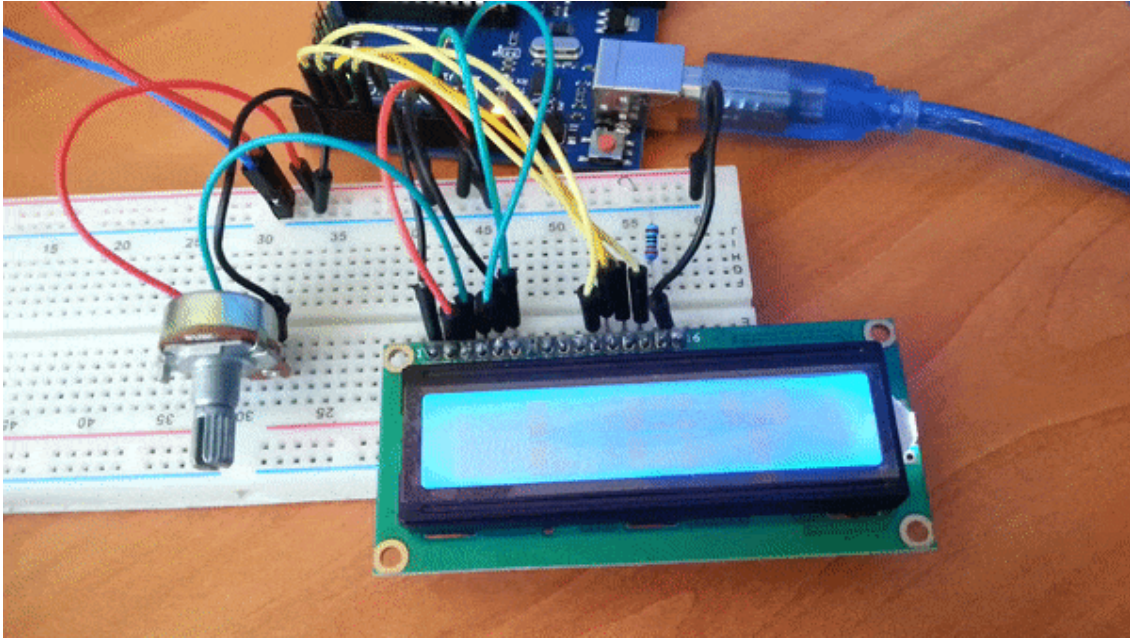


الرسم على شاشة LCD

يمكن عرض جميع الرسائل المكونة من الحروف الانجليزية والأرقام ورموز خاصة مثل (علامات الترقيم، الرموز الرياضية) وكذلك بعض الاحرف اليابانية و اليونانية على شاشة الـ LCD . ولكن كيف يمكننا عرض الحروف العربية أو رموز الخاصة بك على الشاشة؟! في هذا المشروع سنقوم بكتابة كلمات باللغة العربية و رسم رموز مخصصة عن طريق الرسم على شاشة LCD.



القطع المطلوبة:

الأدوات التي تحتاجها لهذا المشروع :



LCD 16×2 شاشة



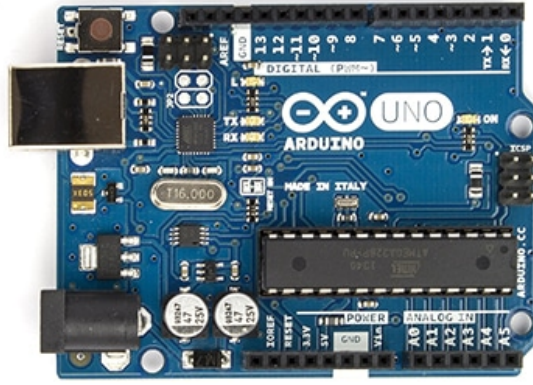
Potentiometer 10K Ω



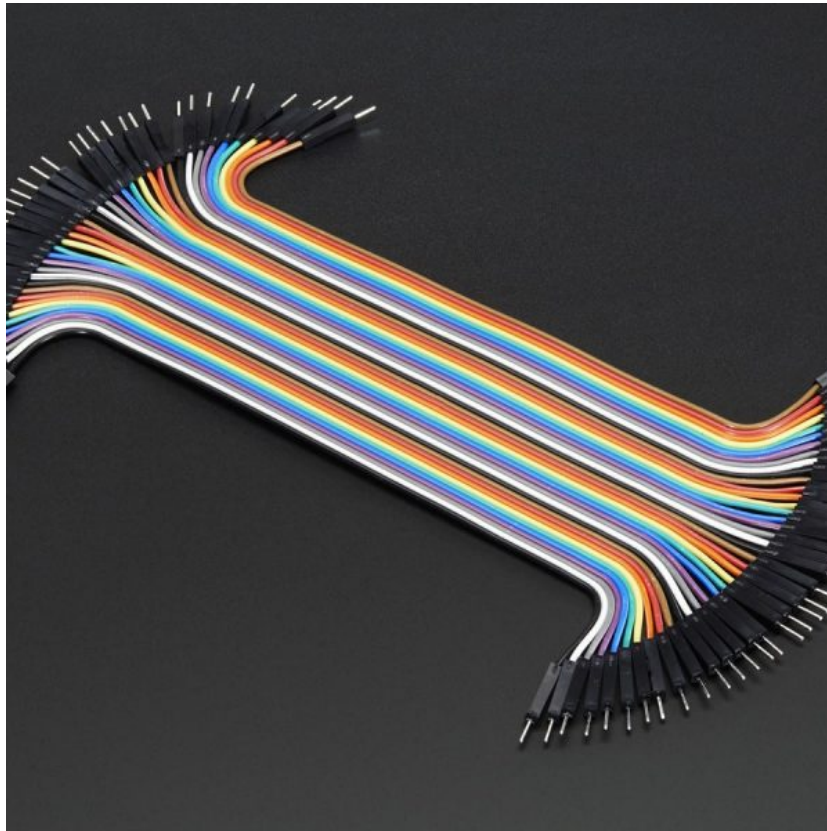
مقاومة 220 اوم



لوح تجارب حجم متوسط (Half size breadboard)



Arduino Uno R3



اسلاك توصيل ذكر/ذكر (Jumper Wires Male Male)

طباعة الحروف العربية او الرموز الخاصة بك على شاشة العرض LCD ليس من الصعب جدا، يتطلب ذلك معرفة CG-RAM لشاشة LCD .

ذاكرة CG-RAM

CG-RAM هي العنصر الأساسي لإنشاء الرسومات و الاحرف الخاصة بك . حيث تقوم بتخزين الرسومات التي تم إنشائها بمجرد اعلانها بالكود .

- حجم ذاكرة CG-RAM هو 64 بايت (64-byte) . ولذلك يمكنك تشكيل 8 احرف في وقت واحد، وكل حرف يتكون من 8 بايت (8-Byte) . ثمانية احرف كل منها 8 بايت (8-byte* 8 characters) تساوي 64-Byte .
- يبدأ عنوان الذاكرة CG-RAM من (0x40 Hex (64 decimal) .

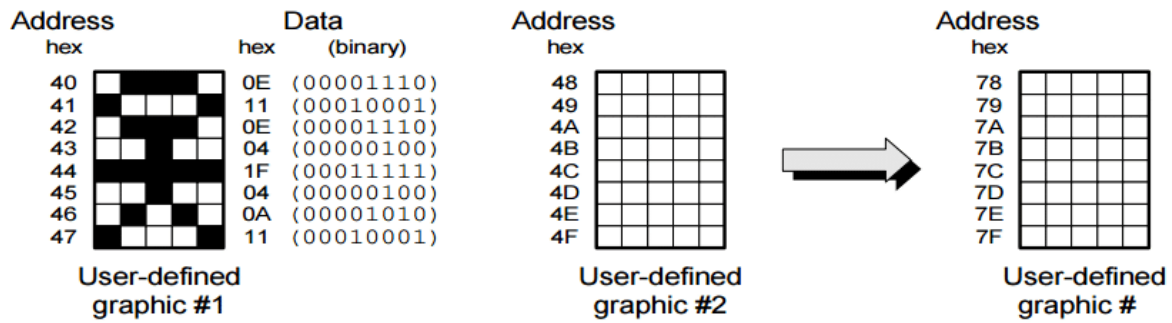
سنقوم بإنشاء الاحرف العربية او رسومات على هذه الذاكرة. بمجرد إنشاء هذه الاحرف يمكننا عرضها على شاشة LCD في اي وقت

عن طريق ارسال امر معين الى شاشة الـ LCD .

في الجدول التالي يتم توضيح عناوين الاحرف و الرسومات التي سيتم إنشائها وأوامر الطباعة الخاصة بهم:

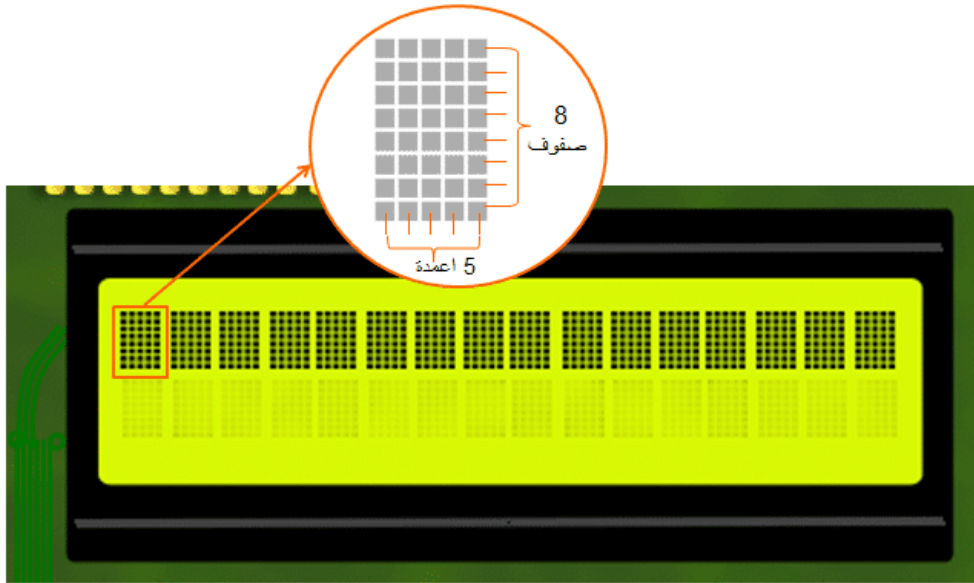
CG-RAM Characters	CG-RAM Address (Hexadecimal)	Commands to display Generated Characters (Decimal)
1 st Character	0x40	0
2 nd Character	0x48	1
3 rd Character	0x50	2
4 th Character	0x58	3
5 th Character	0x60	4
6 th Character	0x68	5
7 th Character	0x70	6
8 th Character	0x78	7

في الجدول اعلاه ستلاحظ العنوان الذي سيبدأ به كل رمز خاصة بك مع امر الطباعة الخاص به. مثلا اول حرف يتم انشاؤه وتخزينه على الموقع من 0x40 الى 0x47 في الذاكرة، ثم يتم طباعته على شاشة الـ LCD بواسطة ارسال الامر 0 الى شاشة الـ LCD. والحرف الثاني تبدأ تخزينه من الموقع 0x48 الى 0x4F ويتم طباعته من خلال ارسال الامر 1 الى الشاشة.

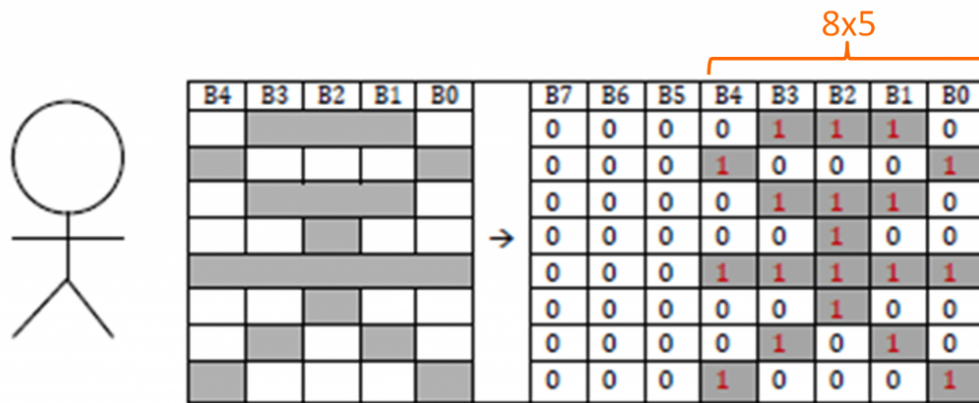


كيف يتم إنشاء أحرف مخصصة في CG-RAM

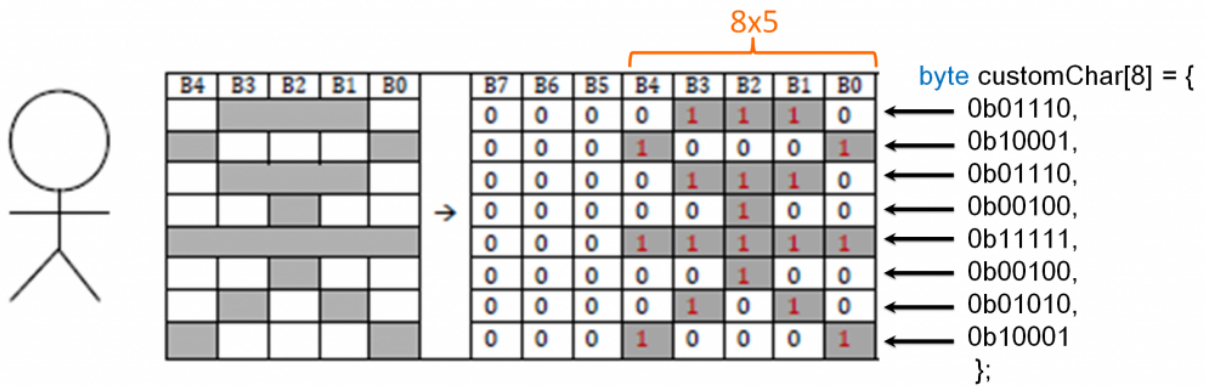
يتم تشكيل كل حرف او رمز على الشاشة في مصفوفة تكون ابعادها 5x8. حيث ان 5 تمثل عدد الاعمدة و 8 تمثل عدد الصفوف.



أولاً: قم برسم مصفوفة 5×8 بكسل، ثم قم برسم الرمز أو الحرف الخاص بك في داخل المصفوفة، ثم قم بوضع على كل خلية ملون قيمة 1 ، و قيمة صفر على الباقي.

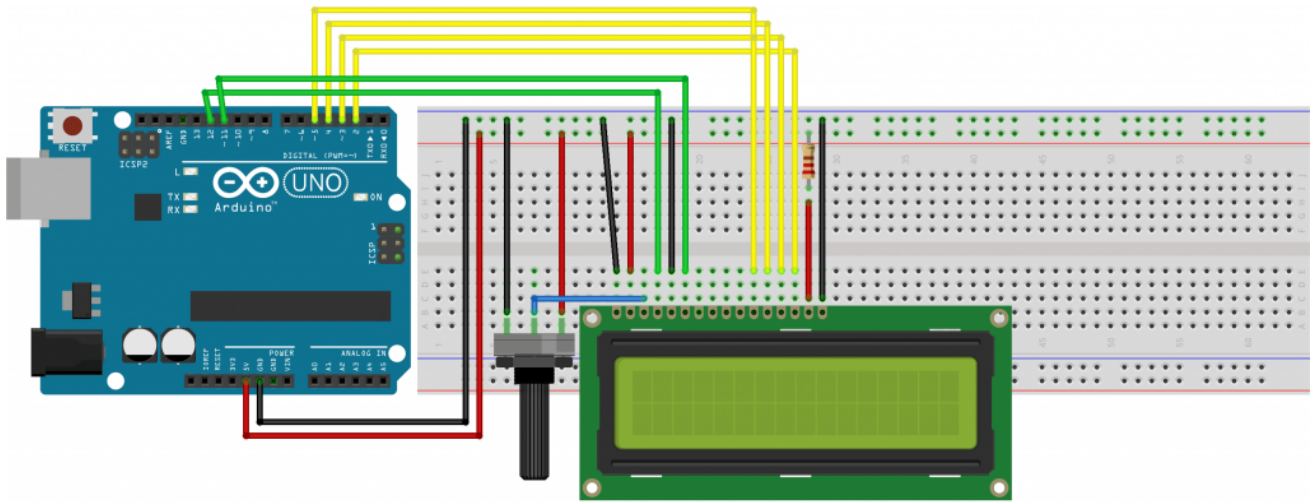


ثانياً: قم بتحويل تلك المصفوفة الى 8-Bytes.



تصميم لوح التجارب

قم بتوصيل الدائرة الكهربائية على لوحة التجارب كما هو موضح بالصورة :



توصيل شاشة الـ LCD :

LCD	Arduino
RS pin	Pin 12
Enable pin	Pin 11
D4 pin	Pin 5
D5 pin	Pin 4
D6 pin	Pin 3
D7 pin	Pin 2

يمكنك العودة إلى الدرس الثاني عشر للإطلاع على طريقة توصيل شاشة الـ LCD .

الكود البرمجي للأردوينو

قم برفع البرنامج التالي على الأردوينو:

```
#include <LiquidCrystal.h>

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

byte M[8] = {
0b000000,
0b000000,
0b00110,
0b01001,
0b11001,
0b00110,
0b000000,
0b000000
};
byte R[8] = {
```

```

0b000000,
0b000000,
0b000010,
0b000010,
0b000011,
0b000100,
0b01000,
0b10000
};
byte H[8] = {
0b000000,
0b01100,
0b10010,
0b000001,
0b11111,
0b000000,
0b000000,
0b000000
};
byte BA[8] = {
0b10000,
0b10000,
0b10000,
0b10010,
0b11111,
0b00000,
0b00100,
0b00000
};
byte logo[8] = {
0b11111,
0b10101,
0b11111,
0b11111,
0b01110,
0b01010,
0b11011,
0b00000
};

void Word1()
{
lcd.setCursor(12,0);
lcd.write(byte(0));
lcd.write(1);
lcd.write(2);
lcd.write(3);
}
void Logo1()
{
lcd.setCursor(10,0);
lcd.write(4);
}

```

```

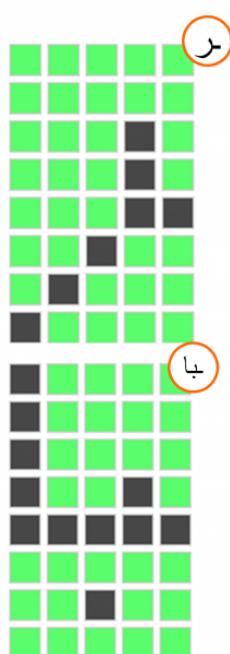
void setup() {
  lcd.createChar(0, BA); // char (بـ)
  lcd.createChar(1, H); //char (حـ)
  lcd.createChar(2, R); //char (رـ)
  lcd.createChar(3, M); //char (مـ)
  lcd.createChar(4, logo);
  lcd.begin(16, 2);
}

void loop() {
  Word1();
  Logo1();
}

```

شرح الكود :

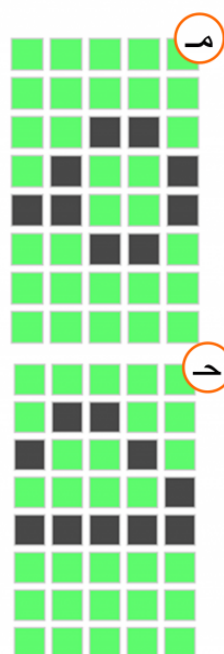
تم رسم الأحرف داخل مصفوفة و تحويلها الى byte-8 :



```

byte customChar[8] = {
  0b000000,
  0b000000,
  0b000010,
  0b000010,
  0b000011,
  0b000100,
  0b01000,
  0b10000
};

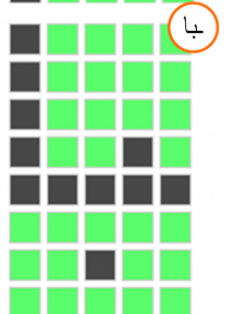
```



```

byte customChar[8] = {
  0b000000,
  0b000000,
  0b00110,
  0b01001,
  0b11001,
  0b00110,
  0b00000,
  0b00000
};

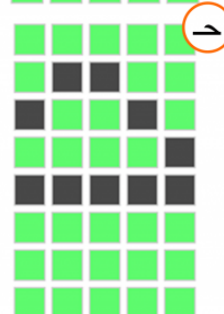
```



```

byte customChar[8] = {
  0b10000,
  0b10000,
  0b10000,
  0b10010,
  0b11111,
  0b00000,
  0b00100,
  0b00000
};

```



```

byte customChar[8] = {
  0b00000,
  0b01100,
  0b10010,
  0b00001,
  0b11111,
  0b00000,
  0b00000,
  0b00000
};

```

```

byte M[8] = {
  0b00000,
  0b00000,
  0b00110,
  0b01001,
  0b11001,
  0b00110,
  0b00000,
  0b00000
};
byte R[8] = {
  0b00000,
  0b00000,
  0b00010,
  0b00010,
  0b00011,

```

```

0b00100,
0b01000,
0b10000
};
byte H[8] = {
0b00000,
0b01100,
0b10010,
0b00001,
0b11111,
0b00000,
0b00000,
0b00000
};
byte BA[8] = {
0b10000,
0b10000,
0b10000,
0b10010,
0b11111,
0b00000,
0b00100,
0b00000
};

```

يتم استخدام createChar لتشكيل رمز يمكن طباعته على شاشة LCD و تكون ابعادة 5x8 بكسل.

```

lcd.createChar(0, BA); // char (ب)
lcd.createChar(1, H); //char ( ح )
lcd.createChar(2, R); //char (ـ)
lcd.createChar(3, M); //char (م)

```

وليتم رسمه على شاشة الـ LCD يتم استخدام الامر write

```

lcd.write(byte(0));
lcd.write(1);
lcd.write(2);
lcd.write(3);

```

بعد رفع الكود البرمجي على لوحة الأردوينو سيتم عرض كلمة مرحبا و رمز على شاشة الـ LCD .

