



وزارة التعليم العالي والبحث العلمي
جامعة الفرات الأوسط التقنية
كلية البوليتكنك النجف
قسم هندسة تقنيات الأجهزة الطبية

المادة: الحاسبات الدقيقة المرحلة: الثانية

تدريسي المادة:

م.د علي ناصر حسين

الايميل الجامعي للتدريسي

ali.jamaluddin@atu.edu.iq

العام الدراسي

2025-2026

المحاضرة 3 - البدء مع Arduino و LED

الاختبار القبلي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. ما الوظيفة الأساسية للأمر **pinMode**؟

- (A) تحديد حالة المنفذ كمدخل أو مخرج
- (B) قراءة الجهد التناظري
- (C) إيقاف البرنامج
- (D) طباعة البيانات

2. ماذا تعني **HIGH** عند التحكم بمخرج رقمي؟

- (A) جهد مرتفع
- (B) جهد سالب
- (C) عدم وجود جهد
- (D) قراءة تناظرية

3. أي دالة تُستخدم لإرسال **HIGH** أو **LOW** إلى **LED**؟

- A) digitalWrite
- B) digitalWrite
- C) analogRead
- D) Serial.print

4. ما وظيفة **delay(1000)**؟

- (A) انتظار 1000 ms
- (B) تشغيل المنفذ 1000 مرة
- (C) قراءة 1000 قيمة
- (D) إيقاف Arduino نهائياً

5. أي جزء من البرنامج يتكرر باستمرار؟

- (A) loop
- (B) setup
- (C) include
- (D) const

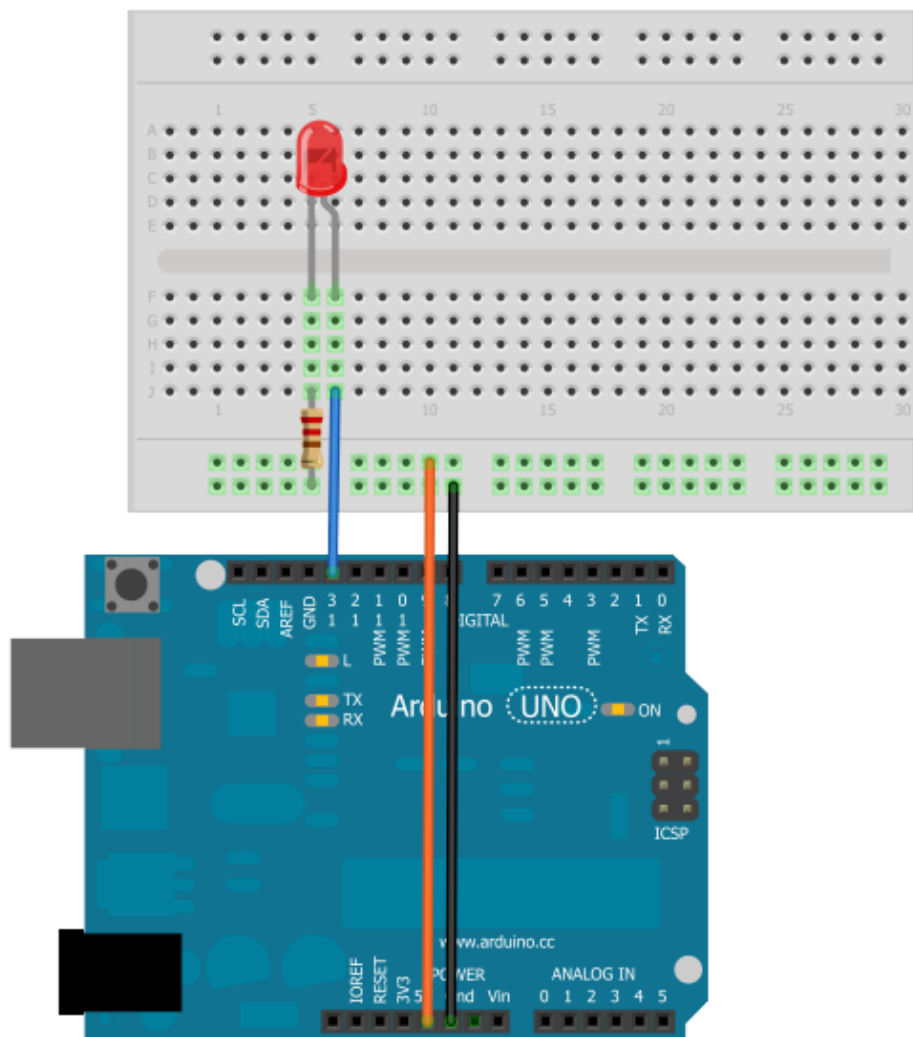
الفصل الثالث

لنبدأ العمل مع اردوينو



Let's get started

المثال الأول: تشغيل دايود ضوئي *Blinking led*



مكونات المثال (1):✓ **بورده اردوينو** Arduino Uno

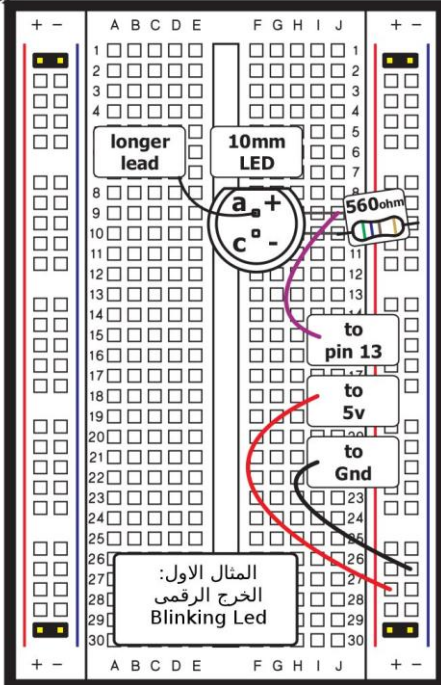
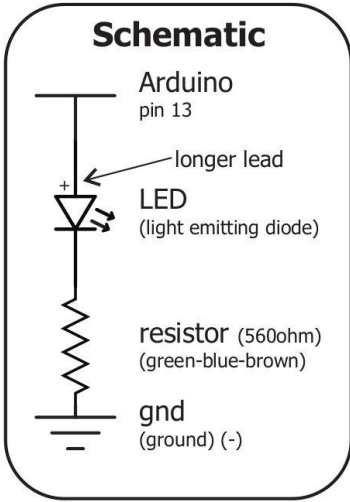
✓ Breadboard

✓ مقاومة كهرييه بقيمه 560 اوم

✓ دايود ضوئى قطر 10 مللى

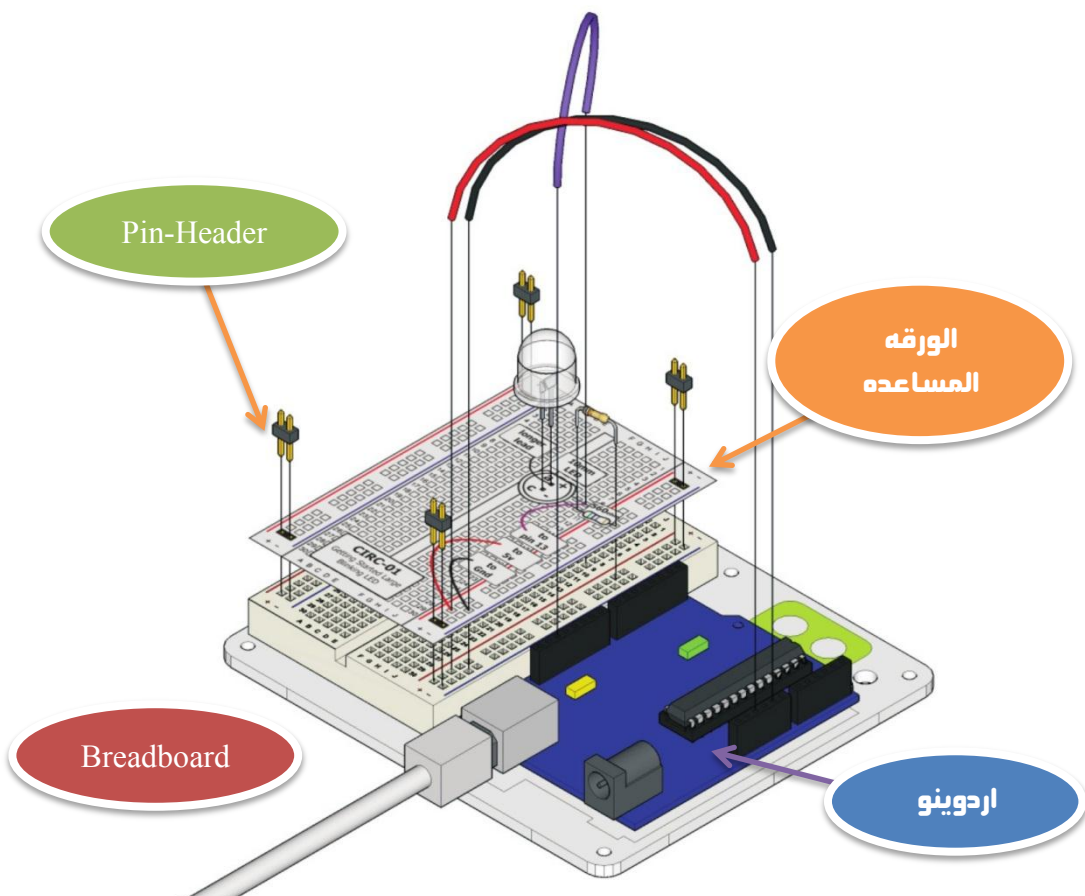
✓ جهاز الكمبيوتر

✓ ورقه المساعده الموجوده فى هذه الصفحه

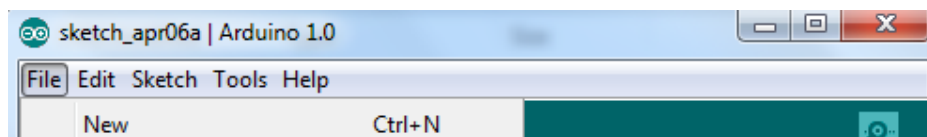


ما هى الورقه المساعده: هى ورقه يتم طباعتها لترشدك الى كيفية وضع المكونات الالكترونيه على لوحه التجارب Breadboard

الخطوه الاولى: قم بطباعه هذه الصفحه (ملحوظه عند الطباعه لا تقيم بتكبير او تصغير حجم الصفحه و استخدم مقياس 100%) ثم قص ورقه المساعده بعد ذلك ضع الورقه فوق Breadboard وثبتها باستخدام 4 وصلات pin-header ثم ابدأ فى توصيل المكونات الالكترونيه فوقها كم فى الصوره



الخطوة الثانية: من بيئة التطوير Arduino IDE قم بعمل ملف جديد من شريط الاوامر السريع و اكتب الاوامر البرمجيّه التاليه فى الجزء المخصص لكتابه الاكواد



// Example 01 : Blinking LED

```
const int LED = 13;
```

```
void setup ( )
```

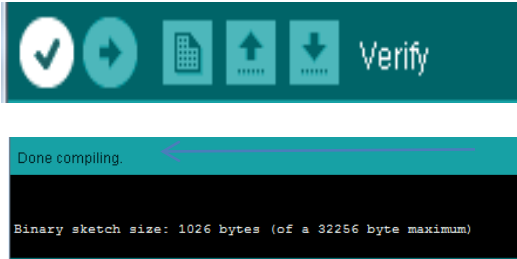
```
{  
  pinMode(LED, OUTPUT);  
}
```

```
void loop()
```

```
{  
  digitalWrite(LED, HIGH);  
  delay(1000);  
  
  digitalWrite(LED, LOW);  
  delay(1000);  
}
```

الخطوة الثالثة: بعد ان تنتهي من كتابه

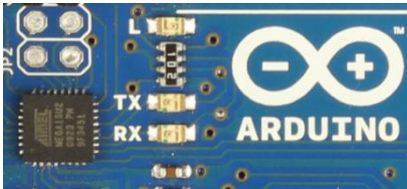
الكود البرمجي اضغط على زر تأكيد Verify فى الشريط الاوامر السريعه فى اعلى ال IDE ثم انتظر حتى تظهر رساله فى الاسفل تخبرك بأن البرنامج قد انتهى من تجهيز الكود ليتم رفعه على **بورده** **الاردوينو**



تأكد ان كابل الاردوينو متصل بالحاسب ثم قم بالضغط على زر " رفع



Upload " و ذلك حتى يتم بدأ عمليه تحميل البرنامج الى المتحكمه الدقيقه و انتظر عده ثوانى ستلاحظ عندها ان ال leds الموجوده على البورده (مكتوب بجانبها Tx/Rx) بدأت تضئ بسرعه عاليه ثم تقف قليل و هذه الاضائه تعنى ان البورده بدأت تستقبل بيانات البرنامج الذى قمنا بكتابته ، بعد ذلك يبدأ البرنامج و يجعل الليد الذى وضعناه يضى و ينطفئ بانتظام كل ثانيه .



مبروك 😊 اول مشروع لك بأستخدام اردوينو، و الان سنقوم بتشريح و فهم الكود البرمجي المستخدم فى هذا المثال.

شرح الكود البرمجي



```
const int LED = 13;
```

يقوم هذا الامر بعمل ثابت **constant** قيمته = 13 واسمه LED



سؤال لماذا كتبنا هذا الثابت؟؟

نقوم باستخدام الثوابت حتى تسهل علينا تسميته المخارج و المداخل

(Input/Output Pins) للمتحكمه الدقيقه، فمثلا فى هذا البرنامج

قمنا بتعريف هذا الثابت حتى نعبر عن المخرج رقم 13 بأن اسمه LED و

بذلك لو اردنا ان نأمر المتحكمه فى اى جزء من اجزاء البرنامج لاحقا

بأن تشغل الليد الموجود على المخرج رقم 13 او ان تقوم بإطفاء ذلك الليد لن نحتاج

عندها لذكر الرقم و انما سنذكر كلمه **LED** مباشره و ستفهم المتحكمه الدقيقه اننا نقصد

بهذا المخرج رقم 13



```
Void setup ( )
```

```
{
pinMode(LED,OUTPUT);
}
```

السطر الثانى من اوامر

البرنامج هو تعريف وظيفه

ال pin رقم 13 و التى نرمز لها

بأسم LED

في برامج المتحكمات الدقيقة يتكون البرنامج من 3 مراحل و هي كالتالي:

المرحلة الاولى: ان تقوم بتعريف اى متغيرات او ثوابت فى البرنامج الذى سوف نقوم بكتابته

المرحلة الثانيه: تتميز المتحكمات الدقيقة ان جميع ال **Digital Pins** يمكن ان تستخدم اما كمخرج او مدخل وفى هذا المرحله نقوم بضبط اعدادات المخارج و المدخل و تعريف المتحكمه ان ال Pin رقم 1 او 12 او 13 مثلا سوف تستخدم كمخرج او ان ال Pin رقم 5 و 6 سوف يستخدمان كمدخل للبيانات و تعتبر الداله (**setup**) هي المرحله الثانيه.



يتم استخدام الامر: **pinMode(pin number, state);** لتحديد وظيفه ال Pin و هل ستعمل كمدخل ام كمخرج حيث يتم كتابه رقم **او** اسم ال Pin اذا قمنا بتعريف ثابت مخصص سابقا، ثم يتم كتابه حاله التشغيل **INPUT او OUPUT** و لاحظ انه يجب كتابه حروف كلمه (مدخل او مخرج) بصوره capital.

ايضا يجب ان تكتب جميع اعدادات المخارج و المدخل بداخل اقواس الداله

{write your configurations here} () setup على سبيل المثال اذا اردنا

ان نخبر المتحكمه ان ال pin العاشره و الحاديه عشر و الثالثه عشر هم مخارج وان ال pin الثانيه هي مدخل فسنقوم بكتابته الاوامر التاليه:

```
void setup ( )
```

```
{
```

```
pinMode(11,OUTPUT); ←
```

```
pinMode(12,OUTPUT); ←
```

```
pinMode(13,OUTPUT); ←
```

```
pinMode(2,INPUT); ←
```

```
}
```

لاحظ : انه يجب بعد ان ننتهي من

كل امر يجب ان نضع علامه الفاصله المنقطوه (;) و اذا نسبتها في اي امر سوف تظهر لك رساله خطأ عندما تقوم بعمل تأكيد verify للبرنامج

المرحلة الثالثة : في هذه المرحلة نقوم بكتابه البرنامج نفسه و ما نريد من المتحكمه ان تفعله بالظبط و في حاله البرنامج السابق ستقوم المتحكمه بتشغيل الليد لمدته ثانيه ثم تطفئه لمدته ثانيه و هكذا مرار و تكرار.

```
void loop( )
```

```
{
```

```
digitalWrite(LED, HIGH); ←
```

```
delay(1000); ←
```

```
digitalWrite(LED, LOW); ←
```

```
delay(1000); ←
```

```
}
```

قم بتشغيل LED

ثم انتظر 1000 ملي ثانيه

قم بإطفاء LED

ثم انتظر 1000 ملي ثانيه

يتم كتابه برنامج المتحكمه داخل الداله `{your program} ( ) void loop` ، فى هذا المثال استخدمنا الامر: `digitalWrite(LED, HIGH);` وهو المسؤول عن تطبيق جهد كهربى اما $HIGH = 5 \text{ Volt}$ او $LOW = 0$ حيث نقوم بكتابه اسم المخرج اولاً ثم الجهد الذى نريد تطبيقه وعندما نريد تشغيل الليد نستخدم `HIGH` لتطبيق جهد 5 فولت وعند اطفائه نستخدم `LOW` لتطبيق جهد = صفر.

اما بالنسبه للأمر: `delay (1000)` فهو يستخدم لاجبار المتحكمه كم من الوقت يجب ان تنتظر قبل تنفيذ الامر التالى و بهذا الامر يمكننا ان نتحكم فى كم من الوقت يجب ان يعمل او يطفى الليد .

و كما شاهدنا فى المثال السابق:

```
digitalWrite(LED, HIGH);
delay(1000);
```

و تلك الاوامر تعنى ان المتحكمه ستطبق جهد يساوى 5 فولت `HIGH` على المخرج `LED` ثم تنتظر 1000 ملى ثانى و نلاحظ انه عند كتابه الوقت فى برامج المتحكمات الدقيقه يتم استخدام الملى ثانيه `millisecond` و التى تعنى جزء من الف من الثانيه و يمكننا استخدام النانو `Nano-second` و المايكرو ثانيه `Micro-Second`

بعد ذلك تقوم المتحكمه بتطبيق جهد `Low` يساوى صفر فولت و الذى يعنى ان الليد سوف ينطفى ثم يظل منطفئ 1000 ملى ثانيه اخرى = واحد ثانيه .

// Example 1 Blinking led ماذا عن اول سطر فى البرنامج؟؟

يحتاج المبرمجين عند كتابه اى برنامج مهما كانت اللغة البرمجيّه الى نظام للتعليقات comments و التعليقات هى جمل تساعد المبرمج على فهم و تنظيم الكود البرمجى و لا تحتسب فى اجزاء البرنامج ، بمعنى اخر سيقوم Arduino IDE بتجاهل التعليقات عن نقل البرنامج الى الصوره التنفيذيه و نقله و تشغيله على المتحكمه الدقيقه .

يمكنك كتابه التعليقات فى اردوينو فى اى مكان تشاء و يمكن اعتبار اى جمله انها تعليق ان قمت باضافه "///" قبل بدايه السطر.....مثال:

```
// Hello I'm Abdallah
// I'm Using comments
// Arduino is Soooooo funny
```

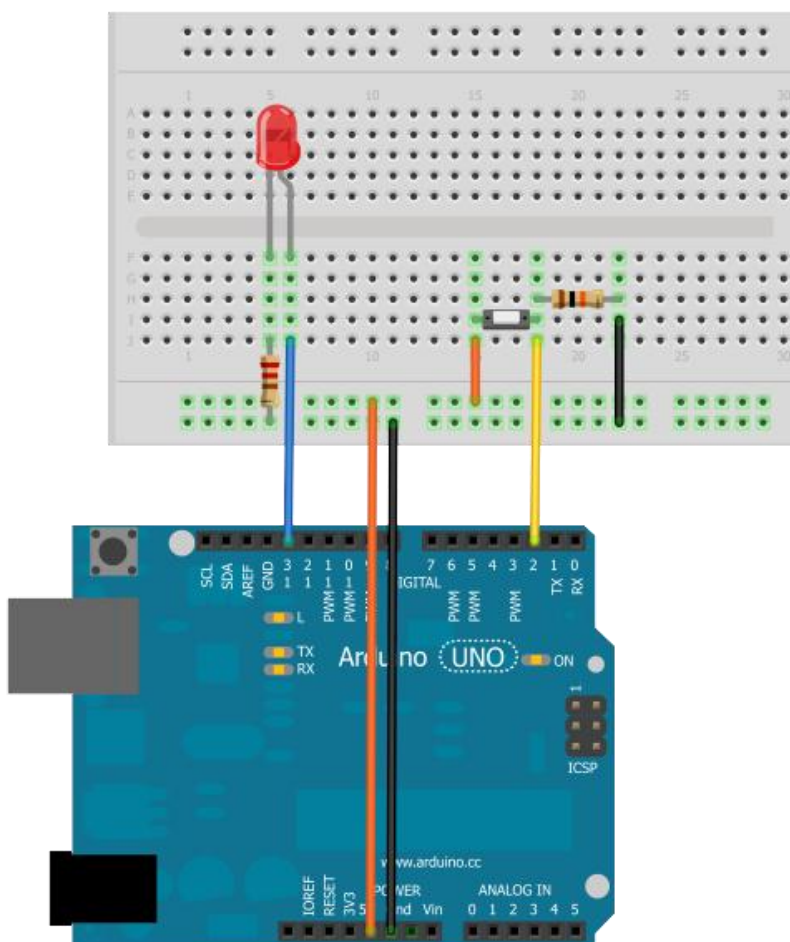
هناك اسلوب اخر لكتابه التعليقات الطويله و هى باستخدام الشرطه المائله و النجمه فى اول الكلام ثم كتابه نجمه و شرطه مائله فى نهايه الكلام.....مثال:

```
/* Hello I'm Abdallah and I Love Arduino, and you
are going to learn The best microcontroller system in
the world (^_^) */
```

ملحوظه: فى الطريقه الاولى لكتابه التعليقات يجب ان تكتب // قبل كل سطر تريد ان يتجاهله بالبرنامج باعتباره تعليق، لكن فى الطريقه الثانيه نكتب /* فى بدايه النص ثم نكتب ما نريد من كلام و نصوص و فى النهايه نختم ب */، لذلك تستخدم الطريقه الثانيه فى التعليقات الطويله و الطريقه الاولى فى التعليقات القصيره التى غالبا ما تكون سطر واحد.

المثال الثاني: تطوير للمثال السابق ليعمل فقط عند ضغط مفتاح (سويتش) من نوع Push button

في هذا المثال سنقوم بتطوير الفكرة السابقة و سنجعل الليد يعمل في حاله اذا ضغط المستخدم على زر من نوع push button وهو اشهر انواع السويشات المستخدمة في الأجهزة الإلكترونية.



مكونات المثال (2):

- بورده اردوينو Arduino uno
- Breadboard
- Push button
- Led 10 mm
- مقاومه 10 كيلو اوم
- مقاومه 560 اوم
- بعض السلوك للتوصيل

قم بوضع السويتش على لوحه التجارب ثم وصل احدى طرفيه بالمدخل 5v باستخدام سلك على بورده اردوينو و الطرف الاخر بالمقاومه العشره كيلو اوم و من نفس النقطه قم بتوصيل سلك للمدخل رقم 2 على **اردوينو** بورده و بعد ذلك قم بتوصيل طرف المقاومه الثانى بالمدخل الارضى .

ما الهدف من هذه الخطوة ؟؟

تعمل تلك الدائره على توفير دخل رقمي Digital input ، عند ضغط الزر تنطلق نبضه بقيمه 5 فولت و التى تعتبرها المتحكمه الدقيقه اشاره من نوع HIGH و عند ترك الزر ينفصل التيار الكهربى و يصبح الدخل يساوى صفر فولت و الذى تعتبره المتحكمه LOW و سوف نستغل هذا الامر لتعديل المثال الاول لكى يعمل فقط عندما يكون الدخل HIGH

الخطوة الثانية : كتابه الأكواد البرمجية التالية في Arduino IDE

نصيحه: قم بكتابه الأوامر البرمجيه بنفسك ولا تأخذها نسخ و لصق من الكتاب و ذلك حتي تدرب نفسك على كتابه الكود

```
// Example_2_Blinking_with_Switch
```

```
const int ledPin = 13;  
const int buttonPin = 2;  
int val;
```

**المرحلة الاولى: تعريف
المتغيرات و الثوابت**

```
void setup ( )
```

```
{  
  pinMode(ledPin, OUTPUT);  
  pinMode(buttonPin, INPUT);  
}
```

**المرحلة الثانية: تعريف و
تحديد المداخل و المخرجات**

```
void loop()
```

```
{  
  val = digitalRead(buttonPin);  
  
  if (val == HIGH)  
  {  
    digitalWrite(ledPin, HIGH);  
    delay(1000);  
    digitalWrite(ledPin, LOW);  
    delay(1000);  
  }  
  
  else {digitalWrite(ledPin, LOW); } }
```

**المرحلة الثالثة: كتابة
اوامر البرنامج الرئيسيه**

الخطوة الثالثة: قم بعمل تأكيد للكود البرمجي عن طريق الضغط على علامه تأكيد Verify و بعد ان تتأكد من ان البرنامج صحيح ولا يحتوى على اخطاء برمجيه و تم تحويله الى الصوره التنفيذيه بنجاح، قم بعد ذلك برفعه للمتحكمه الدقيقه عن طريق زر رفع Upload

شرح الكود البرمجي



المرحلة الاولى: قمنا بتعريف متغير اسمه ledPin و الذى سيرمز الى Pin 13 و

كذلك قمنا بتعريف المتغير المتغير buttonPin و الذى سيرمز الى السويتش الموجود

```
int ledPin = 13;
int buttunPin = 2;
int val = 0;
```

على المدخل الرقمى السابع و قمنا

بتعريف المتغير val و الذى سوف

نستخدمه لاحقا لتخزين حاله الدخل

على المدخل الرقمى الثانى pin 2

المرحلة الثانية: فى هذه المرحلة جعلنا المتحكمه تعامل pin 13 و التى يرمز لها

باسم ledPin على أنها مخرج و

سنستخدمه لتشغيل الليد الموصل به،

ثم قمنا بجعل المتحكمه تعامل

المخرج pin 2 على انه مدخل

رقمى وظيفته استقبال الاشارات

الرقميه

HIGH or LOW

```
void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(inPin, INPUT);
}
```

المرحلة الثالثة:

```
val = digitalRead(buttonPin);
```

في هذا الأمر ستقوم المتحكممة بقياس قيمة الجهد على المدخل pin 2 و تسجل القيمة في المتغير val

ويتم ذلك باستخدام الأمر (**digitalRead**) وإذا كان المستخدم ضغط على السويتش فإن القيمة ستكون 5 فولت = HIGH و ان لم يضغط ستكون القيمة صفر فولت = LOW

```
if (val == HIGH)
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
else {digitalWrite(ledPin, LOW); }
```

في هذا الجزء استخدمنا واحد من أشهر الاوامر في لغات البرمجة وهو الامر (**if...else** = إذا .. مادون ذلك)، يستخدم ذلك الامر لعمل مقارنات و جعل المتحكممة تفعل **شيء ما** إذا حدث **شيء آخر** محدد وفي هذا المثال استخدمنا الامر if كالتالى :

- ستقيس المتحكممة قيمة الجهد على المدخل السابع و تخزن القيمة في المتغير **val**
- إذا كانت القيمة تساوى HIGH و التى تعنى 5 فولت ستقوم المتحكممة بتشغيل الليد لمدة ثانية و اطفائه لمدة ثانية و هكذا باستمرار.
- ما لم تكن القيمة تساوى HIGH و الجهد المطبق على المدخل يساوى صفر ستقوم المتحكممة بتطبيق جهد يساوى صفر على الليد و بذلك لن يعمل الليد و سيظل منطفئ.

المحاضرة 3 - البدء مع Arduino و LED

الاختبار البعدي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. في مثال **Blinking LED**، إذا كان **LED** معرفاً على **pin 13**، فأَي سطر يجعله مخرجاً؟

- ;A) pinMode(LED, OUTPUT)
- ;B) digitalWrite(LED)
- ;C) delay(13)
- ;D) pinMode(LED, INPUT)

2. ماذا يحدث بعد **digitalWrite(LED,HIGH);** ثم **delay(1000);**؟

- (A) يبقى LED مشغلاً لمدة ثانية تقريباً
- (B) ينطفئ فوراً
- (C) يصبح pin مدخلاً
- (D) يُمسح البرنامج

3. ما الغرض من تعريف **const int LED = 13;**؟

- (A) إعطاء اسم ثابت للمنفذ 13
- (B) تغيير جهد المنفذ
- (C) قراءة المنفذ
- (D) إنشاء حلقة

4. أي دالة تُنفذ مرة واحدة عند بدء تشغيل **Arduino**؟

- (A) setup()
- (B) loop()
- (C) delay()
- (D) digitalWrite()

5. ما الذي يحدث إذا أزيلت الفاصلة المنقوطة من نهاية تعليمة برمجية؟

- (A) قد يظهر خطأ عند Verify
- (B) يزداد الجهد
- (C) يتحول المنفذ إلى analog
- (D) يعمل الكود أسرع

6. في مثال الزر، أي دالة تقرأ حالة **Push Button**؟

- A) digitalWrite
- B) digitalWrite
- C) analogWrite
- D) delay

7. إذا كانت **val == HIGH** في مثال **Blinking with Switch**، ماذا يفعل **LED**؟

- (A) يومض وفق أوامر البرنامج
- (B) يبقى مطفأ دائماً
- (C) يتحول إلى مدخل
- (D) لا تُقرأ حالة الزر

8. ما وظيفة **Verify** في **Arduino IDE**؟

(A) فحص الكود بحثاً عن أخطاء قبل الرفع

(B) رفع الكود فقط

(C) مسح اللوحة

(D) قياس الجهد

9. ما وظيفة **Upload**؟

(A) نقل البرنامج إلى لوحة Arduino

(B) فتح Serial Monitor

(C) تغيير نوع المتغير

(D) إيقاف loop

10. أي صيغة صحيحة لتعليق سطر واحد؟

A) // comment

B) ** comment

C) ## comment

D) -- comment

المحاضرة 4 - Push Button وتشغيل/إطفاء LED

الاختبار القبلي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. ما نوع العنصر **Push Button** بالنسبة للأردوينو؟

- (A) جهاز إدخال
- (B) جهاز إخراج
- (C) مصدر طاقة
- (D) شاشة عرض

2. ما الدالة المستخدمة لقراءة حالة الزر؟

- A) digitalWrite
- B) digitalWrite
- C) analogWrite
- D) lcd.print

3. في الدائرة، لماذا تُعرّف منافذ الأزرار **INPUT**؟

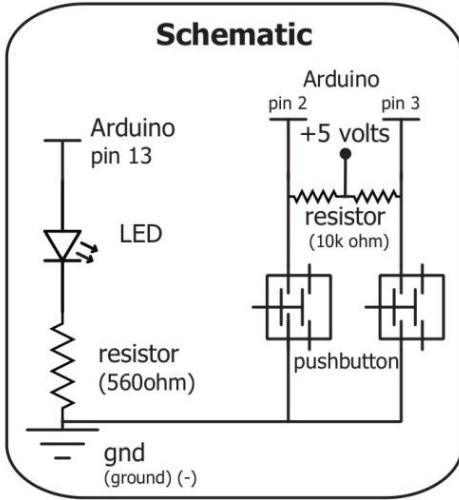
- (A) لأنها تستقبل حالة الضغط
- (B) لأنها تشغل LED
- (C) لأنها تولد 5V
- (D) لأنها تعرض النص

4. ما القيمتان الشائعتان للإشارة الرقمية؟

- A) HIGH و LOW
- B) A0 و A1
- C) 0V و 12V فقط
- D) SDA و SCL

5. ما وظيفة **LED** في المثال؟

- (A) جهاز إخراج يبين حالة التشغيل
- (B) قراءة الزر
- (C) تخزين الكود
- (D) قياس الحرارة

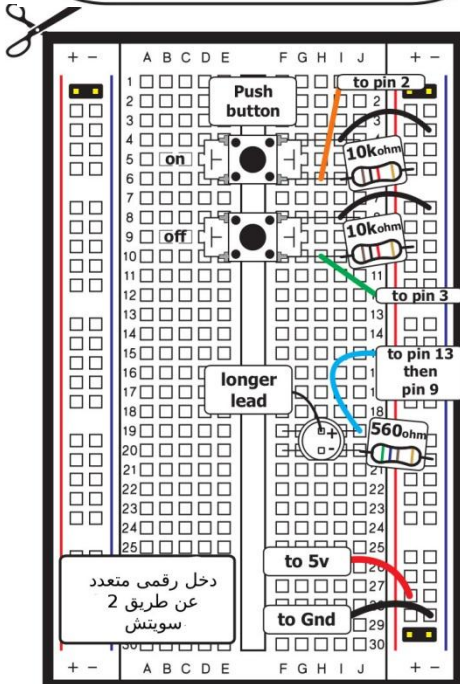
مكونات المثال (3):

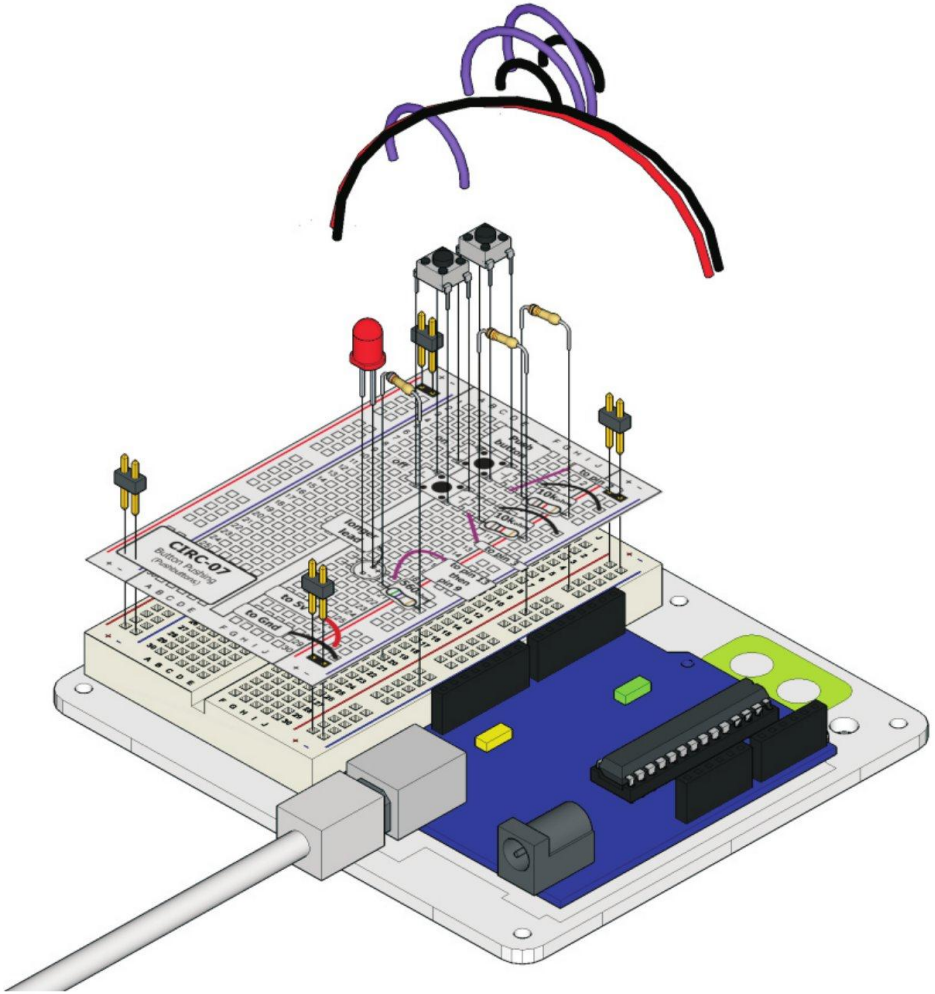
- Arduino uno
- Breadboard
- Led 5 mm
- عدد 2 مقاومة قيمه 10 كيلو اوم
- مقاومة واحده قيمه 560 اوم
- عدد 2 سويتش
- اسلاك توصيل
- نسخه مطبوعه من الورقه المساعد

في هذا المثال سندمج ما تعلمناه في كلا المثالين الاول والثاني ، سنستخدم 2 سويتش من نوع push button وسيتولى السويتش الاول عمليه تشغيل الليد وسيتولى السويتش الاخر عمليه اطفائه.

الخطوه الاولى: قم بطابعه ورقه المساعد و

قصها ثم ثبتها على breadboard باستخدام 4 من ال-Pin-Headers ثم قم بتوصيل المكونات الالكترونيه على الورقه المساعد كما في الصوره





الخطوة الثانية: من بيئه التطوير Arduino IDE قم بعمل ملف جديد من شريط الاوامر السريع و اكتب الاوامر البرمجيّه التاليه فى الجزء المخصص لكتابه الاكواد و بعد الانتهاء ارفع البرنامج الى المتحكمه الدقيقه على البورده و جرب الضغط على السويتش الاول و الثانى و شاهد ماذا سيحدث

```
//Example_3_On_Off_with_2_Switchs_1
```

```
const int ledPin= 13;
```

```
const int inputPin1 = 2;
```

```
const int inputPin2 = 3;
```

```
void setup ( )
```

```
{
```

```
pinMode(ledPin, OUTPUT);
```

```
pinMode(inputPin1, INPUT);
```

```
pinMode(inputPin2, INPUT);
```

```
}
```

تحديد المداخل
والمخارج
للدائرة

```
void loop ( )
```

```
{
```

```
if (digitalRead(inputPin1) == HIGH)
```

```
{digitalWrite(ledPin, LOW); }
```

```
else if (digitalRead(inputPin2) == HIGH)
```

```
{digitalWrite(ledPin, HIGH); }
```

```
}
```

اوامر البرنامج
الرئيسيه

في هذا المثال استخدمنا تعديل بسيط على الأمر if وهو أضافه الأمر else if بدلاً من else فقط وهو أمر يمكننا من أضافه اكثر من احتمال واحد في جملة if .

مثال:

```
If (you like science) {you should read books}
else if (you like sport) {you should play football}
else if (you like art) { you should take art lessons}
else { you should search for a hobby to do in your summer}
```

كما نرى في هذا المثال وضعنا 3 احتمالات و خيار أخير كتالي :

- **إذا** كنت تحب العلوم **عليك أن** تقرأ بعض الكتب
- **أما إذا** كنت تحب الرياضة **عليك أن** تلعب كرة القدم
- **أما إذا** كنت تحب الفن **عليك أن** تتعلم دروس الرسم
- **و إذا لم** يكن لديك هواية **عليك أن** تبحث عن هواية لتشغل بها وقت الإجازة الصيفية

وفي البرنامج السابق استخدمنا احتمال أن المستخدم سوف يضغط على السويتش الأول فتقوم المتحكم الدقيقة بتشغيل الدايود الضوئي و الاحتمال الثاني هو أن يضغط المستخدم السويتش 2 فتقوم المتحكم بإطفاء الدايود الضوئي ... وهكذا

و الآن ما رأيك أن نعدل البرنامج قليل (^_^)

```
//Example_3_On_Off_with_2_Switchs_2
const int ledPin= 13;
const int inputPin1 = 2;
const int inputPin2 = 3;

void setup ( )
{
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin1, INPUT);
  pinMode(inputPin2, INPUT);
}

void loop ( )
{
  if (digitalRead(inputPin1) == HIGH)
  {
    digitalWrite(ledPin, HIGH);
    delay(400);
    digitalWrite(ledPin, LOW);
    delay(400);
  }

  else if (digitalRead(inputPin2) == HIGH)
  {
    digitalWrite(ledPin, HIGH);
    delay(100);
    digitalWrite(ledPin, LOW);
    delay(100);
  }
}
```

مراجعته سريعة على ما تعلمناه فى الامثله السابقه

<i>void setup ()</i> {.....}	الداله المسؤله عن ضبط اعدادات المتحكمه و تحديد المداخل والمخارج
<i>void loop ()</i> {.....}	يتم كتابه البرنامج الرئيسى داخل هذه الداله و يقوم البرنامج باعاده نفسه مره تلو الاخرى كلما اتم تنفيذ جميع الخطوات
<i>int name = value;</i>	امر يستخدم لتعريف متغير بقيمه معينه
<i>const int name = value;</i> example: <i>const int led = 13;</i>	امر يستخدم لتعريف ثابت غير قابل للتغير يشبه الأمر السابق مع أضافه const
<i>pinMode (pin number, State);</i> example 1: <i>pinMode(11,INPUT);</i> example2: <i>pinMode(led,OUTPUT);</i>	الامر المسؤول عن تحديد وظيفه كل pin وضبطها للعمل كمدخل ام كمخرج INPUT or OUTPUT
<i>digitalWrite (pin number, State);</i> example: <i>digitalWrite(13,HIGH);</i>	الامر المسؤول عن اخراج جهد على احدى ال pin بصورة رقميه HIGH (1) or LOW(0)
<i>digitalRead (pin number);</i> example: <i>digitalRead(4);</i>	الامر المسؤول عن قراءه الجهد المطبق على احدى ال pin بصورة رقميه حيث يقوم بقياس الجهد (اذا كان 5 فولت حتى 3.8 فولت يعتبر HIGH واذا كان

	ما بين صفر حتى 1.8 فولت يعتبر (LOW)
<i>delay (time);</i> example1: <i>delay(100);</i> example2: <i>delay(1000);</i>	تأخير بزمان محدد : يستعمل هذا الامر فى تحديد المدة الزمنية للامر الذى يسبقه فى التنفيذ
if (the condition) {....what to do} else if (another condition) {...what to do} else (last condition) {....what to do}	(اذا حدث ... او اذا حدث.. افعل) جمله شرطيه تستخدم لتنفيذ امر ما عندما يتحقق شرط معين ويمكن استخدام الشروط التاليه: == يساوي != لا يساوي <= اقل من أو يساوي >= اكبر من أو يساوي < اقل من > اكبر من

اشهر أنواع المتغيرات التي يمكن استخدامها في اردوينو كالتالي:

النوع	مثال	القيمه
Integers	int led = 13;	أي عدد صحيح من سالب 32768 حتى موجب 32768
Float	float sensor = 12.5;	يستخدم مع الأعداد التي تحتوى كسور
character	char name= hello;	حروف نصيه أو كلمات
long	long variable = 99999999;	رقم طويل جدا يتراوح بين سالب 2,147,483,648 حتى موجب نفس الرقم
byte	byte number= 44;	رقم صغير يتراوح بين (0) إلى 255

لمزيد من المعلومات عن أنواع المتغيرات التي يمكن استخدامها

داخل برامج اردوينو قم بزياء الصفحة التالية

<http://www.arduino.cc/en/Reference/VariableDeclaration>

المحاضرة 4 - Push Button وتشغيل/إطفاء LED

الاختبار البعدي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. في المثال ذي زرّين، كم **Push Button** مستخدم؟

- A) 2
- B) 1
- C) 3
- D) 4

2. ما رقم المنفذ المستخدم للـ **LED** في الكود؟

- A) 13
- B) 2
- C) 3
- D) 9

3. ما رقما منافذ الإدخال للزرّين في الكود؟

- A) 2 و 3
- B) 9 و 10
- C) 12 و 13
- D) A4 و A5

4. أي تعليمة تجعل **ledPin** مخرجاً؟

- A) `pinMode(ledPin, OUTPUT)`
- B) `pinMode(ledPin, INPUT)`
- C) `digitalRead(ledPin)`
- D) `delay(ledPin)`

5. ما الذي تتحقق منه `if (digitalRead(inputPin1) == HIGH)`؟

- A) هل الزر الأول في حالة HIGH
- B) سرعة المحرك
- C) قيمة حساس تناظري
- D) عنوان LCD

6. وفق الكود، ماذا يحدث عند تحقق شرط `inputPin2 == HIGH`؟

- A) يُكتب HIGH على `ledPin`
- B) يُقرأ A0
- C) يُطفأ Arduino
- D) يتوقف loop

7. لماذا نستخدم `else if` بعد الشرط الأول؟

- A) لفحص شرط ثانٍ عند عدم تحقق الأول
- B) لتعريف منفذ
- C) لإنشاء متغير `float`
- D) لإرسال Serial

8. أي من الآتي يمثل خرجاً في الدائرة؟

A) LED

B) Push Button الأول

C) Push Button الثاني

(D) المقاومة فقط

9. إذا عُرِّف `inputPin1` كـ **OUTPUT** بدلاً من **INPUT**، فما المشكلة الأساسية؟

(A) لن يُستخدم كما هو مطلوب لقراءة الزر

(B) سيصبح LCD

(C) سيزيد عدد الأزرار

(D) سيعمل Serial تلقائياً

10. أي جزء من الكود يحتوي أوامر الفحص المستمر للأزرار؟

(A) loop

(B) setup

(C) include

(D) تعريف الثوابت

المحاضرة 5 - المداخل التناظرية والحساسات

الاختبار القبلي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. ما الفرق الأساسي بين الإشارة التناظرية والرقمية؟

(A) التناظرية متغيرة بصورة مستمرة والرقمية لها حالات محددة

(B) لا يوجد فرق

(C) الرقمية فقط للحساسات

(D) التناظرية دائماً صفر

2. أي من الآتي مثال لحساس تناظري في المحاضرة؟

A) LM35/TMP36

B) LED

C) Relay

D) Push Button فقط

3. أي منافذ **Arduino Uno** مخصصة للقراءة التناظرية؟

A) A0 إلى A5

B) 0 إلى 1 فقط

C) SDA فقط

D) 13 فقط

4. ما الدالة المستخدمة لقراءة قيمة تناظرية؟

A) analogRead

B) digitalWrite

C) pinMode فقط

D) delay

5. ماذا يتغير في حساس الحرارة التناظري مع تغير الحرارة؟

A) جهد الخرج

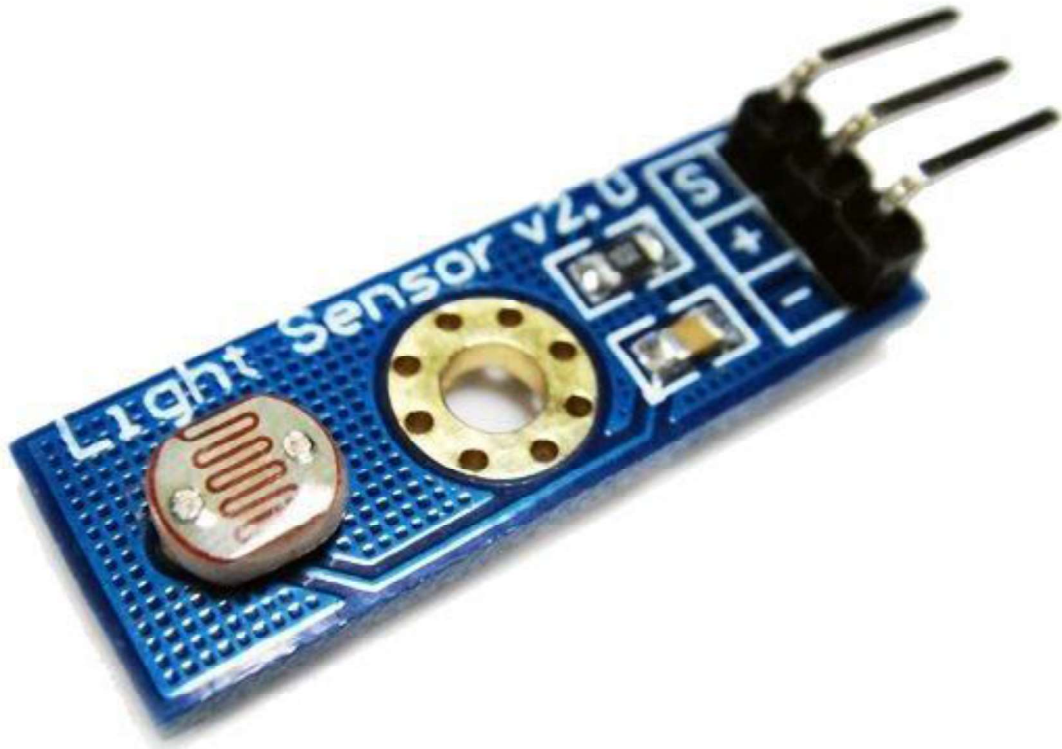
B) عدد الأرجل

C) نوع Arduino

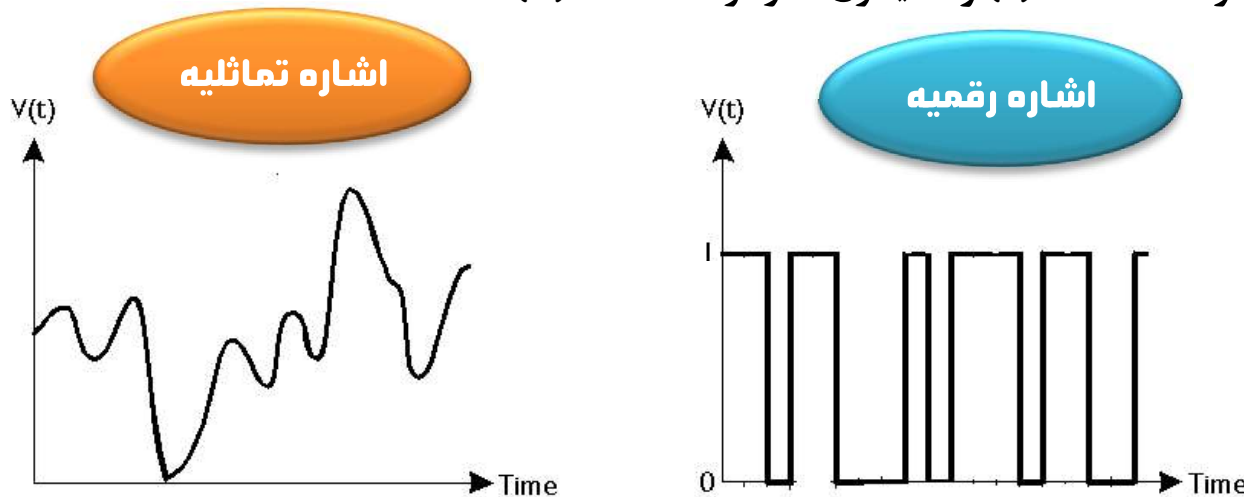
D) عنوان I2C

الفصل الرابع

الدخل و الخرج التماثلي و استخدام المستشعرات
Analog Inputs/Outputs and Sensors



يعرف الدخل او الخرج التماثلي بأنه اى فرق جهد تبدأ قيمته من صفر وقابل للتغير دائما و ليس له حدود قصوى معينه ، على عكس الدخل الرقمي و الذى اما يكون 5 فولت HIGH (1) و اما يكون صفر فولت LOW (0)



ما هي اهميه الاشارات التماثليه Analog Signals؟؟



يعبر الدخل التماثلي عن الاشارات المتغيره مثل فرق الجهد الناتج من مستشعر (حساس sensor) معين نتيجة التغير فى عامل معين، على سبيل المثال : حساس للضوء او ما يعرف بأسم المقاومة الضوئية Photo-Resistor وهى مقاومه كهرييه تتغير قيمتها بتغير مقدار الضوء الذى تتعرض له و اذا قمنا بتوصيل هذا المقاومه على مصدر ثابت للفولت و استخدامنا AVO-meter لقياس فرق الجهد المطبق عليها .. سنجد ان فرق الجهد سيتغير دائما بتغير مقدار الضوء الساقط عليها



في هذا الفصل سنتعرف على بعض اشهر المستشعرات مثل المقاومه الضوئية و مستشعر الحرارة مثل LM35 أو TMP36 هو عبارة عن ترانزستور خاص يتأثر بالحرار و يعطى تغير فى الجهد الكهربى (الفولت) متناسب مع التغير في درجة الحرارة .. حيث يعطي 10 مللى فولت (عشره أجزاء من الف فولت) لكل 0.1 درجة مئوية.... كما سنتعرف على الدخول و الخرج التماثلي و كيفية استخدام المستشعرات مع اردوينو





سنحتاج لهذه التجربة المكونات التالية:

- 1- عدد 2 بطاريه مقاس AAA بقيمه 1.5 فولت
- 2- عدد 1 حامل بطاريات لتوصيل 2 بطاريه
- 3- جهاز قياس متعدد الاستخدامات AVO Meter
- 4- حساس حرارى من أي الأنواع التالية :
TMP 35, TMP36, LM 35

احضر 2 بطاريه بقياس AAA بقيمه 1.5 فولت و ضعهم في
علبه حامل البطاريات لتحصل على قيمه كليه تساوى 3 فولت
ثم وصل الطرف الموجب لحامل البطارية (سلك لونه احمر)
بالطرف الأول من ناحيه اليسار للحساس الحرارى (طرف الدخل الثابت)



ثم قم بتوصيل الطرف الأرضي لحامل البطارية (سلك لونه اسود) بالطرف الأخير
للحساس الحرارى ، بعد ذلك قم بضبط الـ AVO على وضع قياس الفولت و وصل
الطرف الأحمر بالمخرج للحساس الكهربى و الطرف الأسود للطرف الأرضي للحساس
الكهربى



لاحظ قراءة فرق الجهد الظاهر على الـ AVO meter ستجد أنها 0.76 فولت تقريبا **الآن قم** بوضع يدك على الحساس واضغط عليه قليلا (تعمل تلك الحركة على رفع درجة حرارة الحساس و من المعلوم أن درجة حرارة الجسم تساوي 37 درجة مئوية تقريبا وهي اعلى من درجة حرارة جو الغرفة)

ثم لاحظ التغير في قراءة الفولت الناتج كما في الصورة التالية:

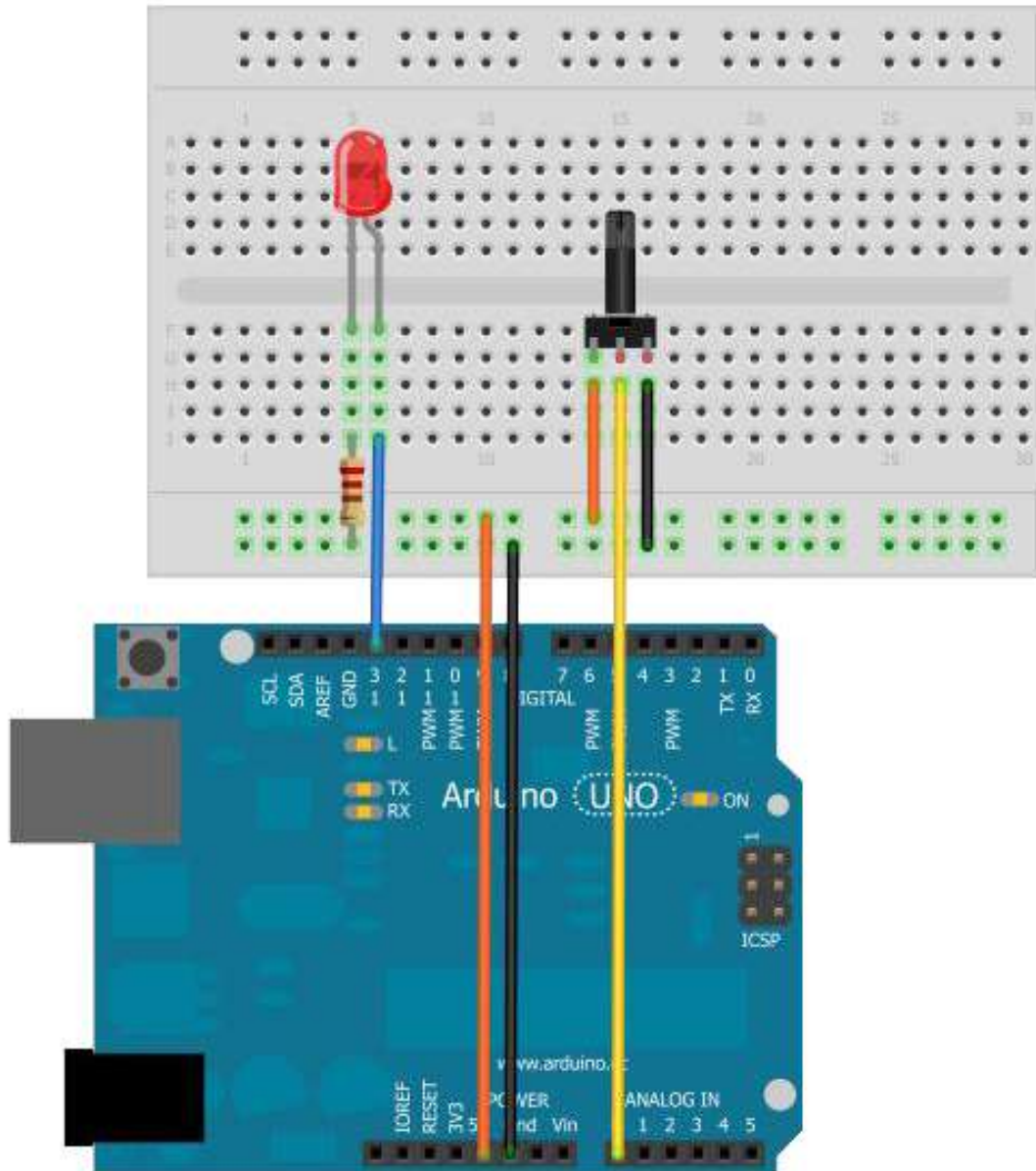


سنجد أن قيمه الفولت قد ازدادت بسبب حرارة اليد

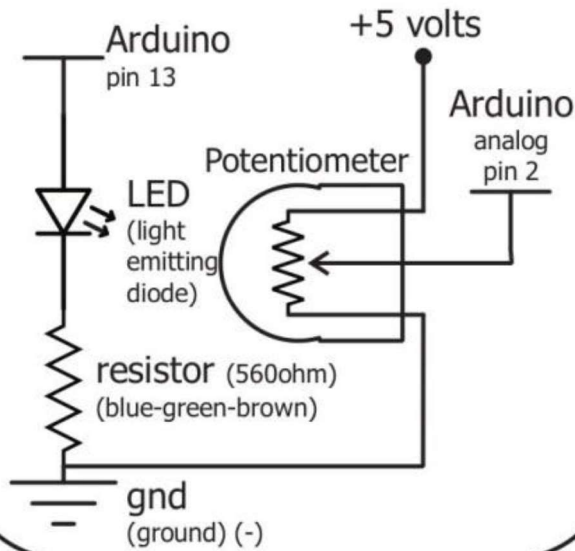
تعمل جميع الحساسات بأنواعها المختلفة بنفس المبدأ حيث يؤدي التغير في عامل بيئي معين إلى تغير المقاومة الكهربائية مما يؤدي إلى التغير في فرق الجهد الخارج من الحساس مما يولد إشارة تماثليه متغيره يمكن قياسها ، والآن لنبدأ مع امثله استخدام الحساسات مع اردوينو

المثال الرابع: استخدام مقاومه متغيره للتحكم فى سرعه اضاءه دايود ضوئى Potentiometer

في هذا المثال سنستخدم مقاومه متغيره للحصول على فرق جهد يمكن تغييره (دخلى تماثلى analog input) و سنجعل الداىود الضوئى يضيئ و ينطفئ بسرعه تعتمد على قيمه الدخلى التماثلى



Schematic

**مكونات المثال (4):**

✓ بورده اردوينو Arduino Uno

✓ لوحه تجارب Breadboard

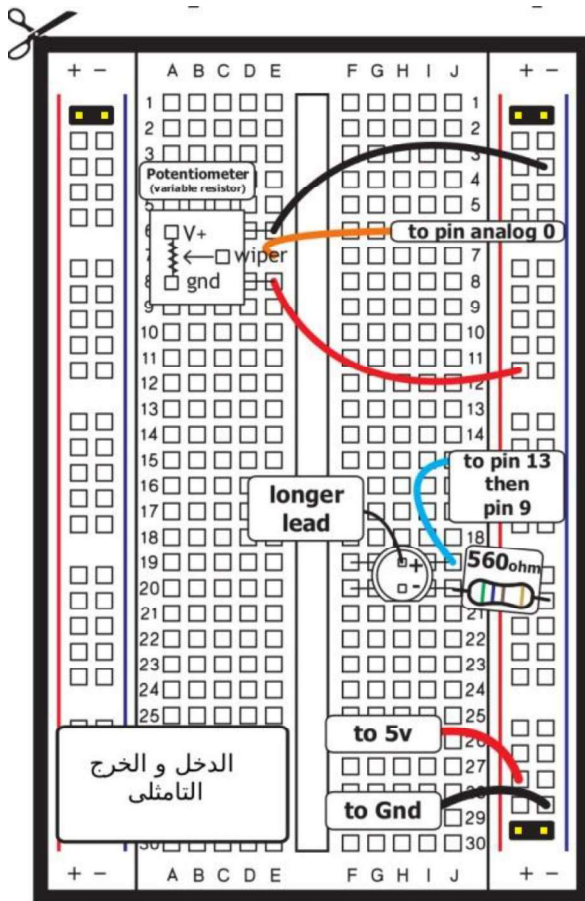
✓ دايود ضوئى led 5mm

✓ مقاومه 560 اوم

✓ مقاومه متغيره 3 اطراف 10 كيلو اوم

✓ أسلاك توصيل

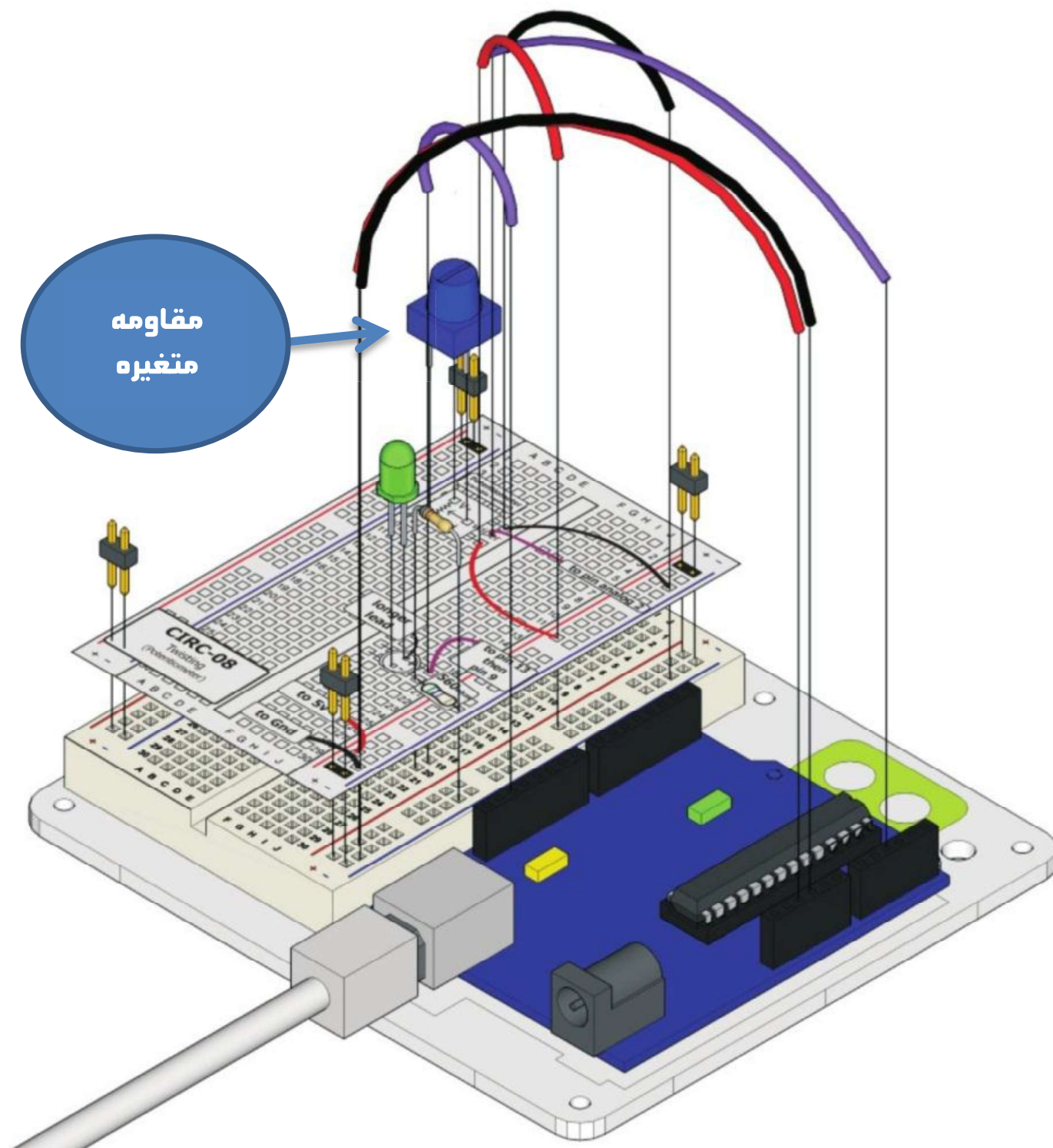
✓ ورقه المساعدة المطبوعة



قم بطباعه الورقه المساعدته ثم وصل

المكونات الالكترونيه على لوحه التجارب

ملحوظه: لا يشترط ان تستخدم مقاومه متغيره صغيره الحجم مثل التى فى الصوره ولك الحرية فى استخدام اى مقاومه متغيره لها 3 اطراف بأى حجم و اى قيمه مثل 100 كيلو اوم او 200 كيلو اوم، ايضا جميع المقاومات المتغيره ذات 3 اطراف يكون الطرف الاوسط هو الطرف المتغير و الذى سنوصله بالمدخل التماثلى A0 على بورده اردوينو كما فى الصوره التاليه



بعد الانتهاء من توصيل المكونات الالكترونيه على لوحه التجارب و بورده اردوينو قم بعمل ملف جديد و اكتب الاوامر البرمجيّه التاليه ثم قم برفع البرنامج الي بورده اردوينو

```
//Example_4_Variable_Resistor_sensor
const int sensorPin = A0;
const int ledPin = 13;
int sensorValue;
```

متغير لتخزين قيمه الجهد
التماثلي

```
void setup ( )
{
  pinMode(ledPin, OUTPUT);
}
```

```
void loop ( )
{
  sensorValue = analogRead(sensorPin);
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  digitalWrite(ledPin, LOW);
  delay(sensorValue);
}
```

شرح الكود البرمجي

في هذا المثال سنتعرف على واحد من أهم الدوال البرمجية الموجودة في لغة أردوينو `analogRead(pin number)` تقوم هذه الدالة بقراءة فرق الجهد بصورة تماثلية و تستطيع المتحكمه الدقيقه ان تقيس فرق جهد من 4.8 مللي فولت حتى 5 فولت تقريبا و تقوم بتحويل الاشاراه التماثليه الى قيمه رقميه من صفر الى 1024 و تسمى هذه العمليه باسم تحويل الاشاره من تماثليه الى رقميه Analog to digital converting

مثلا اذا كان الجهد الداخلى الى A0 يساوى القيم التاليه:

رقمى	4.8 مللى فولت = 1
رقمى	48 مللى فولت = 10
رقمى	480 مللى فولت = 100
رقمى	واحد فولت = 208.33
رقمى	2 فولت = 416.66
رقمى	5 فولت = 1024

و يستخدم الامر التالى فى تخزين قيمه القراءه الناتجه داخل المتغير sensorValue

sensorValue = analogRead(sensorPin);

بعد ذلك تقوم المتحكمه بتشغيل الـ داىود الضوئى فتره زمنيه تساوى قيمه هذا المتغير و تقوم بأطفائه بنفس الزمن مثل المثال الاول blinking led ولكن الزمن فى هذا المثال متغير تبعا لقراءه الجهد التماثل على المدخل A0.

فى هذا المثال استخدمنا مقاومه متغيره يمكننا ان نغير قيمتها عن طريق لف (تدوير) عمود الدوران الخاص بها و بذلك نحصل على اشارته تماثليه (تسمى تلك العمليه تغير متحكم به من الانسان) ، فى المثال القادم سنتعلم نوع جديد من المقاومات المتغيره و هى المقاومه الضوئيه التى تتغير قيمتها بتغير كثافه الضوء الساقط عليها و يسمى هذا تغير بيئى و سوف نستغل ظاهره التغير فى المقاومه بتغير الضوء لنحول المقاومه الى مستشعر للضوء light sensor

المحاضرة 5 - المداخل التناظرية والحساسات

الاختبار البعدي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. أي متحكم ذُكر في لوحة **Arduino Uno** بالمحاضرة؟

- A) ATmega328P
- B) ESP8266
- C) PIC16F84
- D) 8051 فقط

2. كم عدد مداخل **Analog** المذكورة في **Arduino Uno**؟

- A) 6
- B) 2
- C) 4
- D) 16

3. ما وظيفة الرجل الوسطى في حساس الحرارة ذي الثلاث أرجل كما سُرح؟

- A) خرج الحساس
 - B) GND
 - C) مدخل USB
 - D) خرج رقمي ثابت
4. أي رجل توصل بالأرضي؟

- A) GND
- B) VCC
- C) Signal فقط
- D) A5 دائماً

5. في كود حساس الحرارة، **sensorPin** معرف على:

- A) A0
- B) 13
- C) 9
- D) A5

6. ما الدالة التي تقرأ **sensorPin**؟

- A) `analogRead(sensorPin)`
- B) `digitalWrite(sensorPin)`
- C) `Serial.read()`
- D) `lcd.print()`

7. لماذا يُستخدم **float** للمتغير **voltage**؟

- A) لتمثيل قيم قد تحتوي كسوراً
- B) لتمثيل حرف واحد
- C) لتحديد pin كخرج
- D) لإيقاف البرنامج

8. ما وظيفة **Serial.begin(9600)**؟

(A) بدء الاتصال التسلسلي بسرعة 9600

(B) قراءة حساس الضوء

(C) تشغيل LED

(D) تهيئة LCD

9. أين يمكن مشاهدة القيم المرسلة من **Arduino** إلى الحاسوب؟

A) Serial Monitor

B) Breadboard

(C) المقاومة

(D) زر Reset فقط

10. أي عبارة تصف **analogRead** بشكل أدق؟

(A) تحول قراءة المدخل التناظري إلى قيمة رقمية يعالجها البرنامج

(B) تشغيل المحرك مباشرة

(C) تعرض نصاً على LCD

(D) تحدد عنوان I2C

المحاضرة 6 - المقاومة الضوئية LDR

الاختبار القبلي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. ما العنصر المستخدم لاستشعار شدة الضوء؟

- A) Photoresistor / LDR
- B) Servo
- C) Relay
- D) LCD

2. ما الخاصية التي تتأثر بالضوء في المقاومة الضوئية؟

- A) المقاومة الكهربائية
- B) عدد الأرجل
- C) حجم Arduino
- D) تردد USB

3. أي نوع مدخل مناسب لقراءة تغير LDR؟

- A) مدخل تناظري
- B) خرج رقمي فقط
- C) SCL فقط
- D) VCC فقط

4. ما اللوحة المستخدمة في تجربة LDR؟

- A) Arduino Uno
- B) Raspberry Pi فقط
- C) PLC
- D) ESP32 فقط

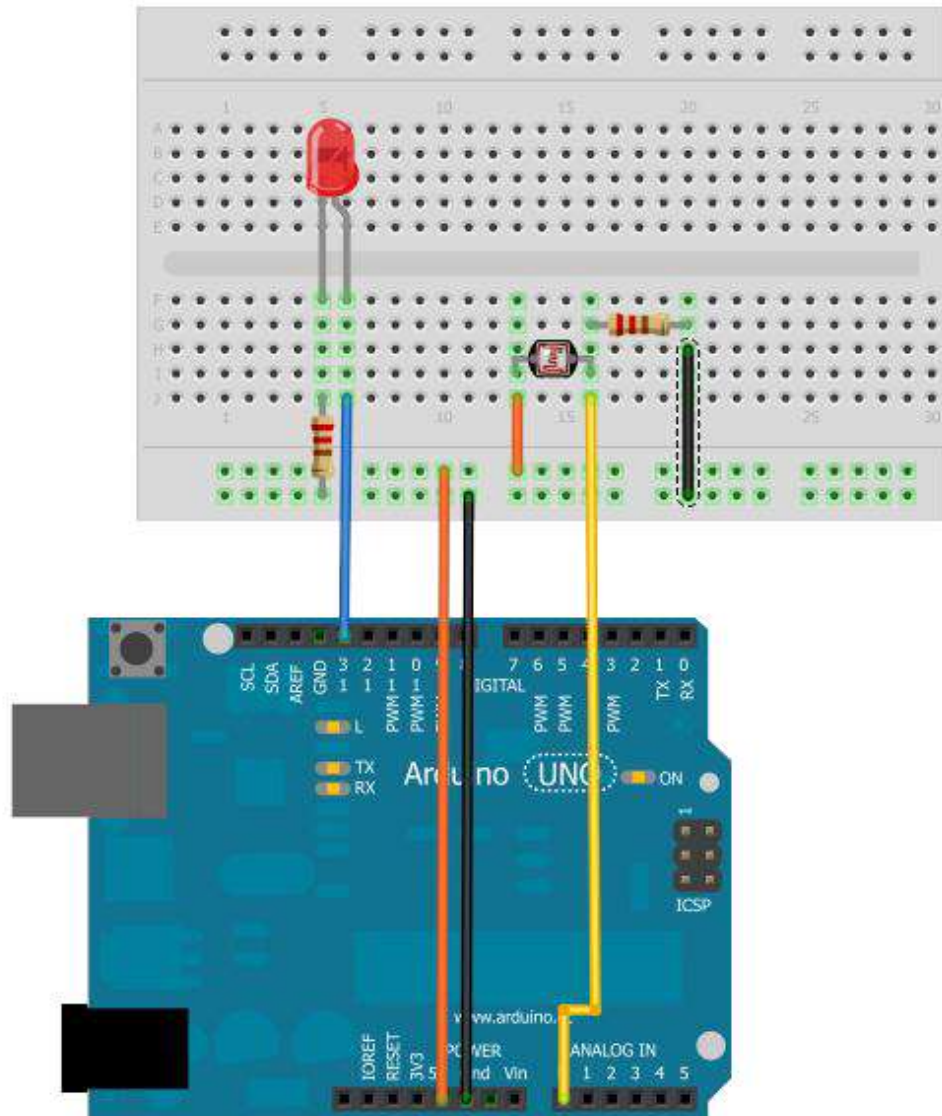
5. ما فائدة Breadboard في التجربة؟

- A) تركيب الدائرة بدون لحام
- B) زيادة الجهد
- C) تخزين الكود
- D) عرض النتائج

المثال الخامس: المقاومه الضويه (حساس الضوء التماثلي)

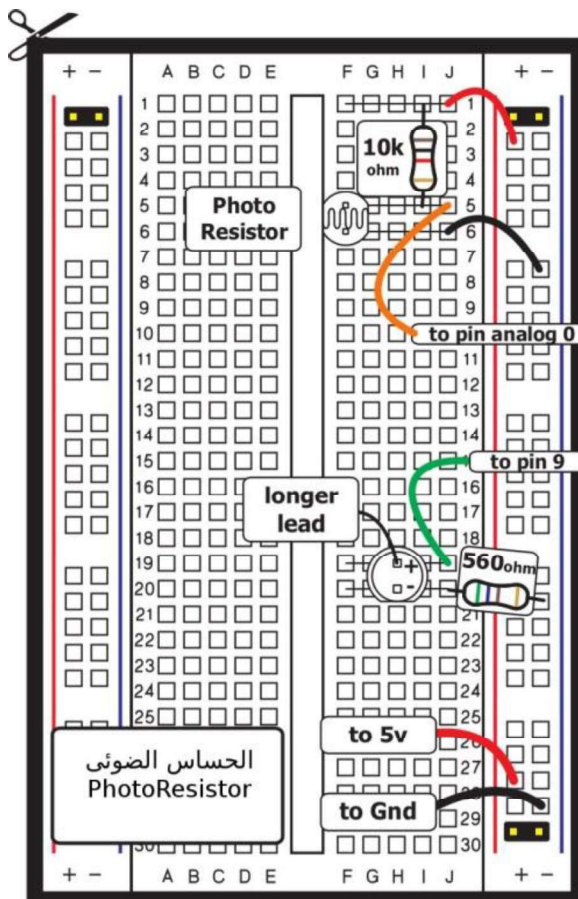
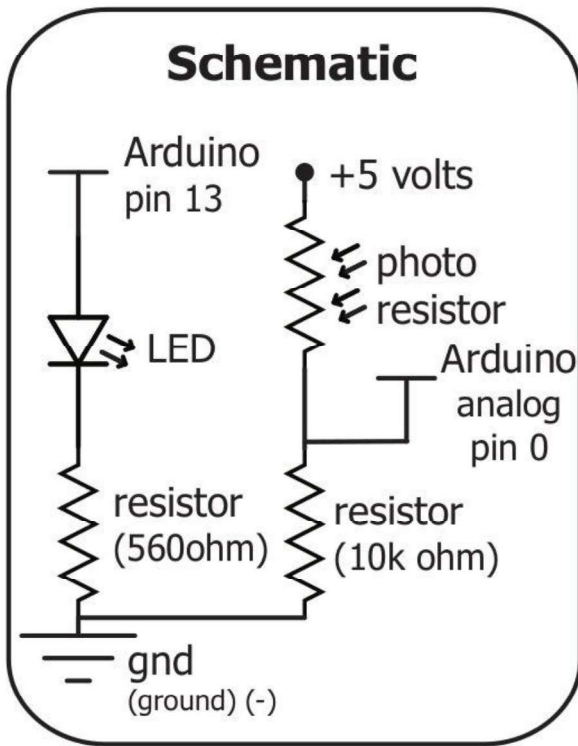
Photo resistor as light sensor

تصنع المقاومات الضوئية عادة من نوع خاص من السيلكون المُعالج، ويعتبر السيلكون المُعالج ماده شبه موصله للكهرباء و تتأثر مدى توصيليته للكهرباء بالضوء لذلك يستخدم في الحساسات الضوئية و الخلايا الشمسية .



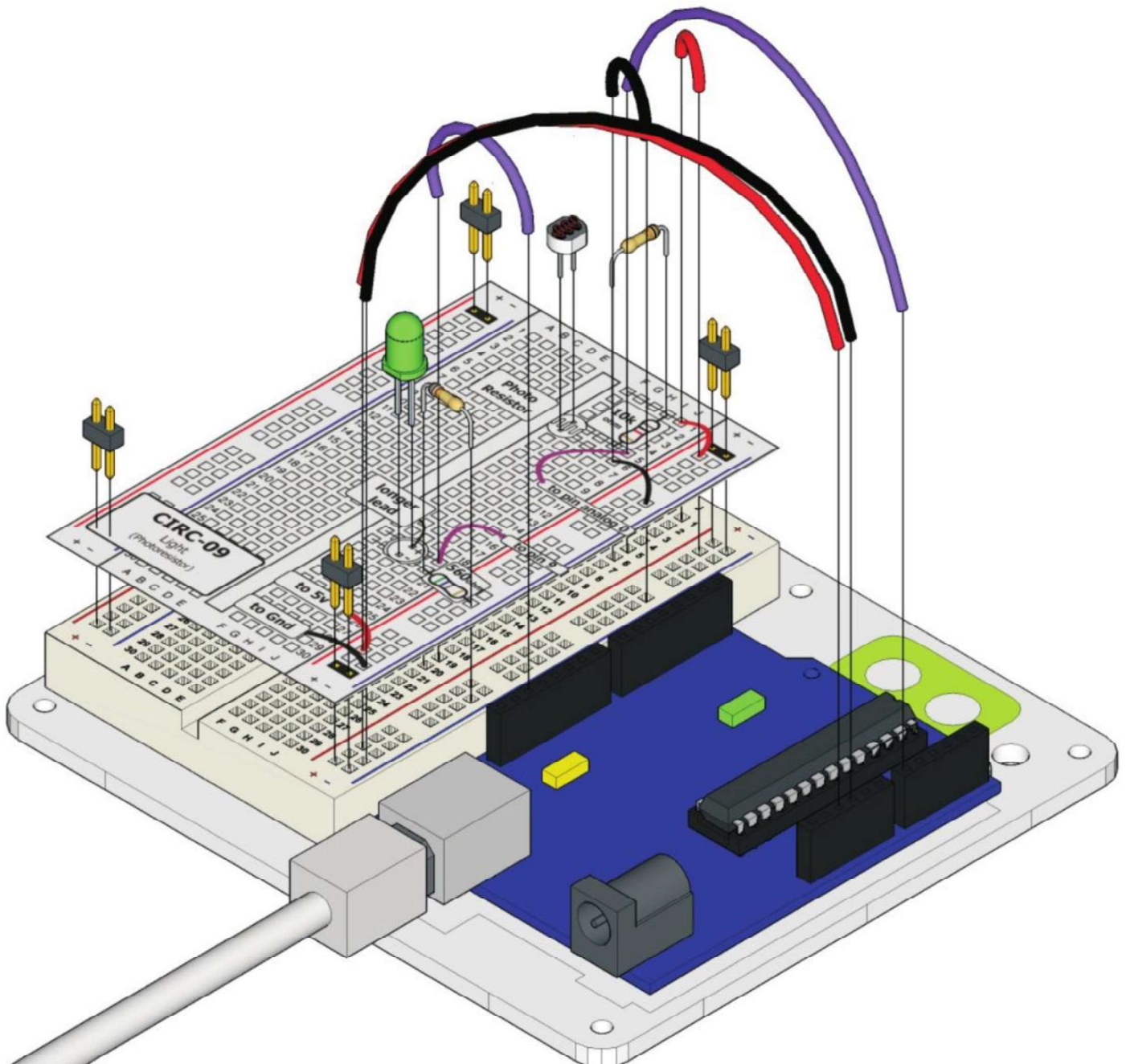
مكونات المثال (5):

- ✓ بورده اردوينو Arduino Uno
- ✓ لوحه تجارب Breadboard
- ✓ دايود ضوئي led 5mm
- ✓ مقاومه 560 اوم
- ✓ مقاومه ضوئيه
- ✓ اسلاك توصيل
- ✓ ورقه المساعدة المطبوعه



تتوفر المقاومات الضوئية في احجام مختلفه تبعا
لمقدار حساسيتها للضوء ويمكن استخدام اى
حجم ممتوفر





```
//Example_5_Light_Sensor_led
const int lightPin = A0;
const int ledPin = 9;
int lightLevel;

void setup ( )
{
  pinMode(ledPin, OUTPUT);
}

void loop ( )
{
  lightLevel = analogRead(lightPin);
  lightLevel = map(lightLevel, 0, 900, 0, 255);
  lightLevel = constrain(lightLevel, 0, 255);
  analogWrite(ledPin, lightLevel);
}
```

قم بتنفيذ البرنامج ورفعه على بورده المتحكمه الدقيقه ثم شاهد ماذا سيحدث للدايود الضوئى عند تعريض المقاومه الضوئيه للضوء، ثم حاول ان تقلل مقدار الضوء الساقط على المقاومه الضوئيه عن طريق تغطيتها بيدك و شاهد ماذا سيحدث الى الدايود الضوئى.

رقم المخرج

قيمه الخرج



الآن سنتعرف `analogWrite (pin number, value);` وهى الداله المسئوله عن توليد خرج تماثلى `analog Output` ويمكن تطبيقها على المخارج التى تدعم خاصيه التعديل الرقمى المعتمد على عرض النبضة (Pulse-Width modulation) وهى المخارج التى يكتب بجانبها علامه `(~)` . وهى 6 مخارج كالتالى: 3,5,6,9,10,11

ما هي خاصيه التعديل الرقمى PWM؟؟

خاصيه تمكّنك من انتاج فرق جهد على هيئة موجة قابله للتغير عن طريق اشارته رقميه، وبذلك تستطيع أن تحول الأوامر الرقمية إلى موجة تماثليه ويمكن استغلال تلك الخاصية في توليد جهد متغير قيمته بين صفر فولت حتى 5 فولت عن طريق استخدام قيم رقميه من 0 حتى 255 مثلاً:

$$255 = 5 \text{ volt}$$

$$128 = 2.5 \text{ volt}$$

$$0 = 0 \text{ volt}$$

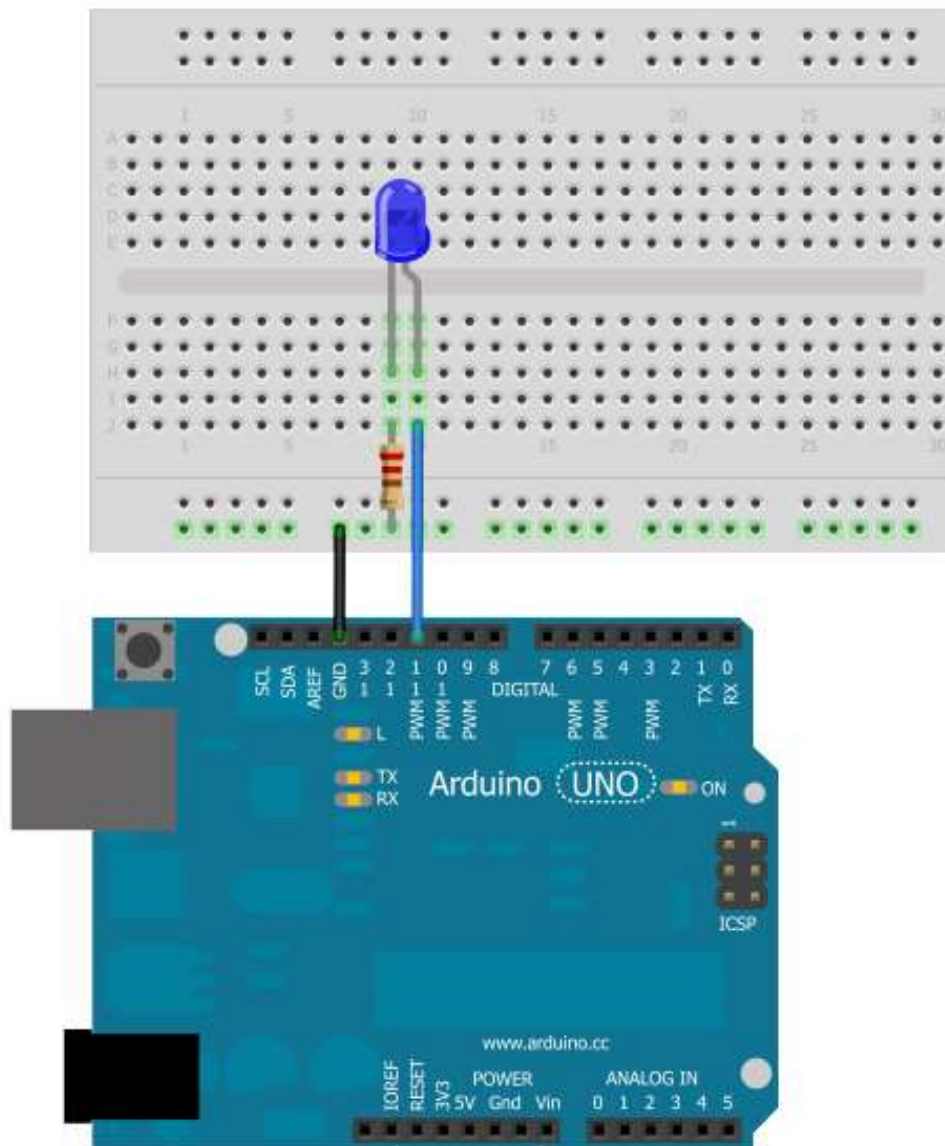
كيف نستفيد من هذه الخاصيه؟؟

الكثير من المكونات الالكترونيه والكهريه تتفاعل مع القيم المختلفه من فرق الجهد بصور مختلفه مثل الداىود الضوئى، مثلاً عندما تطبق 3 فولت على الداىود الضوئى ستجد ان اضاءته ضعيفه واذا رفعت الجهد الى 4 فولت ستجد الاضاءه قد ازدادت واذا وصلت الى 5 فولت ستجد الاضاءه وصلت الى اعلى معدل سطوع.. وهناك مكونات كهريه مثل

المحرك الكهربى (الموتور) و الذى تردداد سرعته بزياده الفولت الكهربى.
في هذا المثال وصلنا الداىود الضوئى على المخرج رقم 9 (الذى يدعم PWM) و
 استخدمنا مقاومه ضوئيه حتى تستشعر كميه الضوء الموجود حولها ثم ترسل الاشاره
 التماثليه الى اردوينو فيقوم بتشغيل الليد بناءا على الاشاره التماثليه من المقاومه الضوئيه .

لنأخذ مثال آخر على خاصيه التعديل الرقمي

المثال السادس:المؤثرات الضوئيه بأستخدام PWM



في هذا المثال سنقوم بتوصيل دايود ضوئي أزرق (اخترت اللون الأزرق لان أضاءته عالية مقارنة بالألوان الأخرى ويمكنك اختيار اللون الأبيض كذلك) إلى المنفذ رقم 11 والذي يدعم خاصية PWM و سنستغل تلك الخاصية في عمل بعض المؤثرات الضوئية الجميلة

والآن نأتي إلى الكود البرمجي:

```
// Example_6_fading_led_light
const int ledPin = 11;
int i = 0;

void setup( )
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  for (i = 0; i < 255; i++)
  {
    analogWrite(ledPin, i);
    delay(10);
  }
  for (i = 255; i > 0; i--)
  {
    analogWrite(ledPin, i);
    delay(10);
  }
}
```

العداد الأول

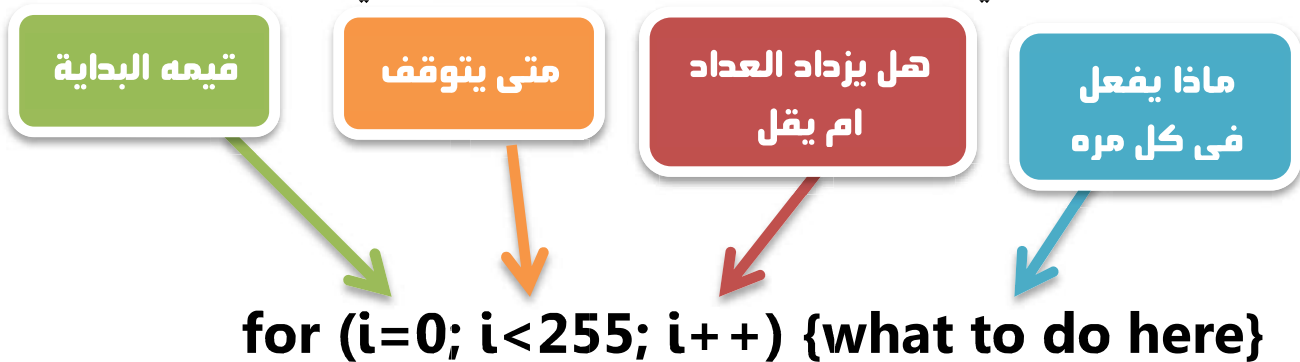
العداد الثاني

بعد الانتهاء من كتابة الكود و رفعه إلى المتحكم الدقيقة.. لاحظ ماذا يحدث للدايود

الضوئي وكيف تزداد إضاءته بهدوء و تنخفض بهدوء ☺

سنتعلم الآن احدى اهم الأوامر البرمجية و هي حلقات التكرار

في هذا المثال شاهدنا امر جديد وهو `for (i =0; i < 10; i++)` يستخدم هذا الأمر في عمل ما يسمى بالحلقة المتكررة `loop` ويمكن استخدام الحلقات في أداء وظيفه تتكرر اكثر من مره بعدد معروف من المرات أو الي ما لا نهاية ويمكن استغلال الحلقات في عمل ما يسمى بالعدادات `counters` وهي برامج صغيره تقوم بالعد التزايدى أو التناقصي كما يريد المبرمج ، و تكون صيغه الأمر كتالي:



في المثال السابق استخدمنا عداد يقوم بعد الأرقام ابتداءً من صفر حتى 255 وفي كل مره يضيف واحد و يخزن القيمة في المتغير (`i`) ثم يقوم بتطبيق جهد تماثلي بقيمه `i` على الداىود الضوئي باستخدام الأمر `analogWrite(ledPin,i)` وبذلك نحصل على ذلك التأثير الضوئي الجميل حيث نرى الداىود الضوئي يبدأ في الوميض بصورة خافته ثم تزداد شدة الإضاءة تدريجيا بزيادة قيمه المتغير (`i`) في العداد الأول.

في العداد الثاني استخدمنا نفس المبدأ لكن بصورة معكوسة بحيث يبدأ العداد من 255 ويظل الرقم يتناقص بمقدار واحد حتى يصل إلى الصفر وفي كل مره يسجل العداد قيمه المتغير (i) و يطبق جهد تماثلي على الدايمود الضوئي بقيمه (i) وهذا ما يجعل الدايمود الضوئي يبدأ في الوميض بشده ثم تنخفض شدة إضاءته تدريجيا بتغير قيمه (i)

قد يتسائل البعض لماذا استخدمنا الأم: delay(10) داخل كل عداد ؟؟

الأجابه تتمثل في التحكم في سرعه اردوينو اثناء تطبيق الجهد التماثلي ولكي يتضح المعنى بصورة عمليه جرب ان تغير الرقم و تجعله مره = 20 بحيث يكتب الأمر delay(20); و مره أخرى تجعله 50 و مره أخرى تجعله = 5 و شاهد ماذا سيحدث للدايمود الضوئي.

المحاضرة 6 - المقاومة الضوئية LDR

الاختبار البعدي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. وفق مخطط التجربة، إلى أي مدخل تناطري تصل إشارة LDR؟

- A) Analog pin 0 / A0
- B) A5
- C) 13
- D) 2

2. ما قيمة المقاومة الموضحة مع LED في مخطط التجربة؟

- A) 560 ohm
- B) 10 ohm
- C) 1 megaohm
- D) 100 kohm

3. ما قيمة المقاومة الموضحة في مقسم الجهد مع LDR؟

- A) 10 kohm
- B) 560 ohm
- C) 1 ohm
- D) 100 ohm

4. ما وظيفة مقسم الجهد مع LDR؟

- A) تحويل تغير المقاومة إلى جهد يمكن قراءته
- B) تشغيل USB
- C) زيادة عدد المنافذ
- D) عرض نص

5. عند تغير شدة الضوء، ماذا نتوقع من قراءة A0؟

- A) تتغير
- B) تبقى ثابتة دائماً
- C) تصبح حرفاً
- D) تتحول إلى عنوان I2C

6. أي عنصر في التجربة يمثل Output بصرياً؟

- A) LED
- B) LDR
- C) المقاومة 10k
- D) A0

7. لماذا تستخدم مقاومة مع LED؟

- A) للحد من التيار
- B) لزيادة عدد الحساسات
- C) لتخزين القراءة
- D) لإنشاء Serial

8. أي وصف أدق للـ LDR؟

(A) مقاومة حساسة للضوء

(B) محرك زاوي

(C) شاشة أحرف

(D) مفتاح ميكانيكي

9. ما مصدر التغذية الظاهر في المخطط للحساس؟

A) 5V

B) 24V

C) 220V

D) 1V فقط

10. إذا كانت قراءة الضوء مطلوبة برمجياً، فأى دالة مناسبة؟

A) analogRead

B) digitalWrite

C) Servo.write

D) lcd.clear

المحاضرة 7 - ربط Arduino بالحاسوب

الاختبار القبلي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. ما وسيلة الاتصال الشائعة بين Arduino Uno والحاسوب في المحاضرة؟

- A) USB/Serial
- B) I2C فقط
- C) Bluetooth فقط
- D) HDMI

2. ما الأداة المستخدمة لمشاهدة البيانات التسلسلية؟

- A) Serial Monitor
- B) LCD فقط
- C) Keypad
- D) Relay

3. ما الأمر الذي يبدأ الاتصال التسلسلي؟

- A) Serial.begin
- B) analogRead
- C) digitalWrite
- D) pinMode

4. ما السرعة المستخدمة في الأمثلة؟

- A) 9600
- B) 100
- C) 255
- D) 1024

5. أي حساس استُخدم لإرسال درجة الحرارة للحاسوب؟

- A) LM35/TMP36
- B) Keypad
- C) Servo
- D) Relay

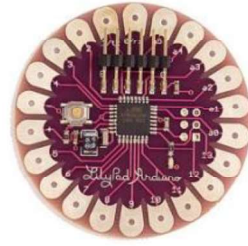
الفصل الخامس

أرسال البيانات و الأوامر من اردوينو إلى الحاسب الآلي و العكس
Computer interfacing with Arduino



جميع بوردرات اردوينو (lilypad, mini) بخلاف

توفر إمكانية إرسال واستقبال البيانات مباشرة مع الحاسب الآلي computer interface وذلك عن طريق منفذ الـ USB ويمكن استغلال هذه الخاصية في الكثير من المشاريع الرائع، وبالنسبة إلى البوردرات التي لا تدعم هذه الخاصية فهي



Liliypad

تحتاج إلى ما يسمى بالـ FTDI interface وهي شريحة صغيرة تستخدم لتبادل البيانات بين الحاسب الآلي والمتحكمات الدقيقة

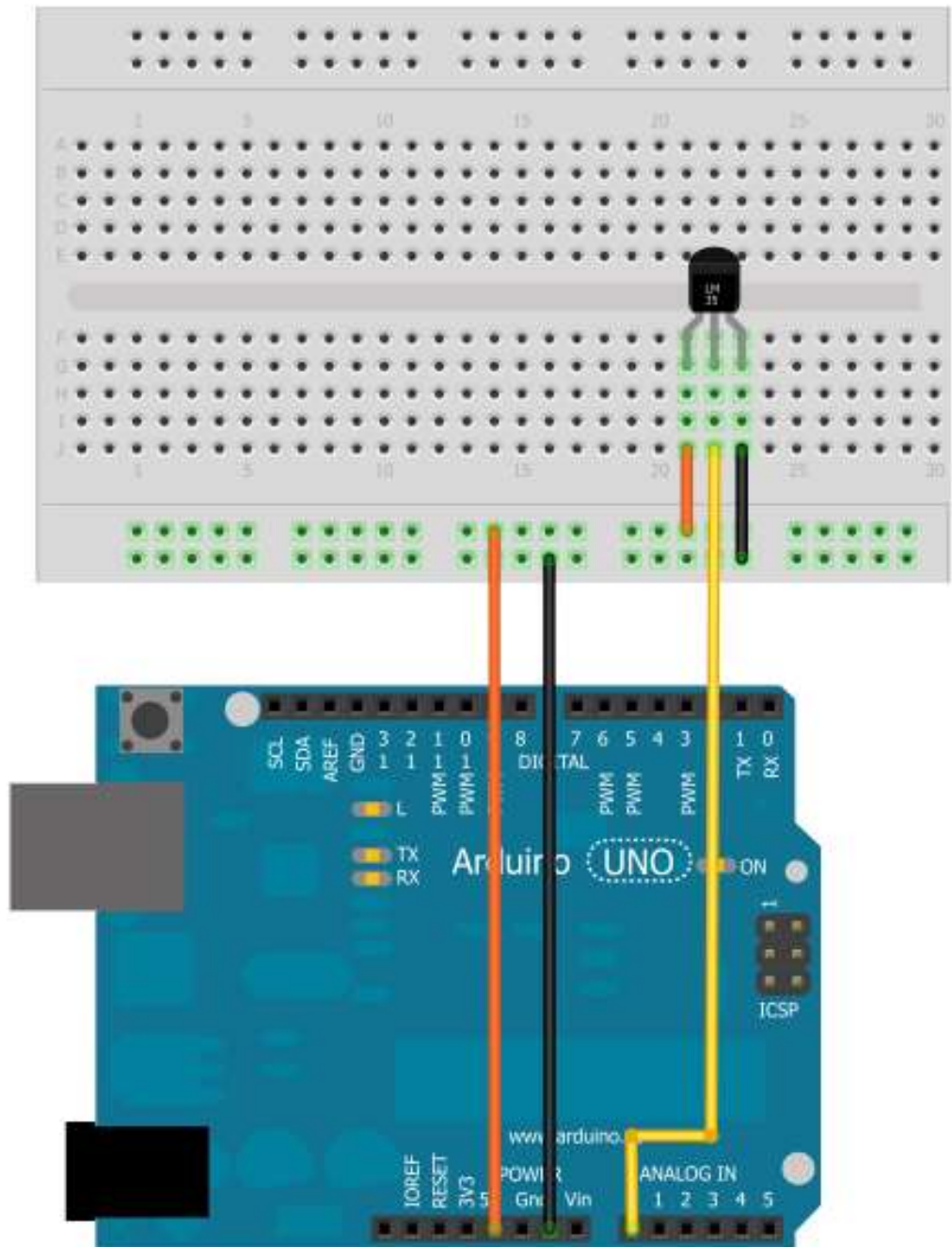


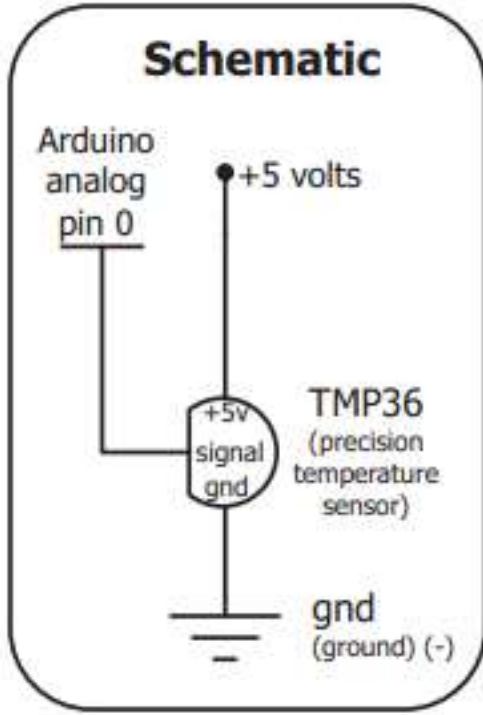
FTDI chip

يتحدث الفصل الخامس عن إمكانية استخدام اردوينو في الحصول على معلومات من الحساسات المختلفة وإرسال النتائج لعرضها على الحاسب الآلي أو اخذ أوامر من الحاسب الآلي لتنفيذها على المكونات الإلكترونية الموصلة على البورده.

في الأمثلة السابقة كنا قد استخدمنا اردوينو في قراءة بعض الحساسات مثل الضوء و الحرارة وإظهار ناتج الحساس على دايود ضوئي ليوضح مقدار هذه القراءة أما الآن فسنستخدم خاصية serial interface لإرسال نتائج القراءة الحساسات إلى الحاسب الآلي حيث يمكن قرائه النواتج بالأرقام وإجراء بعض العمليات الحسابية عليها.

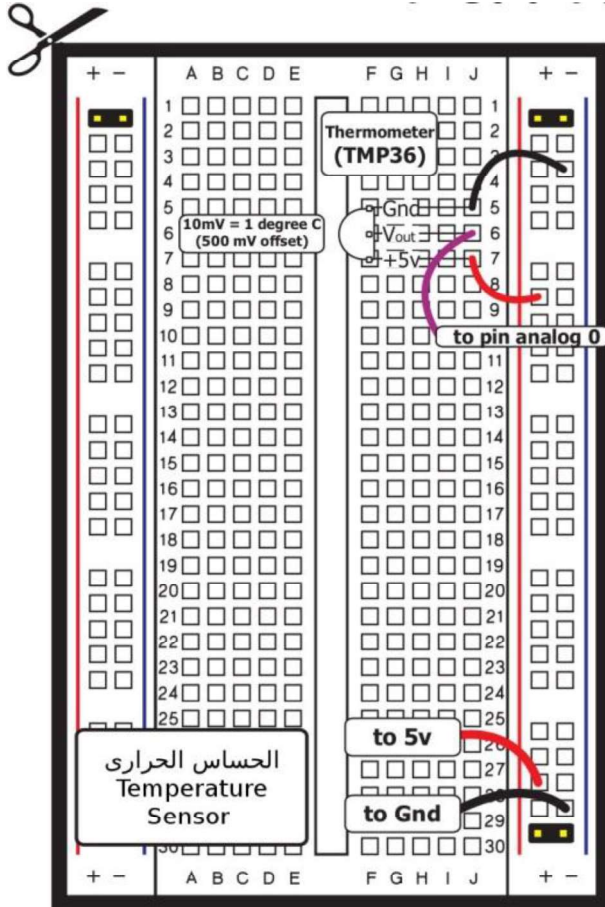
المثال السابع: استخدام حساس الحرارة و إرسال درجة الحرارة إلى الحاسب الآلي



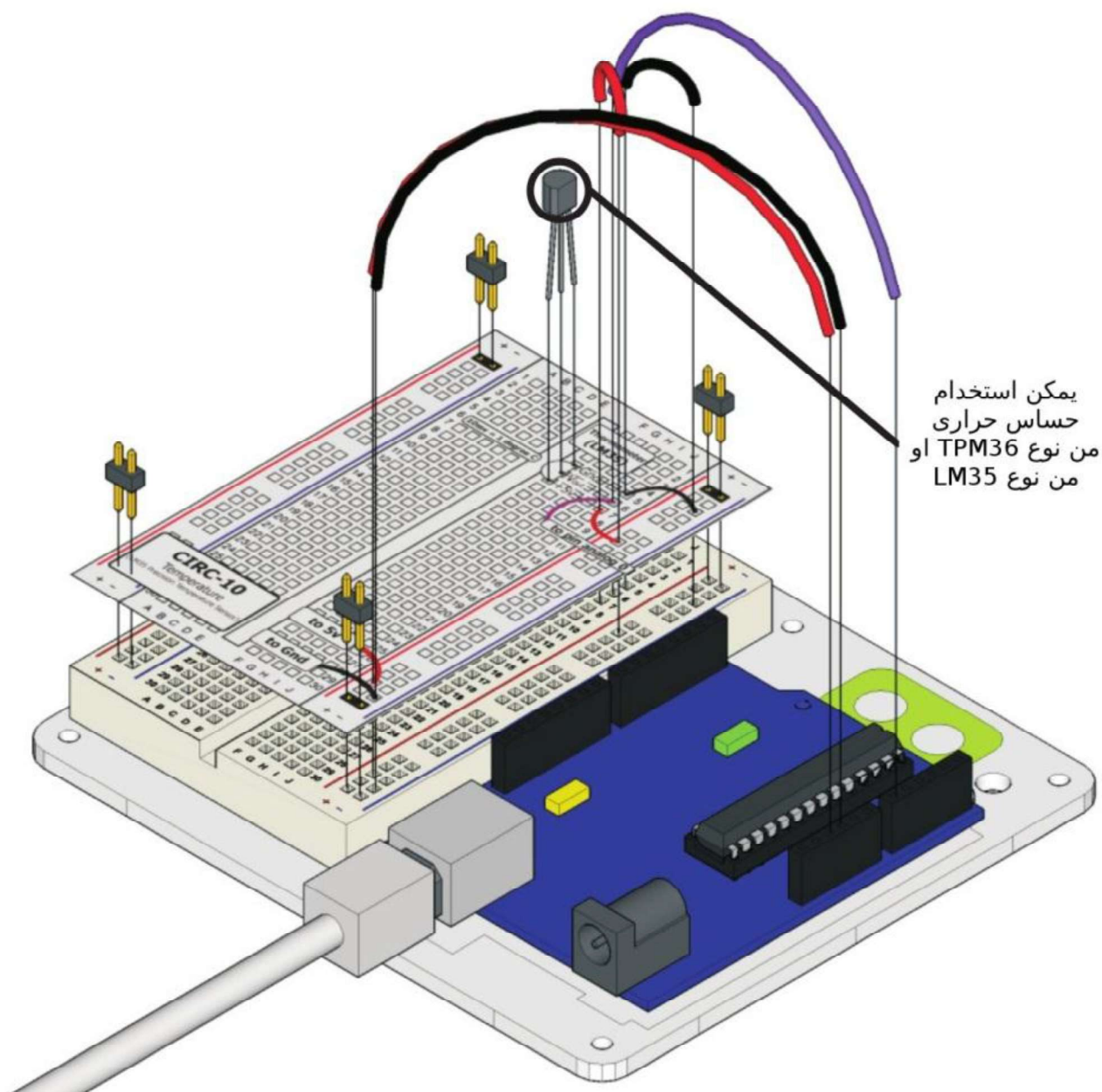


مكونات المثال (7):

- ✓ بورده اردوينو Arduino Uno
- ✓ لوحه تجارب Breadboard
- ✓ حساس حرارى LM35 او TMP 36
- ✓ أسلاك توصيل
- ✓ كابل التوصيل بالـ USB
- ✓ ورقه المساعدة المطبوعة



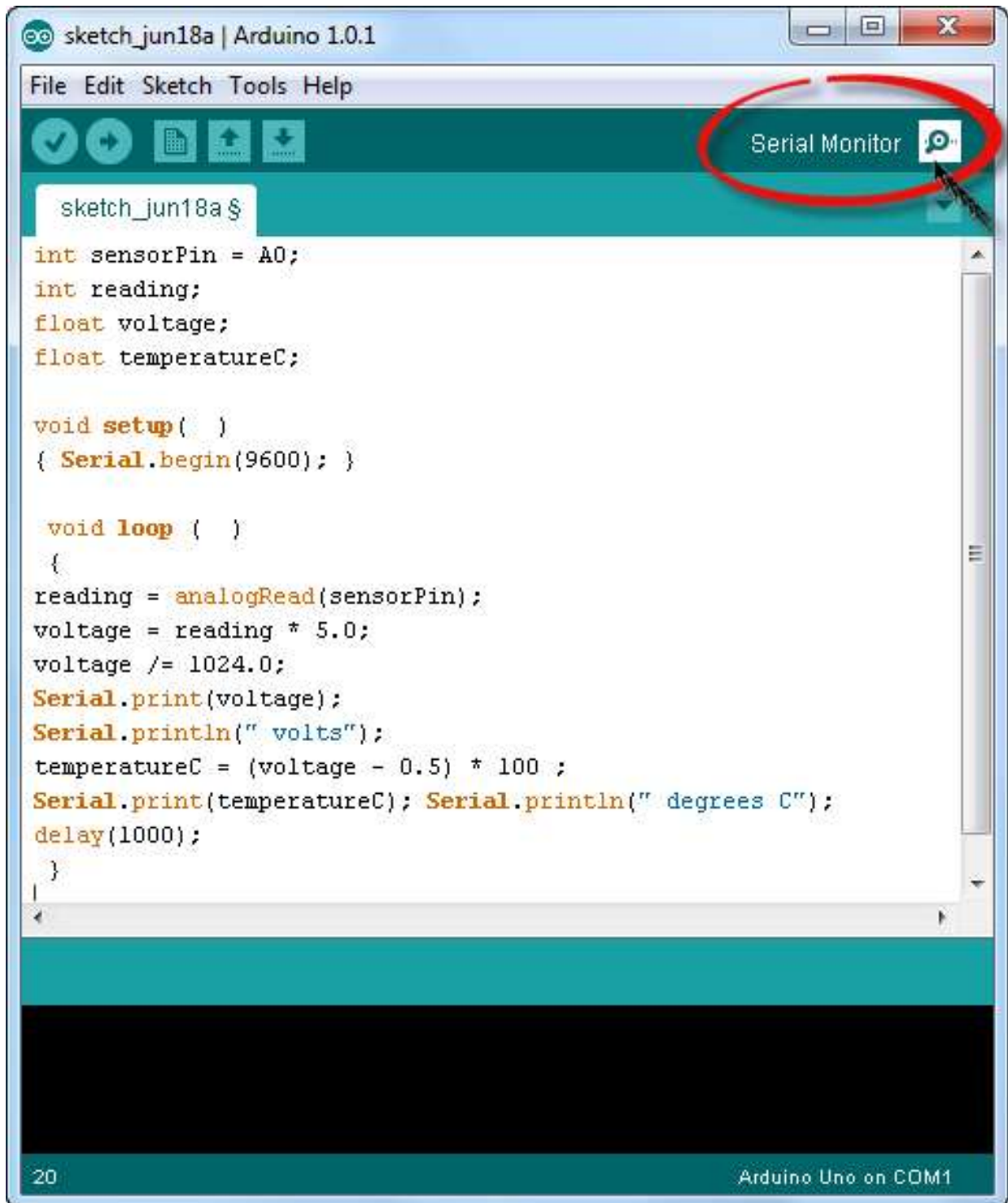
لاحظ انه في حاله استخدامك لحساس حرارى مختلف عن النوعين المذكورين في الأعلى فأن المداخل و المخرج الخاصة بالجهد قد تختلف ويمكنك معرفتها من خلال قراءة الداتا شيت الخاصة بالحساس المستخدم



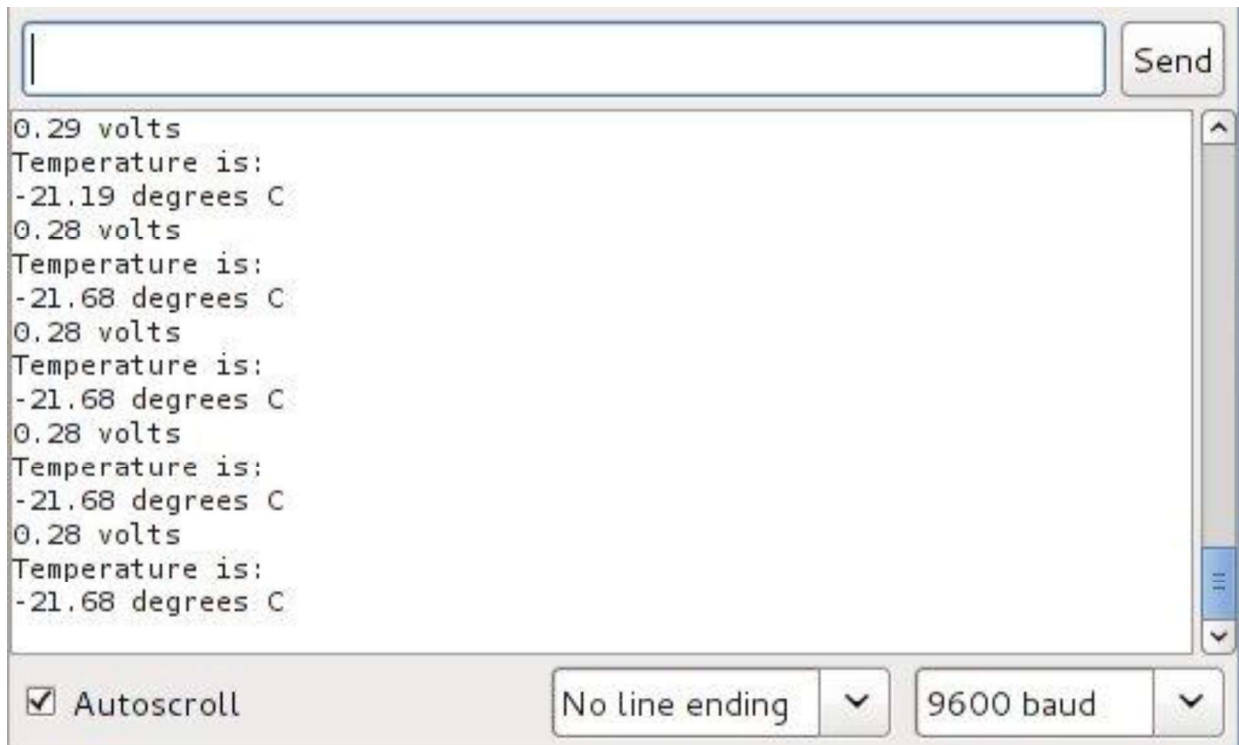
بعد الانتهاء من توصيل المكونات على لوحه التجارب قم بكتابه الأكواد التالية و تأكد
من صحتها عن طريق زر **Verify** ثم ارفعها إلى بورده اردوينو

```
// Example_7_Temperature_Sensor  
const int sensorPin = A0;  
  
int reading;  
  
float voltage;  
  
float temperatureC;  
  
void setup( )  
{ Serial.begin(9600); }  
  
void loop ( )  
{  
  reading = analogRead(sensorPin);  
  voltage = reading * 5.0/1024;  
  Serial.print(voltage);  
  Serial.println(" volts");  
  temperatureC = (voltage - 0.5) * 100 ;  
  Serial.println("Temperature is: ");  
  Serial.print(temperatureC);  
  Serial.println(" degrees C");  
  delay(1000);  
}
```

بعد الانتهاء من كتابه ورفع الكود البرمجي إلى المتحكم نقوم بالضغط على زر Serial Monitor الموجود في شريط الأوامر السريعة كما في الصورة التالية:



ستظهر صفحة خاصه تستقبل البيانات من بورده اردوينو و تقوم بعرض درجه الحرارة كل ثانيه كما سنلاحظ أن الدايدو الضوئي الموجود على بورده اردوينو المكتوب بجانبه Tx بدأ في الوميض بسرعه و ذلك يعنى أن اردوينو يرسل بيانات إلى الحاسب الآلي ...



الآن حاول ن ترفع درجه حراره الحساس عن طريق تقريب شمعه مشتعلة أو عود ثقاب مشتعل أو أي مصدر حرارى متوفر (حذارى أن تمس النار الحساس الحرارى فذلك قد يتلفه ، هناك حساسات حرارية خاصه يمكنها تحمل الحرارة الشديدة لكن الحساس المستخدم في هذا المثال يستطيع تحمل حراره تصل إلى 150 درجه مئوية تقريبا)

ملحوظه: علامه الشرطه (-) التي تكتب قبل درجه الحرارة لا تعنى سالب درجه مئوية و إنما خطأ برمجى مؤقت في أول السطر عند استقبال بيانات من اردوينو

شرح الكود البرمجي

يستخدم الأمر `Serial.begin(9600);` في جعل بورده أردوينو تبدأ التواصل مع الحاسب الآلي عن طريق منفذ الـ USB و بهذا يمكن للأردوينو إرسال أو استقبال بيانات من وإلى الحاسب .

نلاحظ انه هناك متغيران وهما `voltage` و `temperatureC` تم تعريفهم باستخدام الأمر `float` بدلا من استخدام الأمر `int` وذلك لان الحساس الحراري يقيس درجة الحرارة بدقه عالية تصل إلى 0.1 درجة مئوية و من المتوقع أن تكون قيمه الجهد الكهربائي الناتج عنه و درجة الحرارة بالكسور العشرية و ليس الأعداد الصحيحة لذلك استخدمنا الأمر `float` لجعل هذه المتغيرات تقبل قيمه تحتوى على كسور عشريه .

يستخدم الأمر `reading = analogRead(sensorPin);` في تسجيل قراءة الدخل التماثلي على المتغير `reading` و هي القيمة الناتج من `sensorPin` و الذي يرمز إلى المدخل A0 .

كما ذكرنا سابقا ان المتحكم الدقيقة تقوم بتحويل الإشارة التماثلية إلى قيمه رقميه من صفر الى 1024 لذلك استخدمنا الأمر التالي لتحويل هذه القيمة الرقمية إلى قيمه فولتيه نستطيع فهمها

```
voltage = reading * 5/1024;
```

بعد تحويل القيمة الرقمية إلى قيمه فولتيه استخدمنا الأمر: `Serial.print(voltage);` لأرسال هذه القيمة إلى الحاسب الآلي لعرضها على بيئة التطوير باستخدام الـ `Serial` Monitor ثم نستخدم الأمر: `Serial.println("voltage");` ليقوم بطابعه كلمه `voltage` بعد كل قيمه فولت يعرضها على شاشة الحاسب الآلي

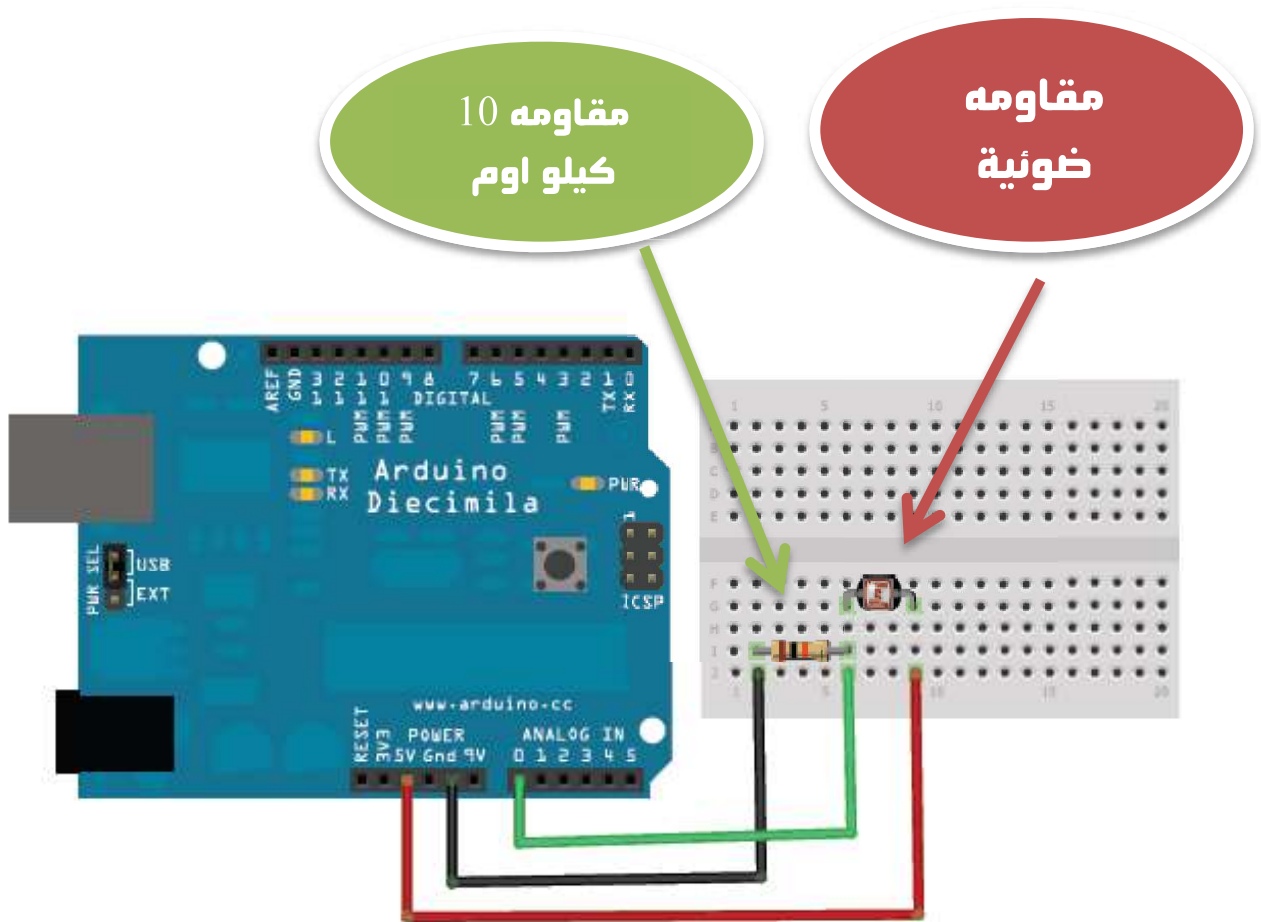
نستخدم الأمر: `temperatureC = (voltage - 0.5) * 100;` لتحويل قيمه الفولت إلى درجه حراره بوحده سي لزيوس Celsius ثم نكرر نفس أوامر `Serial.print` مع درجه الحرارة لعرضها على شاشة الحاسب الآلي

```
Serial.print(temperatureC);
Serial.println(" degrees C ");
```

الأمر الأخير هو: `delay(1000);` وذلك لنجعل المتحكمه الدقيقة تنتظر ثانيه واحده قبل أن تعيد إرسال درجه الحرارة و الفولت إلى الحاسب الآلي مره أخرى.

المثال الثامن: استخدام حساس الضوء و عرض شدة الإضاءة على الحاسب الآلي

هل تتذكر المثال الخامس، سنقوم بأعاده تنفذ ذلك المثال ولكن بدلا من عرض شدة الإضاءة على الدايمود الضوئي led ، سنعرضها على الحاسب الآلي .



```
// Example_8_Light_Sensor_2
const int photocellPin = A0;
int photocellReading;

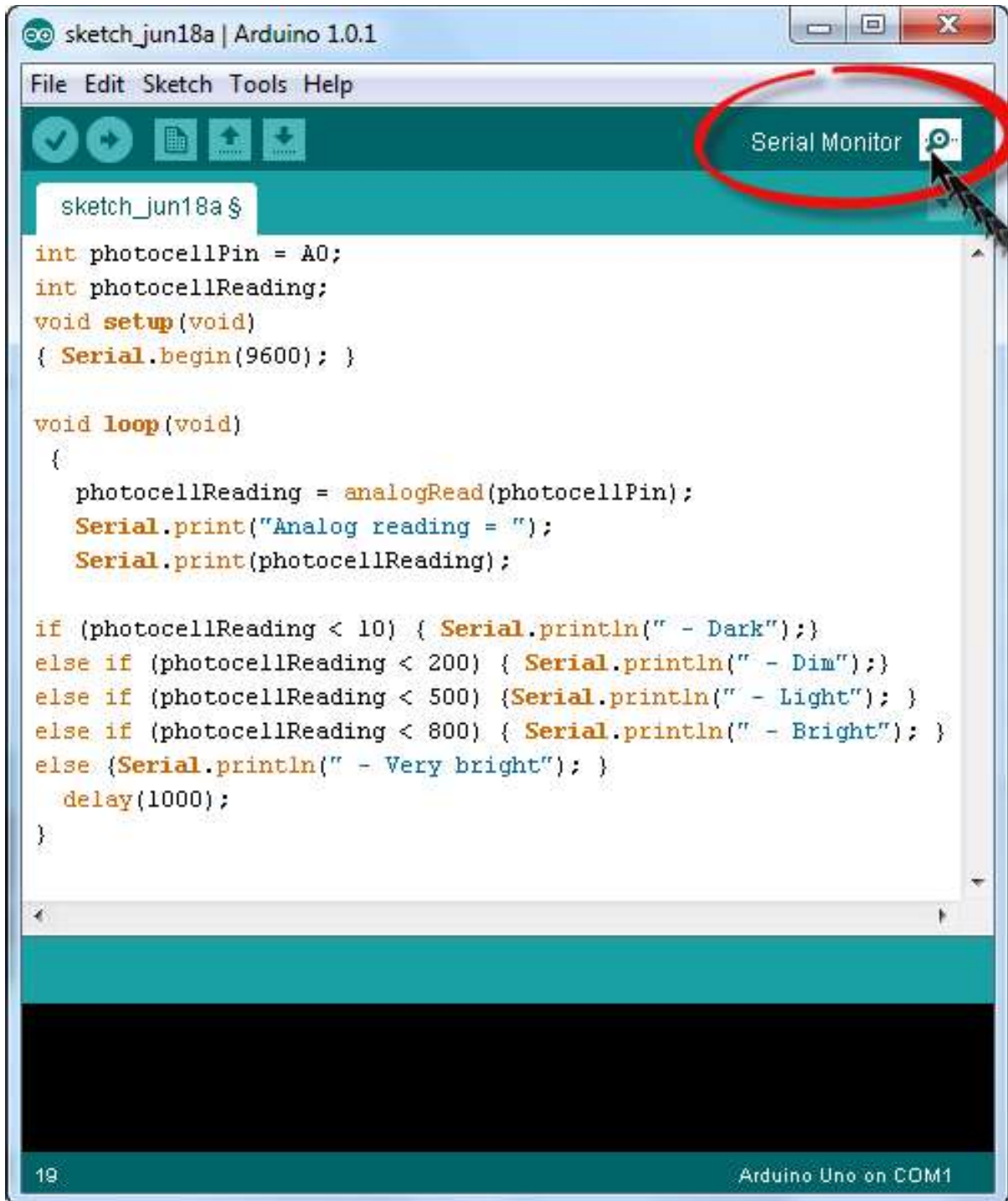
void setup(void)
{ Serial.begin(9600); }

void loop(void)
{
    photocellReading = analogRead(photocellPin);
    Serial.print("Analog reading = ");
    Serial.print(photocellReading);

    if (photocellReading < 10) { Serial.println(" - Dark");}
    else if (photocellReading < 200) { Serial.println(" - Dim");}
    else if (photocellReading < 500) {Serial.println(" - Light"); }
    else if (photocellReading < 800) { Serial.println(" - Bright"); }
    else {Serial.println(" - Very bright"); }

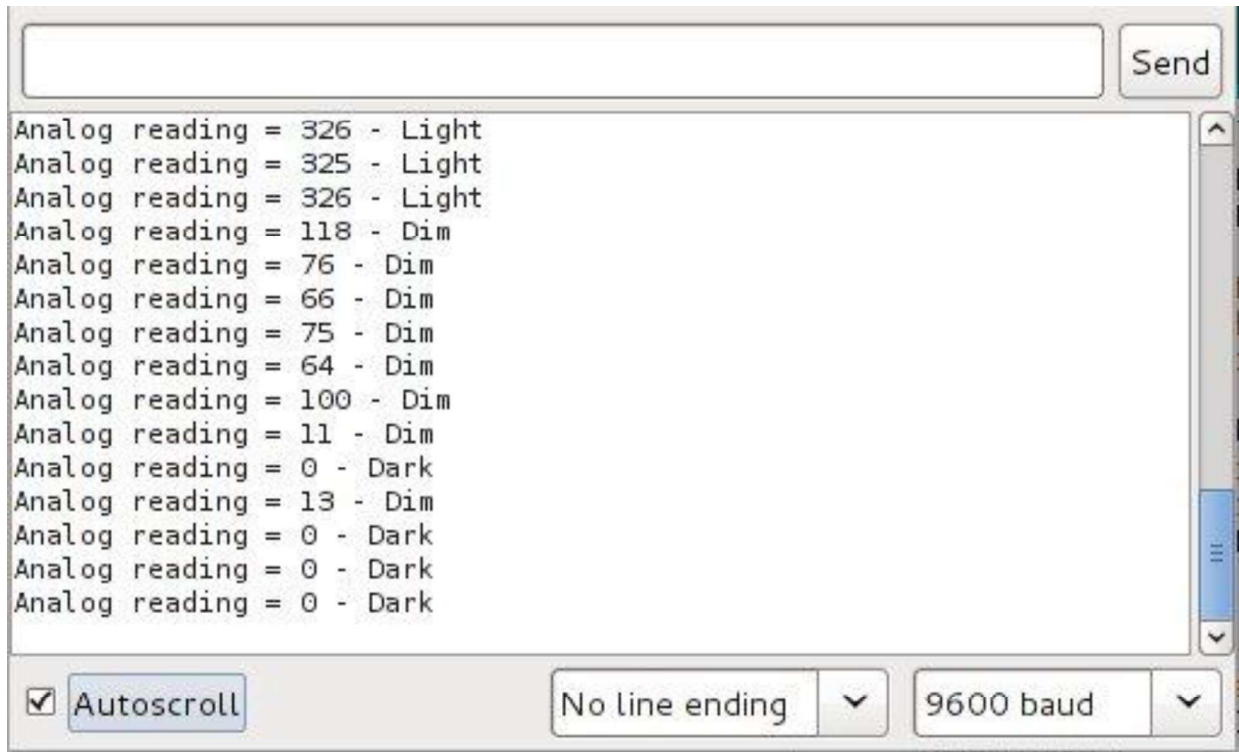
    delay(1000);
}
```

بعد الانتهاء من كتابه ورفع الكود البرمجي إلى المتحكم نقوم بالضغط على زر Serial Monitor الموجود في شريط الأوامر السريعة كما في الصورة التالية :



الآن حاول أن تغير مستوى الإضاءة الذى تتعرض له المقاومة الضوئية عن طريق :

- تعريضها للضوء الساطع مباشرة
- تغطيتها بقطعه قماش خفيفة
- تغطيتها بيديك لتحجب عنها الضوء تماما



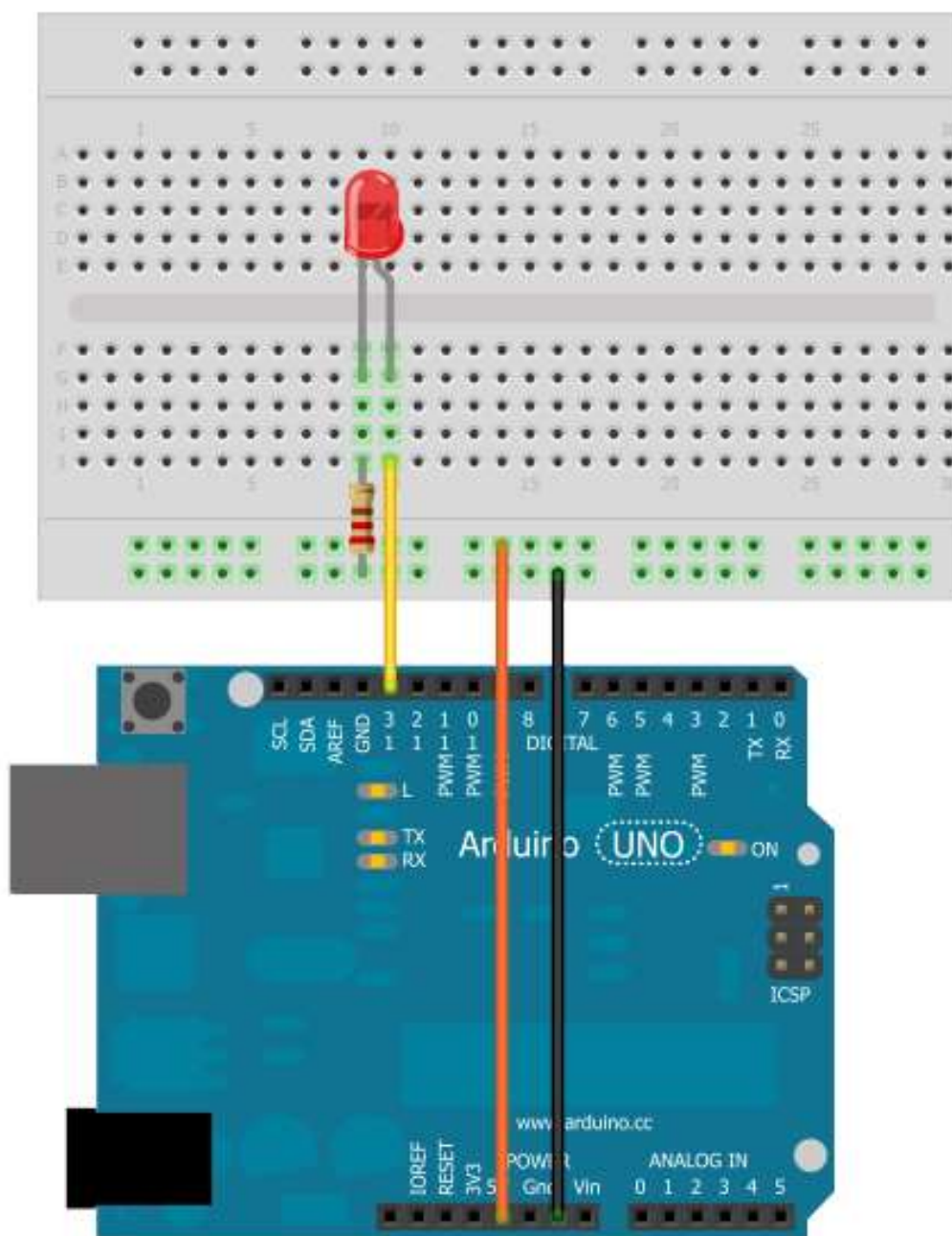
```

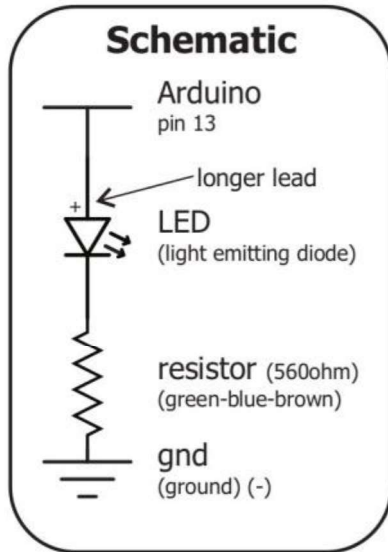
Analog reading = 326 - Light
Analog reading = 325 - Light
Analog reading = 326 - Light
Analog reading = 118 - Dim
Analog reading = 76 - Dim
Analog reading = 66 - Dim
Analog reading = 75 - Dim
Analog reading = 64 - Dim
Analog reading = 100 - Dim
Analog reading = 11 - Dim
Analog reading = 0 - Dark
Analog reading = 13 - Dim
Analog reading = 0 - Dark
Analog reading = 0 - Dark
Analog reading = 0 - Dark
  
```

ملحوظه:

- Dim تعنى إضاءة خافته
- Dark تعنى ظلام دامس
- Light تعنى إضاءة معتدلة
- Bright Light تعنى إضاءة شديده

المثال التاسع: تشغيل دايود ضوئي عن طريق استقبال أمر من الحاسب الآلي





مكونات المثال (9):

✓ بورده اردوينو Arduino Uno

✓ لوحة تجارب

✓ دايود ضوئي led 5mm

✓ مقاومه 560 اوم

✓ أسلاك توصيل

يعتبر المثال التاسع تطوير للمثال الأول والثاني ، وفي هذا المثال سنستخدم الحاسب

الآلي في التحكم في الدايود الضوئي بدلا من السويتش، سنجعل اردوينو يستقبل امر

التشغيل والإطفاء عن طريق الـ USB باستخدام خاصيه الـ Serial Monitor في بيئة

تطوير اردوينو Arduino IDE

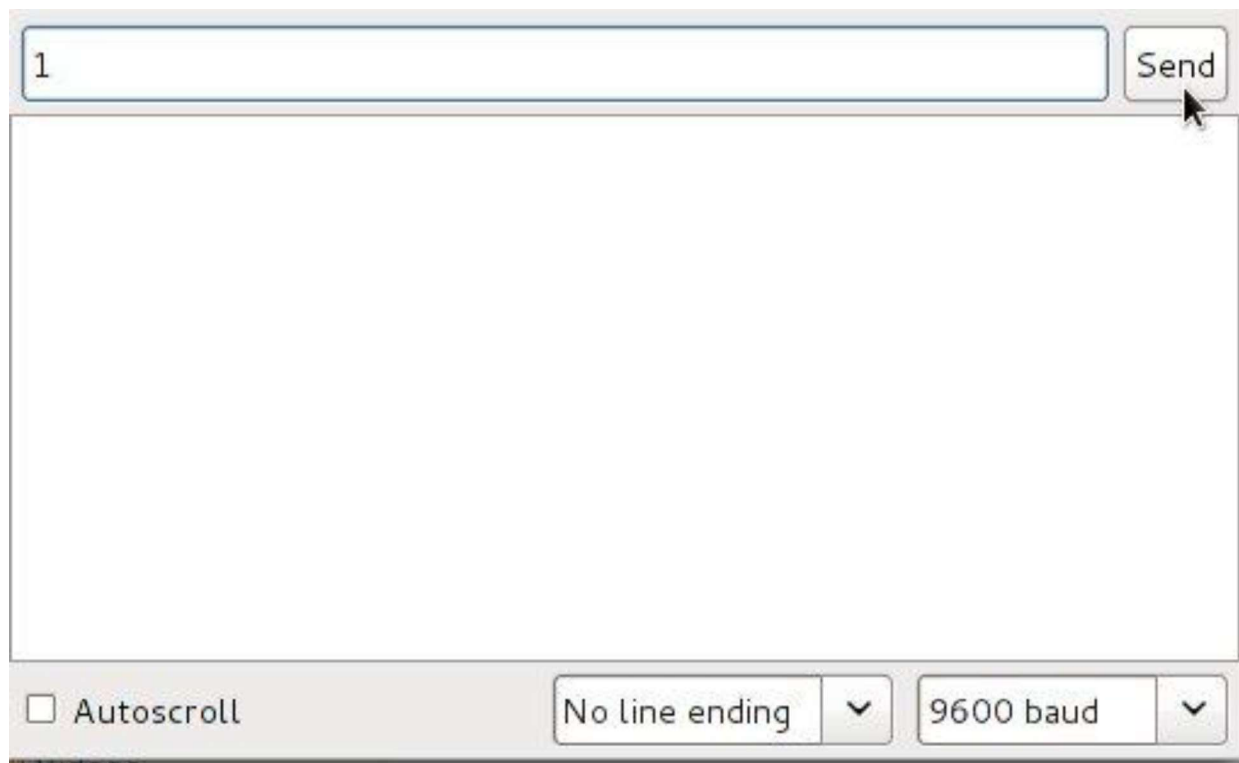
الكود:

```
// Example_9_Computer_Interfacing
int ledPin=13;
int value;

void setup ()
{
  Serial.begin(9600);
  pinMode(ledPin,OUTPUT);
}

void loop ()
{
  value = Serial.read();
  if (value == '1') {digitalWrite(ledPin,HIGH);}
  else if (value == '0') {digitalWrite(ledPin,LOW);}
}
```

بعد الانتهاء من كتابة الكود البرمجي و رفعه إلى اردوينو اضغط على أيقونة serial Monitor ستجد شريط كتابي في اعلى الصفحة قم بكتابه 1 و اضغط زر إرسال Send و شاهد ماذا سيحدث للدايود الضوئي ، ثم اكتب الرقم صفر و اضغط على زر إرسال مره أخرى و شاهد ماذا سيحدث ...



1

Send

☐ Autoscroll

No line ending ▼

9600 baud ▼



0

Send

☐ Autoscroll

No line ending ▼

9600 baud ▼

في هذا المثال استخدمنا الأمر: `Serial.read()` وهو الأمر المستخدم في قراءة البيانات المرسلة من الحاسب الآلي إلى اردوينو عبر منفذ الـ USB وقمنا بإضافة `value =` قبل هذا الأمر وذلك حتى تقوم المتحكمة بقراءة ما يرسل من الـ USB، تخزن هذه القيمة في المتغير `value`

ثم استخدمنا جملة `if.. else if` لوضع شرط:

- إذا كانت قيمة المتغير `value == 1` تقوم المتحكمة بتشغيل الدايود الضوئي
- أما إذا كانت قيمة المتغير `value == 0` تقوم المتحكمة بإطفاء الدايود الضوئي

الآن حاول أن تعدل الكود بنفسك و تريد عدد المخرج التي يمكن التحكم بها عن طريق الحاسب الآلي

مصادر اضافيه للمعلومات:

لمزيد من المعلومات عن الأوامر المستخدمة في الـ Serial communication تفضل الرابط التالي لمرجع الأوامر الرسمي:

<http://arduino.cc/en/Reference/serial>

المحاضرة 7 - ربط Arduino بالحاسوب

الاختبار البعدي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. في كود حساس الحرارة، ما نوع المتغير **voltage**؟

- A) float
- B) char
- C) boolean فقط
- D) byte فقط

2. ما وظيفة **Serial.print**؟

- A) إرسال/عرض بيانات عبر الاتصال التسلسلي
- B) تشغيل محرك
- C) قراءة زر
- D) تهيئة LCD

3. ما وظيفة **Serial.println** مقارنة بـ **Serial.print**؟

- A) تضيف انتقالاً لسطر جديد بعد الطباعة
- B) تقرأ A0
- C) توقف الاتصال
- D) تغير baud rate

4. في مثال حساس الضوء، ما الدالة المستخدمة للحصول على القراءة؟

- A) analogRead(photocellPin)
- B) Serial.read()
- C) digitalWrite
- D) lcd.begin

5. في تصنيف قراءة الضوء، ماذا تعني قراءة أقل من 10 في الكود؟

- A) Dark
- B) Bright
- C) Very bright
- D) Light

6. أي عبارة مستخدمة لاستقبال قيمة من **Serial** في مثال التحكم بالـ **LED**؟

- A) Serial.read()
- B) Serial.print()
- C) analogWrite()
- D) delay()

7. في مثال **Computer Interfacing**، إرسال '1' يؤدي إلى:

- A) تشغيل LED
- B) إطفاء LED
- C) مسح البرنامج
- D) إيقاف Serial

8. إرسال '0' في المثال يؤدي إلى:

(A) إطفاء LED

(B) تشغيل Servo

(C) زيادة الحرارة

(D) فتح LCD

9. ما المنفذ المستخدم للـ LED في مثال **Computer Interfacing**؟

A) 13

B) A0

C) 2

D) 5

10. ما الغرض من **if** و **else if** في مثال **Serial**؟

(A) تنفيذ فعل مختلف حسب القيمة المستلمة

(B) تعريف سرعة الاتصال

(C) إنشاء منفذ Analog

(D) تهيئة حساس الحرارة

المحاضرة 8 - المحركات DC و Servo

الاختبار القبلي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. ما وظيفة المحرك الكهربائي بصورة عامة؟

(A) تحويل الطاقة الكهربائية إلى حركة

(B) عرض النص

(C) قياس الضوء

(D) تخزين البيانات

2. أي نوعين يركز عليهما الملف؟

Servo Motor و A) DC Motor

Keypad و B) LCD

LM35 و C) LDR

Button و D) Relay

3. أي محرك مناسب للتحكم بالزاوية؟

A) Servo Motor

B) DC Motor العادي فقط

C) LDR

D) Relay

4. ما المكتبة المستخدمة مع Servo في المثال؟

A) Servo.h

B) Keypad.h

C) LiquidCrystal.h

D) Wire.h

5. أي دالة تستخدم لكتابة زاوية Servo؟

A) myservo.write

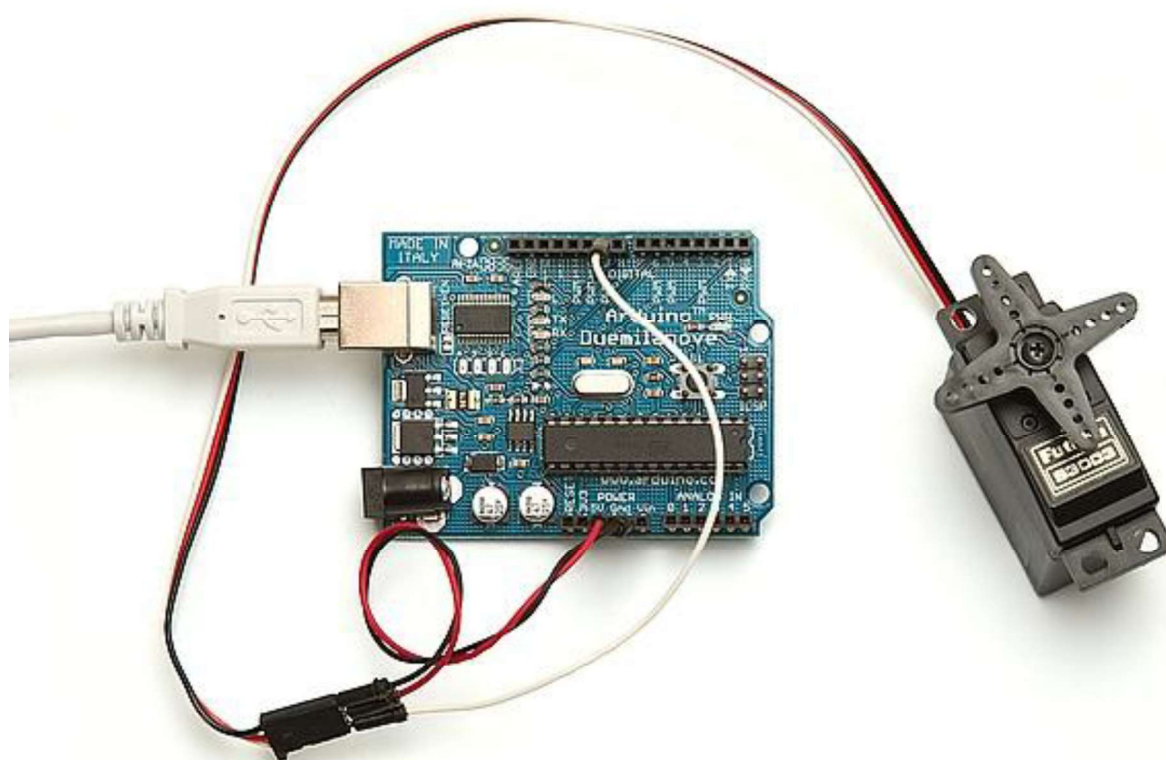
B) analogRead

C) Serial.print

D) lcd.clear

الفصل السادس

أستخدام المحركات The Motors



تعتبر المحركات من أهم العناصر الداخلة في المشاريع الألكترونيه و هي العنصر المسؤول عن تحويل الطاقه الكهربيه الى طاقه حركيه فى صوره دوران .



ستجد المحركات فى كثير من الأجهزة الالكترونيه التى يوجد بها حركه ميكانيكيه مثل: الروبوتات بأنواعها المختلفه سواء كانت صناعيه او روبوتات للترفيه، مشغلات الأقراص، ألعاب الأطفال .. الخ

تنقسم المحركات الكهربيه الى نوعين رئيسيين وهما :

- محركات التيار المستمر (DC – Servo – Stepper)
- محركات التيار المتردد (3 Phase – 1 Phase)

فى هذا الفصل سنتعرض للنوع الأول و هو المحركات التى تعمل بالتيار المستمر مثل ال DC Motor و ال Servo Motor وهما أشهر أنواع المحركات المستخدمه فى مشاريع المتحكمات الدقيقه بأنواعها المختلفه.

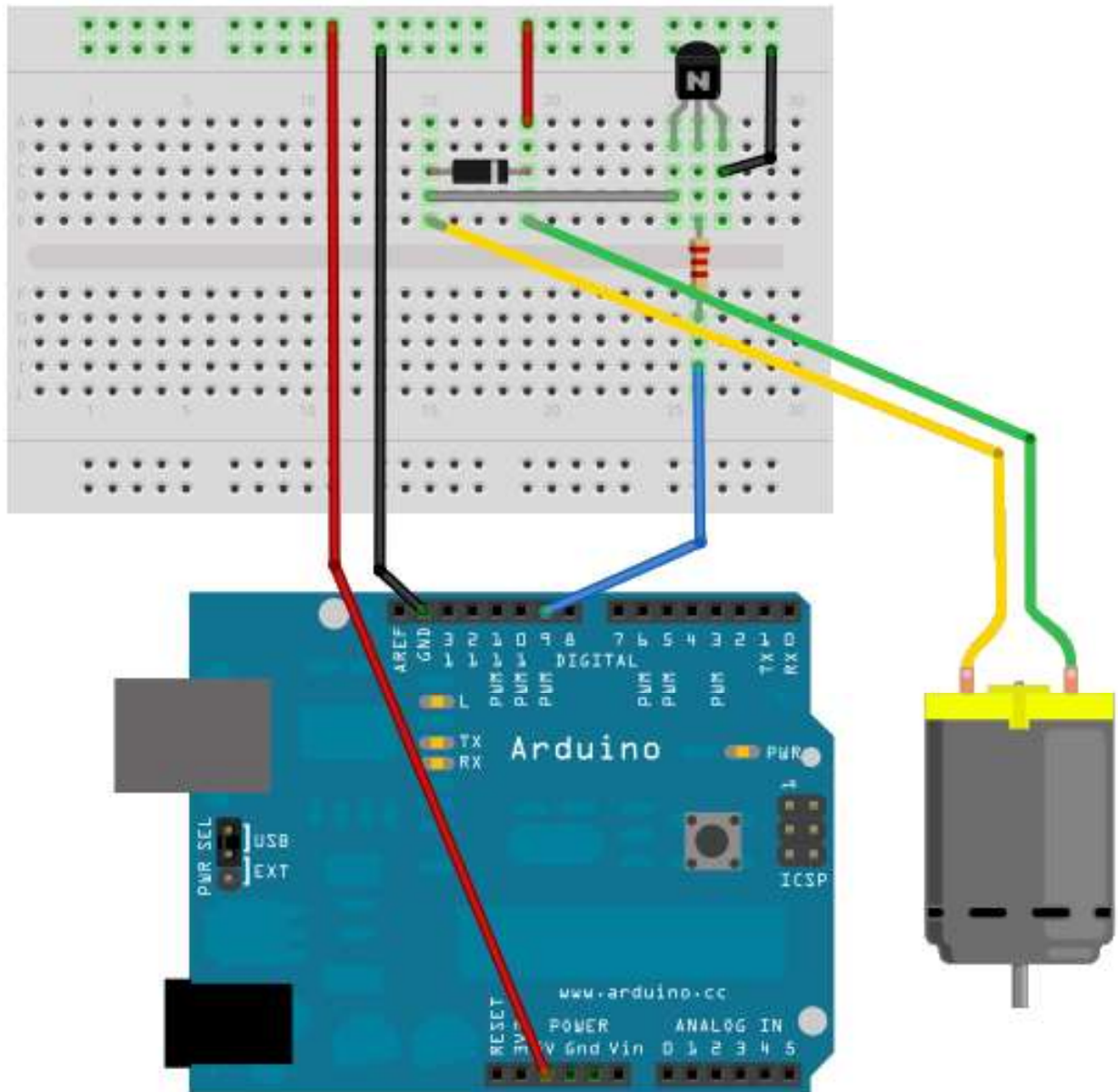


Servo Motor



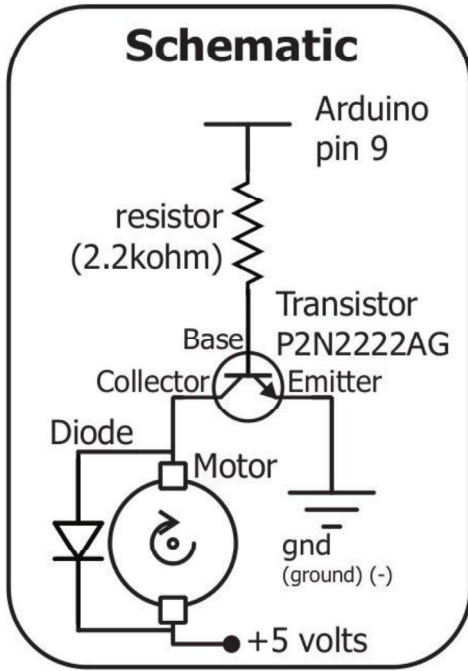
DC Motor

المثال العاشر: استخدام محرك التيار المستمر DC

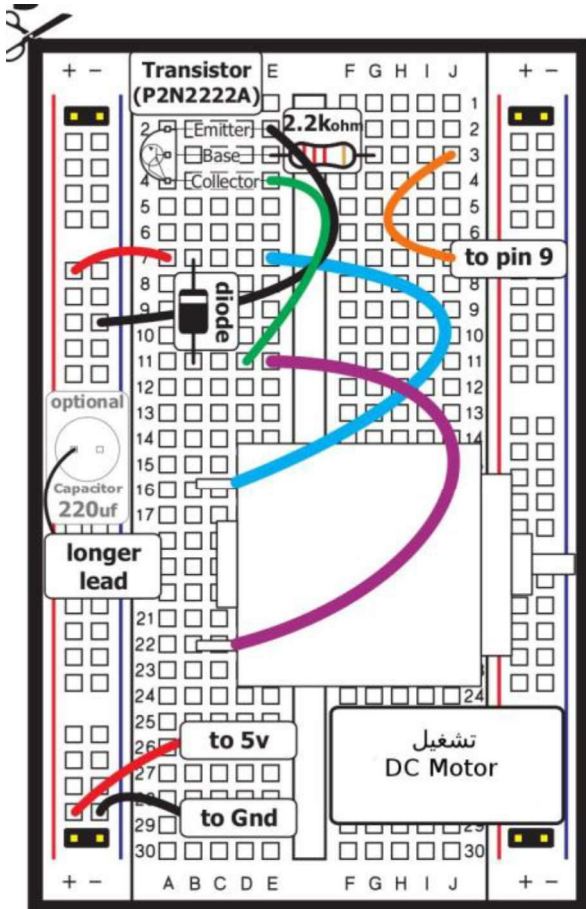


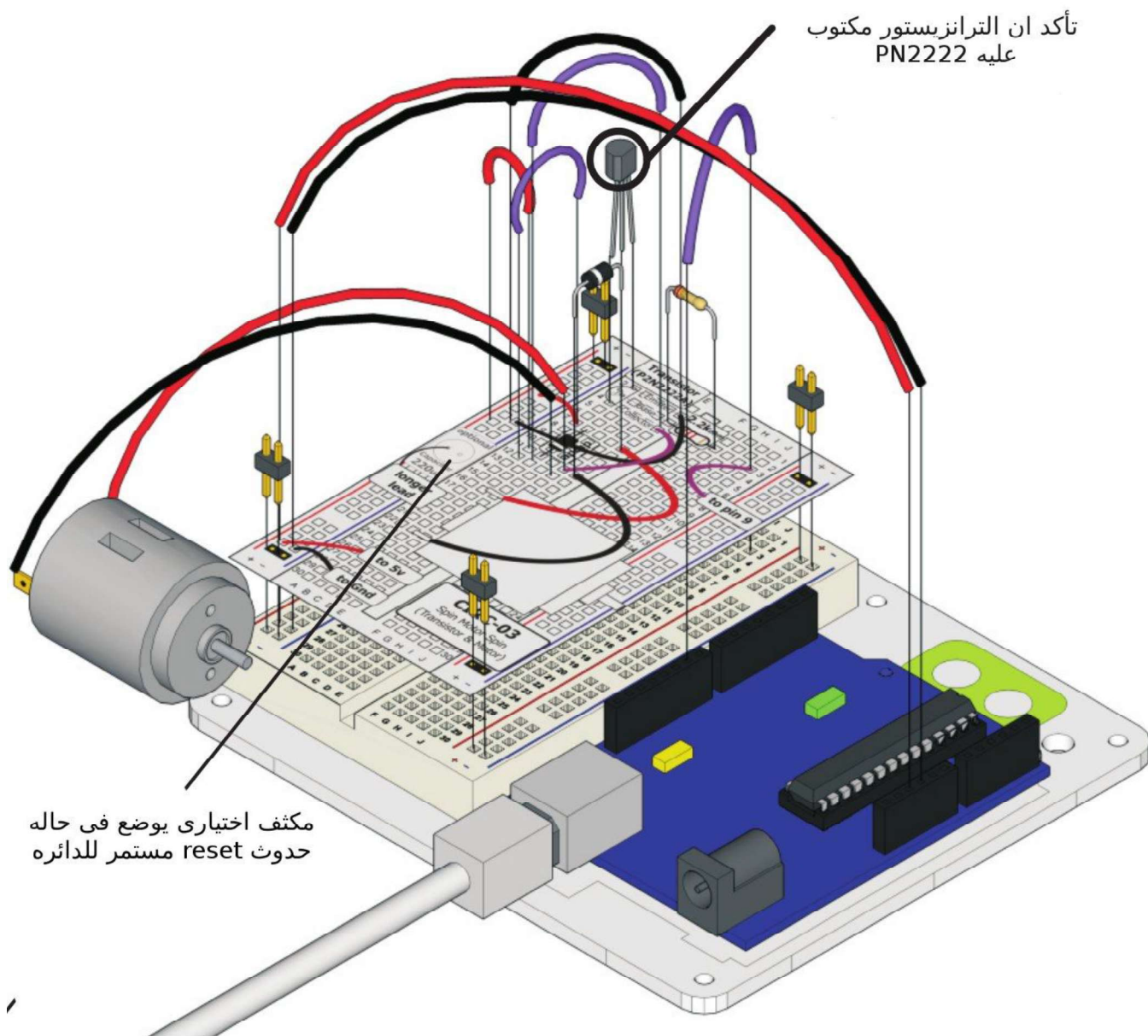
مكونات المثال (7):

- ✓ بورده اردوينو Arduino Uno
- ✓ لوحه تجارب Breadboard
- ✓ محرك تيار مُستمر صغير DC motor
- ✓ ترانزستور 2N2222 او PN2222
- ✓ دايود 1N4001 او اى بديل
- ✓ مقاومه 2.2 كيلو اوم
- ✓ أسلاك توصيل
- ✓ كابل التوصيل بالـ USB
- ✓ ورقه المساعدة المطبوعة



فى هذا المثال سوف نستخدم محرك تيار مستمر من الحجم الصغير و الذى يوجد عاده فى لعب الأطفال و يعمل بفرق جهد يبدأ من 3 فولت و اقصاه 9 فولت و ستجد مثل هذه المحركات متوفره فى محلات المكونات الألكترونيه او فى اى من لعب اطفال القديمه التى تحتوى على محركات ☺





بعد الانتهاء من توصيل المكونات على لوح التجارب قم بكتابه الأكواد التالية ثم ارفعها
إلى بورده اردوينو

```
//Example_10_DC_Motor
int motorPin = 9 ;
int onTime = 2500 ;
int offTime = 1000 ;

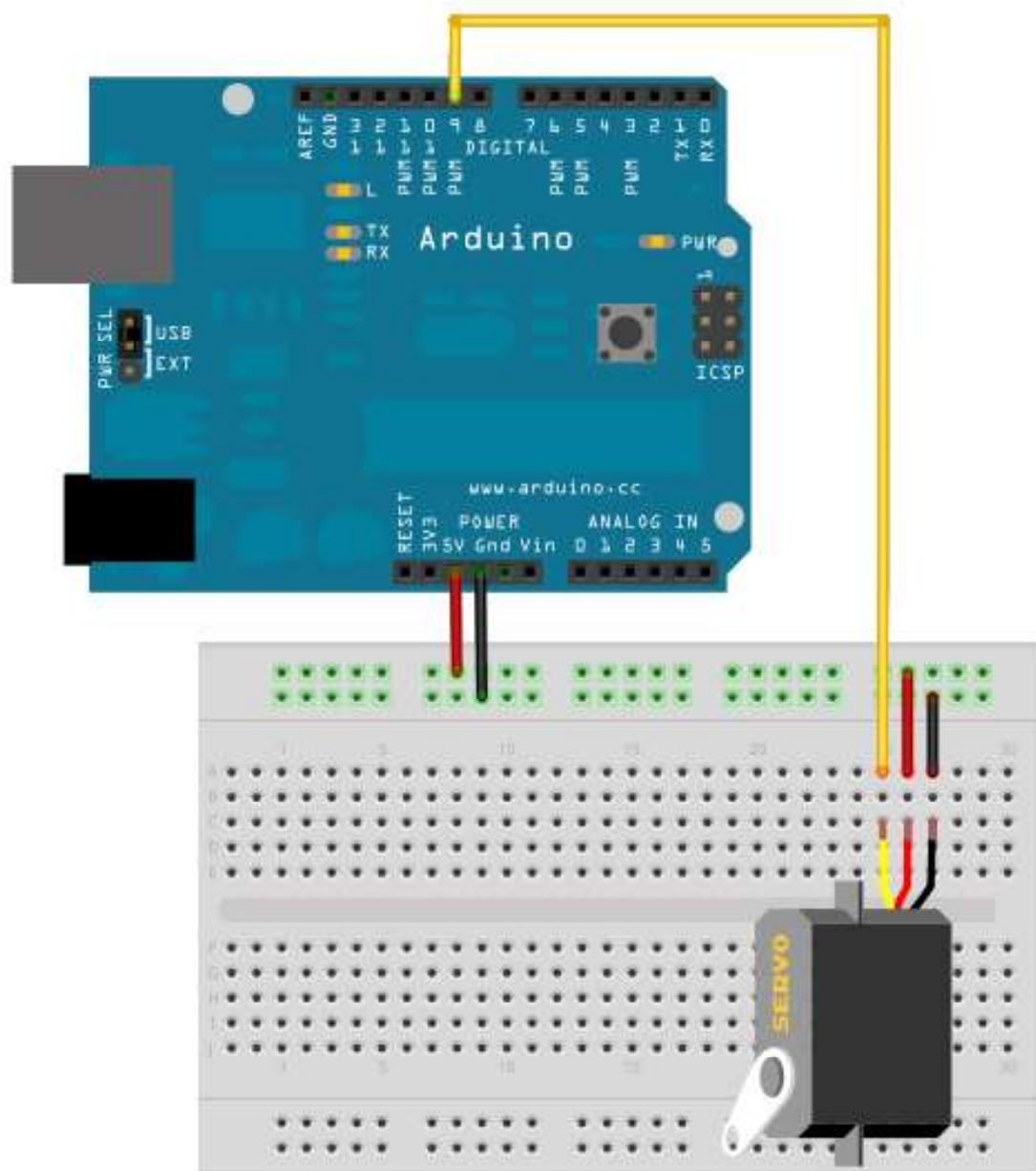
void setup ( )
{pinMode(motorPin, OUTPUT); }

void loop ( )
{
  analogWrite(motorPin,100);
  delay(onTime);
  digitalWrite(motorPin, LOW);
  delay(offTime);

  analogWrite(motorPin,190);
  delay(onTime);
  digitalWrite(motorPin, LOW);
  delay(offTime);

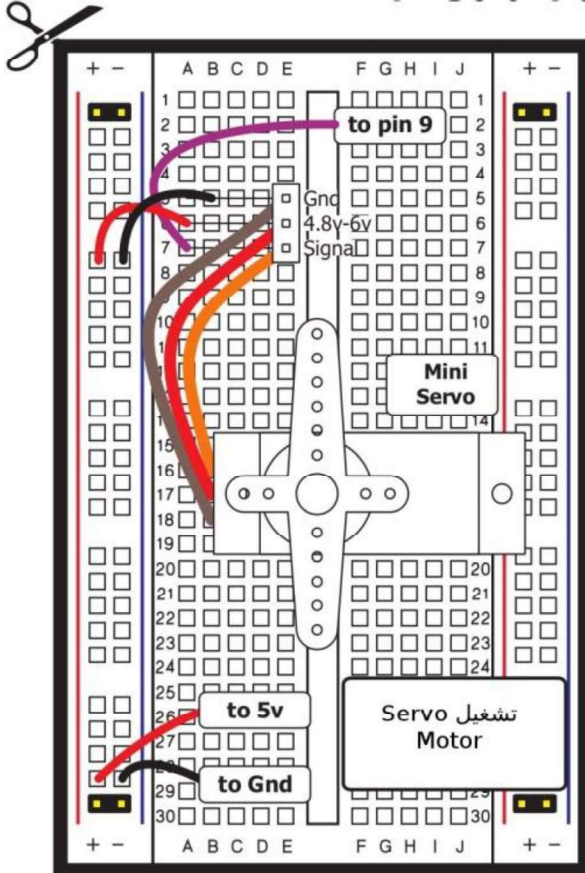
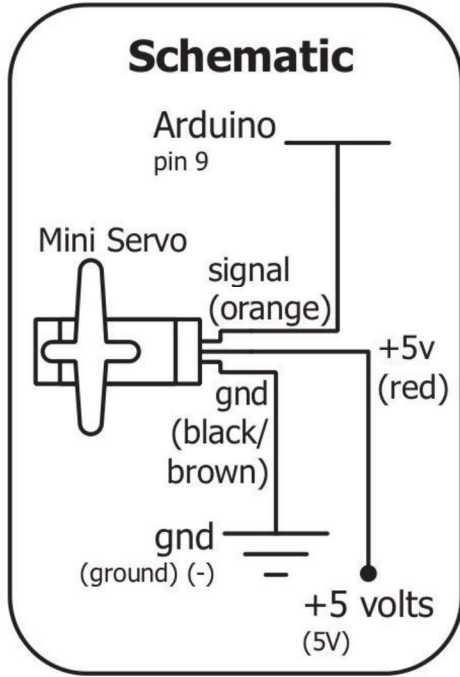
  analogWrite(motorPin,255);
  delay(onTime);
  digitalWrite(motorPin, LOW);
  delay(offTime);
}
```

المثال الحادي عشر: استخدام محرك سيرفو

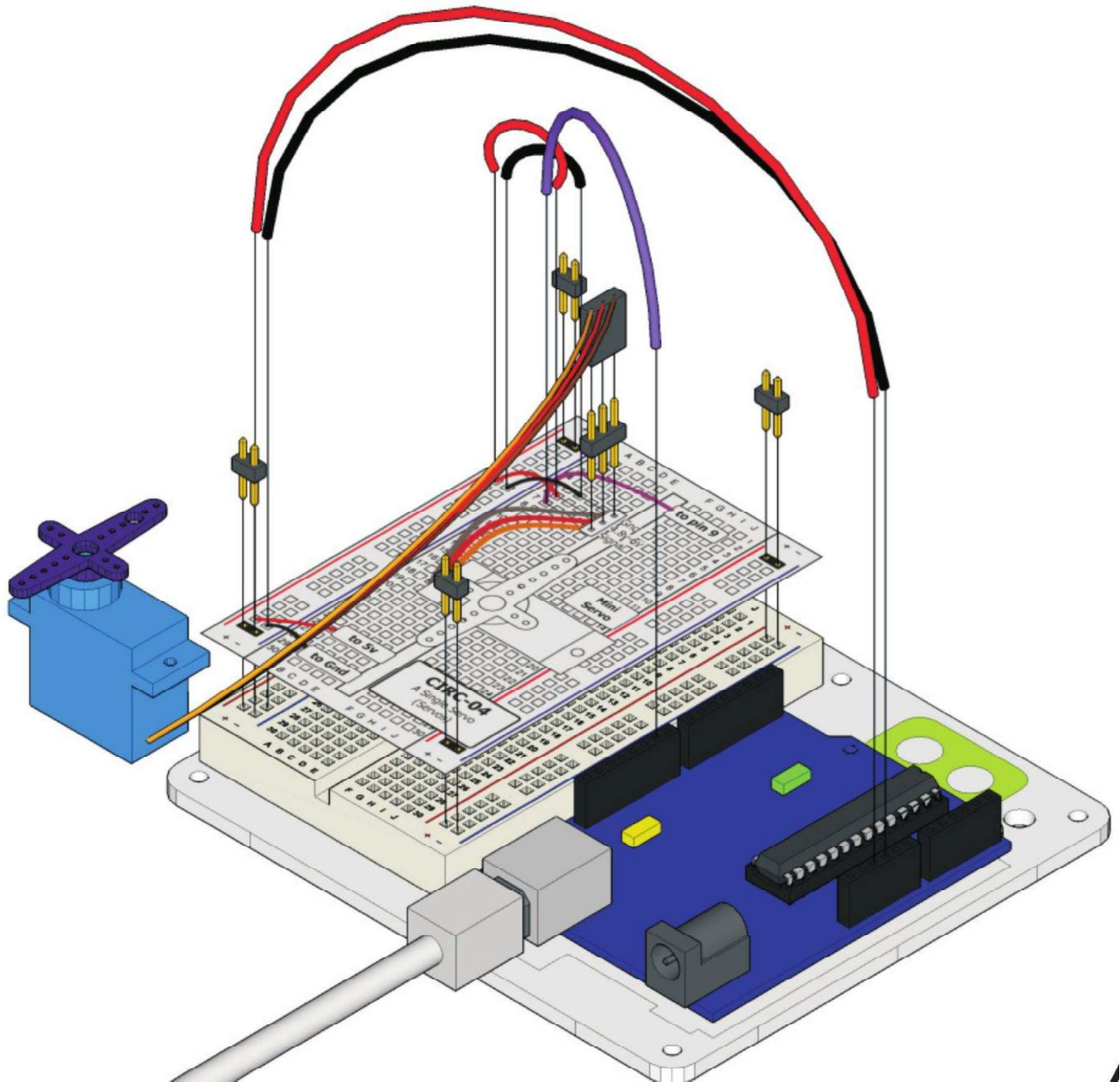


مكونات المثال (7):

- ✓ بورده اردوينو Arduino Uno
- ✓ لوحه تجارب Breadboard
- ✓ محرك تيار مُستمر - سيرفو Servo motor
- ✓ أسلاك توصيل
- ✓ كابل التوصيل بالـ USB
- ✓ ورقه المساعدة المطبوعة



في هذا المثال سوف نستخدم محرك تيار مستمر من نوع سيرفو موتور والذي يعمل بفرق جهد يبدأ من 4 فولت و اقصاه 12 فولت و ستجد مثل هذه المحركات متوفره في محلات المكونات الألكترونيه ☺



بعد الانتهاء من توصيل المكونات على لوح التجارب قم بكتابه الأكواد التالية ثم ارفعها إلى بورده اردوينو (ملحوظة: يمكنك ان تجد الكود البرمجي في قائمه الأمثله الجاهزه فى برنامج اردوينو sweep - servo - examples)

```
//Example_11_Servo_Motor
```

```
#include <Servo.h>
```

مكتبة اضافيه

```
Servo myservo;
```

```
int pos = 0;
```

```
void setup()
```

```
{
```

```
  myservo.attach(9);
```

```
}
```

```
void loop()
```

```
{
```

```
  for(pos = 0; pos < 180; pos += 1)
```

```
  {
```

```
    myservo.write(pos);
```

```
    delay(15);
```

```
  }
```

```
  for(pos = 180; pos >= 1; pos -= 1)
```

```
  {
```

```
    myservo.write(pos);
```

```
    delay(15);
```

```
  }
```

```
}
```

المحاضرة 8 - المحركات DC و Servo

الاختبار البعدي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. ما رقم المنفذ المستخدم لـ **DC Motor** في الكود؟

- A) 9
- B) 13
- C) 2
- D) A0

2. ما الدالة المستخدمة لتغيير قيمة قيادة **DC Motor** في المثال؟

- A) analogWrite
- B) analogRead
- C) Serial.read
- D) lcd.print

3. ما أعلى قيمة **analogWrite** مستخدمة في المثال؟

- A) 255
- B) 190
- C) 100
- D) 1024

4. ما قيمة **onTime** في الكود؟

- A) 2500 ms
- B) 1000 ms
- C) 15 ms
- D) 9600 ms

5. ما قيمة **offTime**؟

- A) 1000 ms
- B) 2500 ms
- C) 180 ms
- D) 255 ms

6. ما وظيفة **myservo.attach(9)**؟

- A) ربط كائن Servo بالمنفذ 9
- B) قراءة المنفذ 9
- C) إطفاء المحرك
- D) بدء Serial

7. إلى أي زاوية تقريباً تتحرك الحلقة الأولى في مثال **Servo**؟

- A) من 0 باتجاه 180 درجة
- B) من 180 إلى 360
- C) تبقى عند 0
- D) من 90 إلى 100 فقط

8. ما قيمة **delay** داخل حركة **Servo**؟

- A) 15 ms
- B) 1000 ms
- C) 2500 ms
- D) 9600 ms

9. أي مكون مذكور مع دائرة **DC Motor**؟

- A) PN2222/2N2222
- B) LCD backpack
- C) Keypad matrix
- D) LM35 فقط

10. أي عنصر حماية مذكور ضمن مكونات دائرة **DC Motor**؟

- A) 1N4001 diode
- B) LDR
- C) Potentiometer فقط
- D) SDA

المحاضرة 9 - Character LCD

الاختبار القبلي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. ما الفرق بين Character LCD و Graphical LCD؟

- (A) الأولى لعرض الأحرف والثانية للرسومات
- (B) لا يوجد فرق
- (C) الأولى محرك
- (D) الثانية حساس

2. ماذا يعني 2×16 في Character LCD؟

- (A) 16 حرفاً في كل سطر و 16 سطراً
- (B) 16 سطراً وعمودان
- (C) 16 لوناً
- (D) 16 فولت

3. أي مكتبة مستخدمة في مثال LCD؟

- (A) LiquidCrystal.h
- (B) Servo.h
- (C) Keypad.h
- (D) EEPROM.h

4. ما وظيفة `lcd.begin(16,2)`؟

- (A) تهيئة أبعاد الشاشة
- (B) مسح الشاشة
- (C) قراءة زر
- (D) تشغيل محرك

5. أي أمر يعرض نصاً على LCD؟

- (A) `lcd.print`
- (B) `lcd.read`
- (C) `digitalRead`
- (D) `Serial.read`

الفصل السابع

وسائل الادخال و الاخراج المتطورة

Advanced Inputs/Outputs



فى الفصول السابقة اعتمدنا بشكل اساسى على ابسط وسائل الادخال والاخراج للتعامل مع اردوينو مثل السويتش (ادخال Input) والدايود الضوئى (اخراج output)



فى هذا الفصل سنستعرض بعض المكونات الاكثر تطورا للتفاعل مع المتحكمات الدقيقة فى مشاريع واقعيه يمكن استخدامها فى حياتنا اليوميه، سيتطرق هذا الفصل الى شرح المكونات التاليه:

- شاشات عرض الكريستال السائل LCD: liquid crystal Display
- لوحة ادخال الارقام Keypad
- مصفوفه الدايود الضوئى Led matrix
- المتممات Relays



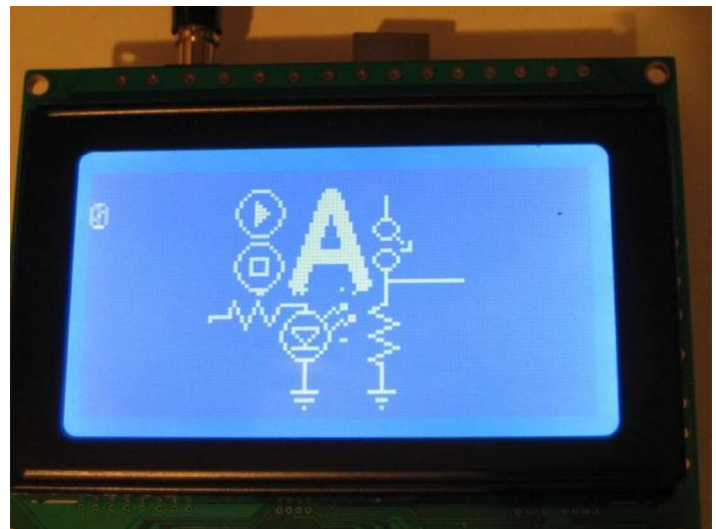
شاشات عرض الكريستال السائل : تتكون هذه الشاشات من زجاج الكريستال المعالج و تتوفر هذه الشاشات بأحجام و انواع مختلفه و سوف نستعرض منها نوعين اساسيين و هما:

- شاشات العرض المعتمده على الحروف Character LCD
- شاشات العرض المعتمده على الرسومات Graphical LCD

Character LCD



Graphical LCD



توفر شاشات عرض المعتمدة على الحروف Character LCD امكانيه اخراج اى نصوص تتكون من حروف او ارقام او رموز (مثل التى تكتب على لوحة المفاتيح فى الحاسب الآلى) و تتوفر بأحجام مختلفه و ألوان مختلفه مثل:

Green 16x2 lcd

Blue 16x2 lcd

Green 20x4 lcd

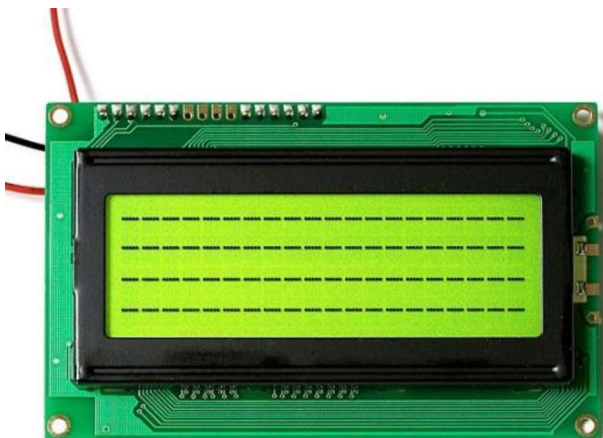
و يمثل الرقم 16x2 عدد السطور (2) الحروف التى يمكن كتابتها فى كل سطر (16) حرف كما تتوفر بألوان مختلفه كما فى الصور التاليه:



Green 16x2 LCD



Blue 16x2 LCD

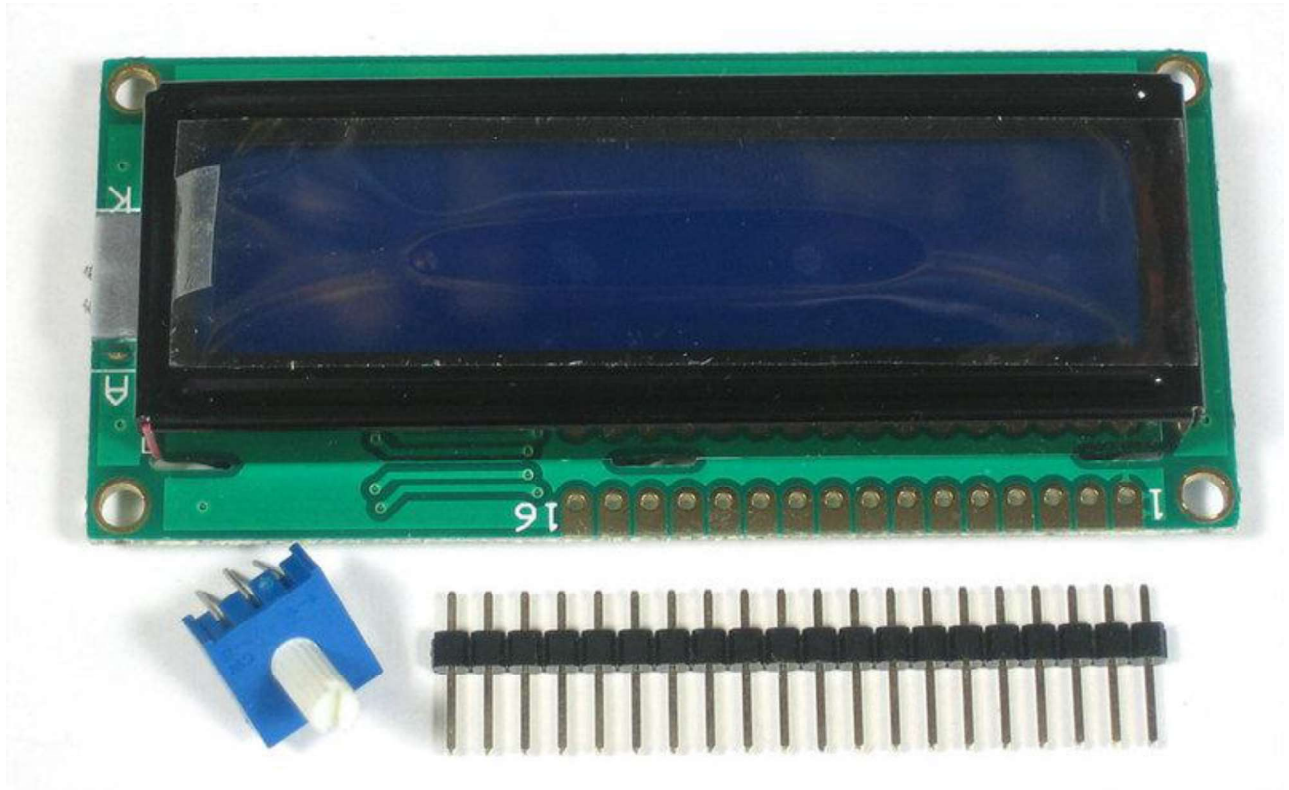


Green 20x4 LCD



Black 16x2 LCD

المثال العاشر: توصيل شاشة Character LCD بمقاس 16x2

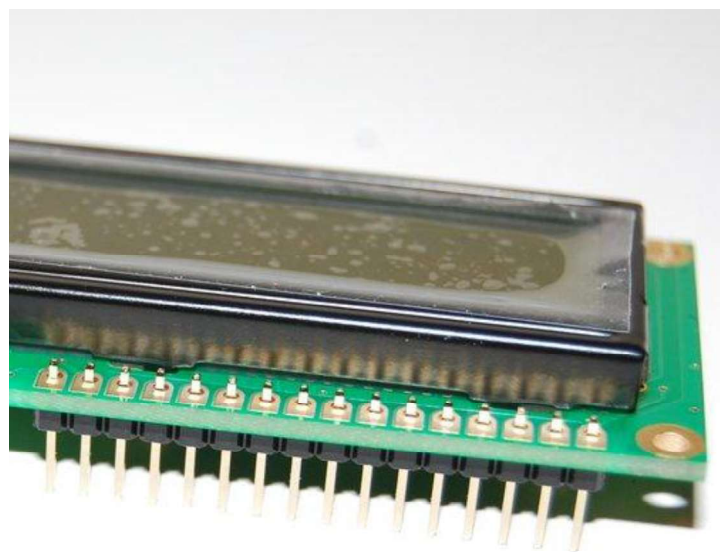
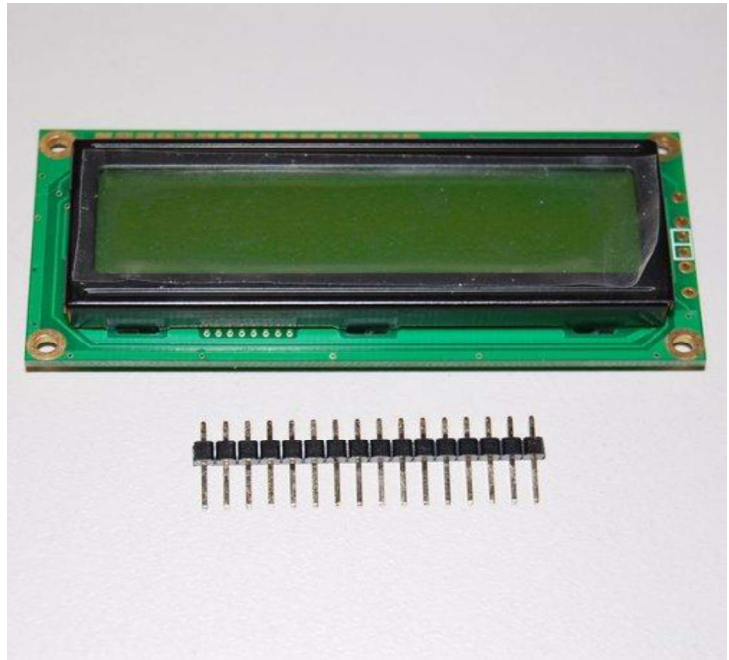


سنحتاج المكونات التاليه:

- شاشة بقياس 16x2 لون اضائه ازرق (او اى لون تفضله انت)
- اعمده توصيل نحاسيه قصيره 16 نقطه copper Pin Headers 16 point
- مقاومه متغيره بقيمه 10 كيلو اوم
- مكواه لحام
- قصدير لحام

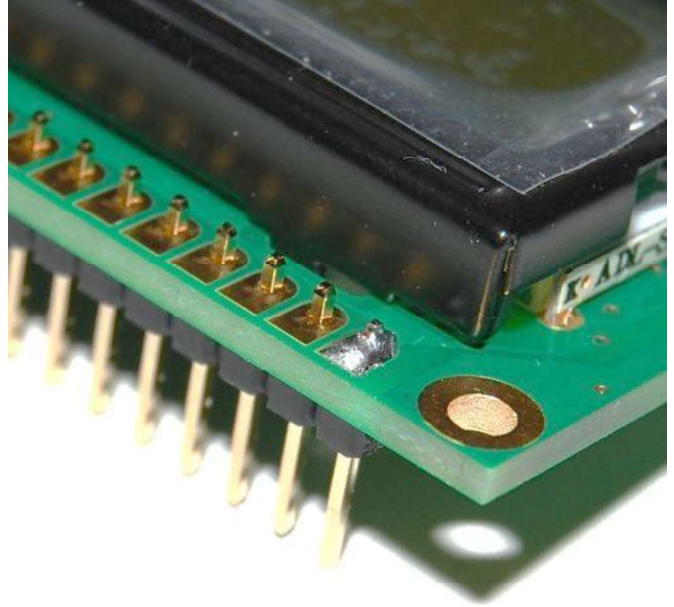
الخطوة الأولى هي لحام نقاط التوصيل Pin Headers بنقاط لحام الشاشة و يتم استخدام مكواه اللحام و القصدير في هذا العملية:

اذا اردت تعلم لحام المكونات الإلكترونية توجهه إلى الينك التالي
<http://www.aaroncake.net/electronics/solder.htm>



قم بلحام اول نقطه توصيل باستخدام
الكاويه والقصدير وانتظر 5 ثواني حتى
تبرد نقطه التوصيل

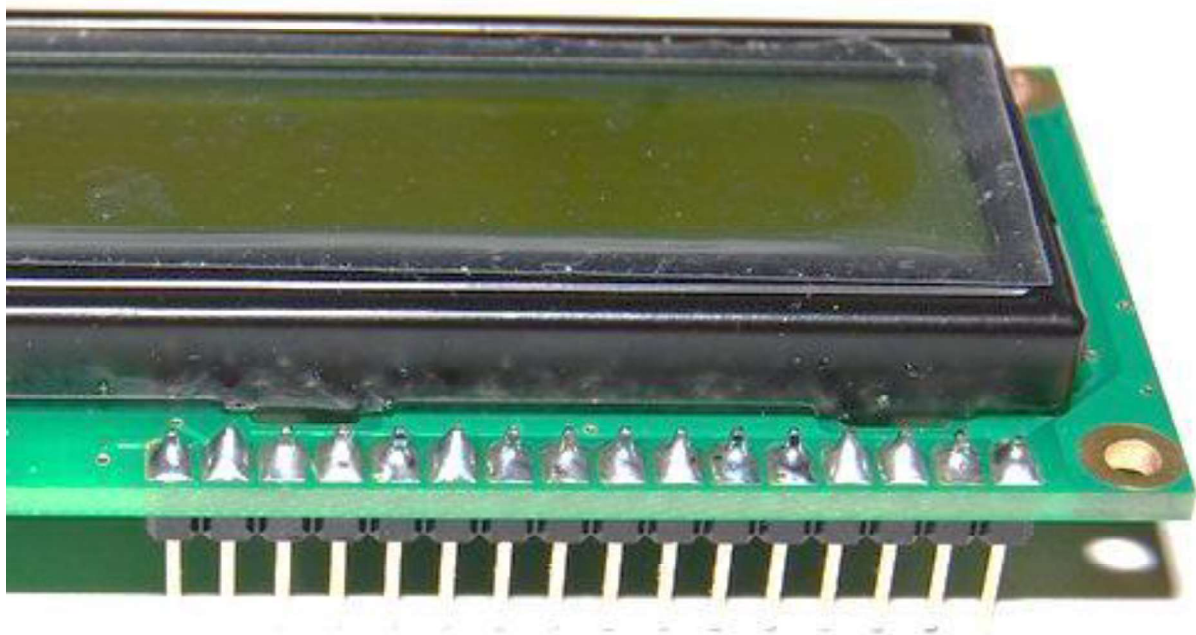
تنبيه: لا تحاول لمس نقطه اللحام بيدك
بعد اللحام مباشره فقد يؤدى ذلك الى
حدوث حروق للجلد من حراره اللحام



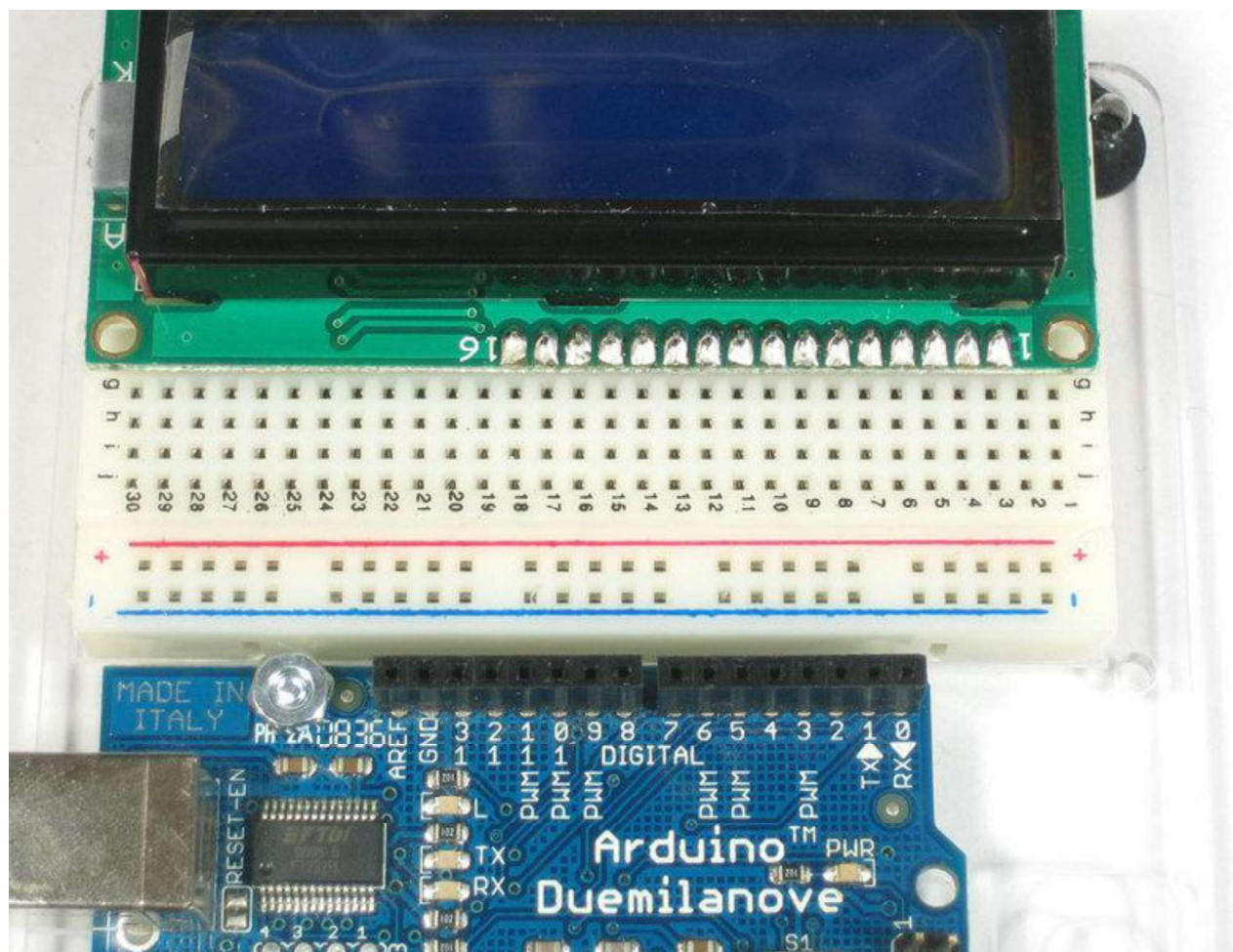
بعد ذلك قم بلحام نقطه التوصيل الاخيره
وذلك حتى تقوم بتثبيت pin headers
من كلا الجانبين



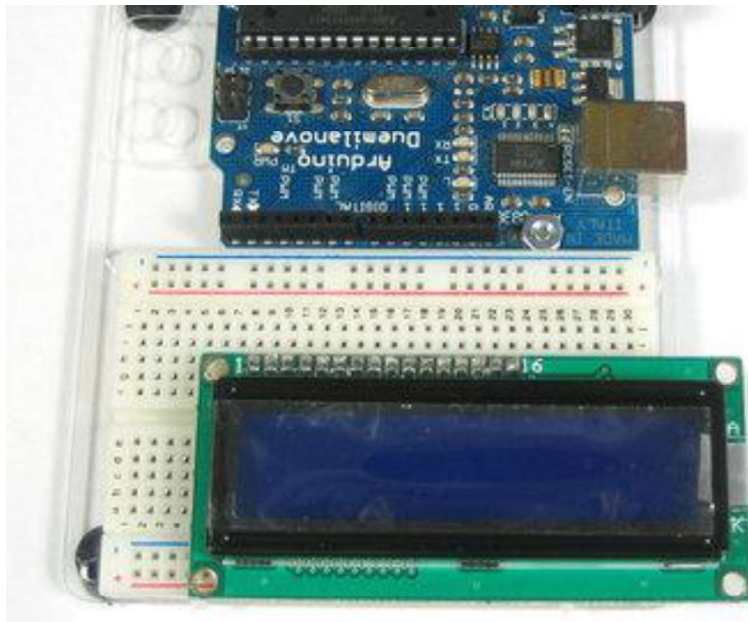
قم بتكرار نفس العمليه السابقه مع جميع نقاط التوصيل حتى تنتهى من 16
نقطه كما فى الصوره التاليه



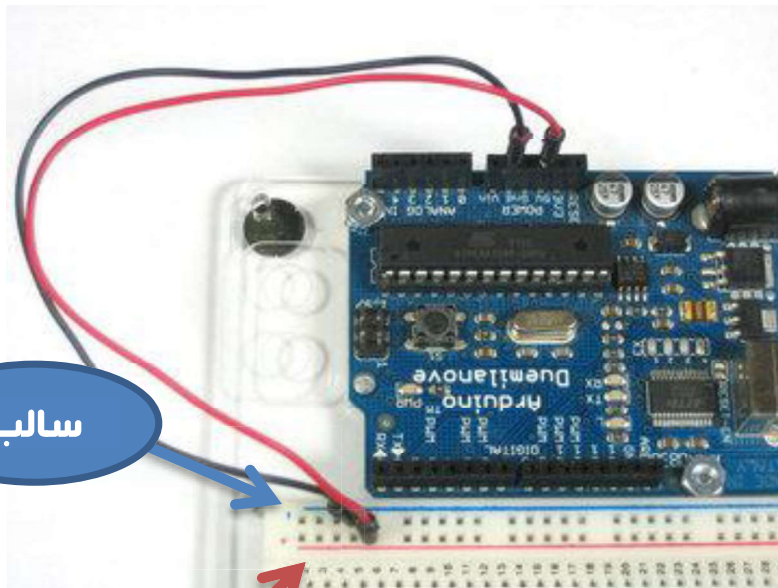
بعد الانتهاء من اللحام قم بتركيب الـ pin header على لوحة التجارب كما في الصورة



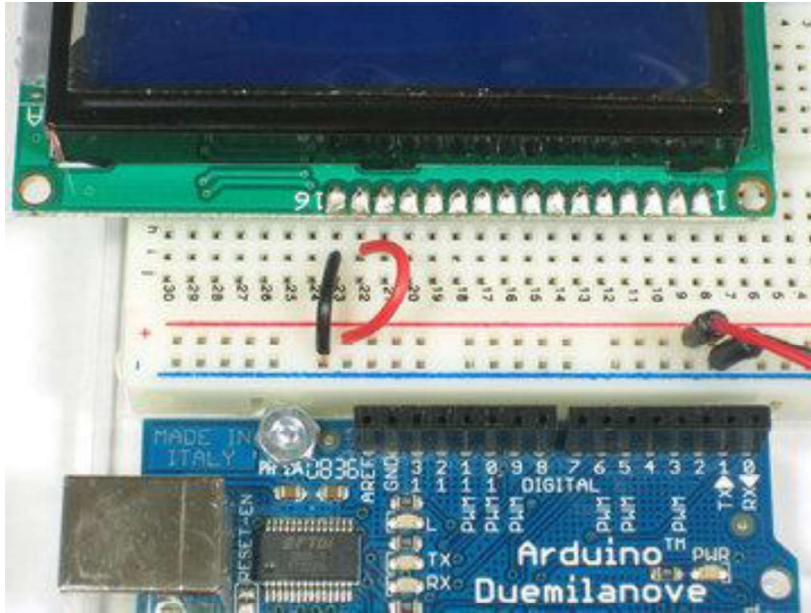
يفضل استخدام قاعده تثبيت بلاستيكيه تستخدم فى تثبيت اردوينو و لوحة التجارب معا
المثل التى تستخدم فى الصورة التاليه



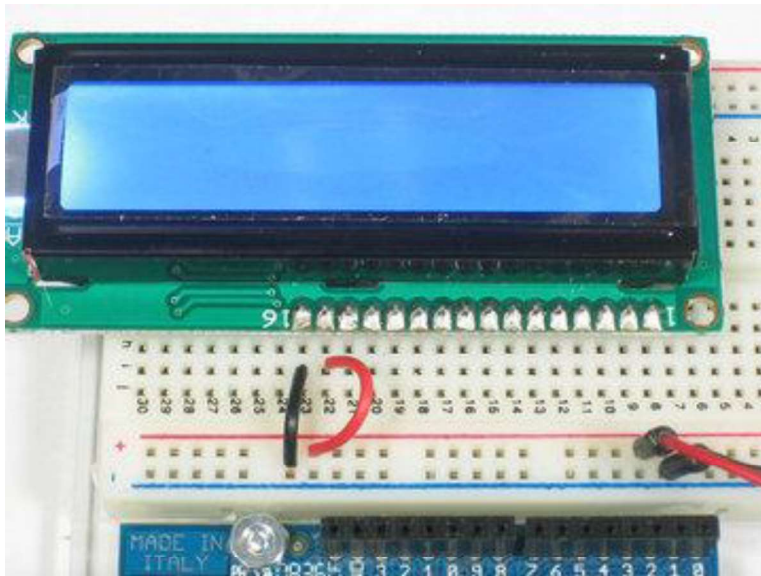
قاعده الثبيت
البلاستيكيه تحمل
اردوينو و لوحه
التجارب و الشاشه
معا



- الان نقوم
بتوصيل المخرج 5v على
خط الموجب الاحمر و
الطرف GND على الخط
الارضى فى لوحه
التجارب



- بعد ذلك نقوم بتوصيل المدخل رقم 16 في شاشة العرض على الخط الارضى و نوصل المدخل رقم 15 على الخط الموجب 5 v كما في الصورة التاليه :

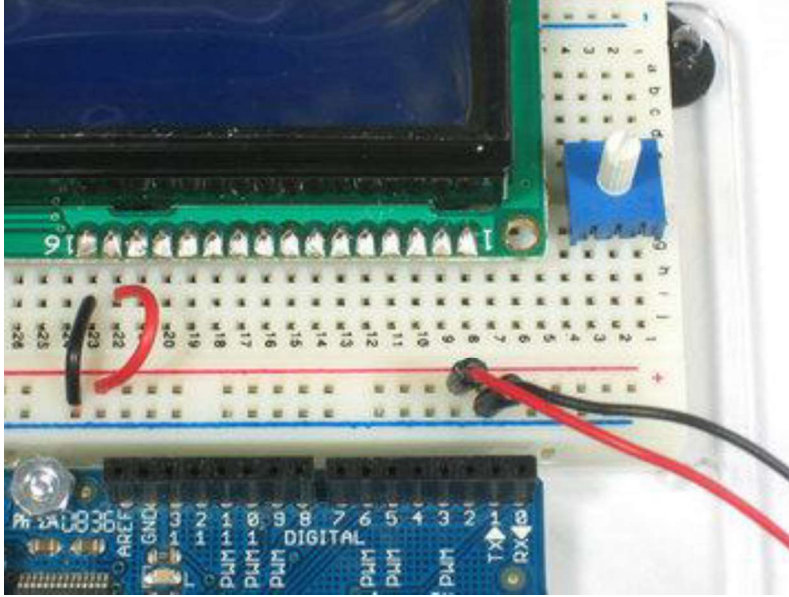


- قم بتوصيل اردوينو بالبطاريه او كابل الـ USB ثم لاحظ ان الضوء الخلفى للشاشه بدأ فى العمل كما فى الصورة التاليه :

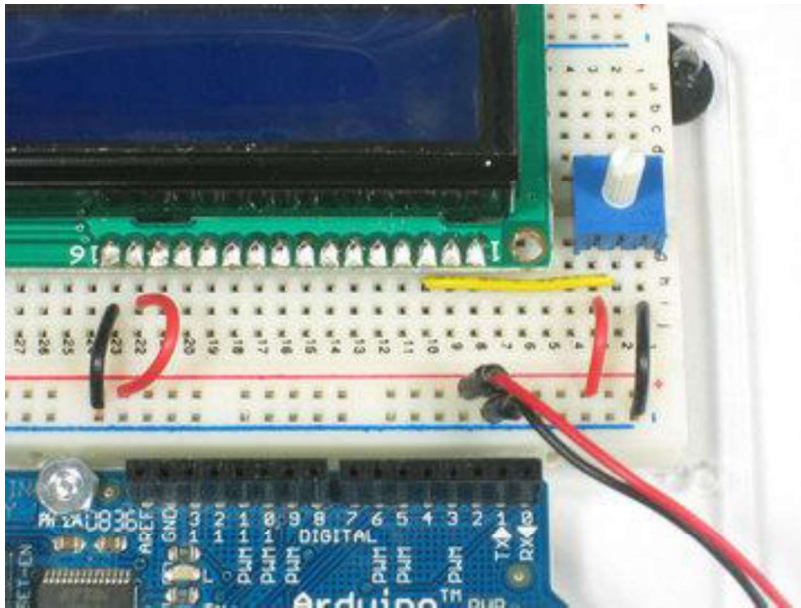
ملحوظه يختلف لون الاضائه على حسب الشاشه المستخدمه و تتوفر الالوان التاليه:

- الاحمر - الأخضر -
- الازرق - الأبيض -

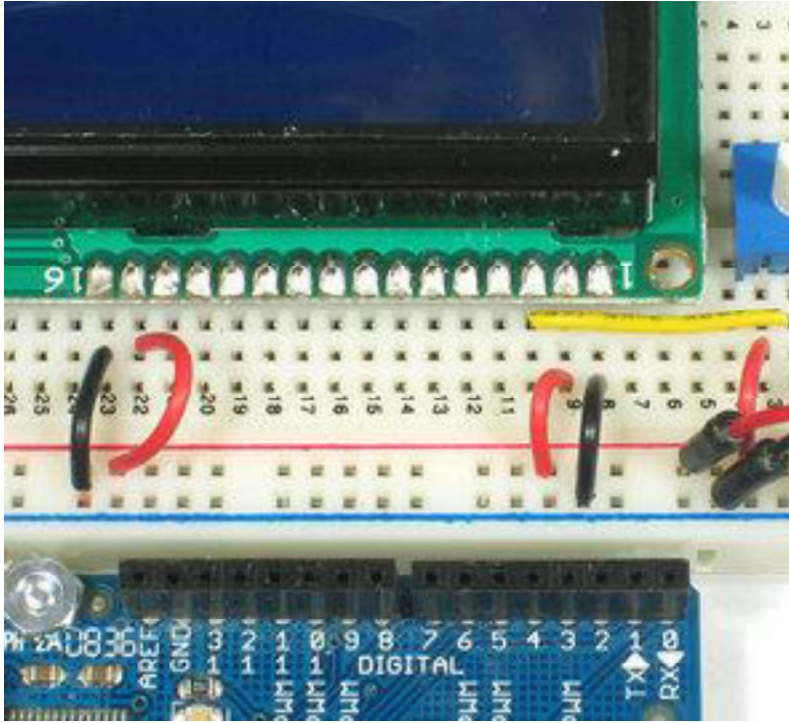
تركيب المقاومه المتغيره للتحكم فى شدة الاضاءة



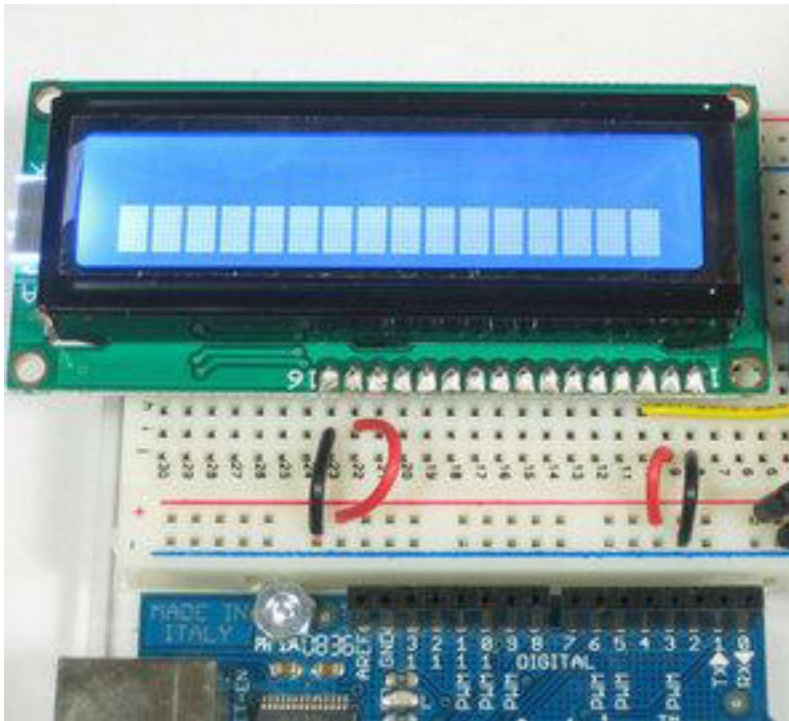
تعتبر تلك الخطوه اختياريه و
يمكنك ان تهملها اذا اردت.
الهدف من تركيب المقاومه
المتغيره هو التحكم فى التيار
الكهربى الداخلى الى الشاشه
وبالتالى التحكم فى شدة
السطوع (الاضاءه) الناتجه



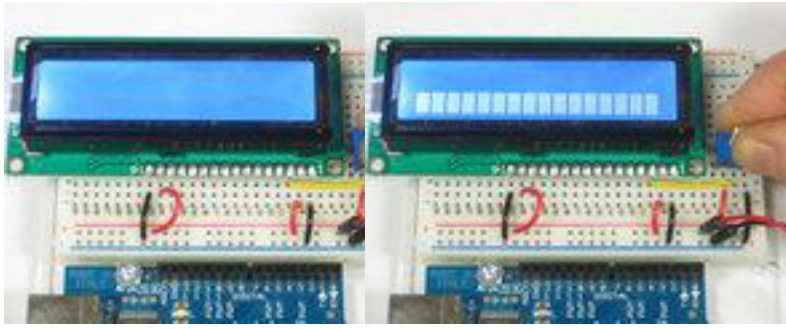
- قم بتوصيل احد
الاطراف الجانبيه للمقاومه
المتغيره بالخط الموجب و
الطرف الجانبى الاخر بالخط
السالب ثم قم بتوصيل الطرف
الاوسط للمقاومه المتغيره
بالمدخل رقم 3 على الشاشه
كما فى الصوره التاليه :



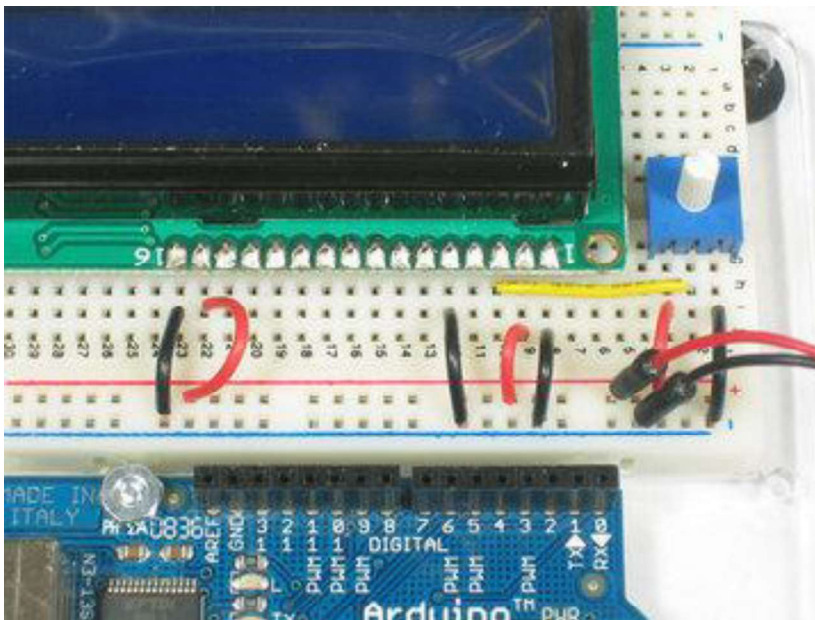
- قم بتوصيل المدخل رقم 1 على شاشة العرض بالخط الارض و المدخل رقم 2 بالخط الموجب على لوحة التجارب



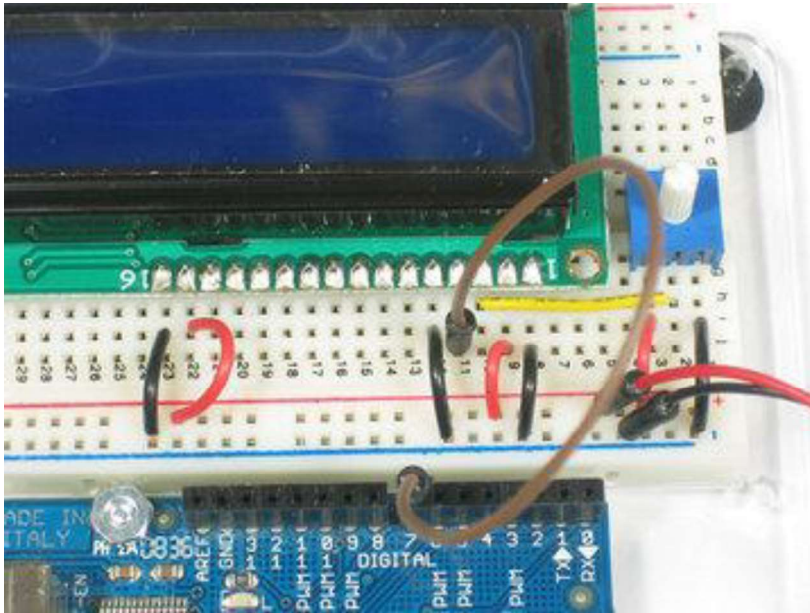
- الان قم بوصيل البطاريه مره اخرى و قم بتدوير عصا الدوران فى المقاومه المتغيره ولاحظ ماذا سيحدث للاضاءه الخلفيه لشاشه العرض



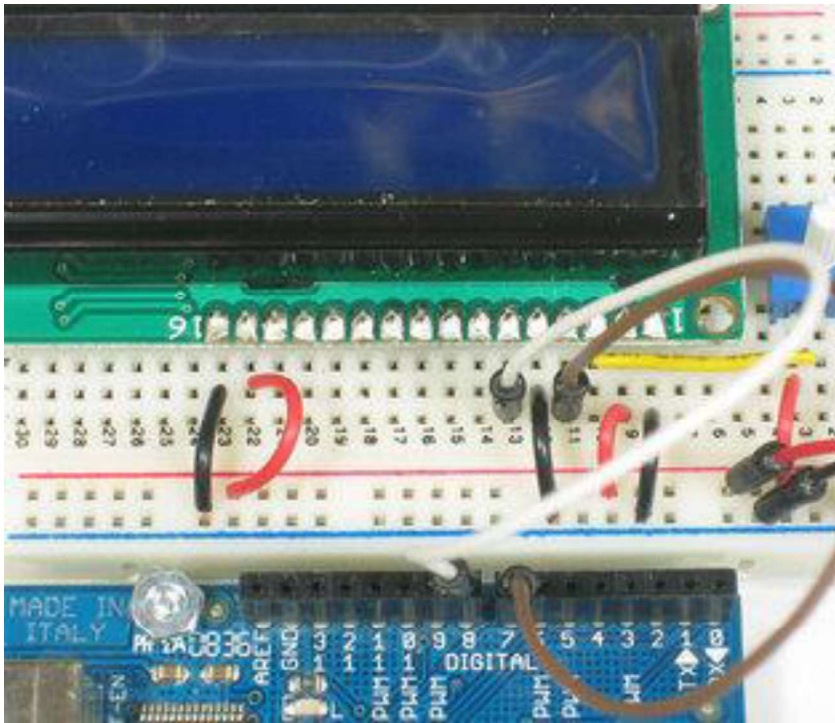
جميع الخطوات السابقة كانت تهدف لتوصيل الشاشة بالطاقة و بالمقاومه الضوئيه التي تتحكم فى مدى سطوع الشاشة و شدة الاضاءه الخلفيه ، الخطوات التاليه ستوضح كيف توصل منافذ نقل البيانات من اردوينو الى الشاشة.



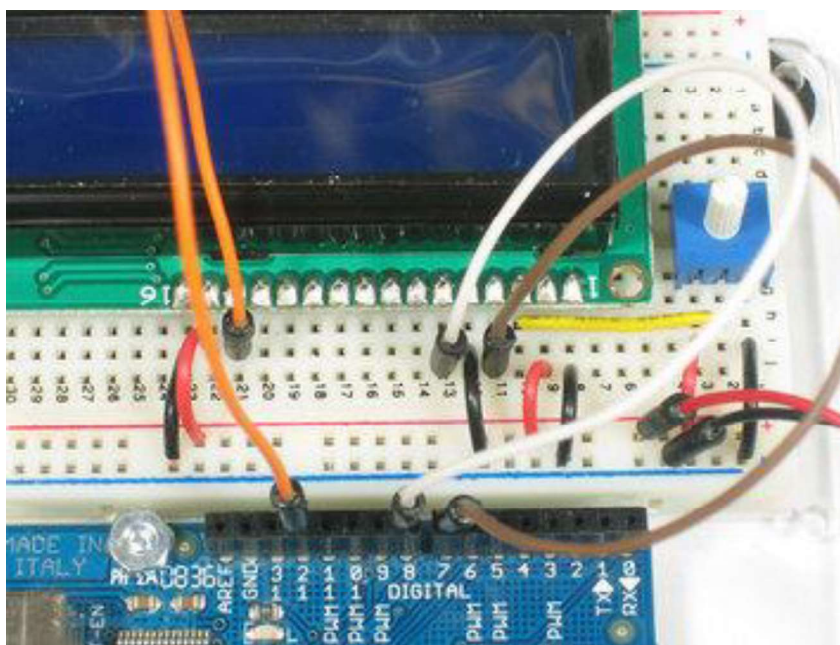
فى بعض المشاريع قد تستخدم المدخل رقم 5 والذى يسمى RW لكن فى حالتنا هذا سنقوم بتوصيله بالخط الارضى



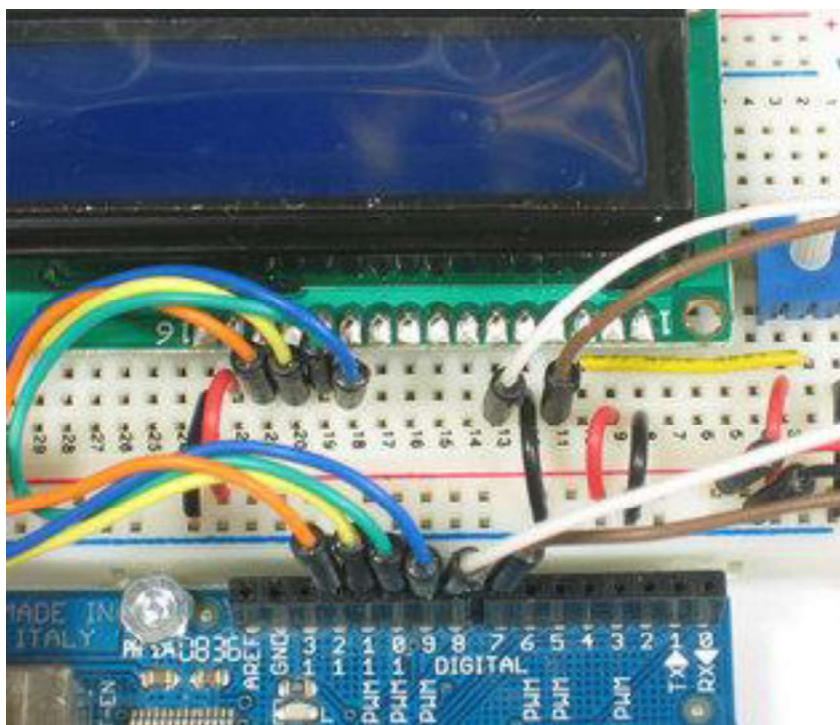
قم بتوصيل المدخل
رقم 4 في شاشة العرض
على المخرج رقم 7
في بورده اردوينو



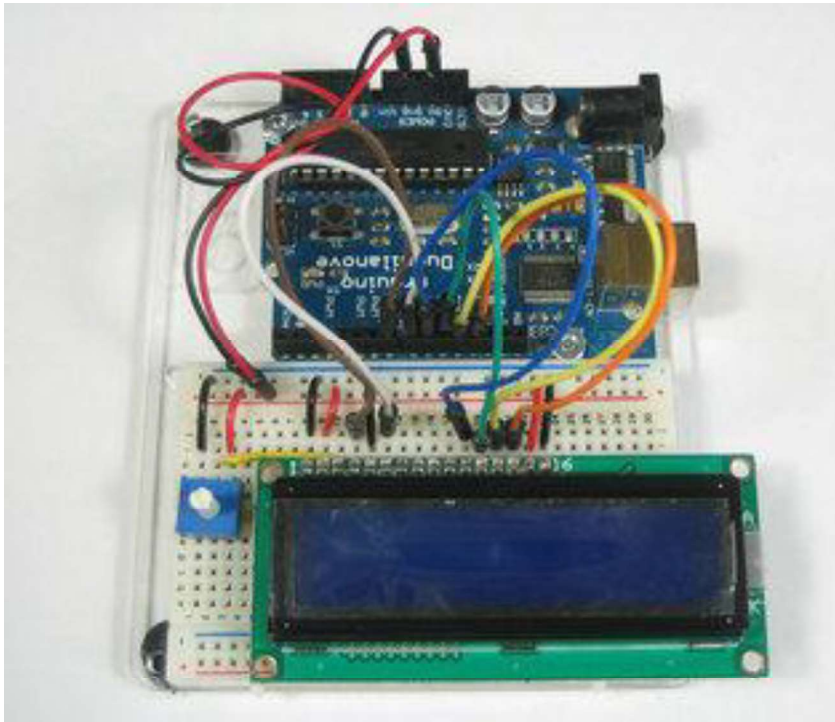
بعد ذلك نقوم بتوصيل
المدخل رقم 6 في
شاشة العرض على
المخرج رقم 8 في
بورده اردوينو



قم بتوصيل المدخل
رقم 14 في شاشه
العرض على المخرج
رقم 12 في بورده
اردوينو



الخطوه الاخير هـ
توصيل المداخل رقم
13,12,11 بالمخارج
رقم 11,10,10 على
الموجوده على
اردوينو بنفس الترتيب
كما في الصوره التاليه



الشكل النهائي بعد
توصيل جميع الاطراف
اللازمه لتشغيل الشاشه
مع اردوينو.

الان حان وقت كتابه الكود البرمجى

توفر بيئه تطوير اردوينو العديد من الامثله الجاهزه و يمكنك ان تستخدم مثال الشاشه
الجاهز من خلال فتح قائمه الامثله كالتالى:

File→Examples→LiquidCrystal→HelloWorld

سنحتاج ان نعدل السطر البرمجى التالى :

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

ليصبح هكذا:

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

بعد تعديل السطر قم بتأكيد و رفع الكود الى بورده اردوينو كما فعلت فى الامثله السابقه

الكود البرمجي بعد التعديل:

```
//Example_12_LCD_16x2

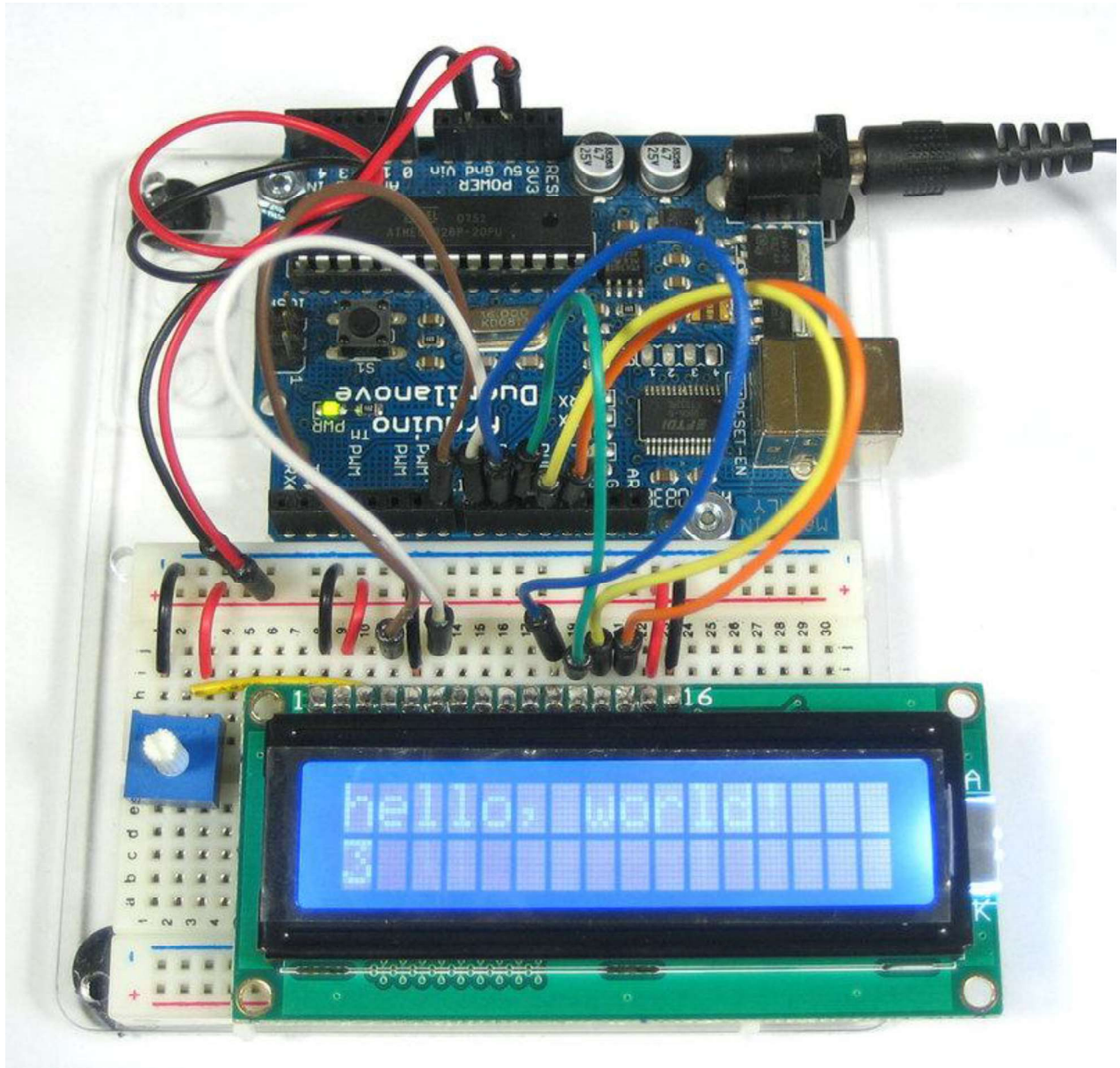
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

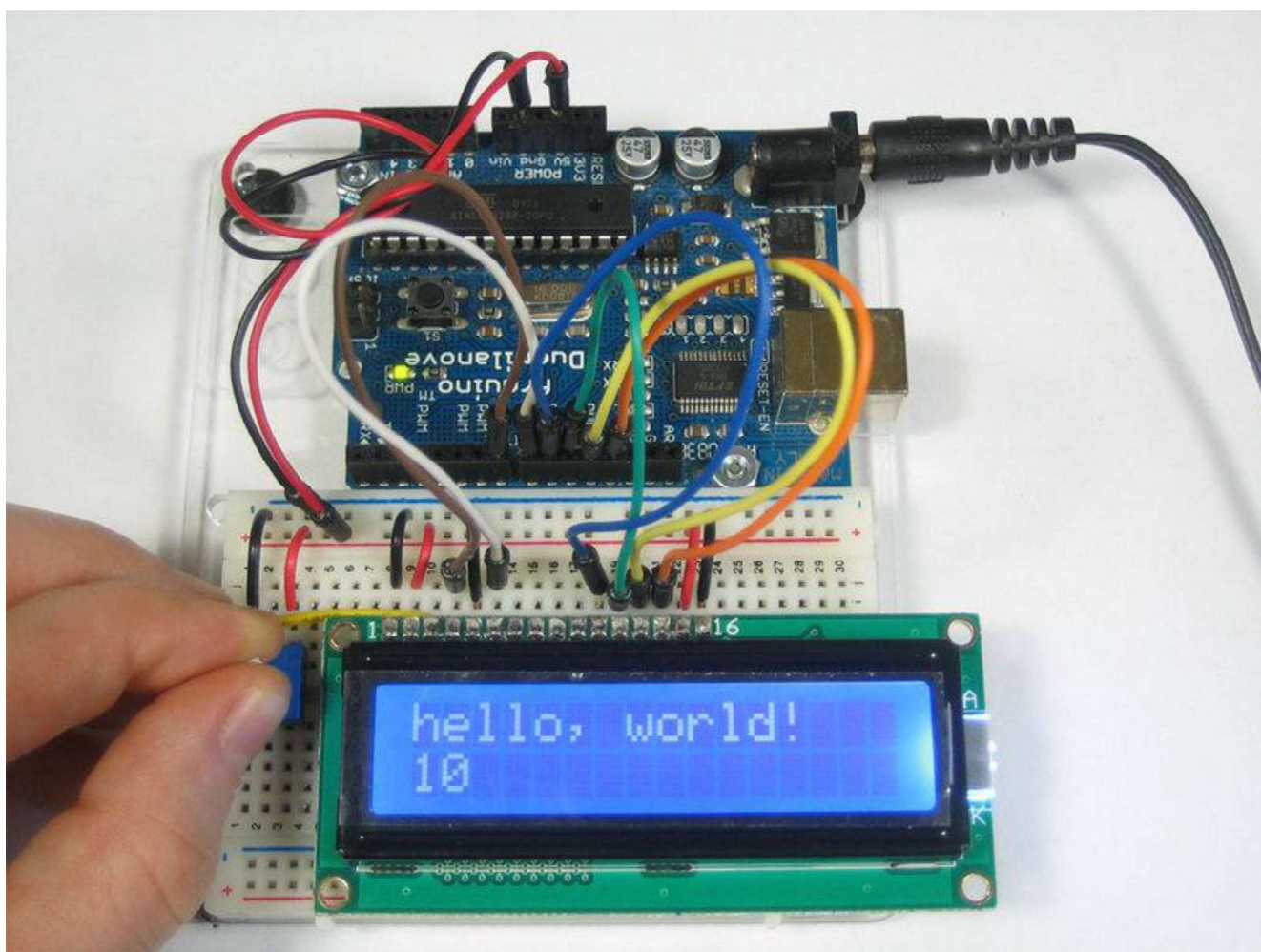
void setup()
{
    lcd.begin(16, 2);
    lcd.print("hello, world!");
}

void loop()
{
    lcd.setCursor(0, 1);
    lcd.print(millis()/1000);
}
```

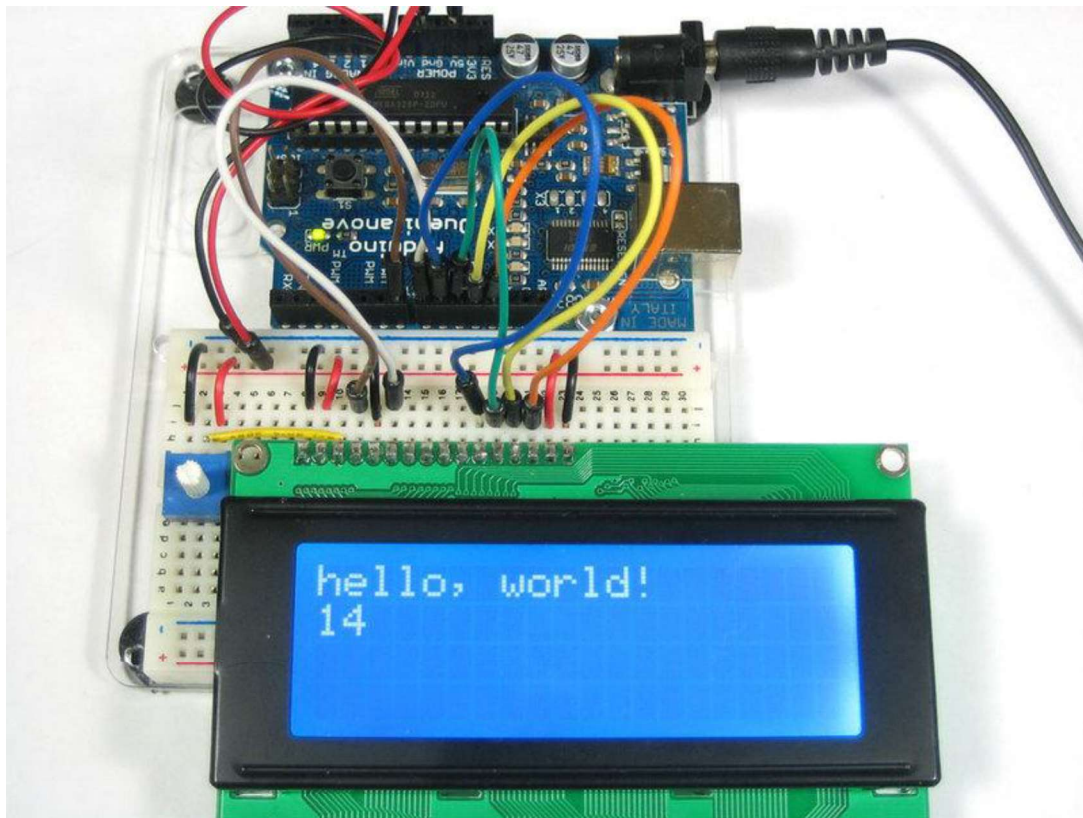
الشكل النهائي للمثال العاشر (^_^)

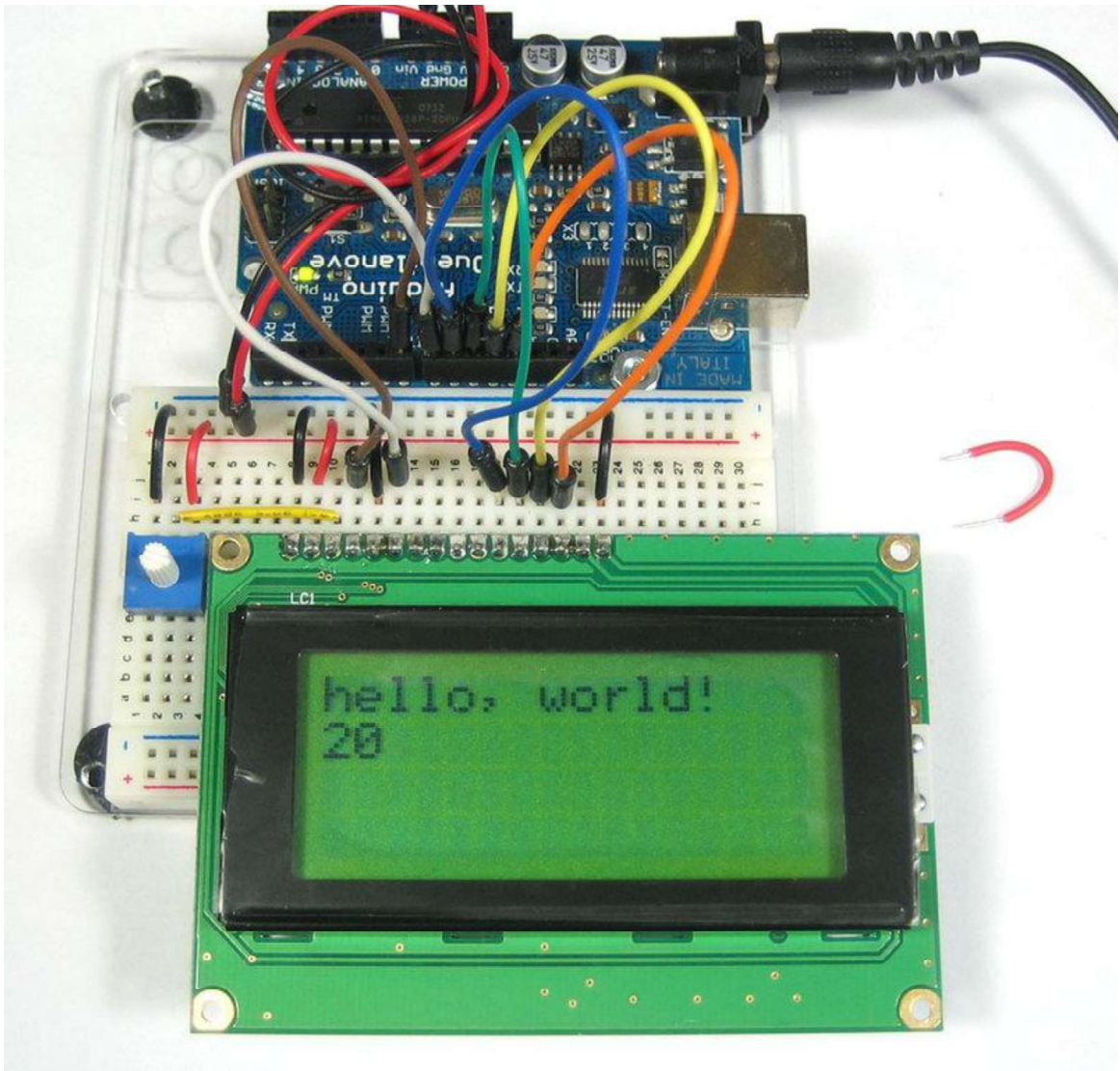


تستطيع تعديل مستوى سطوع الشاشة عن طريق تغير قيمه المقاومه المتغيره



كما يمكنك استخدام اى حجم و لون تفضله من الشاشات Character LCD مثل الحجم الاكبر فى عدد الحروف 20x4 كما فى الصور التاليه





مصادر اضافيه للمعلومات:

لمزيد من المعلومات و الأوامر المستخدمه مع شاشات LCD

<http://arduino.cc/en/Tutorial/LiquidCrystal>

لعمل اشكال مرسومه على شاشه من نوع character LCD

<http://www.instructables.com/id/Controlling-a-character-LCD-with-an-Arduino>

لتوصيل شاشه هواتف نوكيا الملونه Nokia 6100 LCD و Nokia 2600

<http://www.instructables.com/id/How-To-Use-a-Nokia-Color-LCD/>

المحاضرة 9 - Character LCD

الاختبار البعدي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. أي من الآتي مثال على **Character LCD** مذكور في المحاضرة؟

- A) 16×2
- B) LDR
- C) Servo
- D) Push Button

2. ما التعريف المستخدم لكائن **LCD** في مثال المحاضرة؟

- A) LiquidCrystal lcd(7,8,9,10,11,12)
- B) Servo lcd
- C) Keypad lcd
- D) Serial lcd

3. ما النص الذي يطبعه **setup** في المثال؟

- A) hello, world
- B) Arduino UNO
- C) Key Pressed
- D) Dark

4. ما وظيفة **lcd.setCursor(0,1)**؟

- A) وضع المؤشر في بداية السطر الثاني
- B) مسح الشاشة
- C) إيقاف الإضاءة
- D) تهيئة USB

5. ماذا يطبع **millis()/1000** في **loop**؟

- A) الزمن بالثواني تقريباً منذ بدء التشغيل
- B) درجة الحرارة
- C) رقم المنفذ
- D) عنوان I2C

6. أي دالة تُنفذ مرة واحدة في مثال **LCD**؟

- A) setup()
- B) loop()
- C) millis()
- D) print()

7. أي دالة تتكرر وتحديث القيمة المعروضة؟

- A) loop()
- B) include
- C) begin فقط
- D) constructor

8. ما المكتبة التي يجب تضمينها في بداية المثال؟

<A) #include <LiquidCrystal.h

<B) #include <Servo.h

<C) #include <Keypad.h

<D) #include <SPI.h

9. أي حجم آخر من **Character LCD** مذكور في المحاضرة؟

A) 20×4

B) 2×2 فقط

C) 8×1 فقط

D) 32×32 فقط

10. ما الغرض من **Pin Headers** في توصيل الشاشة؟

A) تسهيل التوصيل الكهربائي لأطراف الشاشة

B) قياس الحرارة

C) قراءة الضوء

D) تشغيل Serial

محاضرة LCD I2C 16×2

الاختبار القبلي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. كم سطرًا تعرض شاشة LCD 16×2؟

- (A) سطران
- (B) سطر واحد
- (C) 4 أسطر
- (D) 16 سطرًا

2. كم حرفاً يمكن أن يحتوي كل سطر حسب المحاضرة؟

- (A) 16
- (B) 2
- (C) 20
- (D) 32

3. ما البروتوكول المستخدم للتواصل مع الشاشة؟

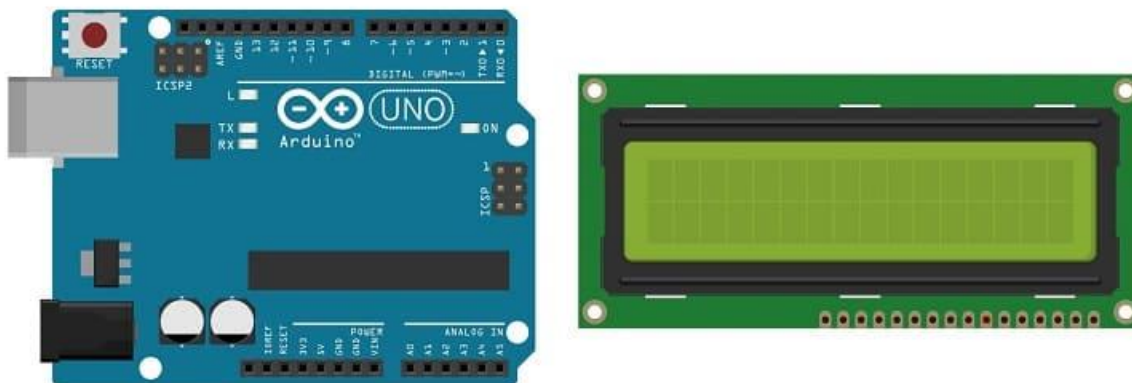
- (A) I2C
- (B) USB فقط
- (C) SPI فقط
- (D) UART فقط

4. ما فائدة I2C في هذه الشاشة؟

- (A) تقليل عدد الأسلاك المطلوبة
 - (B) زيادة حجم الشاشة
 - (C) زيادة عدد الأسطر
 - (D) تحويلها إلى حساس
5. ما نوع البيانات التي يمكن عرضها؟

- (A) نصوص وقراءات ورسائل
- (B) فيديو فقط
- (C) صوت فقط
- (D) ملفات تنفيذية

عرض النص على شاشة LCD 2X16 I2C المتصلة بلوحة الأردوينو



محاور الدرس

- 1- ما هي شاشة LCD I2C 2X16 ؟
- 2- كيف تعرض لوحة الأردوينو النص في شاشة LCD I2C 2X16 ؟
- 3- المكونات المطلوبة لعرض النص المتصل بالاردينو على شاشة LCD I2C 2X16
- 4- مخطط توصيلات لوحة الاردوينو مع شاشة LCD I2C 2X16
- 5- برمجة لوحة الاردوينو لعرض النص على شاشة LCD I2C 2X16

ما هي شاشة LCD I2C 2X16 ؟

شاشة LCD I2C 2x16 هي نوع من شاشات الكريستال السائل LCD تستطيع عرض سطرين كل منهما يحتوي على 16 حرفاً. تستخدم هذه الشاشة بروتوكول الاتصال I2C (Inter-Integrated Circuit) لنقل البيانات، مما يسهل الأسلاك المطلوبة للاتصال بها مع ميكروكونترولر مثل الأردوينو.

هذه الوحدة عادة ما تشمل واجهة I2C مدمجة تسمى " I2C backpack" أو وحدة I2C تعمل كواجهة بين الشاشة والميكروكونترولر، مما يمكن من التواصل بسهولة عبر بروتوكول I2C.

المواصفات الرئيسية لهذه الشاشة:

2 X16 : تعني أن الشاشة قادرة على عرض سطرين كل منهما يتكون من 16 حرفاً. يمكن عرض النصوص والبيانات على هذه السطور.

I2C: تستخدم بروتوكول الاتصال I2C الذي يسمح بنقل البيانات بين الشاشة والميكروكونترولر باستخدام أسلاك أقل، مما يسهل الاتصال والتحكم بها.

تُستخدم شاشات LCD هذا النوع في مشاريع الإلكترونيات لعرض المعلومات مثل قراءات الحساسات، الرسائل، حالات الأنظمة، أو أي بيانات تناسب الحد الأقصى للحروف المعروضة في السطرين. وتتيح إمكانية التواصل عبر بروتوكول I2C سهولة التكامل مع ميكروكونترولر مثل الأردوينو، مما يسهل الأسلاك المستخدمة ويجعلها مريحة لعرض المعلومات دون استهلاك عدد كبير من دبابيس الميكروكونترولر.

كيف تعرض لوحة الأردوينو النص في شاشة LCD I2C 2X16 ؟

لعرض النص على شاشة LCD I2C 2x16 باستخدام لوحة الأردوينو، يجب توصيل الشاشة باللوحة وكتابة الكود المناسب لعرض النص المطلوب. تأكد من تثبيت مكتبة "LiquidCrystal_I2C" في بيئة تطوير Arduino IDE قبل البدء.

التوصيل:

قم بتوصيل شاشة LCD I2C 2x16 بلوحة الأردوينو بالشكل التالي:

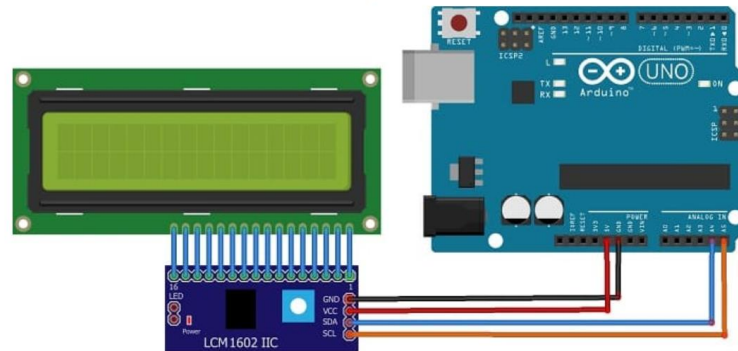
دبوس SDA (خط البيانات): قم بتوصيله بدبوس A4 على لوحة الأردوينو.

دبوس SCL (خط الساعة): قم بتوصيله بدبوس A5 على لوحة الأردوينو.

VCC: قم بتوصيله بـ 5V على لوحة الأردوينو.

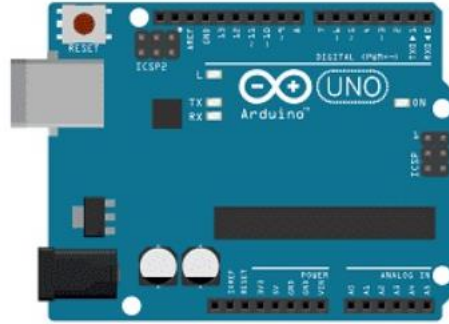
GND: قم بتوصيله بـ GND على لوحة الأردوينو.

مخطط توصيلات لوحة الاردوينو مع شاشة LCD I2C 2X16



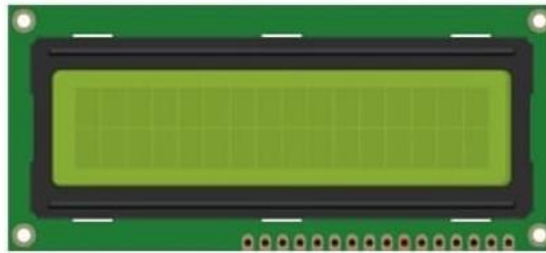
البرنامج:

اكتب البرنامج الذي يحدد ما تريد عرضه على شاشة LCD.
ستخدم وظائف المكتبة لكتابة النص، وتحريك المؤشر، ومسح الشاشة، وما إلى ذلك.
المكونات المطلوبة لعرض النص المتصل بالاردوينو على شاشة LCD I2C 2X16
لوحة الاردوينو:



لوحة الأردوينو هي لوحة إلكترونية تستخدم في تطوير الأجهزة الإلكترونية التفاعلية والمشاريع البرمجية. توفر الأردوينو بيئة برمجية واجهة سهلة للمبتدئين والمحترفين لتطوير الأجهزة والمشاريع المختلفة.

شاشة: LCD I2C 2X16

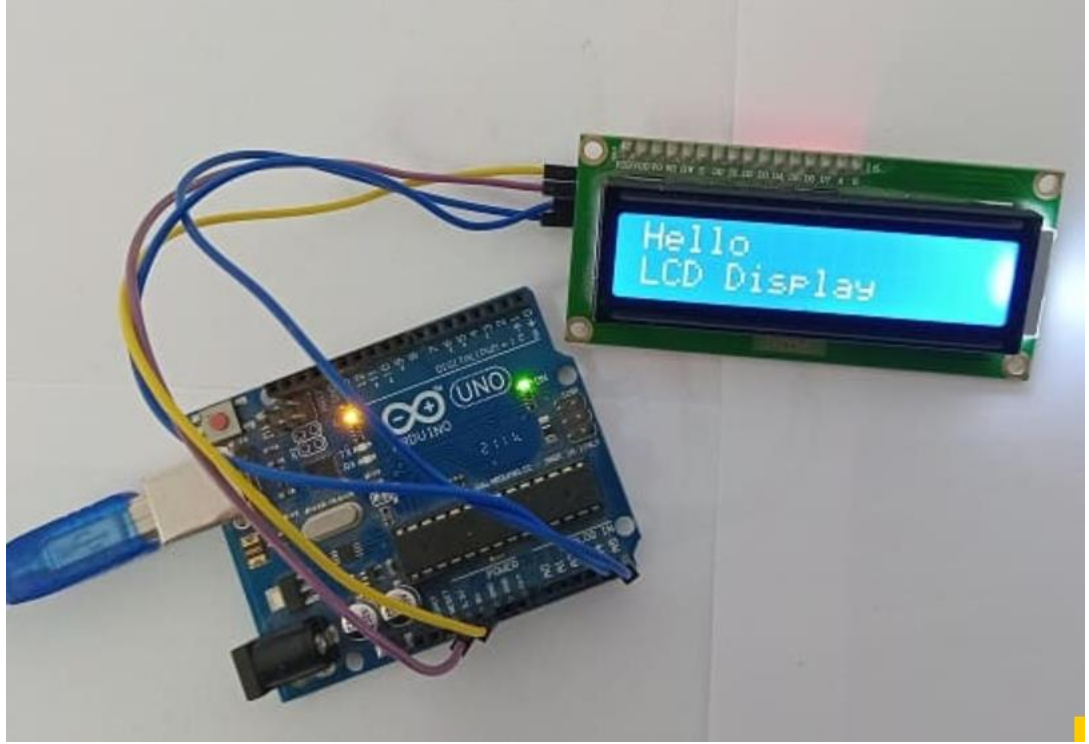


شاشة LCD تعتمد على تواصل I2C لعرض البيانات.

(Jumper wires) وصلات الأسلاك:



لتوصيل شاشة LCD بالاردوينو.



هذا المخطط يعرض الاتصالات الرئيسية بين لوحة الأردوينو Uno وشاشة LCD I2C 2x16
يجب أن تربط هذه الأسلاك بشكل صحيح حسب الأدوات التي تستخدمها وذلك باتباع الخطوات التالية:
قم بتوصيل دبوس VCC الخاص بالشاشة بـ 5V من الأردوينو.

قم بتوصيل طرف GND الخاص بالشاشة بـ GND الخاص بالأردوينو.

قم بتوصيل طرف SDA الخاص بالشاشة بالدبوس A4 (التناظري 4) الخاص بالأردوينو.

قم بتوصيل دبوس SCL الخاص بالشاشة بالدبوس A5 (التناظري 5) من الأردوينو.

برمجة لوحة الاردوينو لعرض النص على شاشة LCD I2C 2X16

بالطبع! هذا مثال بسيط لعرض نص على شاشة LCD I2C 2x16 باستخدام لوحة الأردوينو ومكتبة
LiquidCrystal_I2C.

```
1 #include <LiquidCrystal_I2C.h>
2 // إنشاء كائن من فئة LiquidCrystal_I2C
3 LiquidCrystal_I2C lcd(0x27, 20, 4);
4
5 void setup()
6 {
7   lcd.init(); // تهيئة الشاشة
8 }
9
10 void loop()
11 {
12   lcd.backlight(); // تشغيل الإضاءة الخلفية
13   lcd.clear(); // مسح محتويات شاشة LCD
14   lcd.setCursor(0, 0); // البداية من العمود الأول والصف الأول
15   lcd.print("Hello"); // النص الذي تريد عرضه
16   lcd.setCursor(0,1); // البداية من العمود الأول والصف الثاني
17   lcd.print("LCD Display"); // النص الذي تريد عرضه
18   delay(2000);
19   lcd.clear(); // مسح محتويات شاشة LCD
20   lcd.setCursor(0, 0); // البداية من العمود الأول والصف الأول
21   lcd.print("connected to"); // النص الذي تريد عرضه
22   lcd.setCursor(0,1); // البداية من العمود الأول والصف الثاني
23   lcd.print("Arduino UNO"); // النص الذي تريد عرضه
24   delay(2000);
25 }
```

محاضرة LCD I2C 16×2

الاختبار البعدي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. إلى أي منفذ يوصل SDA في Arduino Uno حسب المحاضرة؟

- A) A4
- B) A5
- C) 13
- D) 2

2. إلى أي منفذ يوصل SCL؟

- A) A5
- B) A4
- C) 9
- D) 3

3. إلى ماذا يوصل VCC؟

- A) 5V
- B) GND
- C) A0
- D) 13

4. إلى ماذا يوصل GND؟

- A) GND
- B) 5V
- C) SDA
- D) SCL

5. ما المكتبة المستخدمة في مثال البرمجة؟

- A) LiquidCrystal_I2C
- B) Servo
- C) Keypad
- D) EEPROM

6. ما وظيفة lcd.init()؟

- A) تهيئة الشاشة
- B) مسح الشاشة فقط
- C) قراءة زر
- D) تشغيل محرك

7. ما وظيفة lcd.backlight()؟

- A) تشغيل الإضاءة الخلفية
- B) إطفاء Arduino
- C) تغيير العنوان
- D) قراءة SDA

8. ما وظيفة lcd.clear()؟

(A) مسح محتويات الشاشة

(B) تحريك المؤشر فقط

(C) بدء Serial

(D) قراءة SCL

9. ما وظيفة `lcd.setCursor(0,1)`؟

(A) وضع المؤشر ببداية السطر الثاني

(B) إطفاء الشاشة

(C) تغيير 5V

(D) إنشاء مكتبة

10. أي أمر يطبع النص على الشاشة؟

A) `lcd.print`

B) `analogRead`

C) `digitalWrite`

D) `Serial.read`

محاضرة Keypad مع Arduino

الاختبار القبلي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. ما نوع لوحة المفاتيح المشروحة؟

- A) Matrix Keypad
- B) Mechanical keyboard للحاسوب
- C) Touchscreen
- D) LCD

2. كم مفتاحاً في Keypad 4×4؟

- A) 16
- B) 8
- C) 12
- D) 4

3. كيف تُرتب مفاتيح 4×4؟

- A) 4 صفوف و4 أعمدة
- B) 8 صفوف
- C) 16 عموداً
- D) صف واحد

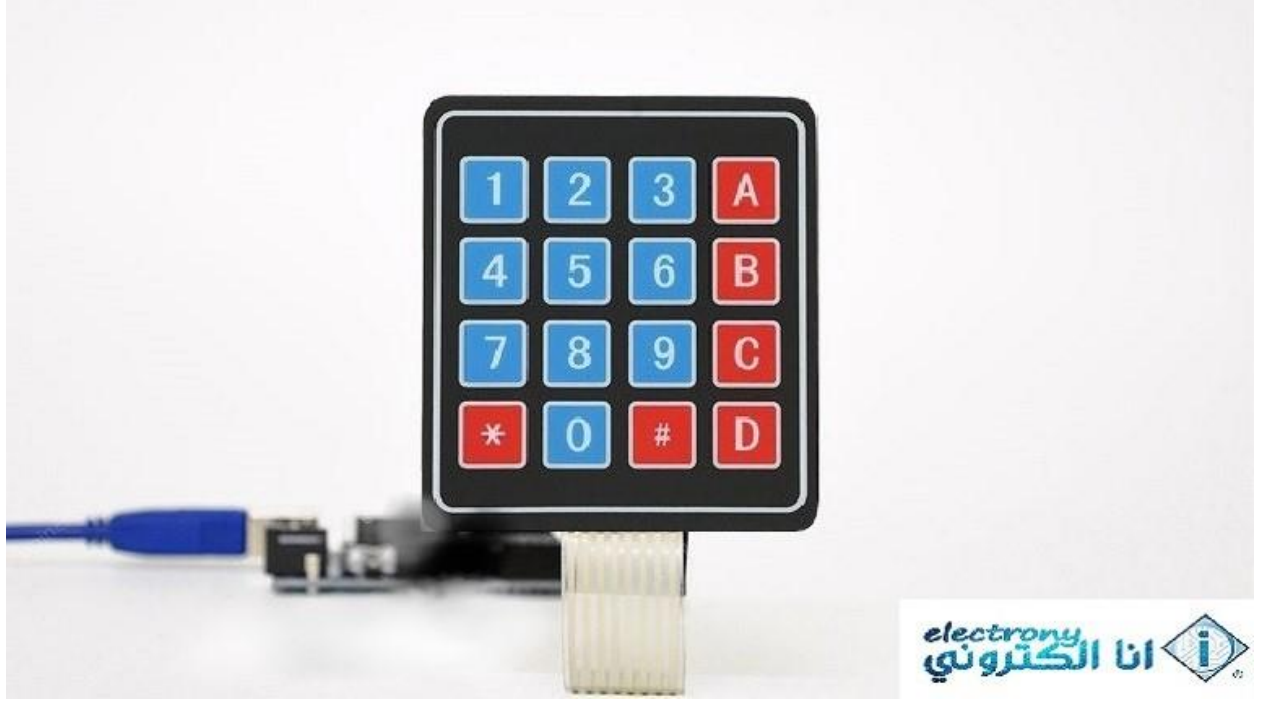
4. ما فائدة Matrix Keypad؟

- A) تقليل عدد منافذ المتحكم المطلوبة
- B) زيادة الجهد
- C) عرض الرسومات
- D) قياس الحرارة

5. ما المكتبة المستخدمة في الكود؟

- A) Keypad.h
- B) Servo.h
- C) LiquidCrystal.h
- D) SPI.h

إستخدام لوحة المفاتيح Keypad مع الاردوينو



ما هي لوحة المفاتيح وكيفية مواجهتها مع الاردوينو

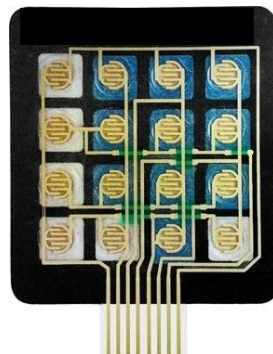
تعد لوحات مفاتيح Matrix نوع من أنواع لوحات المفاتيح التي نراها على الهواتف المحمولة، والآلات الحاسبة، وأفران الميكروويف، وأقفال الأبواب، وما إلى ذلك أي إنها موجودة في كل مكان تقريبًا. ومع ذلك، فهي طريقة رائعة للسماح للمستخدمين بالتفاعل مع المشروع الخاص بهم وغالبًا ما تكون هناك حاجة للتنقل في القوائم، ولإدخال كلمة في كلمات المرور والتحكم في الروبوتات وما شابه ذلك.

نظرة عامة على لوحة المفاتيح المرنة:

تصنع لوحات المفاتيح الغشائية من مادة غشاء رقيقة ومرنة. قد تأتي في أحجام 1×4 ، 4×4 ، 3×4 وما إلى ذلك. بغض النظر عن حجمها، فإنها تعمل بنفس الطريقة. أحد الأشياء الرائعة عنها هو أنها تأتي مع دعم لاصق حتى تتمكن من إرفاقه بأي شيء تقريبًا. عليك فقط أن تقشر ورقة التراجع.



لنأخذ لوحة المفاتيح 4×4 كمثال. لديها مجموع 16 مفاتيح. يوجد أسفل كل مفتاح غشاء خاص بجميع مفاتيح الغشاء هذه مرتبطة ببعضها البعض مع توصيل مكونة مصفوفة من 4×4 شبكة.



إذا كنت قد استخدمت 16 زر ضغط فردي، فستحتاج إلى 17 دبوس إدخال “زر واحد لكل مفتاح” لجعلها تعمل. ومع ذلك، في حالة استخدام لوحة مفاتيح، ما عليك سوى 8 دبابيس متحكم (4 أعمدة و 4 صفوف) لمسح من خلال لوحة.

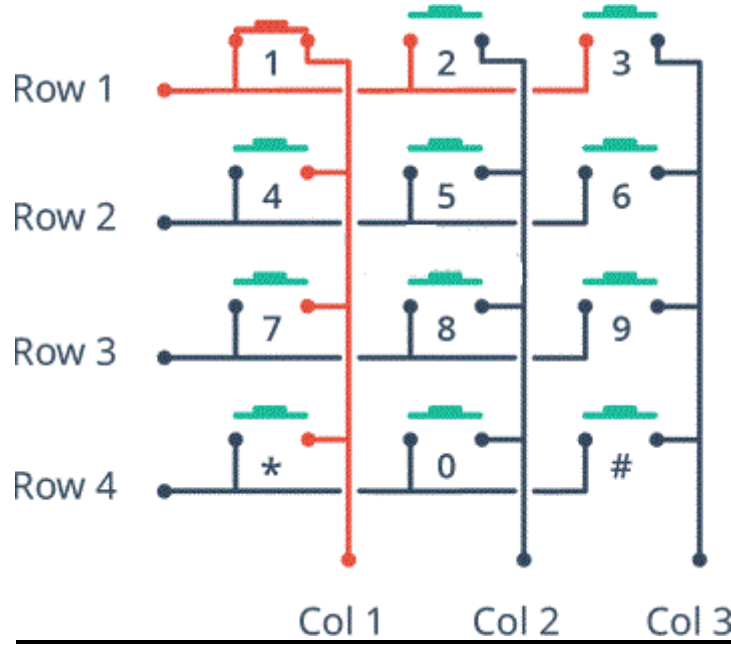
أطراف ومنافذ لوحة المفاتيح من النوع 3 × 4 و 4 × 4 :

أطراف غشاء لوحة المفاتيح 3 × 4 و 4 × 4 على النحو التالي.



كيف تعمل لوحة المفاتيح وكيف تقوم بمسحها؟

مبدأ العمل بسيط جداً. يؤدي الضغط على زر إلى قصر أحد خطوط الصفوف وأحد خطوط العمود، مما يسمح للتيار بالتدفق بينهما. على سبيل المثال، عند الضغط على المفتاح “4”، يتم اختصار العمود 1 والصف 2.

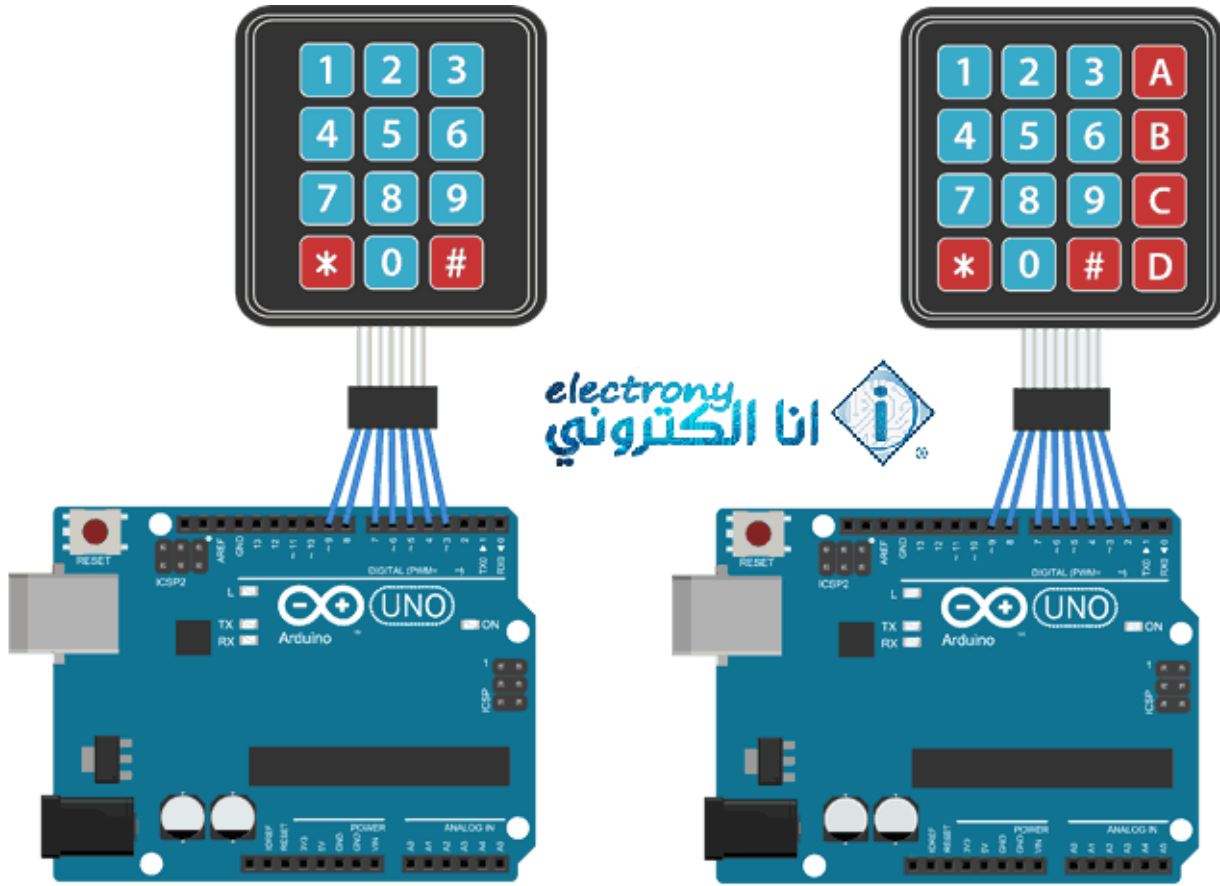


يمكن للمتحكم الدقيق مسح هذه الخطوط لمعرفة حالة الضغط على الزر. للقيام بذلك، فإنه يتبع الإجراء أدناه.

1. المتحكم يعين أو يضبط كل خطوط العمود والصف للإدخال.
2. ثم، تلتقط الإشارة من الصف وتعيينه أو تعتبره دخل عالي أو مرتفع.
3. بعد ذلك، يتحقق من خطوط الأعمدة واحدًا تلو الآخر.
4. إذا ظل اتصال العمود منخفضًا، فلن يتم الضغط على الزر الموجود في الصف.
5. إذا كان الأمر مرتفعًا، يعرف المتحكم الدقيق الصف الذي تم تعيينه، وأي عمود تم اكتشافه عاليًا عند تحديده.
6. أخيرًا، يعرف الزر الذي تم الضغط عليه والذي يتوافق مع الصف والعمود المكتشف.

الأسلاك - ربط لوحة المفاتيح مع الاردوينو:

الآن بعد أن عرفنا كل شيء عن لوحة المفاتيح الغشائية، يمكننا البدء في ربطها مع الاردوينو. الاتصالات واضحة ومباشرة. ابدأ بتوصيل pin 1 من لوحة المفاتيح بالرقم 9 الرقمي على الاردوينو. استمر الآن في توصيل المنافذ إلى اليسار مثل 2 مع 8، 3 مع 7، إلخ.



الكود البرمجي - لوحة المفاتيح:

سوف تمنحك الأكواد التالية فهماً تاماً لكيفية اكتشاف مكابس المفاتيح باستخدام لوحة مفاتيح غشاء 4×3 و 4×4 ويمكن أن تكون بمثابة أساس لمزيد من التجارب والمشاريع العملية.

Code for 4×3 Keypad

```
#include <Keypad.h>

const byte ROWS = 4; //four rows
const byte COLS = 3; //three columns

char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {5, 4, 3}; //connect to the column pinouts of the keypad

//Create an object of keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
  Serial.begin(9600);
}

void loop(){
  char key = keypad.getKey();// Read the key

  // Print if key pressed
  if (key){
    Serial.print("Key Pressed : ");
    Serial.println(key);
  }
}
```

Code for 4×4 Keypad

```
#include <Keypad.h>

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns

char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

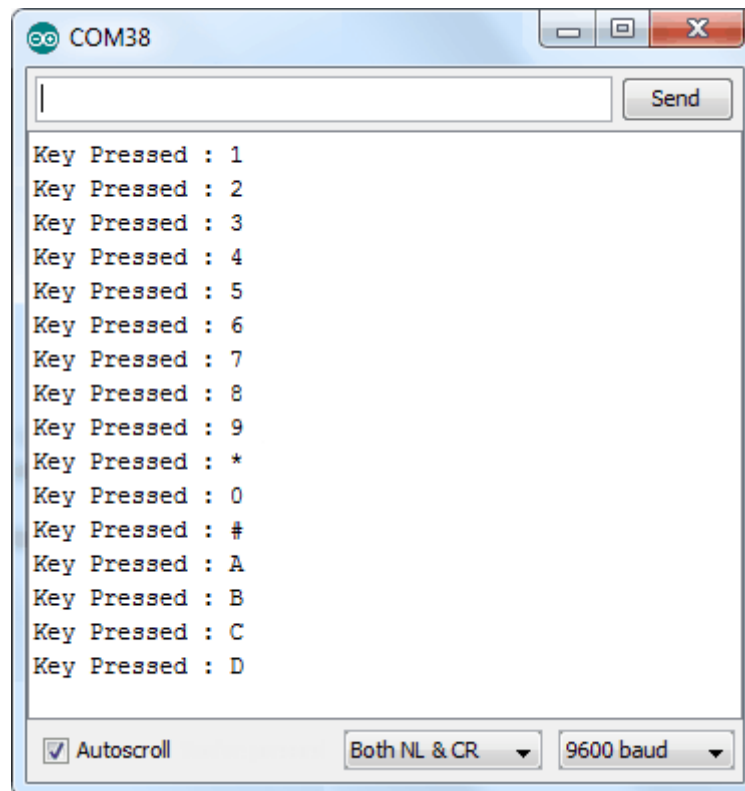
byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad

//Create an object of keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
  Serial.begin(9600);
}

void loop(){
  char key = keypad.getKey();// Read the key

  // Print if key pressed
  if (key){
    Serial.print("Key Pressed : ");
    Serial.println(key);
  }
}
```



محاضرة Keypad مع Arduino

الاختبار البعدي - اختر الإجابة الصحيحة

الاسم: _____ الشعبة: _____ التاريخ: _____

1. كم دبوس متحكم تحتاج Keypad 4×4 حسب المحاضرة؟

- A) 8
- B) 16
- C) 17
- D) 4

2. عند ضغط مفتاح، ماذا يحدث كهربائياً؟

- (A) يتم وصل خط صف بخط عمود
- (B) تزداد التغذية إلى 12V
- (C) يُمسح البرنامج
- (D) يتوقف Serial

3. عند ضغط المفتاح 4 في المثال، أي تقاطع دُكر؟

- (A) العمود 1 والصف 2
- (B) العمود 4 والصف 4
- (C) الصف 1 فقط
- (D) العمود 2 فقط

4. في كود 3×4، كم عدد **ROWS**؟

- A) 4
- B) 3
- C) 7
- D) 12

5. في كود 3×4، كم عدد **COLS**؟

- A) 3
- B) 4
- C) 8
- D) 16

6. ما الدالة التي تقرأ المفتاح المضغوط؟

- (A) keypad.getKey
- (B) digitalWrite
- (C) lcd.print
- (D) Servo.write

7. ما سرعة **Serial** المستخدمة في المثال؟

- A) 9600
- B) 255
- C) 1000
- D) 115

8. في كود 4×4، ما الأحرف الموجودة في العمود الرابع؟

A) A,B,C,D

B) W,X,Y,Z

C) E,F,G,H

D) لا توجد أحرف

9. ما منافذ الصفوف في المثال البرمجي؟

A) 9,8,7,6

B) 5,4,3,2

C) A0-A3

D) 10,11,12,13

10. ما منافذ الأعمدة في مثال 4×4 ؟

A) 5,4,3,2

B) 9,8,7,6

C) A4,A5

D) 0,1 فقط

المحاضرة 3 - البدء مع Arduino و LED

القبلي: 1-A-5 2-A 3-A 4-A

البعدي: 1-A-10 2-A 3-A 4-A 5-A 6-A 7-A 8-A 9-A

المحاضرة 4 - Push Button وتشغيل/إطفاء LED

القبلي: 1-A-5 2-A 3-A 4-A

البعدي: 1-A-10 2-A 3-A 4-A 5-A 6-A 7-A 8-A 9-A

المحاضرة 5 - المداخل التناظرية والحساسات

القبلي: 1-A-5 2-A 3-A 4-A

البعدي: 1-A-10 2-A 3-A 4-A 5-A 6-A 7-A 8-A 9-A

المحاضرة 6 - المقاومة الضوئية LDR

القبلي: 1-A-5 2-A 3-A 4-A

البعدي: 1-A-10 2-A 3-A 4-A 5-A 6-A 7-A 8-A 9-A

المحاضرة 7 - ربط Arduino بالحاسوب

القبلي: 1-A-5 2-A 3-A 4-A

البعدي: 1-A-10 2-A 3-A 4-A 5-A 6-A 7-A 8-A 9-A

المحاضرة 8 - المحركات DC و Servo

القبلي: 1-A-5 2-A 3-A 4-A

البعدي: 1-A-10 2-A 3-A 4-A 5-A 6-A 7-A 8-A 9-A

المحاضرة 9 - Character LCD

القبلي: 1-A-5 2-A 3-A 4-A

البعدي: 1-A-10 2-A 3-A 4-A 5-A 6-A 7-A 8-A 9-A

محاضرة LCD I2C 16×2

القبلي: 1-A-5 2-A 3-A 4-A

البعدي: 1-A-10 2-A 3-A 4-A 5-A 6-A 7-A 8-A 9-A

محاضرة Keypad مع Arduino

القبلي: 1-A-5 2-A 3-A 4-A

البعدي: 1-A-10 2-A 3-A 4-A 5-A 6-A 7-A 8-A 9-A