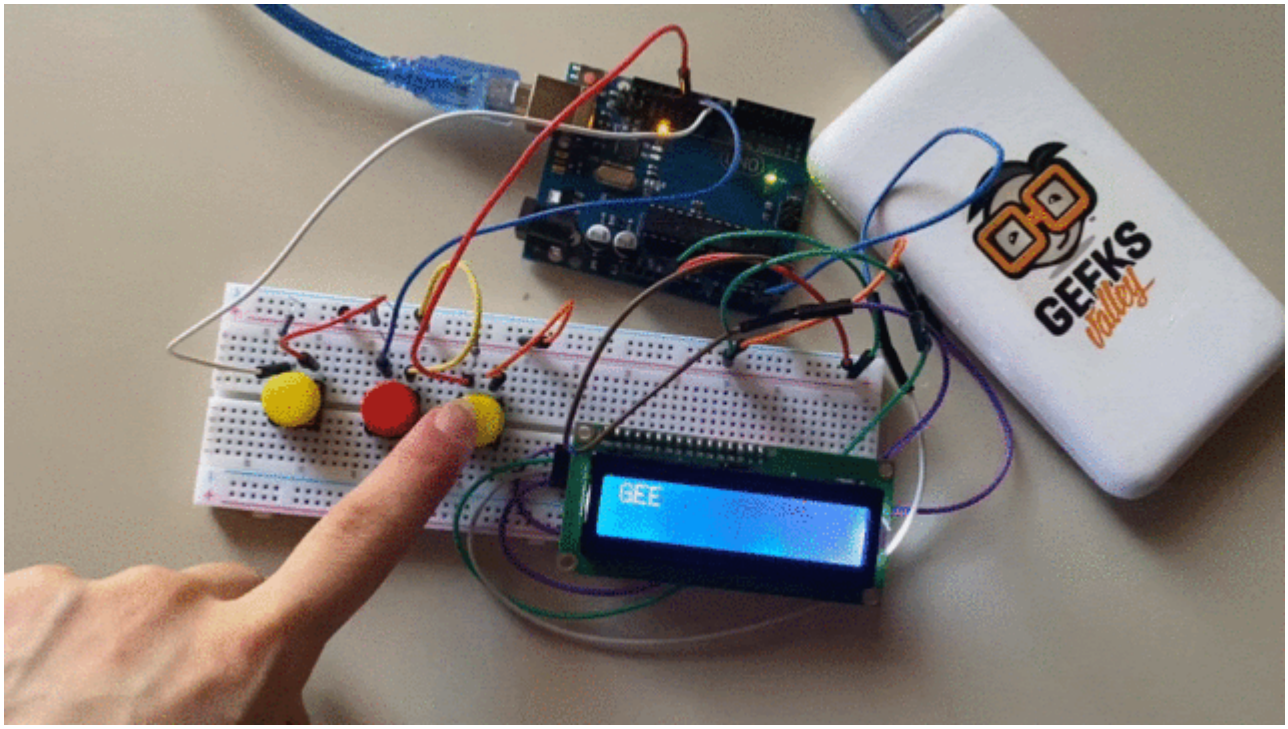
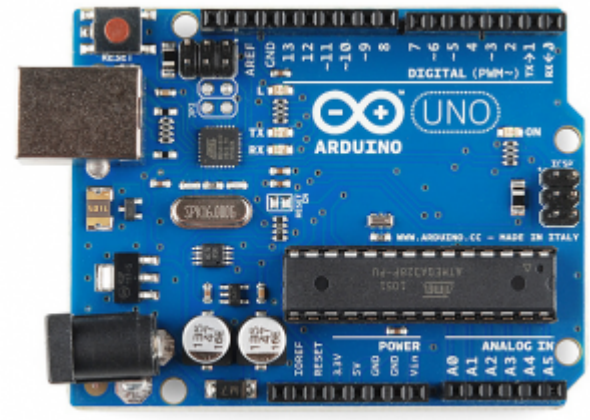
# ترجمة شفرة مورس على الشاشة باستخدام الاردوينو

## مقدمة

تستخدم شفرة مورس قديمًا للتواصل بطريقة مشفرة، خاصة في الشؤون العسكرية والبرقيات الدولية ومحطات الراديو، في هذا الدرس ستتعلم إدخال الرموز المستخدمة في شفرة مورس عن طريق الأزرار وطباعتها على الشاشة الكرستالية باستخدام الاردوينو.

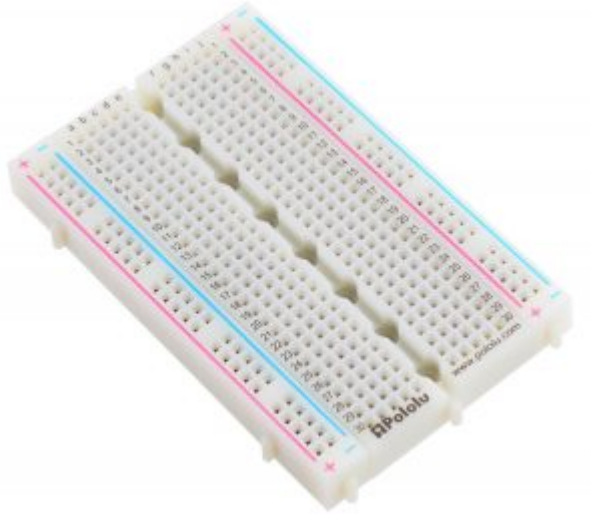



## المواد والأدوات



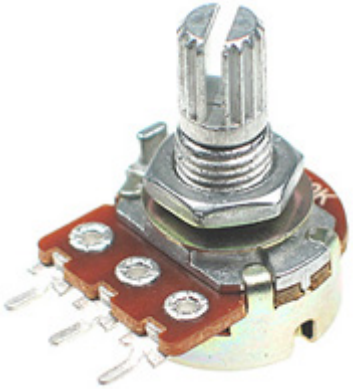
1× اردوينو اونو

1× سلك الاردوينو



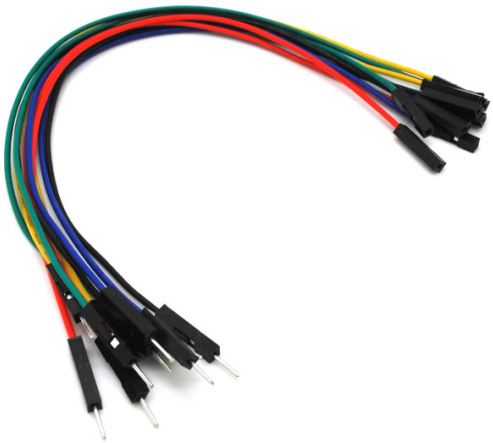
1× لوحة تجارب – حجم كبير

1× شاشة كرستالية (LCD 2×16 )



1× مقاومة متغيرة

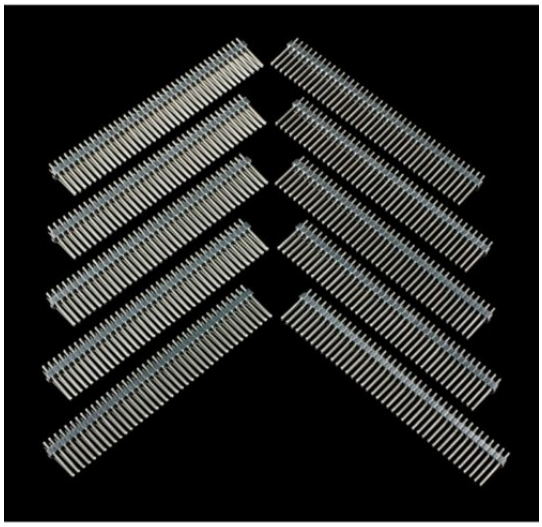
حزمة أسلاك توصيل (ذكر- ذكر)



حزمة أسلاك توصيل (ذكر – أنثى)

3× مقاومة 220 Ω

3× أزرار

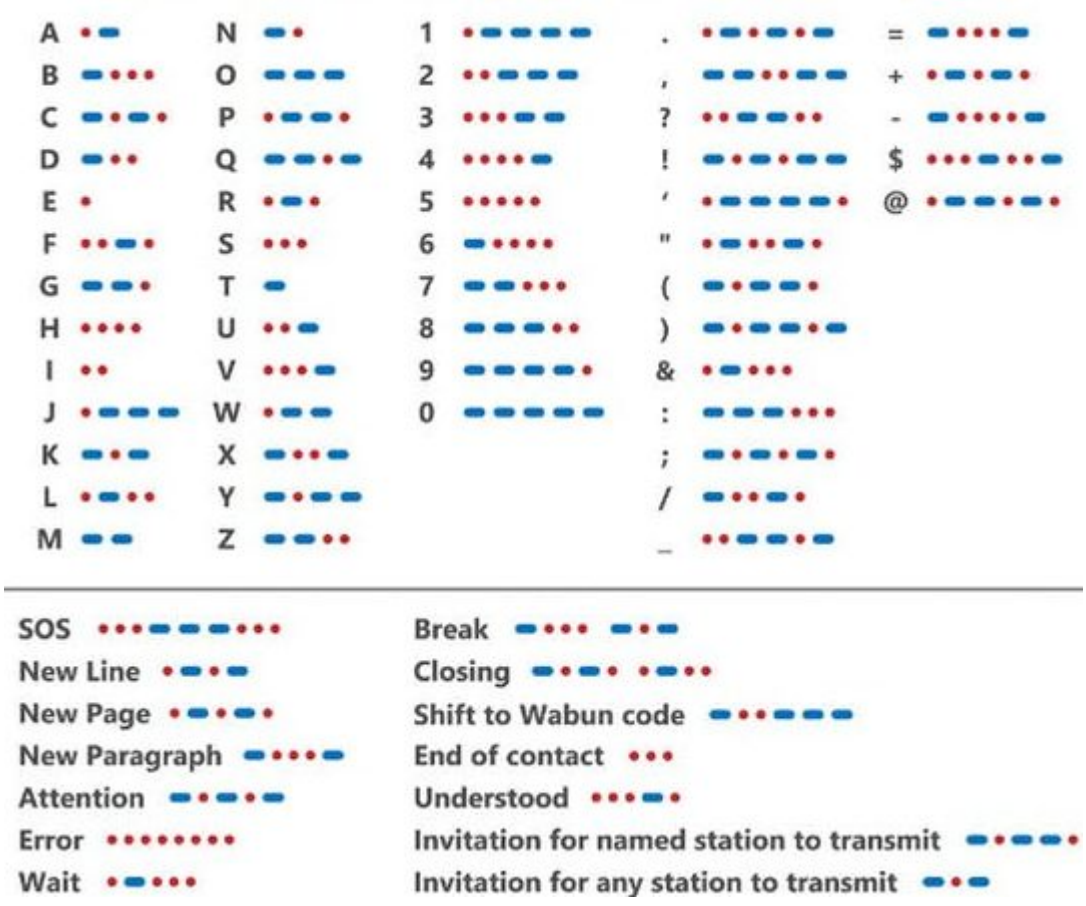


1× 40 رأس دبوس

## شفرة مورس

هي نوع من الشفرات التي تستخدم في إرسال المعلومات التلغرافية باستخدام العناصر المتتابعة المكونة من النقطة والشرطة.

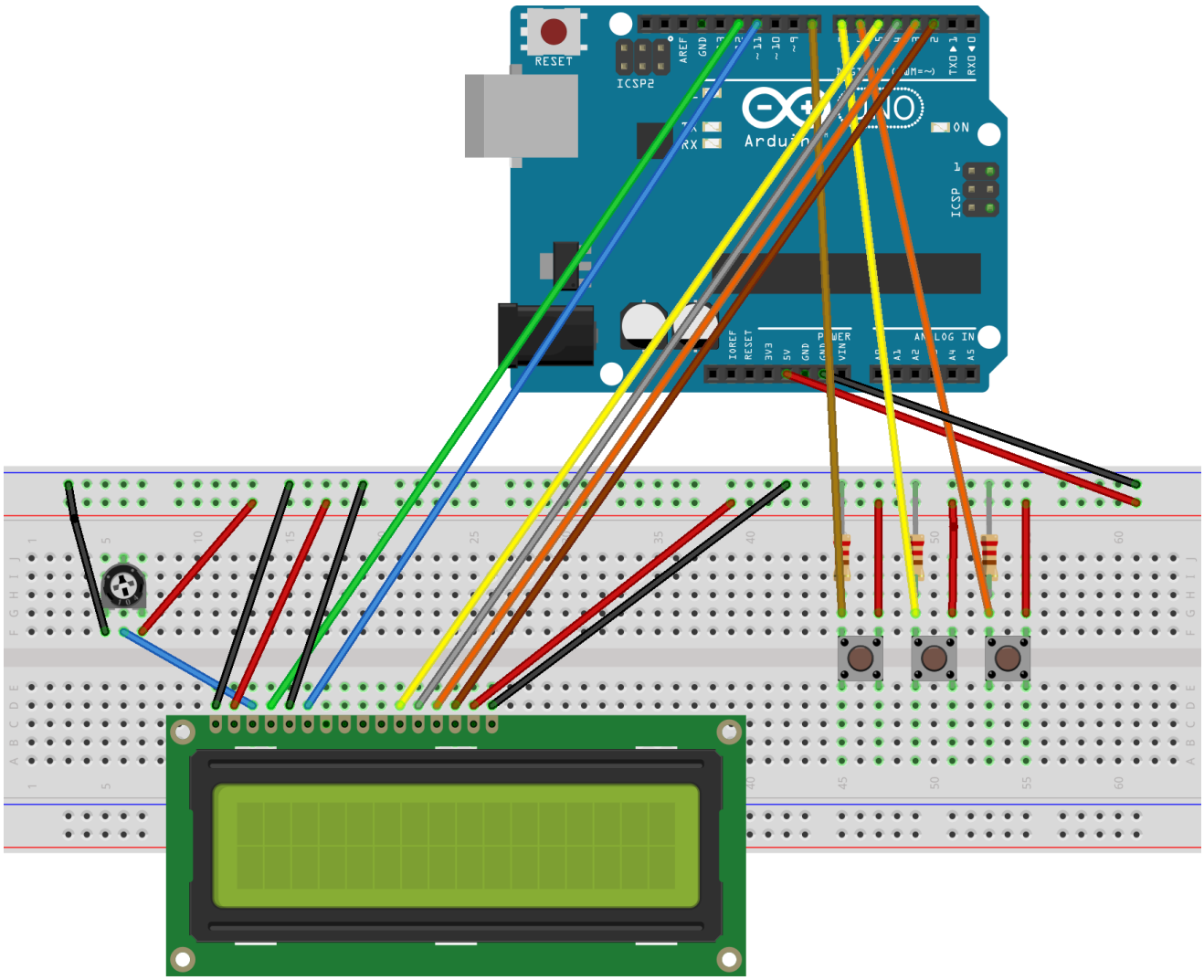
وتتم ترجمة شفرة مورس إلى حروف وأرقام ورموز أخرى.



## توصيل الدائرة

لمعرفة المزيد حول الشاشة الكرستالية يمكنك الرجوع للدرس التحكم بالشاشة الكرستالية LCD

لابد من تلحيم المنافذ مع الشاشة الكرستالية، للمزيد حول اللحام يمكنك الرجوع للدرس تعلم كيفية التلحيم – تلحيم القطع باللوحة الإلكترونية




## الكود البرمجي

```
#include <LiquidCrystal.h>
#define BUTTON1PIN 6
#define BUTTON2PIN 8
#define BUTTON3PIN 7
#define DISPLAY_NUMOFCOLUMNS 16
int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int Button1State = 0;
int Button1LastState = 0;
int Button2State = 0;
int Button2LastState = 0;
int Button3State = 0;
```

```
int Button3LastState = 0;
String tonesBuffer;
String text;
String expectedText;
String symbolsAlphabet[][2] =
{
{ ".-","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" },
{ ".----","1" },
{ "..---","2" },
{ "...--","3" },
{ "....-","4" },
{ ".....","5" },
{ "-....","6" },
{ "--...","7" },
{ "---..","8" },
{ "----.","9" },
{ "-----","0"}
};
char getToneFromButtonStates()
{
if (!Button1State && Button1LastState)
return '-';
if (!Button2State && Button2LastState)
return '.';
if (!Button3State && Button3LastState)
return ' ';
return (char)0;
}
```

```
char getSymbolFromBuffer()
{
if (tonesBuffer == "")
return ' ';
for (int i = 0; i < sizeof symbolsAlphabet / sizeof symbolsAlphabet[0]; i++)
if (tonesBuffer == symbolsAlphabet[i][0])
return symbolsAlphabet[i][1][0];
return (char)0; }
 void extractActionFromTonesBuffer() {
 if (tonesBuffer == ".....")
 text.remove(text.length() - 1, 1);
 if (tonesBuffer == "------") text = ""; }
void setup()
 { lcd.clear();
 lcd.begin(16,2);
 lcd.print("Morse");
 lcd.setCursor(0, 1);
lcd.print("Code");
 pinMode(BUTTON1PIN, INPUT);
pinMode(BUTTON2PIN, INPUT);
pinMode(BUTTON3PIN, INPUT); }
 void loop() {
Button1State = digitalRead(BUTTON1PIN);
 Button2State = digitalRead(BUTTON2PIN);
 Button3State = digitalRead(BUTTON3PIN);
 char tone = getToneFromButtonStates();
if (tone != (char)0)
{ if (tone == ' ')
{ char symbol = getSymbolFromBuffer();
if (symbol != (char)0) { text += symbol;
if (text.length() > DISPLAY_NUMOFCOLUMNS) {
 text = (String)symbol; }
 } else { extractActionFromTonesBuffer(); } tonesBuffer = "";
 } else {
 tonesBuffer += tone;
 if (tonesBuffer.length() > DISPLAY_NUMOFCOLUMNS)
{ tonesBuffer = (String)tone;
}
} lcd.clear();
 lcd.print(text);
 lcd.setCursor(0, 1);
lcd.print(tonesBuffer); }
 Button1LastState = Button1State;
Button2LastState = Button2State;
Button3LastState = Button3State; }
```

## شرح الكود البرمجي

هذا السطر يستدعي مكتبة الشاشة الكرستالية.

نستطيع تحميلها بتتبع المسار التالي:

Sketch > Include libraries > Manage libraries

ثم نكتب بخانة البحث Liquid crystal by Arduino

ثم نضغط على Install.

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

هذه الأسطر توضح منافذ الاردوينو التي ستستخدمها للربط في هذا المشروع.

```
#define BUTTON1PIN 6
#define BUTTON2PIN 8
#define BUTTON3PIN 7
#define DISPLAY_NUMOFCOLUMNS 16
```

في هذه الأسطر قمنا بإنشاء متغيرات لتخزين الحالة الحالية والسابقة للأزرار؛ لضمان كتابة الضغطة مرة واحدة فقط عند الضغط على الزر.

```
int Button1State = 0;
int Button1LastState = 0;
int Button2State = 0;
int Button2LastState = 0;
int Button3State = 0;
int Button3LastState = 0;
```

أيضًا أنشئنا متغيرات لتخزين التسلسل الحالي للضغطات والرمز التابع لكل ضغطة، التسلسل الحالي إما نقطة أو شرطة.

ترجمة الرموز في شفرة مورس ستكون إما رقم أو حرف أو رموز خاصة.

```
String tonesBuffer;
String text;
String expectedText;
```

ومتغير symbolsAlphabet من نوع string تمت تهيئته ويحتوي على ترجمة الرموز وتم تخزينها على شكل مصفوفة ثنائية الأبعاد.

```
String symbolsAlphabet[][2] =
{
{ ".-","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" },
{ ".----","1" },
{ "..---","2" },
{ "...--","3" },
{ "....-","4" },
{ ".....","5" },
{ "-....","6" },
{ "--...","7" },
{ "---..","8" },
{ "----.","9" },
{ "-----","0"}
};
```

في دالة ()setup، التي ستبدأ عند بدء التشغيل، تتم تهيئة الشاشة وتشغيل الإضاءة الخلفية وطباعة النص "Morse code" هنا علينا تعريف المنافذ إلى مدخلات للاتصال الرقمي باستخدام الأزرار.

```
void setup() {
clear.lcd();
lcd.begin(16,2);
lcd.print("Morse");
lcd.setCursor(0, 1);
lcd.print("Code");
pinMode(BUTTON1PIN, INPUT);
pinMode(BUTTON2PIN, INPUT);
pinMode(BUTTON3PIN, INPUT);
}
```

تبدأ دالة ()loop بقراءة حالات الأزرار واستدعاء دالة ()getToneFromButtonStates.

```
void loop() {
Button1State = digitalRead(BUTTON1PIN);
Button2State = digitalRead(BUTTON2PIN);
Button3State = digitalRead(BUTTON3PIN);
char tone = getToneFromButtonStates();
```

يتم تحليل الضغطة هنا، إذا كانت لا تحمل قيم بوضع Low (لم يضغط المستخدم على الزر) فلن يحدث شيء.

إذا كانت تحمل قيم (ضغط المستخدم على أي زر)، يأتي تحليل آخر، يتفرع البرنامج إلى قسم "تم الضغط على النقطة" وقسم " "تم الضغط على الشرطة".

```
if (tone != (char)0) {
        if (tone == ' ')
        {
        }
        else
       {
        }
        lcd.clear();
        lcd.print(text);
        lcd.setCursor(0, 1);
        lcd.print(tonesBuffer);
}
Button1LastState = Button1State;
Button2LastState = Button2State;
Button3LastState = Button3State;
```

تعمل الدالة getSymbolFromBuffer() على إرجاع القراءات التي تم الحصول عليها من الازرار إذا ضغط المستخدم الزر الذي بالمنتصف سيتم إنهاء المخزن المؤقت وترجمة القيم.

إذا كان المخزن المؤقت فارغًا وتم الضغط على الزر الذي بالمنتصف فستفسر دالة getSymbolFromBuffer() بأنه لا يوجد مخرجات لطباعتها على الشاشة.

إذا لم يكن المخزن المؤقت فارغًا، يتم البحث عن ترجمة الرموز المدخلة ويتم إرجاع ترجمة الرمز الذي تم العثور عليه.

إذا كان هناك رمز ولم يتم العثور على ترجمة الرمز، فسيتم إرجاع قيمة فارغة.

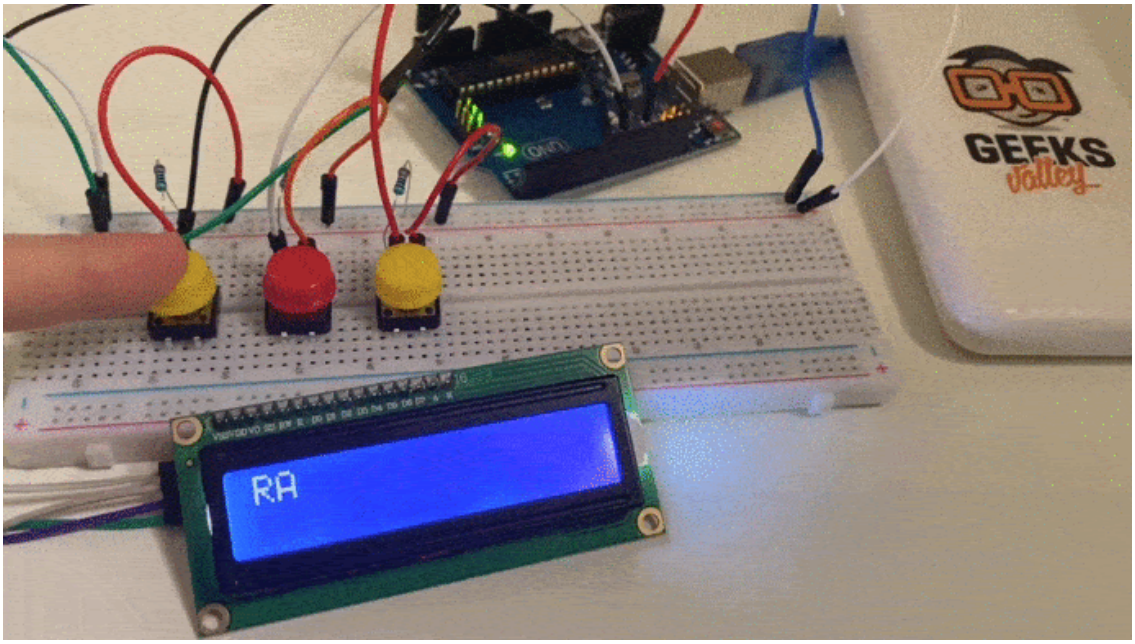
```
char getSymbolFromBuffer()
{
if (tonesBuffer == "")
return ' ';
for (int i = 0; i < sizeof symbolsAlphabet / sizeof symbolsAlphabet[0]; i++)
if (tonesBuffer == symbolsAlphabet[0])
return symbolsAlphabet[1][0];
return (char)0;
}
```

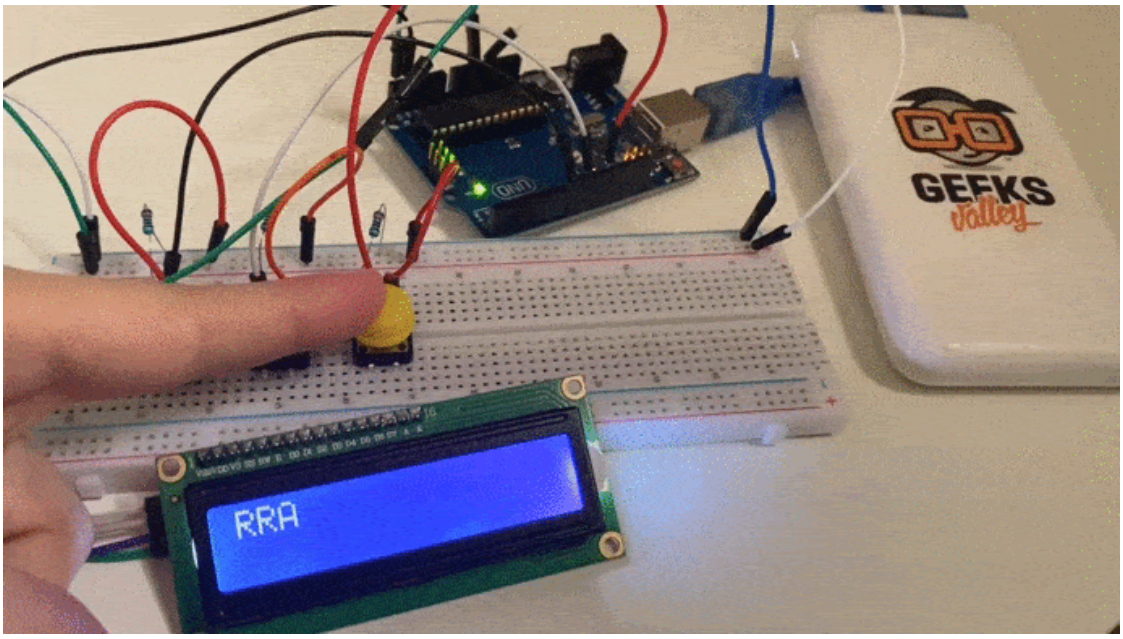
في هذه الدالة سيكون هناك رموز خاصة في شفرة مورس لها تفسير معين.

عند إدخال المستخدم 6 نقط متتابعة فذلك يعني أن المستخدم يريد حذف آخر حرف تمت كتابته.

عند إدخال المستخدم 6 شرطات متتابعة فذلك يعني أن المستخدم يريد حذف كامل النص المكتوب.

```
void extractActionFromTonesBuffer()
{
if (tonesBuffer == "......")
text.remove(text.length() - 1, 1);
if (tonesBuffer == "------")
text = "";
}
```


RA
GEEKS
Valley


RRA
GEEKS
Valley