras[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U',
'V', 'W', 'X', 'Y', 'Z'};
String letras_morse[] = {".-", "-...", "-.-.", "-..", ".", "-.-.", "--.", "...", ".---", "-.-", "-..", "--",
".", "---", "-.-.", "--.-", "-.", "...", "-", "-.-", "-.-.", "-.-", "-.-.", "-.-."};

void setup() {
  lcd.begin(16, 2);
  pinMode(boton, INPUT_PULLUP); // Configurar el pin del botón como entrada con
  resistencia pull-up interna
}

void loop() {
  if (digitalRead(boton) == LOW) { // Verificar si se ha presionado el botón String
  codigo_morse = obtenerCodigoMorse(); // Obtener el código Morse ingresado String
  mensaje = traducirCodigoMorse(codigo_morse); // Traducir el código Morse a español
  mostrarMensaje(mensaje); // Mostrar el mensaje en el LCD
  delay(2000); // Esperar 2 segundos antes de volver a traducir
  }
}

String obtenerCodigoMorse() {
  String codigo = "";
  // Aquí puedes implementar la lógica para obtener el código Morse a través del botón.
  // Puedes utilizar la función digitalRead() para leer el estado del botón. // Por
  simplicidad, asumiremos que el código Morse se ingresa como una cadena de
  caracteres.
  codigo = "-.-. --- -- . / -- --- .- ... . / .- / -.-. --- -.. . / -- --- .- ... .";
  return codigo;
}

String traducirCodigoMorse(String codigo_morse) {
  // El código de traducción se mantiene igual
  // ...
}

void mostrarMensaje(String mensaje) {
  // El código para mostrar el mensaje en el LCD se mantiene igual
  // ...
}

```

6.

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

```

int boton = 2; // Pin del botón
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Configuración del LCD

char letras[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U',
'V', 'W', 'X', 'Y', 'Z'};
String letras_morse[] = {".-", "-...", "-.-.", "-..", ".", "-.-.", "--.", "...", "--", ".---", "-.-", "-..", "--",
".", "---", ".--", "-.-", "-.", "...", "-", "-.-", "-.-", "-.-", "-.-", "-.-", "-.-"},

String codigo = "";
String traduccion = "";

void setup() {
  pinMode(boton, INPUT_PULLUP);
  lcd.begin(16, 2);
  lcd.print("Ingrese el codigo");
}

void loop() {
  if (digitalRead(boton) == LOW) {
    delay(50); // Debounce
    if (digitalRead(boton) == LOW) {
      codigo += "-";
      lcd.setCursor(0, 1);
      lcd.print(codigo);
      delay(500);
    }
  } else {
    delay(50); // Debounce
    if (digitalRead(boton) == HIGH) {
      codigo += ".";
      lcd.setCursor(0, 1);
      lcd.print(codigo);
      delay(500);
    }
  }
}

if (codigo.length() >= 5) {
  traduccion = traducirCodigoMorse(codigo);
  lcd.clear();
  lcd.print("Traduccion:");
  lcd.setCursor(0, 1);
  lcd.print(traduccion);
  delay(2000);
  codigo = "";
  traduccion = "";
}

```

```
lcd.clear();
lcd.print("Ingrese el codigo");
}
}
```

```
String traducirCodigoMorse(String codigo) {
    for (int i = 0; i < 26; i++) {
        if (codigo == letras_morse[i]) {
            return String(letras[i]);
        }
    }
    return "No encontrado";
}
```

Martes 31/10

estuvimos trabados ya que el código no funcionaba y otra vez las IAs no daban soluciones

al final llamamos al profe y nos ayudo un poco con el código:
arreglar errores pequeños

nos ayudó probando muchas cosas a agregar un pedazo de código que hace que cada vez que se apriete el botón se encienda una led (pero por el pulseln solo se puede leer cuando se termina de apretar)

```
void comienzo(){  
    micro = pulseIn(boton, LOW, 3000000);  
    digitalWrite(led, HIGH);  
    delay(20);  
    digitalWrite(led, LOW);  
}
```

Martes 7/11

probamos mas codigos:

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

char

```
letras[]={'A','B','C','D','E','F','G','H','I','J','K','L','M','N','Ñ','O','P','Q','R','S','T','U','V','W','X','Y','Z'}
; String
```

```
letras_morse=[".-","-...","-.","..","...", "-.--","....","..","---","-.",".-.", "--","-.","--.", "---",".-.", "
```

```
String letra="";
int boton = 9 ;
int cant;
float micro;
float seg;
bool presion = true;
char codigo[5];
LiquidCrystal lcd(7,6,2,3,4,5);

void setup()
{
  pinMode(boton, INPUT);
  Serial.begin(9600);
  lcd.begin(16,2);
}

void loop()
{
  presion = true;
  comienzo();
  //Serial.println(presion);
  if (!presion){
    convertir_caracteres();
  }
  if (cant >= 1){
    Serial.print("Código Morse recibido: ");
    for (int i = 0; i < cant; i++) {
      letra = letra+codigo[i];
    }
    Serial.println(letra);
    for (int i=0;i<27;i++){
      if (letra == letras_morse[i]){
        lcd.setCursor(0,0);
        lcd.print(letras[i]);
        Serial.println(letras[i]);
        delay(300);
      }
    }
  }
  letra = "";
  for (int i = 0; i < 5; i++){
    codigo[i]='0';
  }
  cant = 0;
}
```



```

numeros_morse={"-----","..---","...--","....-",".....","-....","--...","---..","----.", "-----"};
String letra="";
int boton = 11 ;
long tiempo;
int cant;
float micro;
float seg;
bool presion = true;
char codigo[5];

```

```

LiquidCrystal_I2C lcd(0x27,16,2);

```

```

void setup()

```

```

{
  pinMode(boton, INPUT);
  lcd.init();
  lcd.backlight();
  Serial.begin(9600);
  lcd.begin(16,2);
}

```

```

void loop()

```

```

{
  presion = true;
  comienzo();
  //Serial.println(presion);
  if (!presion){
    convertir_caracteres();
  }
  if (cant >= 1){
    Serial.print("Código Morse recibido:
    "); for (int i = 0; i < cant; i++) {
      letra = letra+codigo[i];
    }
    Serial.println(letra);
    for (int i=0;i<27;i++){
      if (letra == letras_morse[i]){
        lcd.setCursor(0,0);
        lcd.print(letras[i]);
        Serial.println(letras[i]);

        lcd.setCursor(0,1);
        lcd.print(letra);
      }
    }
  }
  letra = "";
}

```

```

    for (int i = 0; i < 5; i++){
        codigo[i]='0';
    }
    cant = 0;
}

void comienzo(){
    micro = pulseIn(boton, LOW);
    seg = micro / 1000000;
    if (seg > 0.00 && seg < 0.20){
        codigo[0]='.';
        presion = !presion;
        cant ++;
        Serial.println(codigo[0]);
    }else if(seg > 0.20){
        codigo[0]='-';
        presion = !presion;
        cant ++;
        Serial.println(codigo[0]);
    }
}

void convertir_caracteres(){
    for (int i=1; i<5; i++){
        micro = pulseIn(boton, LOW);
        seg = micro / 1000000;
        if (seg > 0.00 && seg < 0.200){
            codigo[i]='.';
            cant ++;
            Serial.println(codigo[i]);
        }else if(seg > 0.200 ){
            codigo[i]='-';
            cant++;
            Serial.println(codigo[i]);
        }else {
            i=6;
        }
        presion = !presion;
    }
}

```

Y también lo armamos físico con la ayuda del profe:

En teoría ya está terminado

Martes 14/11

llamamos a l profe y terminamos el proyecto: - ajustamos bien los tiempos
- arreglamos detalles

código:

agregamos

String palabra;

int tem;

int coutPalabra =0;

bool toque = true;

void loop

for

if

palabra = palabra + letras[i];

if(coutPalabra >= 8 || tem == 8)

{

palabra = "";

coutPalabra=0;

toque = true;

tem = 0;

}

letra = "";

for (int i = 0; i < 5; i++){

codigo[i]='0';

}

cant = 0;

//Serial.print("Frase:");

//Serial.println(palabra);

lcd.setCursor(0,1);

lcd.print(palabra);

void comienzo(){

Serial.println(toque);

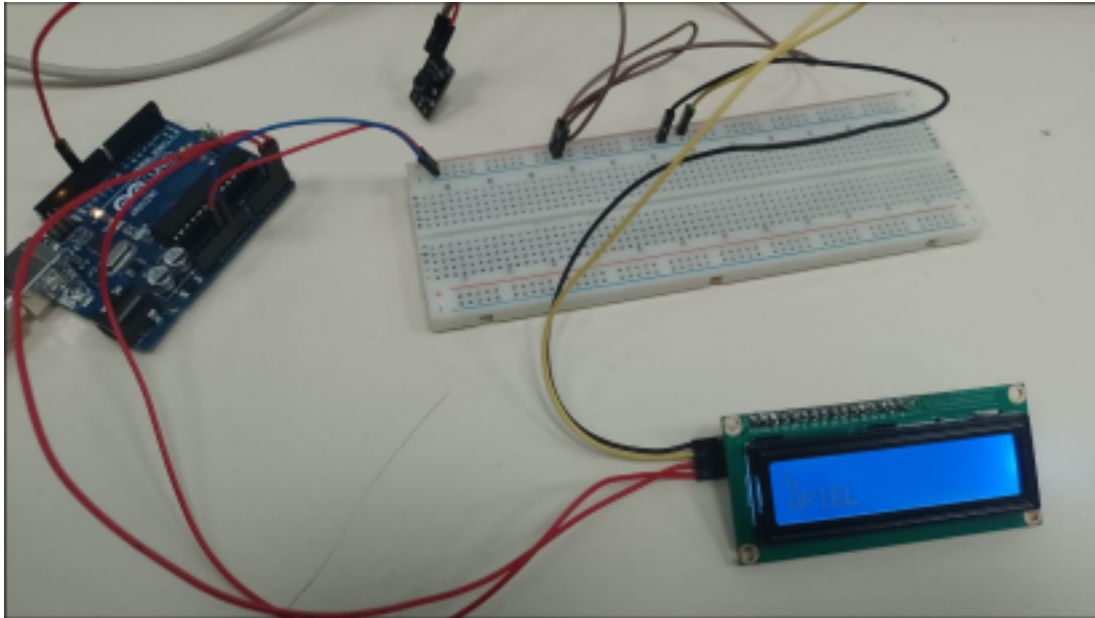
if (!toque){

tem ++;


```
}  
Serial.println("Inicio");
```

```
coutPalabra++;
```

fisico:



Escribi mi nombre ❓❓

