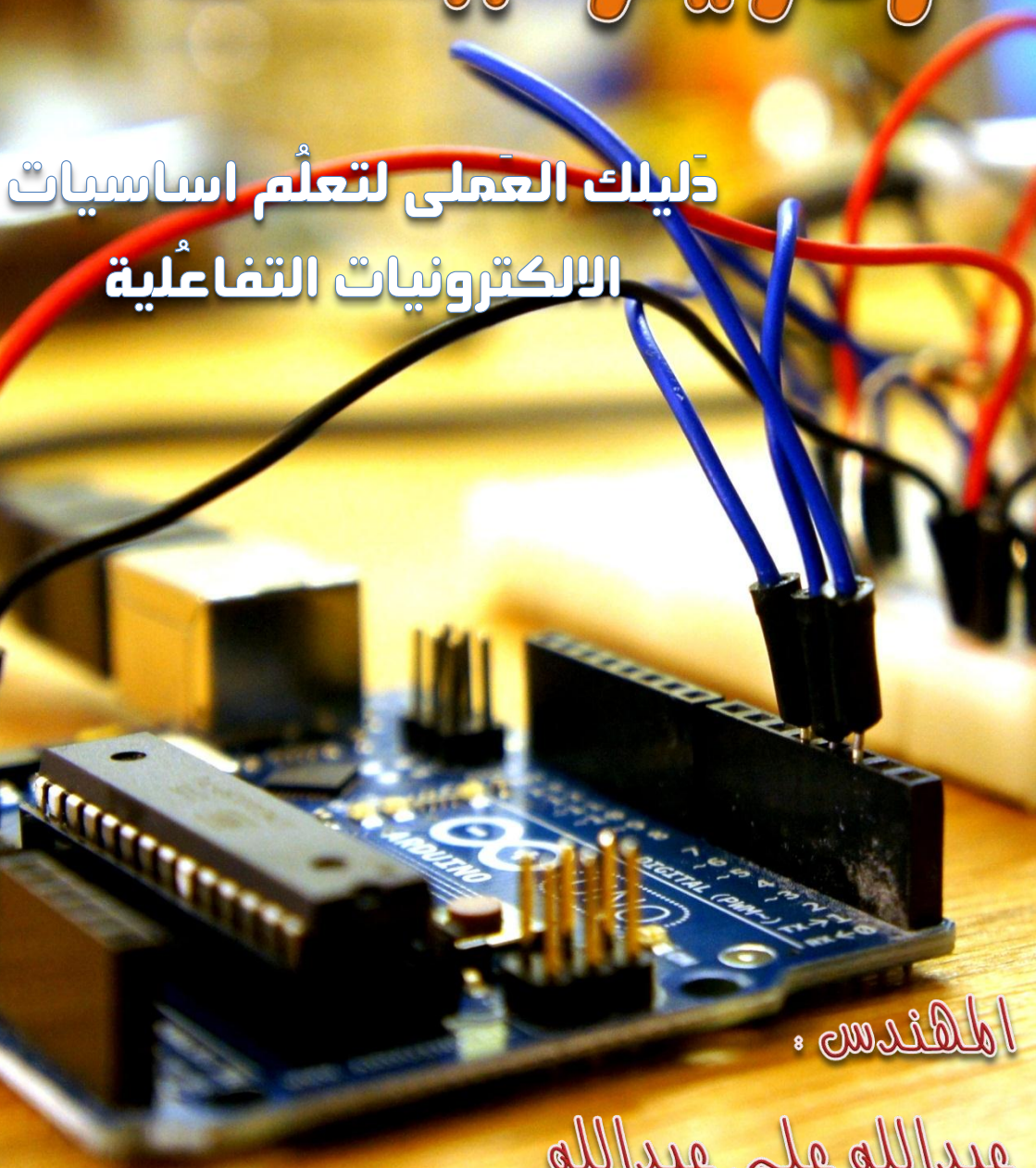


# اردوينو ببساطة

دليلك العملي لتعلم اساسيات  
الالكترونيات التفاعلية



المهندس :

عبدالله علي عبدالله

# رخصة الكتاب،

الكتاب يوضع للرخصة المره مفتوحه المصدر، بصورته الألكترونيه مجاناً  
و يمكنك ان تنشره او تُعده كيفما تشاء شرط أن تذكر المصدر،

- الصور ثلاثيه الابعار المستخرمه فى الكتاب مأفوضه من موقع [oomlout.co.uk](http://oomlout.co.uk) و توضع ايضاً للرخصة المره مفتوحه المصدر،
- يمكنك ان تستغل ذلك العمل بصوره تجاريه بشرط المصول على اذن مباشر من المؤلف.

اذا اعجبك الكتاب وأهبيت دعم الكاتب و المزيد من الكتب

العريه المره مفتوحه المصدر، فلا تتردد بالتواصل معى

[abdallah.ali.abdallah.elmasry@gmail.com](mailto:abdallah.ali.abdallah.elmasry@gmail.com)

# اهداء

الى والدى العزيزين....

الى والدتى العزيزة....

الى كل من ساهم فى نشر المعرفة بحريه

[www.networkset.net](http://www.networkset.net)

أيمن النعيم

[www.aabouzaid.com](http://www.aabouzaid.com)

أحمد محمد أبو زيد

[www.shabayek.com](http://www.shabayek.com)

رءوف شبايك

[linuxac.org](http://linuxac.org)

مجتمع لينكس العرب

## فهرس الكتاب

- رخصه الكتاب..... ( 1 )
- معلومات حول الكتاب..... ( 5 )
- الفصل الأول: مقدمه عن اردوينو و المتحكيمات الدقيقة..... (7)
- الفصل الثاني: التجهيزات..... (21)
- الفصل الثالث: ابدأ مع اردوينو ..... (43)
- الفصل الرابع: استخدام الحساسات ..... (70)
- الفصل الخامس: التواصل مع الحاسب الآلى..... (92)
- الفصل السادس: استخدام المحركات ..... (113)
- الفصل السابع: وسائل الأدخال و الاخراج المتقدمه..... (125)
- الفصل الثامن: أعطيه اردوينو ..... (166)
- الفصل التاسع: تاريخ تطور اردوينو..... (173)
- تعلم و شارك الاخرين Fablab Egypt..... (179)
- ملحق المراجع: مراجع (كتب + مواقع تعليميه)..... (183)

قال الله ﷻ في القرآن الكريم:

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

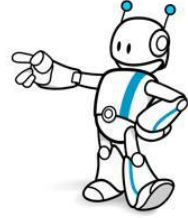
أَقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿١﴾ خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ ﴿٢﴾

أَقْرَأْ وَرَبُّكَ الْأَكْرَمُ ﴿٣﴾ الَّذِي عَلَّمَ بِالْقَلَمِ ﴿٤﴾ عَلَّمَ الْإِنْسَانَ

مَا لَمْ يَعْلَمْ ﴿٥﴾

## الفئات المستهدفة من الكتاب

الى كل من يريد تعلم تقنيات التحكم الآلى بسهولة و يسر سواء كان لك معرفه تقنيه او خبره بالمتحكمات الدقيقه او حتى ما زلت فى بدايه تعلم هذا المجال و الى كل من لديه الشغف لمعرفة الجيل الجديد من المتحكمات الدقيقه مفتوحه المصدر



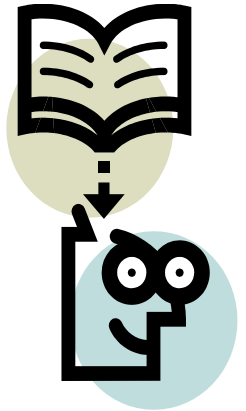
## هل احتاج الى خبره فى مجال الإلكترونيات و البرمجة حتى اقرأ هذا الكتاب ؟



- بعض المعرفه باساسيات الالكترونيات و البرمجه محبذه جدا لانيها ليست شرط ضرورى للتعلم فلقد قمت بكتابه المعلومات بأسلوب سهل و ميسر قدر الامكان و يمكنك اذا واجهت صعوبه ما ان تراسلنى على البريد الالكترونى ☺

## ما الذى سوف اتعلمه فى هذا الكتاب؟؟

- ✓ كيف تصمم الإلكترونيات التفاعلية بنفسك
- ✓ العمل على بيئة تطوير اردوينو Arduino IDE
- ✓ التحكم الالكترونى لكل ما هو حولك من اجهزه الكترونيه

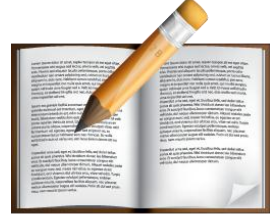




**سيجعلك الكتاب** تكتسب المهارات الأوليه لفهم الأكواد البرمجيّه المستخدمه في العديد من المشاريع التي تعتمد على **اردوينو** ، ولقد اضفت العديد من المصادر الخارجيه التي تحتوى على العديد من المشاريع الرائعه و التي يمكنك ان تنفذها بنفسك بعد قراءة الكتاب

## كيف تم تصميم الكتاب ؟

**تم تصميم** الكتاب بأسلوب يعتمد على الشرح بالأمثله Examples و حيث يتم تقديم المعلومات على هيئه أمثله صغيره و يقدم كل مثال جزء جديد في تعلم لغه برمجته اردوينو ، كما ستجد في نهاية كل فصل صفحه مخصصه لكي تكتب ملاحظاتك الشخصية.



اذا كنت من مستخدمي ويندوز يمكنك أن تكتب داخل ملفات الكتب الإلكترونية من نوع PDF باستخدام برنامج Foxit PDF و اذا كنت تقرأ الكتاب باستخدام لوح الكتروني Tablet مثل Ipad يمكنك استخدام برنامج Good Reader مع ملاحظه ان الكتاب متوافق مع جميع اجهزه التابلت المختلفه التي تمتلك شاشه بقياس 7 بوصة او اكبر .

# الفصل الأول

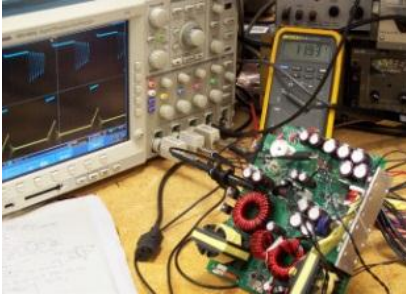
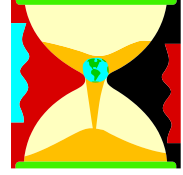
مقدمه الى عالم اردوينو و المتحكمات الدقيقه

Introduction to Arduino & microcontrollers



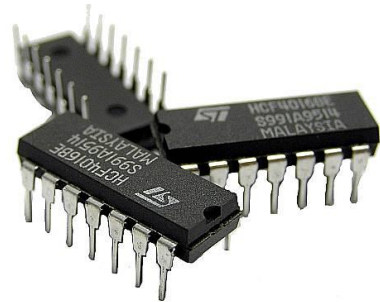


**منذ زمن** ليس بعيد كان العمل على صناعة دائره الكترونيه للقيام بوظيفه معينه يعنى بناء تصميم الكترونى معقد من مكونات مثل المقاومات، المكثفات، الملفات، الترانزستور... الخ.



**كانت** الدوائر الالكترونيه ثابتة التصميم و اعاده تغير او تعديل جزء بسيط فيها كان يعنى الكثير من عمليات معقدة مثل اللحام و قطع الاسلاك و اعاده النظر فى المخططات الالكترونيه و الكثير من الامور المزعجه و التى ادت إلى اقتصار وظيفه تطوير المنتجات الالكترونيه على مجموعه من المهندسين المتخصصين فقط.

**بفضل** التطور التكنولوجى فى مجال اشباه الموصلات و اختراع الدوائر المدمجه Integrated Circuits (IC) اصبح من الممكن وضع دائره الكترونيه كامله على شريحة صغيره حجمها قد لا يتجاوز رأس الدبوس حتى انه فى الوقت الحالى هناك دوائر الكترونيه حجمها يقدر بالنانومتر Nano Meter و التى لا يمكن رؤيتها الا بأستخدام مكبرات ضوئيه خاصه.



**كما ادى** تطور Integrated Circuits الى ظهور جيل خاص من الدوائر الالكترونيه يسمى المتحكمات الدقيقه **Micro Controllers** وهى اشبه بكمبيوتر مصغر قابل للبرمجه لاداء مجموعه من الوظائف مثل قرائه درجه حراره، التحكم فى محرك كهربى، او حتى اداره خطوط الانتاج فى المصانع الكبرى، و كل ذلك يتم ببساطه عن طريق اوامر برمجيّه و بذلك تحولت تقنيه صناعه الدوائر و الانظمه الالكترونيه من التصميم الالكترونى البحت المعتمد على المكونات الصلبه فقط الى اوامر برمجيّه يمكن لاي فرد ان يكتبها و يصممها بنفسه بسهوله و يسر.



**تتميز** المتحكمات الدقيقه ايضا بأمكانيه التغير و التعديل فى اى وقت ، بكل بساطه لو اردت ان تغير شيئاً ما فى مشروعك يمكنك ذلك بالتعديل فى السطور البرمجيّه و اعاده وضع الاوامر الجديده على المتحكمه الدقيقه و تجربتها اكثر من مره و هكذا الى ان تصل بمشروعك للهدف المنشود ...



**اردوينو Arduino** هي بورده الكترونيه مفتوحه

المصدر Open Hardware لتطوير الكثير من الافكار و المشاريع المتعلقه بالتحكم الآلى بصوره سهله و بسيطه عن طريق استخدام لغه برمجيه مفتوحه المصدر Arduino C و يتم برمجته المتحكمه الموجوده على البورده بأستخدام

برنامج خاص يسمى **Arduino IDE: Integrated Development Environment**

والذى يتوفر مجاناً للتحميل من موقع اردوينو الرسمى

<http://arduino.cc>



## ما معنى ان اردوينو مفتوح المصدر؟؟

**يعنى انه** يمكنك الاطلاع و التعديل على

التصميمات الهندسية و الشفرات المصدرية

Source Codes لكل من بوردهات **اردوينو**

المختلفة Arduino و Arduino Boards

IDE بما يتناسب معك و يمكنك ايضاً

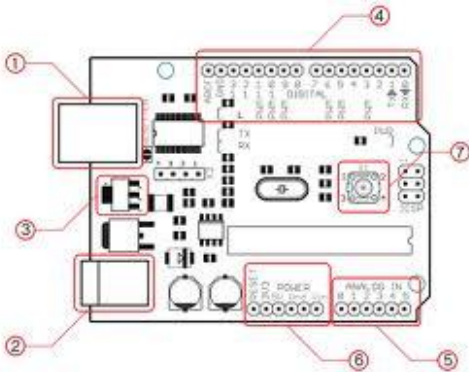
تطوير لغة برمجيه Arduino C بحريه تامه و

الاطلاع على الشفرات المصدرية الخاصة بها

كما ان كل هذه المميزات و البرمجيات مجانيه تماماً على غرار بعض البيئات التطويره

مثل **Mikro C** و التى تتطلب منك شراء رخصه مكلفه تصل فى بعض الاحيان الى الآف

الدولارات لاستخدامها.



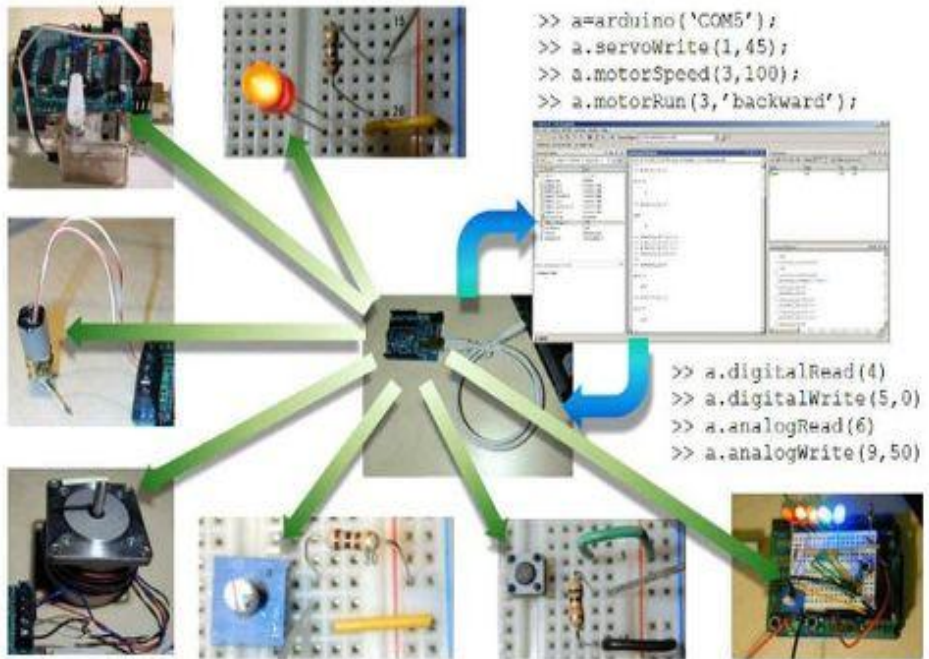
**ما يميز بوردات اردوينو** عن باقي البوردات التطويريه للمتحكمات الدقيقه الاخرى Micro Controllers Development boards هو مدى السهوله التعامل معها و بساطه اللغه البرمجييه و التي عمل فريق من ايطاليا على تطويرها منذ عام 2005 حتى الان ، و لقد تم اشتقاق لغه اردوينو البرمجييه من لغه processing و لغه "سى C" و التي تعد اساس لغات البرمجه الحديثه و صاحبه ثوره تقنيه البرمجيات.



**قد يظن** البعض ان اردوينو مصمم للهواه فقط لآكن هذا ليس صحيح لقد تم تطويره ليناسب جميع المستويات ابتداء من الهواه و انتهاء بالمشاريع المتطوره و الدليل انه هناك مميزات جباره تجعل

**اردوينو** على قمة المتحكمات الدقيقه و هى امكانيه دمجها فى مشاريع يتم برمجتها بلغات هندسيه متطوره مثل MATLAB و لغه Java حيث ستجد مكتبات برمجييه فى جاهزه للغه MATLAB و الجافا خاصه بالتعامل مع **اردوينو**، و ستجد فى صفحه المراجع بعض الموارد التي تعلمك كيفيه استخدام الاردوينو مع المات لاب، كما يمكنك ايضا استخدام لغه Java و لغه VB.NET فى التواصل مع بوردات **اردوينو** المختلفه

**على سبيل المثال:** يمكنك عمل دائره استشعار للطقس و الحراره بالاردوينو و ارسال البيانات الى MATLAB على الحاسب الآلى لاجراء التحليلات الحساييه المتطوره المتوفر بلغه MATLAB.



**من اكثر المشاريع** التي اثاروا اعجابى كانت مستكشف زلازل باستخدام الاردوينو و لغه برمجته MATLAB و كانت من تصميم شاب فى السادسه عشر من عمره فى الولايات المتحده حيث قام بعمل وحده استشعار و تحليل للزلازل بتكلفه لا تتجاوز 80 دولار باستخدام بعض المستشعرات و بورده **اردوينو** فقط مع العلم ان الدوائر الالكترونيه المماثله قد تتكلف اكثر من 2000 دولار س

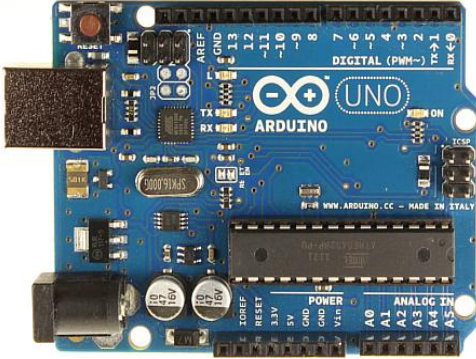
## انواع بوردات اردوينو Arduino Boards



Arduino UNO, Arduino Mega, Arduino Nano, Arduino Mini, Arduino LilyPad, Arduino Duemilanove, Boardduino

**تختلف** البوردات عن بعضها البعض من ناحية عدد المخارج و المداخل و التي تحدد عدد الاجهزه التي يمكن التحكم بها و عدد الحساسات Sensors التي يمكن دمجها مع البورده و كذلك نوع المتحكمه الدقيقه و سرعه المعالج الموجود بداخلها و امكانيه

تبدالها ام لا و سوف نتناول في هذا الكتاب بورده **Arduino Uno**



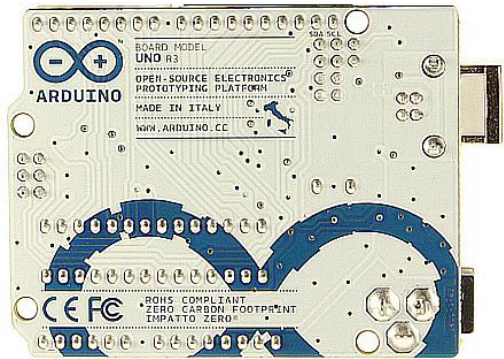
## نظرة عامه على دائره

### Arduino Uno

**دائرة الكترونيه** صغيره تستخدم فى برمجته متحكمه من شركه اتمل ATmega328 و توفر هذه الدائرة منافذ لتوصيل المكونات

الالكترونيه الى المتحكمه مباشره عن طريق 14 (مدخل | مخرج) من النوع الرقمي Digital In/out من هذه ال 14 يوجد 6 يمكن استخدامها كمخارج PWM أو ما يعرف **بالتعديل الرقمي المعتمد على عرض النبضه (Pulse-Width modulation)** و سوف نتحدث عن هذه الخاصيه فى فصل كامل يسمى الدخل و الخرج التماثلى Analog Inputs & Outputs

**ايضا** تحوي الدائرة على مهتز كريستال Crystal Oscillator بتردد 16MHz، بالاضافه الى مدخل USB من أجل التواصل مع الحاسب، وهناك مدخل للطاقة منفصل، بالاضافه الى ICSP header والذي يعتبر طريقه اضافيه لبرمجة المتحكمه وهي لا تزال موصله

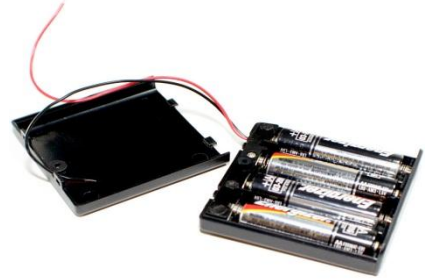
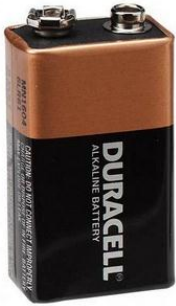


بالبورده (بخلاف الUSB) و يمكنك ان تعتبر بورده اردوينو هذه بورده تطوير و برمجته مصغره و مهيئه للاستخدام المباشر Development Board فهى تقريبا تحوي كل ما تحتاج لكي تعمل عليها سواء عن طريق منفذ الUSB أو عن طريق مصدر خارجى للطاقة مثل البطاريه.

## Power up امداد الدائرة بالطاقة



**يمكنك** امداد الدائرة بالطاقة إما من خلال منفذ الـ USB فقط أو عن طريق استخدام مصدر خارجي للكهرباء كمحول AC\DC ليمد الدائرة بالجهد اللازم للعمل او حتى عن طريق بطارية 9 فولت او 4 بطاريات 1.5 فولت حيث يتم توصيل طرفي البطارية الى مدخل الارضي Gnd و الـ Vin في الدائرة.



**تستطيع** الدائرة العمل على جهد يتراوح بين الـ 6-20 فولت، لكن يجب الانتباه حيث انه اذا قمنا بتأمين جهد أقل من 7 فولت فإن المخرج المتحكمه Pin 5V قد لا يستطيع تأمين جهد خرج يبلغ الـ 5 فولت المطلوب وقد يؤدي الى عدم استقرار الدائرة، اما اذا قمنا بتزويد الدائرة بجهد أعلى من 12 فولت فإنه قد يؤثر على عنصر تنظيم الجهد voltage regulator ويؤدي الى ارتفاع درجة حرارته مما يؤدي الى تلف البورده ، لذا فإن مجال الجهد الذي يفضل استخدامه هو من 7 الى 12 فولت.

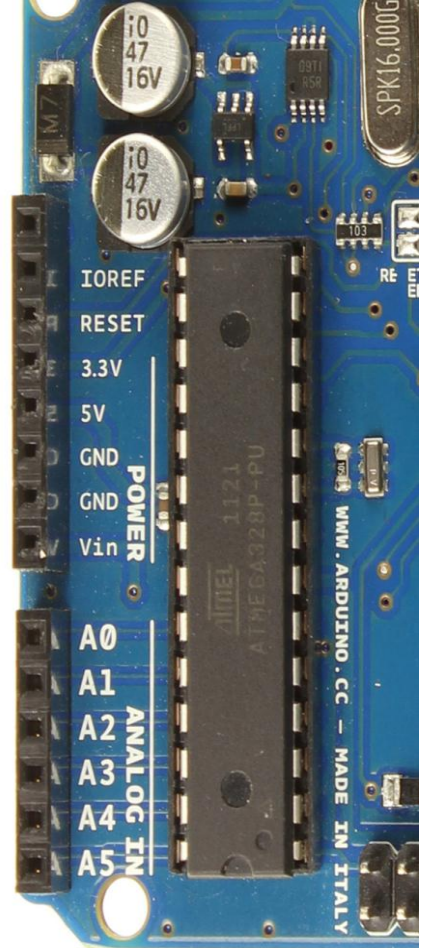




## مخارج و مداخل الطاقة الكهربيه للمتكمه

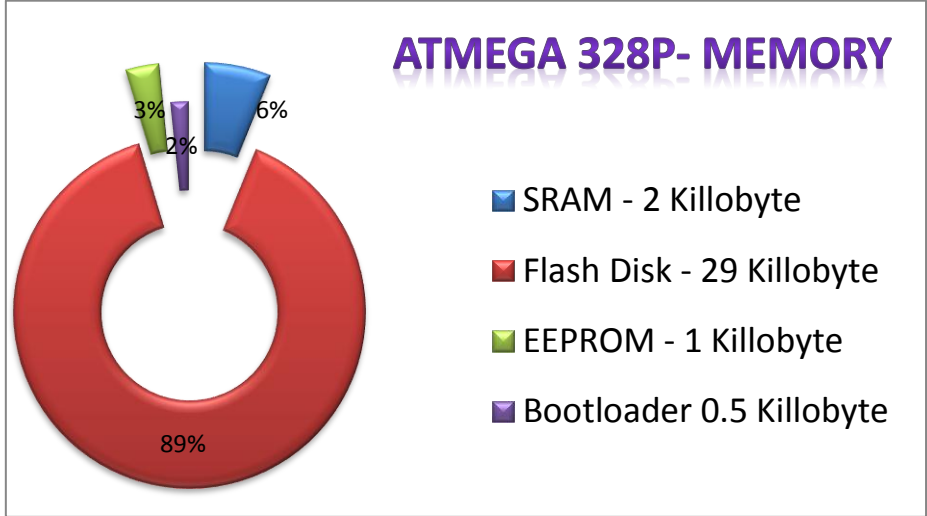
### Power Inputs/Outputs

- **Vin** : جهد الدخل عندما نستخدم مصدر طاقة خارجي، يمكننا تأمين الجهد من خلال هذا المدخل ، إذا كنا نقوم بتأمين الطاقة للدائرة من خلال مدخل المحول يمكننا الوصول له من خلال هذا المدخل أيضاً.
- **V5** : جهد منظم يستخدم لتأمين الطاقة للعناصر المستخدمه على الدائرة و سوف نستخدمه لتوفير الطاقه للقطع الالكترونيه التى سنضيفها، قد يأتي هذا الجهد من خلال Vin عبر منظم جهد داخلي أو تأمينه من خلال منفذ ال USB أو أي مصدر جهد منظم بقيمه 5 فولت.
- **V3.3** : مصدر للجهد بقيمة 3.3 فولت مؤمن من قبل منظم الجهد الداخلي للدائرة و أقصى قيمة لسحب التيار من خلال هذا الخط هو 50 ميلي أمبير.
- **GND** : الخط الارضي.



## المعالج الدقيق و الذاكرة Microprocessor

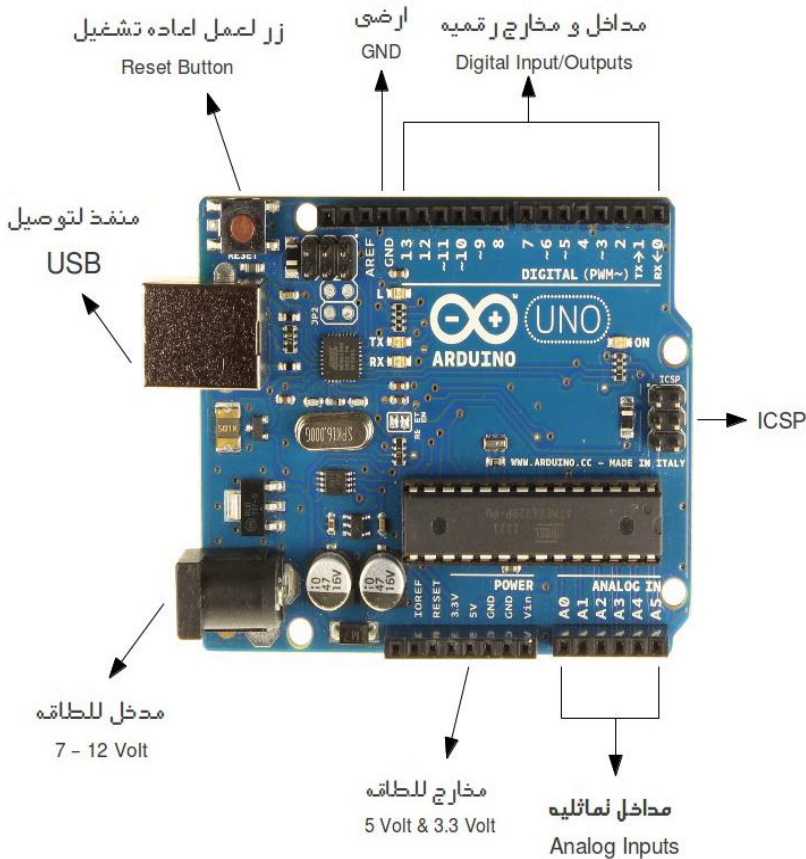
المتحكمات الدقيقة اشبهه بوحده حاسب آلى صغيره الحجم و تحوي المتحكمه الدقيقه  
ATmega328 على معالج بسرعه **16 ميجا** هرتز و ذاكرة كليه تساوى **32 Kilo Byte**



- **Boot loader** : السوفت وير المسئول عن كيفيه فهم الدائره للغه **Arduino C**
- **SRAM** : تعتبر الذاكره المستخدمه فى تسجيل المتغيرات بصوره مؤقتة.
- **Flash Disk** : مساحه تخزينيه تستخدم فى تخزين البرنامج الذى سنكتبه لتشغيل المتحكمه ، فى الوهله الاولى قد يبدو هذا الرقم صغير جدا لكنه فى الحقيقه كافي لكتابه الكثير من الاوامر كما سنرى فى الامثله القادمه.
- **EEPROM** : الذاكره المسئوله عن تسجيل بعض المتغيرات بصوره دائمه داخل المتحكمه و تظل محتفظه بقيمتها حتى بعد فصل الكهرباء و يمكننا ان نعتبرها مثل السواقه **Hard Disk** فى الكمبيوتر الشخصى.

## مداخل و مخارج التحكم (I/O) Input & Output Pins

**يمكن** تخصيص الخطوط الرقمية الاربعة عشر (Digital Pins 14) كمداخل أو مخارج وذلك باستخدام الاوامر البرمجيّه كما سنرى في الفصل الثاني و تعمل هذه الخطوط على جهد اقصاه 5 فولت وكل خط يمكن أن يؤمن سحب للتيار بحدود ال 40 ميلي أمبير، وهناك 6 خطوط دخل تماثلية Analog ومعنونة من A0 الى A5، بشكل افتراضي تستطيع هذه المداخل قياس جهد من صفر حتى 5 فولت.



**ملاحظات شخصية :**

هذه الصفحة مخصصة لكتابه ملاحظاتك الشخصيه عن الفصل الاول :

اكتب ملاحظتك  
هنا

## من أقوال العظماء :

الْعِلْمُ خَزَائِنٌ وَمِفْتَاحُهُ السُّؤَالُ ، فَسَلُّوا يَرْحَمَكُمُ  
 اللَّهُ ، فَإِنَّهُ يُؤَجِّرُ فِيهِ أَرْبَعَةٌ : السَّائِلُ وَالْمُسْتَمِعُ ،  
 وَالْمُعَلِّمُ ، وَالْمُجَابُّ لَهُمْ

رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ

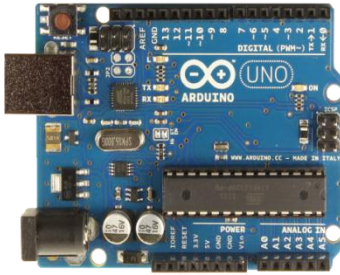
# الفصل الثاني: التجهيزات

اولا: المكونات الماديه Hardware & Tools



## كيف تبدأ استخدام اردوينو و ماذا ستحتاج

سوف نستخدم بورده من نوع Arduino UNO و ستحتاج للمكونات التاليه كبدايه:  
(ملحوظه: في بعض الامثله ستحتاج اضافات لهذه المكونات)



### Arduino Uno V.3

هي البورده المسئوله عن استقبال الأوامر البرمجيه من الحاسب الآلي و كتابتها في ذاكره المتحكمه الدقيقه ATmega328 الموضوعه بداخلها



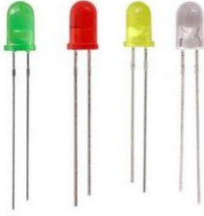
### USB printer cable (1 Meter)

كابل تابعه لتوصيل البورده بالحاسب الآلي، يمكنك شراء اي نوع و يفضل ان يكون قصير قدر الامكان حتى لا يسبب ازعاج اثناء العمل، طول " واحد متر " سيكون اكثر من كافي



### لوحة التجارب (Breadboard)

بورده خاصه تستخدم لتركيب و فك المكونات الالكترونييه عليها بسهوله دون الحاجه الى لحام، يمكنك شراء اي حجم و يفضل Half Sized Breadboard



### دايود ضوئي (الوان مختلفة) Colored Leds

بعض الليدات و هي اشبه بالمصابيح الصغيره و تعمل على تحويل التيار الكهربى الى ضوء، يفضل ان تشتري 10 ليدات بألوان و احجام مختلفه و تتوفره بألوان مثل: **الاحمر، الأخضر، البرتقالى، الأصفر، الابيض و الازرق**



### مقاومات Resistors

سنحتاج الى مقاومات للتحكم فى التيار الكهربى على مداخل و مخارج المتحكمه الدقيقه و سنحتاج المقاومات التاليه:

**Resistor 560 Ohm (x5)**

**Resistor 10 Killo-Ohm (x5)**



### AVO Meter (multi-meter device)

**(اختياري)** جهاز قياس كهربى متعدد الاستخدامات و يستطيع قياس المقاومات و فرق الجهد و شدة التيار الكهربى

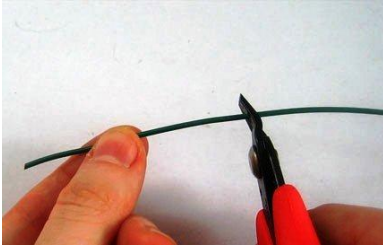


### بعض الاسلاك للتوصيلات بين المكونات Wires

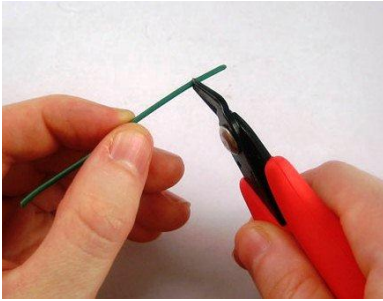
ستستخدم فى توصيل المكونات ببعضها على لوحه التجارب و ستتعلم كيف نقوم بتجهيز السلك بنفسك كما سنرى فى الصفحه التاليه.



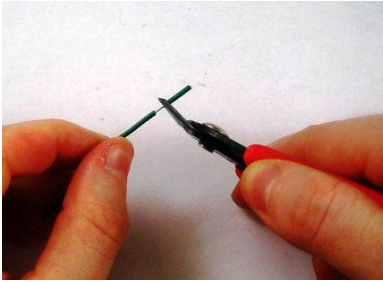
## كيف تُجهز أسلاك التوصيل



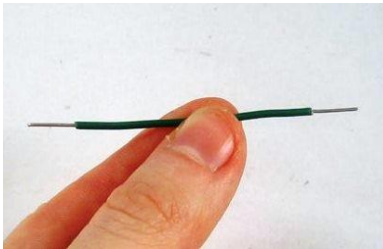
يمكنك الحصول على وصلات عاليه الجوده عن طريق استخدام قطعه من كابل Cat5 المستخدم في وصلات شبكات الحاسب الآلي و تستطيع شراءه من محلات مستلزمات الكمبيوتر و الشبكات .



قم باستخدام قشاره الاسلاك و قص واحد سنتي متر من الجهتين للسلك المراد تجهيزه كما في الصورة الاولى، ثم كرر نفس العمليه مره اخرى ولكن باطوال مختلفه حتى تحصل على مجموعه اسلاك جاهزه للاستخدام



اذا كنت لا تفضل ان تقوم بهذه العمليه يمكنك شراء مجموعه اسلاك مقشره جاهزه للاستخدام مع ال breadboard و ستجدها في محلات المكونات الالكترونيه



**اذا كنت** في مصر و ترغب في شراء تلك المكونات هناك 3 مواقع مصريه على الانترنت تستطيع طلب تلك المكونات منها و سيتم شحنها لباب منزلك في غضون يومين و سيتم اضافته تكلفه الشحن على سعر المكونات.



<http://www.fut-electronics.com>

<http://egyrobots.com>

<http://www.ram-e-shop.com>

اما اذا كنت تفضل زياره بعض المحال بنفسك



يمكنك الذهاب الى محل رام الكترونيكس RAM Electronics الموجود في "باب اللوق" و محل فيوتشر الكترونيكس Future Electronics الموجود امام **كلييه هندسه عين شمس** في ميدان **عبده باشا** في العباسيه .

و اذا كنت في احدى الدول العربيه او اى دوله اخرى لا يتوفر بها محلات تباع هذه المكونات (خاصه **اردوينو**) يمكنك زياره الموقع التالي و ستجد عليه الكثير من المنتجات الخاصه ب**اردوينو** مع الشحن المجاني

<http://www.buyincoins.com>



كما يمكنك زياره قائمه الموزعين العالميين ل**اردوينو** لتبحث عن موزع محلي قريب منك عبر الرابط التالي:

<http://arduino.cc/hu/Main/Buy>



## محتويات علبة Arduino Uno

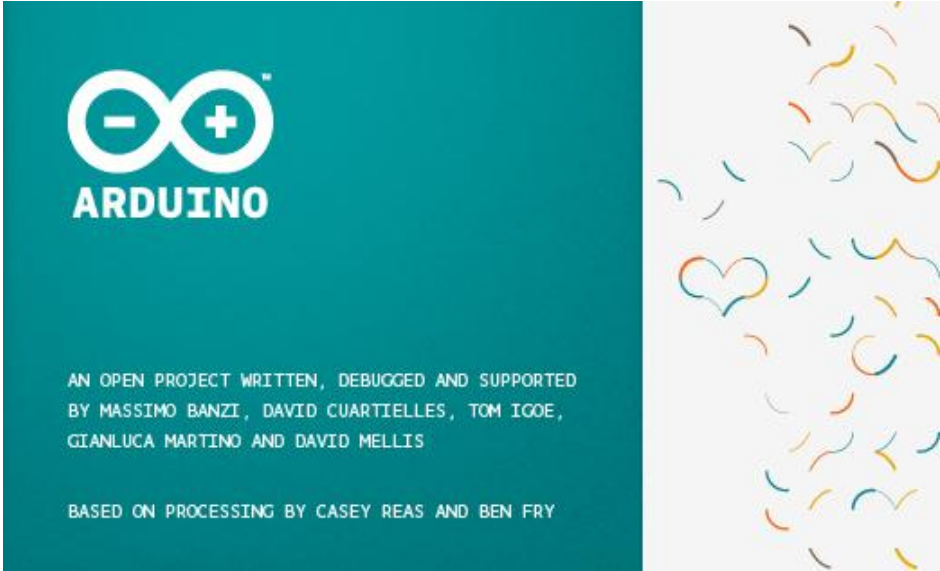
ملحوظه قد تختلف مكونات العلبة من بلد الى اخرى على حسب المورد ورقم  
اصداره البورده كما قد تحتوى على ملصقات اضافيه لاردوينو

## ثانيا: تجهيز البرمجيات



Software's

## بيئة تطوير اردوينو (IDE) Arduino



**تعتبر** بيئة التطوير **Arduino IDE** الاداه المستخدمه في كتابه الاكواد البرمجيّه بلغه **Arduino C** و تحويلها بعد ذلك الى صيغه تنفيذيه يمكن وضعها على المتحكمه الدقيقه الموجوده على البورده .

**تتميز** بيئة تطوير اردوينو بالبساطه و السهوله في التعامل فهي تكاد تخلو من اي تعقيدات في المظهر العام و تحتوي فقط على ما يحتاجه المبرمج ليبدأ تطوير برامج بلغه **اردوينو سي Arduino C** كما انها تستخدم في نفس الوقت لرفع البرنامج مباشره الى المتحكمه الدقيقه و بذلك لن تحتاج الى برنامج اخر مخصص لرفع الصيغه التنفيذيه للبورده.



**سيتم الشرح** على بيئه تطوير اردوينو النسخه الخاصه **بويندوز** بأعتبراره اكثر انظمه التشغيل شيوعا و شهره بين المستخدمين بالرغم من انى انصح بشده بأستخدام لينكس لانه مجاني و سريع و آمن بالاضافه الى انه **مفتوح المصدر Open Source** ،  
و فى جميع الاحوال يمكنك العمل على اى نظام تشغيل تحب.

**من يريد** التعرف اكثر على نظام تشغيل لينكس هناك كتاب عربى مجاني [ **اوبنتو ببساطه** ] من تأليف:  
**أحمد محمد أبوزيد** و يشرح استخدام نظام تشغيل **لينكس اوبنتو Ubuntu** بأسلوب سهل و ممتع و يمكنك تحميله مجانا من على الموقع الرسمى للكتاب :

<http://www.simplyubuntu.com/>



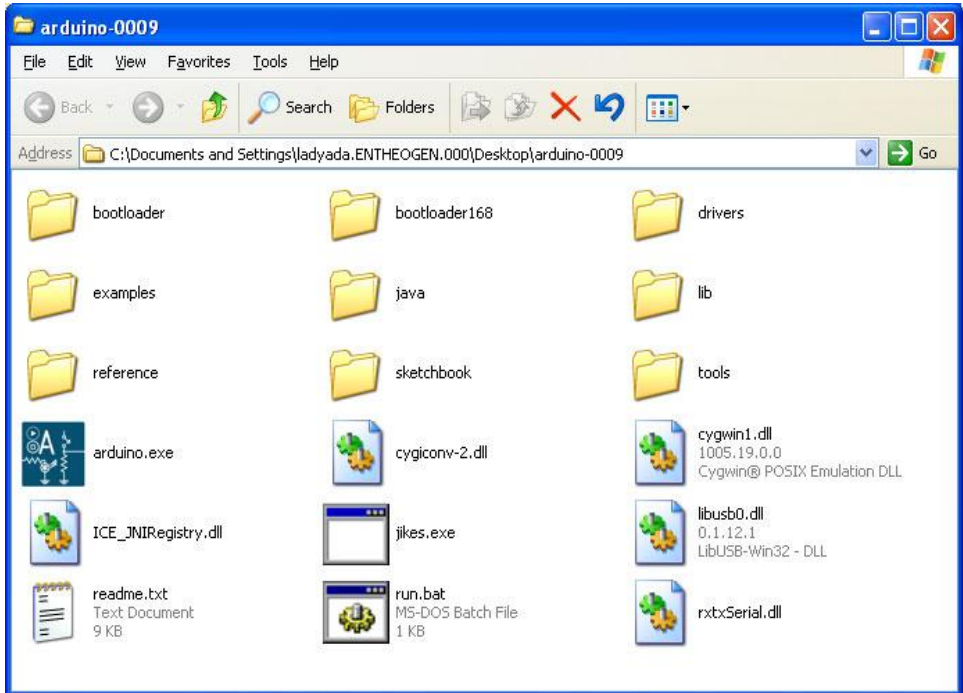
## تنزيل بيئه التطوير Arduino IDE

**تتوفر** بيئه التطوير البرمجيّه لجميع انظمه التشغيل المختلفه Windows, Linux, MAC ويمكنك تحميلها من الموقع الرسمي لمطوري اردوينو على الرابط التالي:

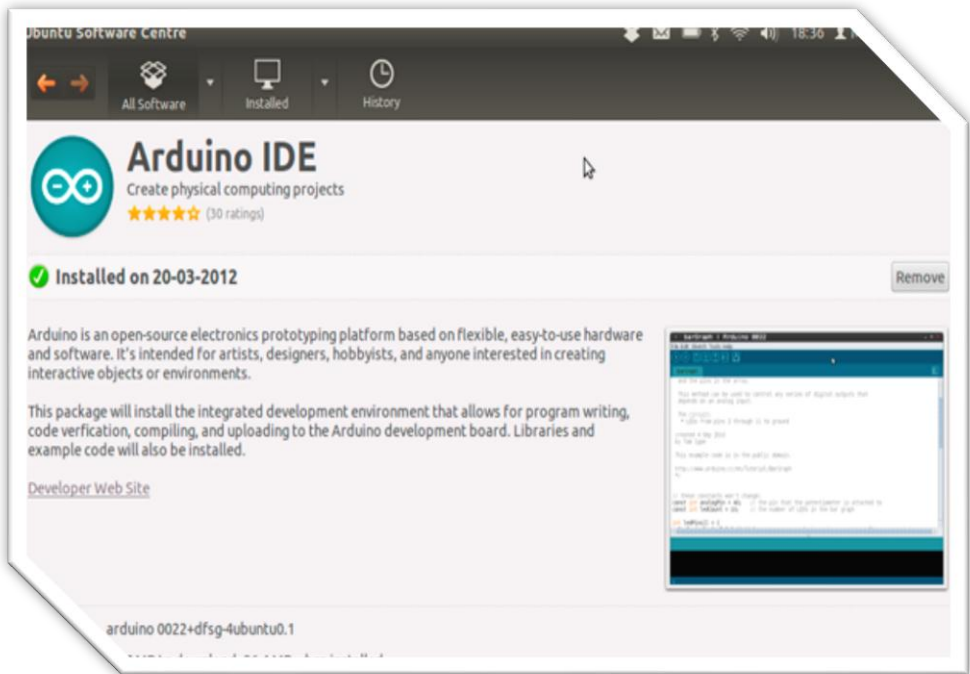
<http://arduino.cc/hu/Main/Software>



**بالنسبه** لمستخدمي ويندوز او ماك قم بتنزيل الاصداره الخاصه بهما من الرابط اعلاه و ستجد ملف مضغوط، قم بفك الضغط بأستخدام احدى البرامج مثل 7zip ثم قم بتشغيل ملف `arduino.exe`



**لمستخدمي** نظام لينكس اوبنتو Ubuntu يمكنك استخدام مركز البرمجيات Software center فقط ابحث عن Arduino IDE ثم اضغط زر Install و بالنسبة لمستخدمي انظمة لينكس الاخرى مثل فيدورا fedora و نسخة اوبن سوزي Open SUSE ستجدو بيئه التطوير موجوده فى مركز البرمجيات الخاصه بالتوزيعه ان لم تستطع ايجاده يمكنك تحميل الملف الخاص بلينكس من على الموقع الرسمى.



ملحوظه: تعمل البيئه التطويره لاردوينو بصوره اسرع على لينكس اكثر من ويندوز لذلك انصح اذا كان جهازك المستخدم فى عمليه البرمجه ذو امكانيات متوسطه او ضعيفه من ناحيه Hardware يفضل ان تستخدم احدى نسخ لينكس



**بعد الانتهاء** من تنصيب Arduino IDE قم  
بالخطوات التاليه لتوصيل البورده بالكمبيوتر:



1

• قم بتوصيل بورده اردوينو بالكابل

• ادخل الطرف الاخر فى فتحة USB

فى الحاسب الآلى و انتظر قليلا حتى

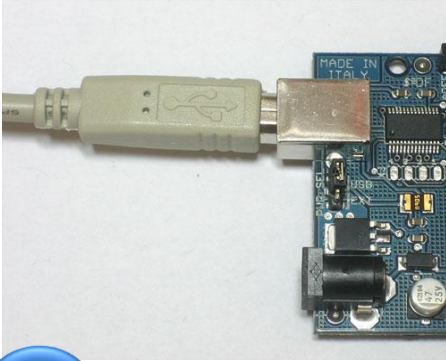
تظهر لك رساله تفيد بأن جهاز الحاسب

تعرف على قطعه عتاد جديده Found

Arduino Uno Board بعد ظهور

تلك الرساله يأتى الوقت لبدأ العمل

على البيئه التطويريه



2



3



4

## التعرف على الواجهه الرسوميه لبيئه التطوير



تكون بيئه التطوير البرمجيّه من واجهه بسيطه و تنقسم الى اربعه اجزاء رئيسيه :

**الاول:** شريط القوائم

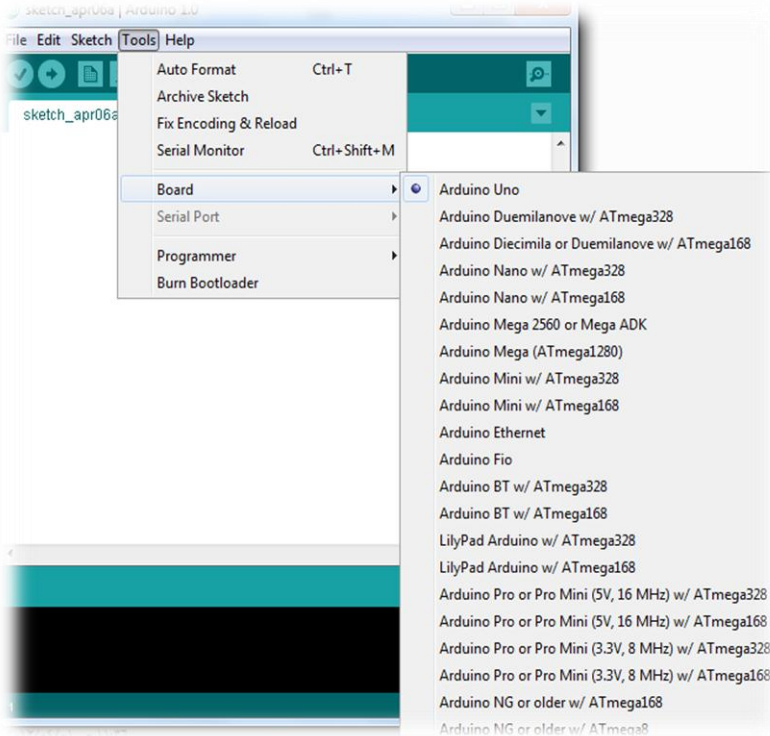
**الثاني:** شريط الاوامر السريعه

**الثالث:** منطقه كتابه الاكواد البرمجيّه

**الرابع:** الجزء الخاص بعرض التنبيهات و الاخطاء البرمجيّه في بيئه التطوير

## تجهيز بيئه التطوير

**اول** خطوه لتجهيز البيئه التطويره هي اختيار البورده التي سنتعامل معها و يتم ذلك عن طريق الذهاب الى قائمه Tools و اختيار Board ثم نختار نوع البورده المتوفره لدينا و في هذه الحاله سنختار Arduino Uno كما في الصوره التاليه



**ثاني** خطوه تتم فى حاله ظهور هذه الرساله بعد توصيل اردوينو و التى تفيد بأن الجهاز لم يجد تعريفات لكى يستخدم اردوينو (تظهر غالبا مع مستخدمى ويندوز Vista او Windows 7) و يجب ان نقوم بتنصيب التعريفات كالتالى:



Device driver software was not successfully installed

Click here for details.

EN

اولا: افتح مدير الاجهزه Device Manager كما فى الصور التاليه



Compu

Open



Manage

Map network drive...

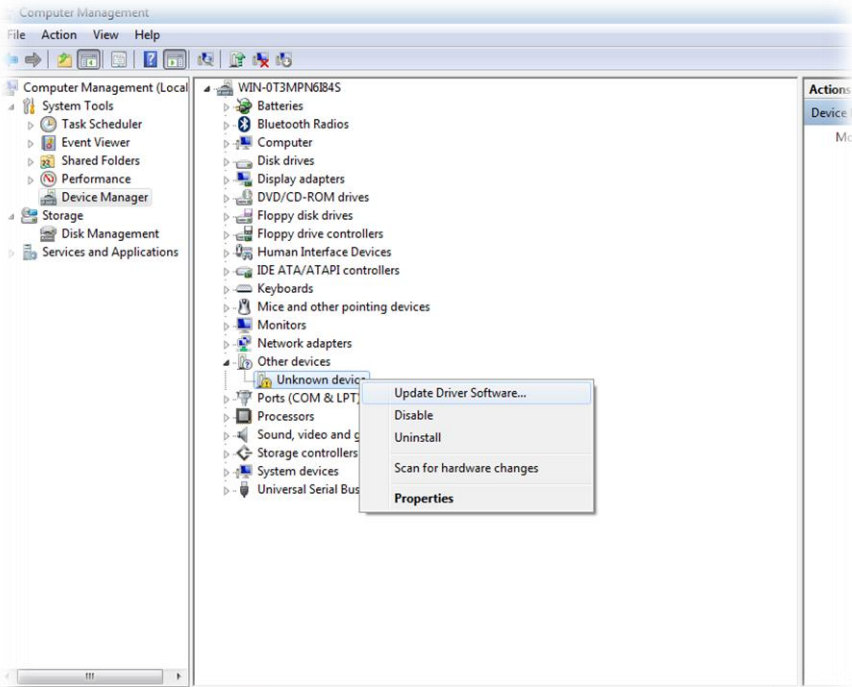
Disconnect network drive...

Create shortcut

Delete

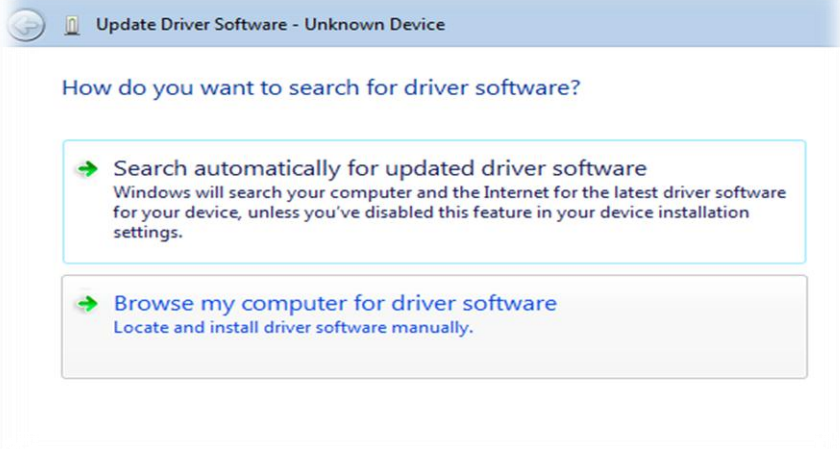
Rename

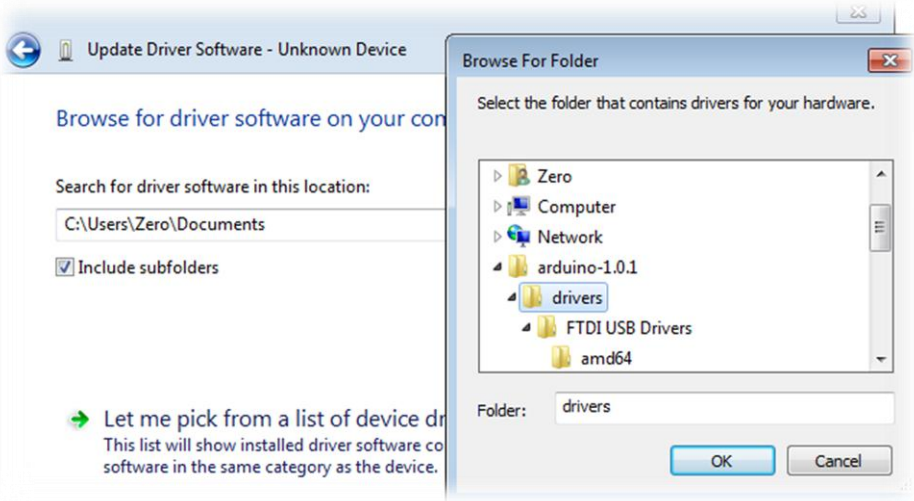
Properties



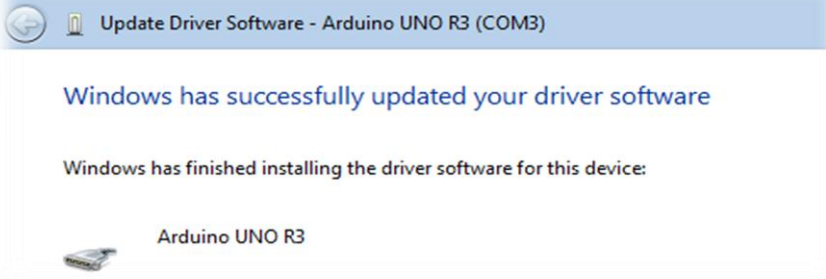
Launches the Update Driver Software Wizard for the selected device.

اختر "تصفح الجهاز لاختيار التعريفات" ثم اختر الفولدر الذى يوجد به برنامج اردوينو



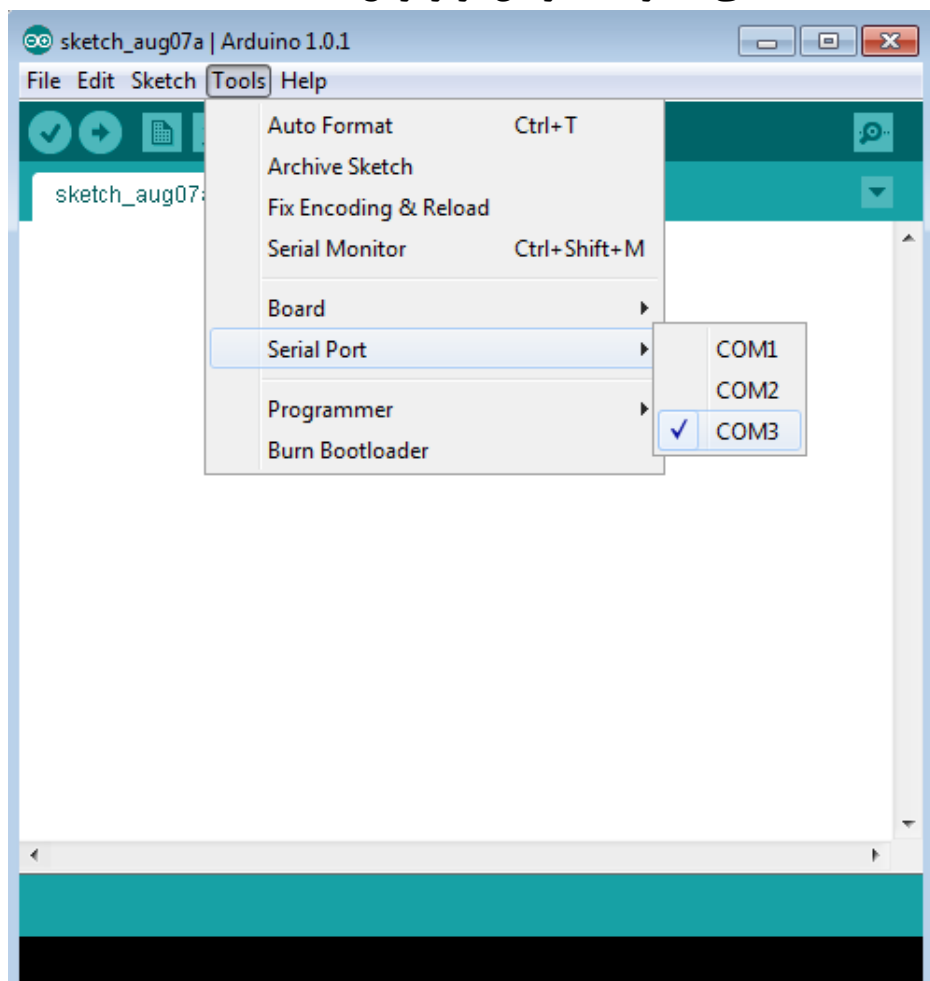


بعدها سوف تظهر رساله تسألك عن "إذا ما كنت ترغب في تنصيب هذه التعريفات ام لا؟  
" اضغط Ok وانتظر قليلا حتى تنتهي عملية تعريف اردوينو بنجاح كما في الصورة التالية

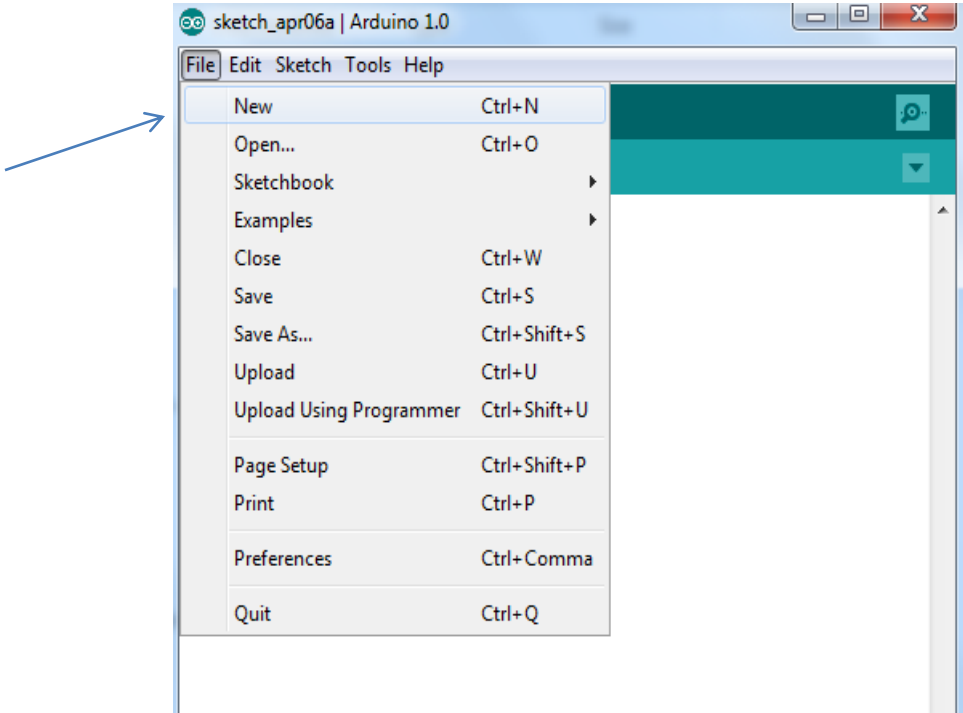


بعد الانتهاء من تنصيب التعريف لاحظ رقم المنفذ المستخدم في اردوينو من مدير  
الاجهزه كما في الصورة التاليه و التي تظهر رقم المنفذ COM3 ، تذكر ذلك الرقم فسوف  
نحتاجه لاحقاً ( مع ملاحظه ان ذلك الرقم قد يختلف عندك) .

الخطوة الأخيرة هي اختيار منفذ التوصيل بأردوينو من قائمه Tools – Serial Port



**بعد ذلك** تستطيع البدء فى كتابه اول برنامج لك على الاردوينو وذلك عن طريق اختيار New من قائمه File الموجوده بشريط القوائم كما فى الصوره التاليه

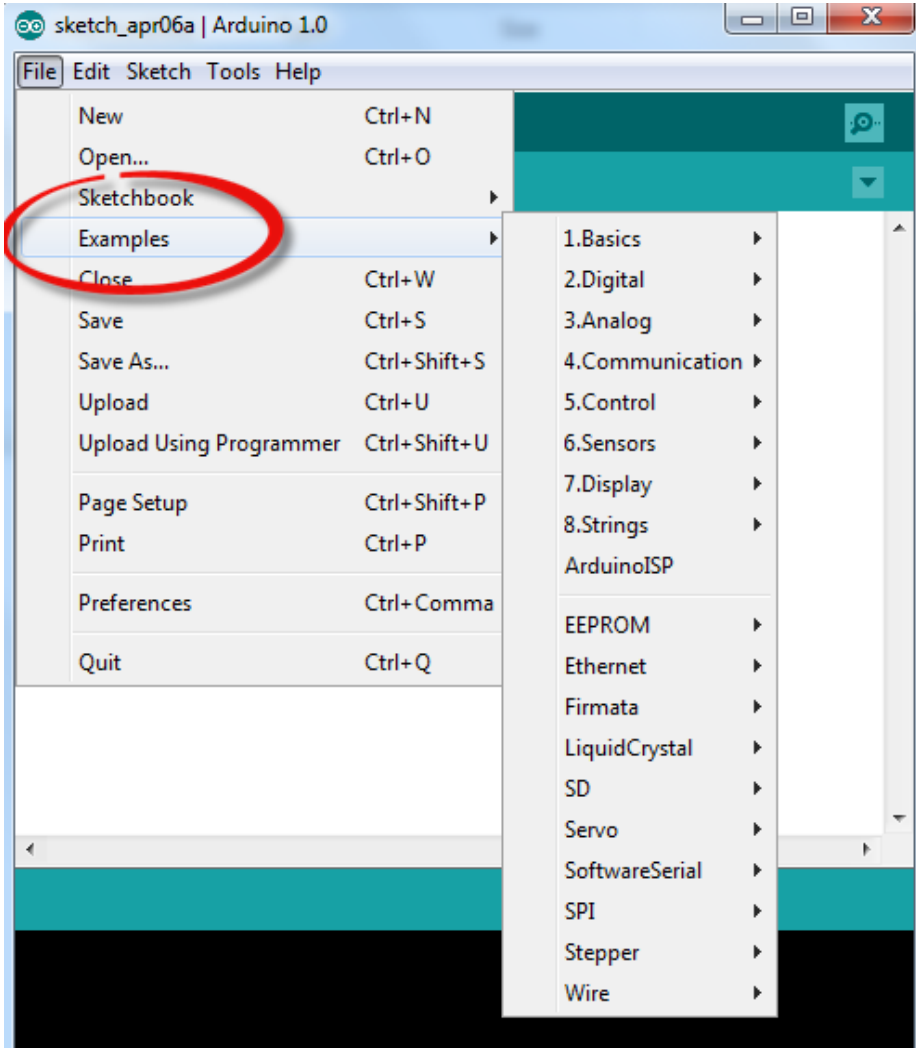


**او يمكنك** اختيار New مباشره من شريط الاوامر السريعه





**ايضا** توفر لك البيئه التطويريه مجموعه من الامثله البرمجيّه الجاهزه و التي تستطيع الوصول اليها من خلال قائمه Examples الموجوده في قائمه File الرئيسيّه كما في الصوره التاليه



**ملاحظات شخصية :**

هذه الصفحة مخصصة لكتابه ملاحظاتك الشخصيه عن الفصل الثاني :

اكتب ملاحظتك  
هنا

## من أقوال الحكماء:

كُنْ شَدِيدَ التَّسَامُحِ مَعَ مَنْ خَالَفَكَ الرَّأْيَ،  
فَإِنْ لَمْ يَكُنْ رَأْيُهُ كُلَّ الصَّوَابِ فَلَا تَكُنْ أَنْتَ  
كُلَّ الْخَطَا بِتَشْبُثِكَ بِرَأْيِكَ

فرانسوا ماري أرويه المعروف باسم فولتير

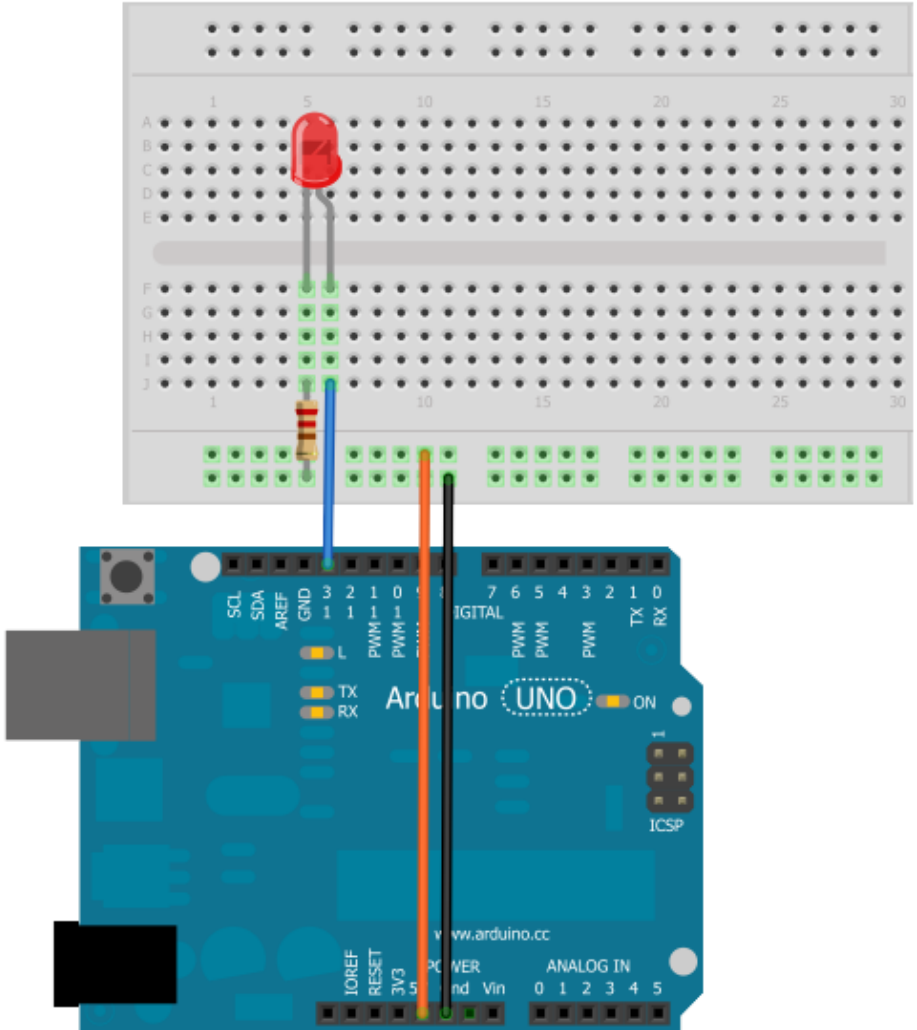
# الفصل الثالث

لنبدأ العمل مع اردوينو



Let's get started

## المثال الأول: تشغيل دايود ضوئي



**مكونات المثال (1):**

✓ بورده اردوينو Arduino Uno

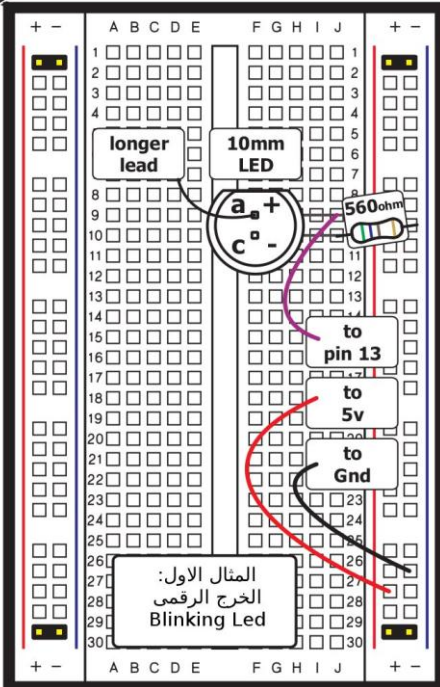
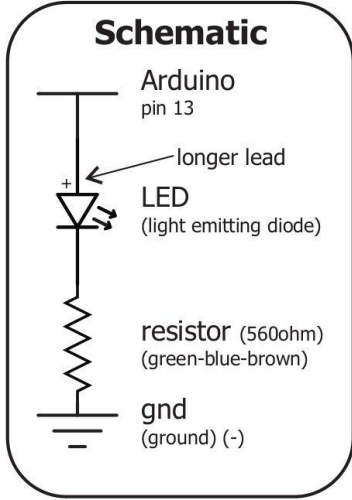
✓ Breadboard

✓ مقاومه كهريبه بقيمه 560 اوم

✓ دايمود ضوئى قطر 10 مللى

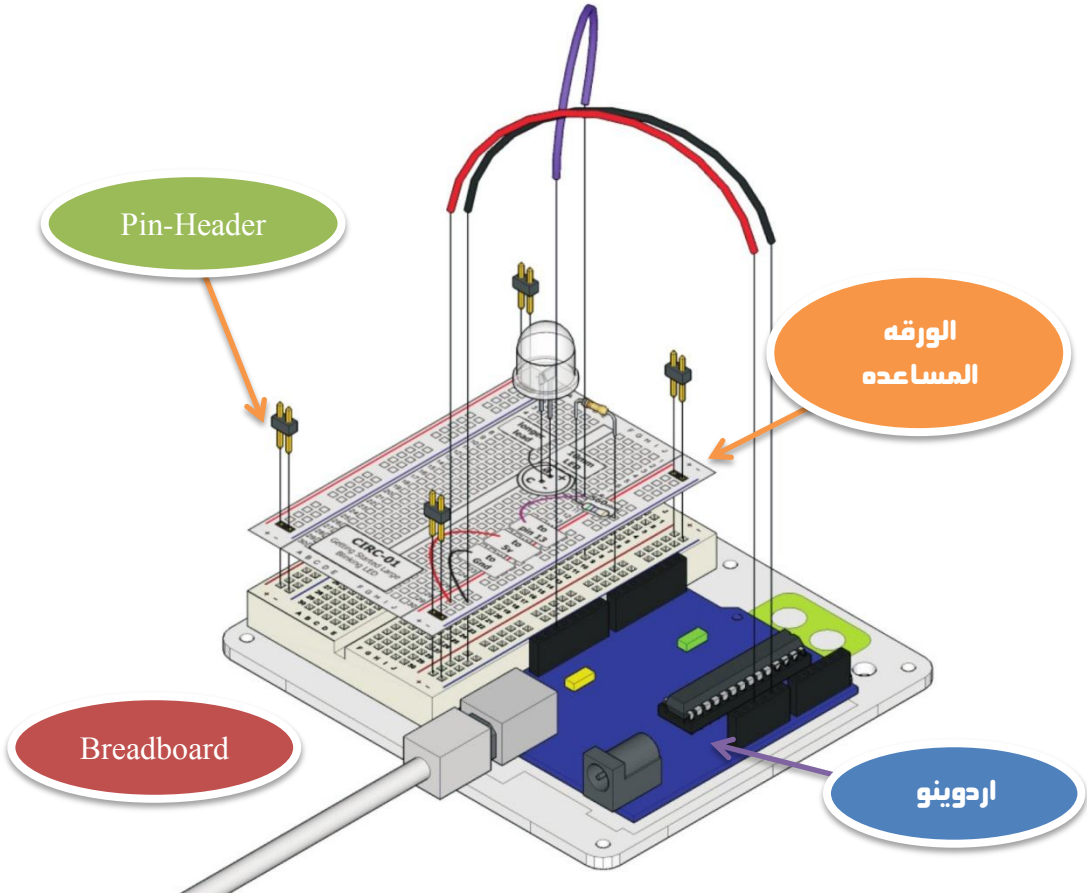
✓ جهاز الكمبيوتر

✓ ورقه المساعده الموجوده فى هذه الصفحه

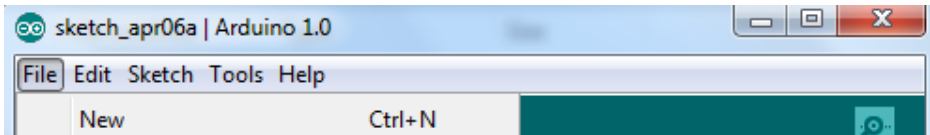


ما هى الورقه المساعده: هى ورقه يتم طباعتها لترشدك الى كيفية وضع المكونات الالكترونيه على لوحه التجارب Breadboard

**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 و ثبتها باستخدام 4 وصلات pin-header ثم ابدأ فى توصيل المكونات الالكترونيه فوقها كم فى الصوره



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



```
// Example 01 : Blinking LED
```

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

```
void setup ( )
```

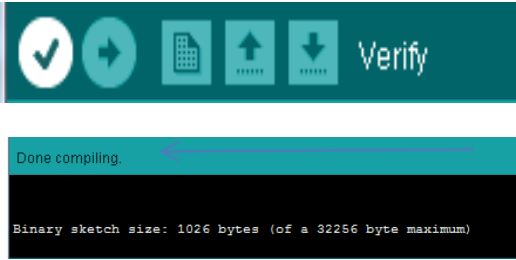
```
{  
  pinMode(LED, OUTPUT);  
}
```

```
void loop()
```

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



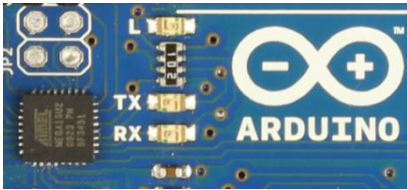
**الخطوة الثالثة:** بعد ان تنتهي من كتابه الكود البرمجي اضغط على زر تأكيد Verify فى الشريط الاوامر السريعه فى اعلى ال IDE ثم انتظر حتى تظهر رساله فى الاسفل تخبرك بأن البرنامج قد انتهى من تجهيز الكود ليتم رفعه على **بورده الاردوينو**



**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

LED

## سؤال ماذا كتبنا هذا الثابت؟؟

نقوم باستخدام الثوابت حتى تسهل علينا تسميه 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 بأطفاء 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 الثاني 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 اي متغيرات او ثوابت في البرنامج الذي سوف نقوم بكتابته

**المرحلة الثانيه:** تتميز المتحكمات الدقيقة ان جميع ال **Digital Pins** يمكن ان تستخدم اما كمخرج او مدخل وفي هذا المرحله نقوم بضبط اعدادات المخارج و المداخر 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 المداخر 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 المنقوطة (;) 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 بأطفاء 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 و عندما نريد تشغيل الليد نستخدم HIGH لتطبيق جهد 5 فولت و عند 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 منطفى 1000 ملى ثانيه اخرى = واحد ثانيه .

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

**يحتاج** المبرمجين عند كتابه اى برنامج مهما كانت اللغه البرمجيّه الى نظام للتعليقات comments و التعليقات هي جمل تساعد المبرمج على فهم و تنظيم الكود البرمجي و لا تحتسب في 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
```

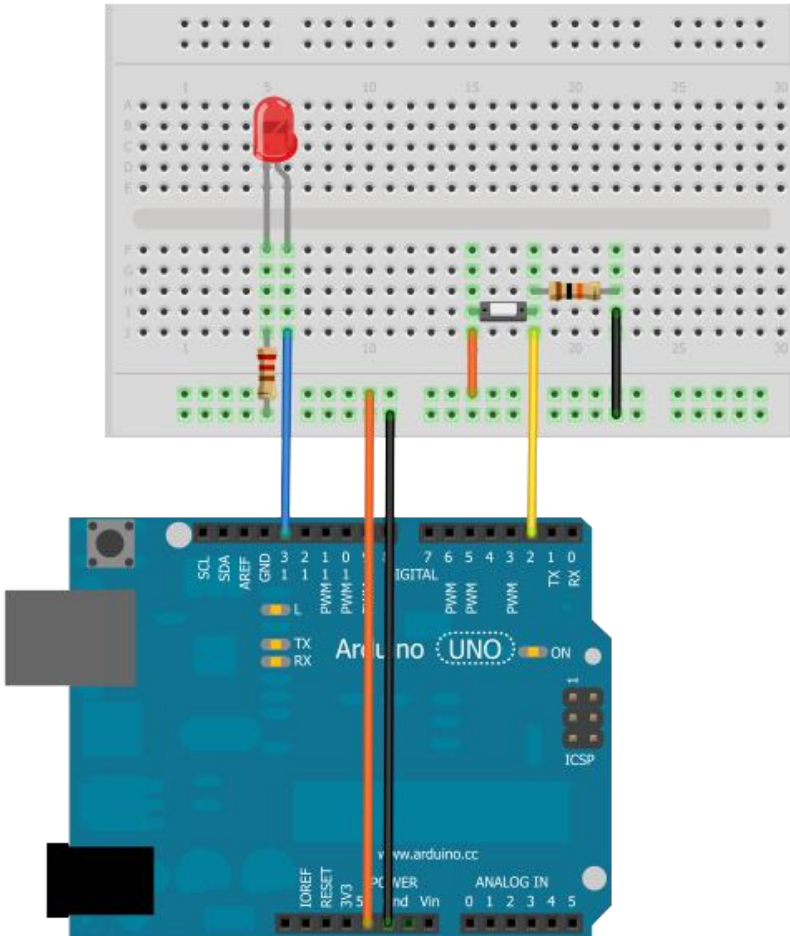
هناك اسلوب اخر لكتابه التعليقات الطويله و هي باستخدام الشرطه المائله و النجمه فى اول الكلام ثم كتابه نجمه و شرطه مائله فى نهايه الكلام.....مثال:

```
/* 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 السويتشات 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 لوحة 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 الثاني 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 تدرّب 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 الثانيه: 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 الثالثه: كتابه 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 يحتوي على اخطاء برمجيّه 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 الذي سيرمز الى Pin 13 و

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

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

على المدخل الرقمي السابع و قمنا

بتعريف المتغير val و الذي سوف

نستخدمه لاحقا لتخزين حاله الدخل

على المدخل الرقمي الثاني pin 2

**المرحلة الثانية:** في هذه المرحلة جعلنا المتحكمه تعامل pin 13 و التي يرمز لها

باسم ledPin على أنها مخرج و

سنستخدمه لتشغيل الليد الموصل به،

ثم قمنا بجعل المتحكمه تعامل

المخرج pin 2 على انه مدخل

رقمي وظيفته استقبال الاشارات

الرقميه

HIGH or LOW

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

## المرحلة الثالثة:

```
val = digitalRead(buttunPin);
```

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

ويتم ذلك باستخدام الأمر ( `digitalRead` ) وإذا كان المستخدم ضغط على السويتش

فأن 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

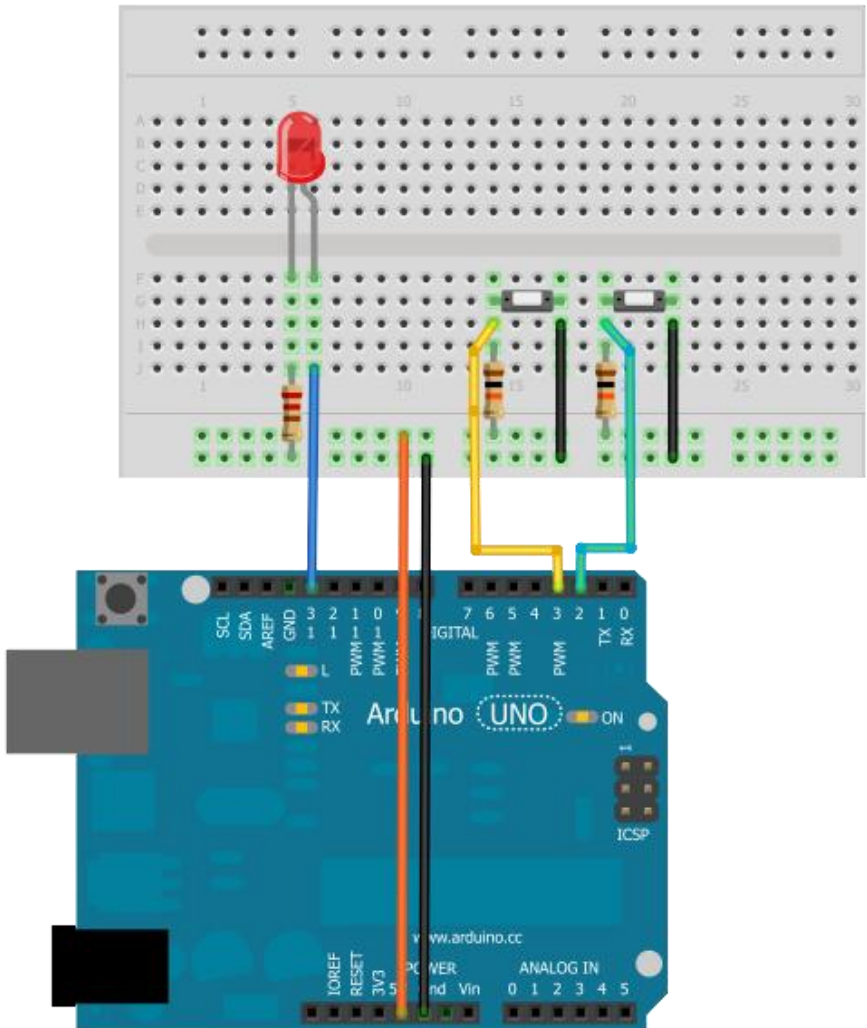
( `if...else` = إذا .. مادون ذلك )، يستخدم ذلك الامر لعمل مقارنات و جعل

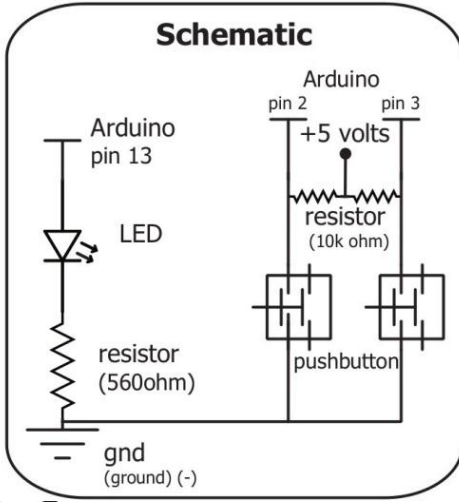
المتحكممة تفعل **شيء ما** إذا حدث **شيء آخر** محدد وفي هذا المثال استخدمنا الامر `if`

كالتالي :

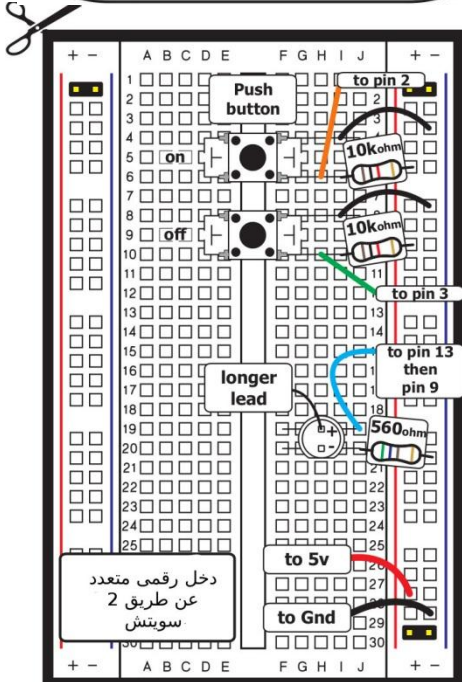
- ستقيس المتحكممة قيمة الجهد على المدخل السابع و تخزن القيمة في المتغير `val`
- إذا كانت القيمة تساوى HIGH و التي تعنى 5 فولت ستقوم المتحكممة بتشغيل الليد لمدة ثانيه و اطفائه لمدة ثانيه وهكذا باستمرار.
- ما لم تكن القيمة تساوى HIGH و الجهد المطبق على المدخل يساوى صفر ستقوم المتحكممة بتطبيق جهد يساوى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

## المثال الثالث: تطوير المثال السابق يعمل بـ 2 سويتش من نوع Push Button واحد من اجل التشغيل ON و الثانى من اجل الاغلاق OFF



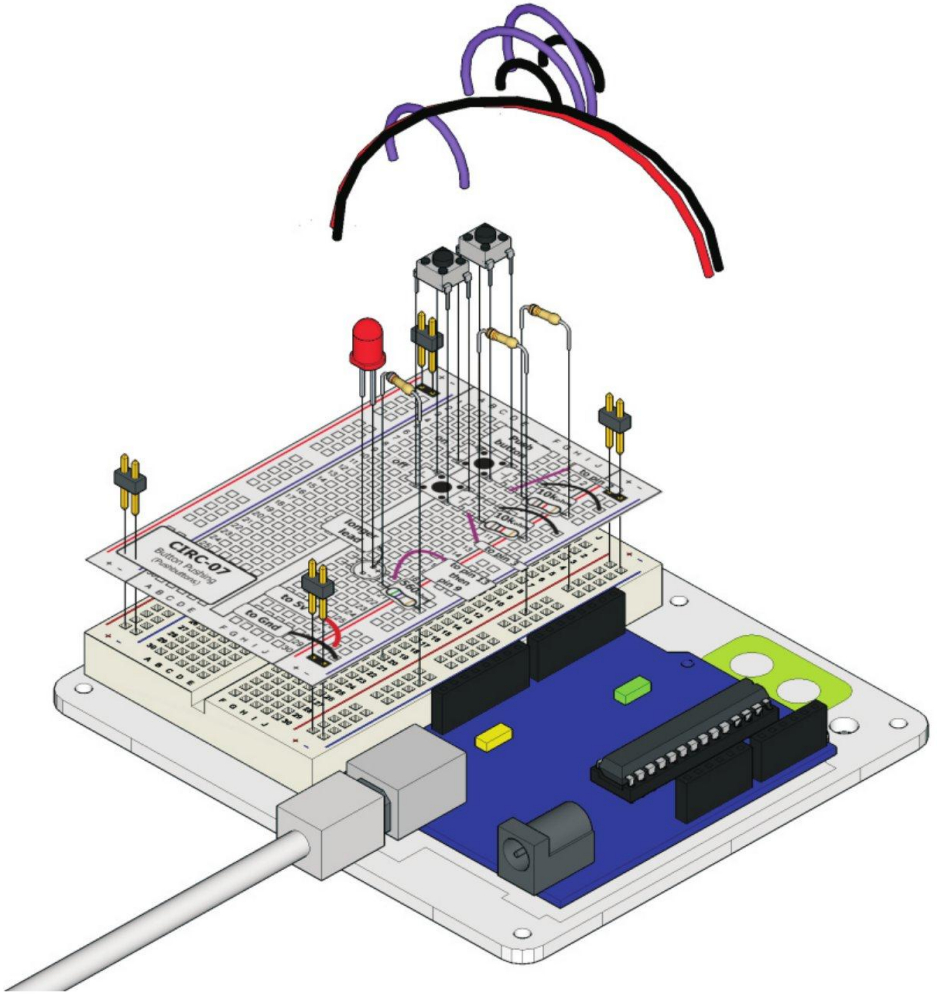
**مكونات المثال (3):**

- Arduino uno
- Breadboard
- Led 5 mm
- عدد 2 مقاومه قيمه 10 كيلو اوم
- مقاومه 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 الأولى:** 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 السريعه 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  
للدائره

```
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 ترى 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 كره 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 المتحكممة الدقيقة بتشغيل الدايدود الضوئي و الاحتمال الثاني هو أن يضغط المستخدم السويتش 2 فتقوم المتحكممة بإطفاء الدايدود 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 في الامثله السابقه

<pre>void setup ( ) {.....}</pre>	<p>الداله المسؤله عن ضبط اعدادات المتحكمه و تحديد المداخل و المخرجات</p>
<pre>void loop ( ) {.....}</pre>	<p>يتم كتابه البرنامج الرئيسى داخل هذه الداله و يقوم البرنامج باعاده نفسه مره تلو الاخرى كلما اتم تنفيذ جميع الخطوات</p>
<pre>int name = value;</pre>	<p>امر يستخدم لتعريف متغير بقيمه معينه</p>
<pre>const int name = value;</pre> <p><b>example:</b> <code>const int led = 13;</code></p>	<p>امر يستخدم لتعريف ثابت غير قابل للتغير يشبه الامر السابق مع اضافته const</p>
<pre>pinMode (pin number, State);</pre> <p><b>example 1:</b> <code>pinMode(11,INPUT);</code>  <b>example 2:</b> <code>pinMode(led,OUTPUT);</code></p>	<p>الامر المسؤل عن تحديد وظيفه كل pin و ضبطها للعمل كمدخل ام كمخرج INPUT or OUTPUT</p>
<pre>digitalWrite (pin number, State);</pre> <p><b>example:</b> <code>digitalWrite(13,HIGH);</code></p>	<p>الامر المسؤل عن اخراج جهد على احدى ال pin بصوره رقميه HIGH (1) or LOW(0)</p>
<pre>digitalRead (pin number);</pre> <p><b>example:</b> <code>digitalRead(4);</code></p>	<p>الامر المسؤل عن قراءه الجهد المطبق على احدى ال pin بصوره رقميه حيث يقوم بقياس الجهد (اذا كان 5 فولت حتى 3.8 فولت يعتبر HIGH و اذا كان</p>

	ما بين صفر حتى 1.8 فولت يعتبر (LOW)
<i>delay (time):</i>  <b>example 1:</b> <code>delay(100);</code> <b>example 2:</b> <code>delay(1000);</code>	تأخير بزمان محدد : يستعمل هذا الامر فى تحديد المده الزمنيه للامر الذى يسبقه فى التنفيذ
<b>if (the condition)</b> {....what to do}  <b>else if (another condition)</b> {...what to do}  <b>else (last condition)</b> {....what to do}	(اذا حدث ... او اذا حدث.. افعل....) جمله شرطيه تستخدم لتنفيذ امر ما عندما يتحقق شرط معين و يمكن استخدام الشروط التاليه:  == يساوي != لا يساوي <= أقل من أو يساوي >= أكبر من أو يساوي < أقل من > أكبر من

## اشهر أنواع المتغيرات التي يمكن استخدامها في اردوينو كالتالي:

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

لمزيد من المعلومات عن أنواع المتغيرات التي يمكن استخدامها

داخل برامج اردوينو قم بزياره الصفحة التالية

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

## ملاحظات شخصية :

هذه الصفحة مخصصة لكتابه ملاحظتك الشخصية عن الفصل الثالث :

اكتب ملاحظتك  
هنا

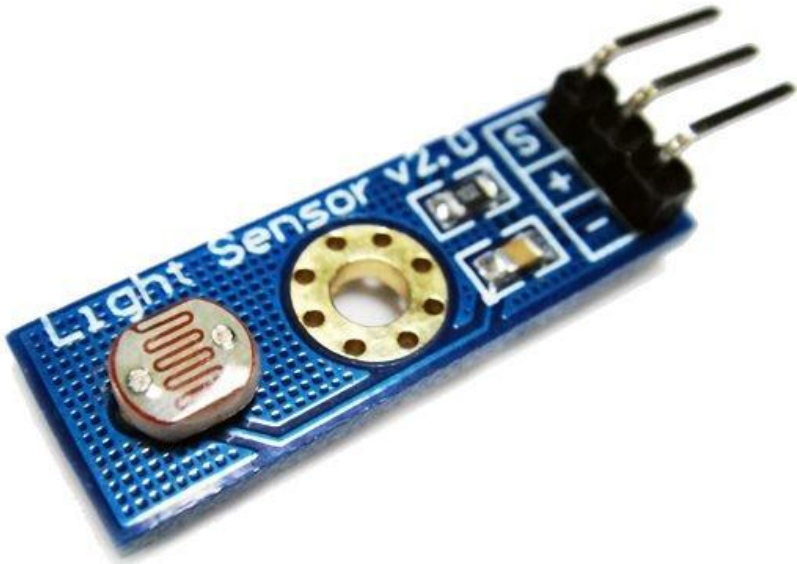
## من أقوال الحكماء:

ذو العَقلِ يَشقى في النِّعيمِ بِعَقْلِهِ  
وَأخو الجَهالةِ في الشَّقَاوَةِ يَنعمُ

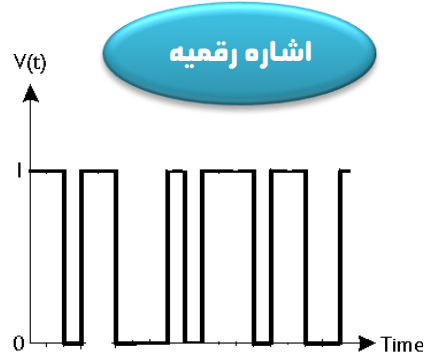
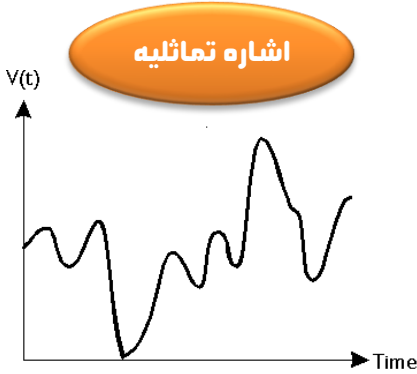
أبو الطيب المتنبي

# 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 وليس له حدود قصوى معينه ، على عكس الدخل الرقمي والذي اما يكون 5 فولت HIGH (1) واما يكون صفر فولت LOW (0)



## ما هي أهميه الاشارات التماثليه Analog Signals؟؟

يعبر الدخل التماثلي عن الاشارات المتغيره مثل فرق الجهد الناتج من مستشعر (حساس sensor) معين نتيجة التغير في عامل معين، على سبيل المثال : حساس للضوء او ما يعرف بأسم المقاومه الضوئيه 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





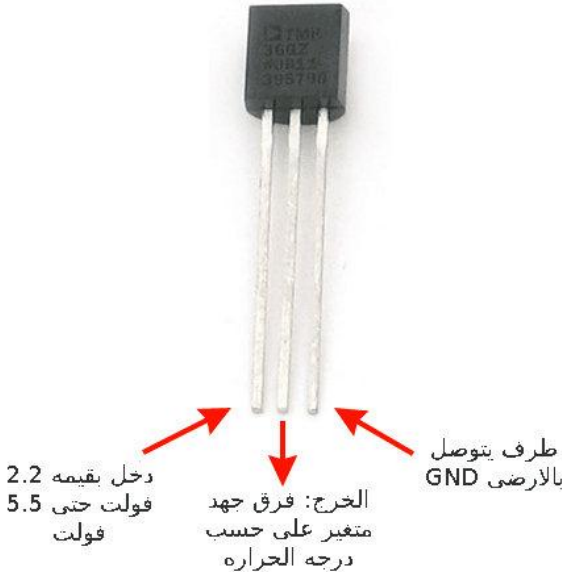


## كيف تولد الحساسات الإشارة التماثلية؟؟

لنأخذ حساس الحرارة كمثال: حساس الحرارة مثل LM35, TMP35, TMP36 ما هو الا ترانزستور ذا حساسية عالية للتغير في الحرارة و من المعروف أن الترانزستور يصنع من ماده السيلكون التي تتغير مدى قابليتها لتوصيل الكهرباء بتغير الحرارة حيث يزداد التيار الكهربى المار فيها بزياده الحرارة و يمكننا إجراء تجربه بسيطة لتتعرف اكثر على طريقه عمل ذلك الحساس.

يحتوى الحساس على ثلاث ارجل وهى كالتالى:

- 1- منفذ الدخل و يتم توصيل جهد ثابت بين 2.2 فولت حتى 5.5 فولت
- 2- منفذ الخرج و هو المنفذ الذى نحصل منه على قراءة الحساس
- 3- منفذ الأرضي و يتم توصيله بأى نقطه ارضى GND





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 ناحية 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 درجه حراره الحساس و من المعلوم أن درجه حراره الجسم تساوى 37 درجه مئوية تقريبا و هي اعلى من درجه حراره جو الغرفة )

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

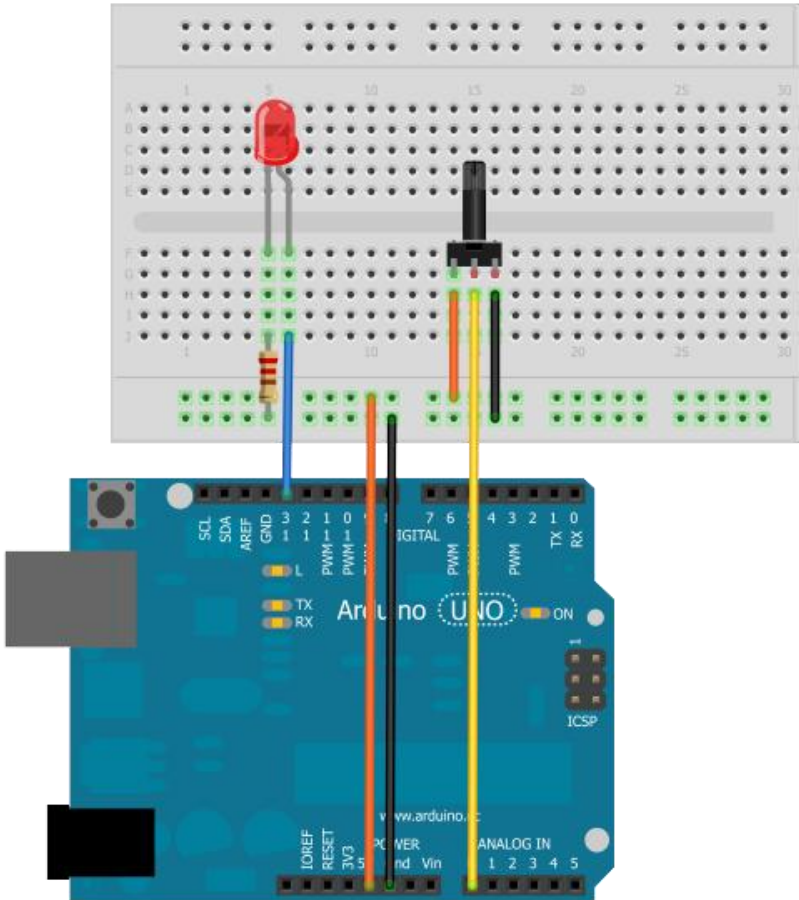


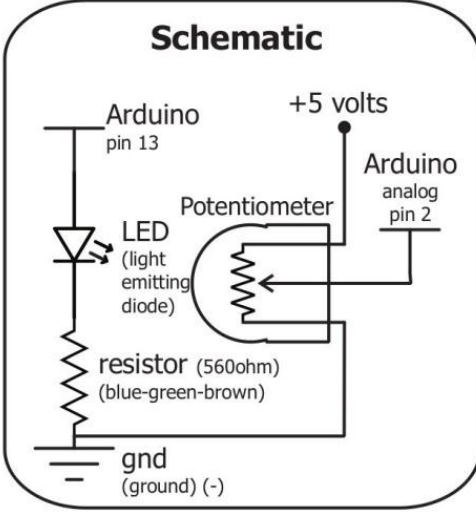
**سنجد أن** قيمه الفولت قد ازدادت بسبب حراره اليد

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 ، و الآن لنبدأ مع امثله استخدام الحساسات مع اردوينو

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

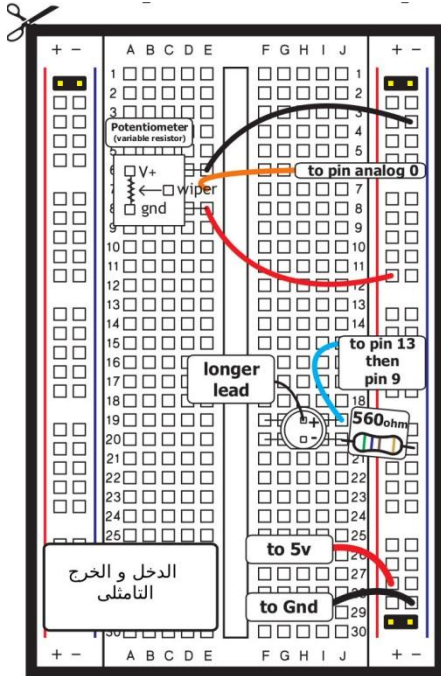
في هذا المثال سنستخدم مقاومه متغيره للحصول على فرق جهد يمكن تغييره (دخلى تماثلى analog input) و سنجعل الدايدو الضوئى يضيئ و ينطفئ بسرعه تعتمد على قيمه الدخلى التماثلى





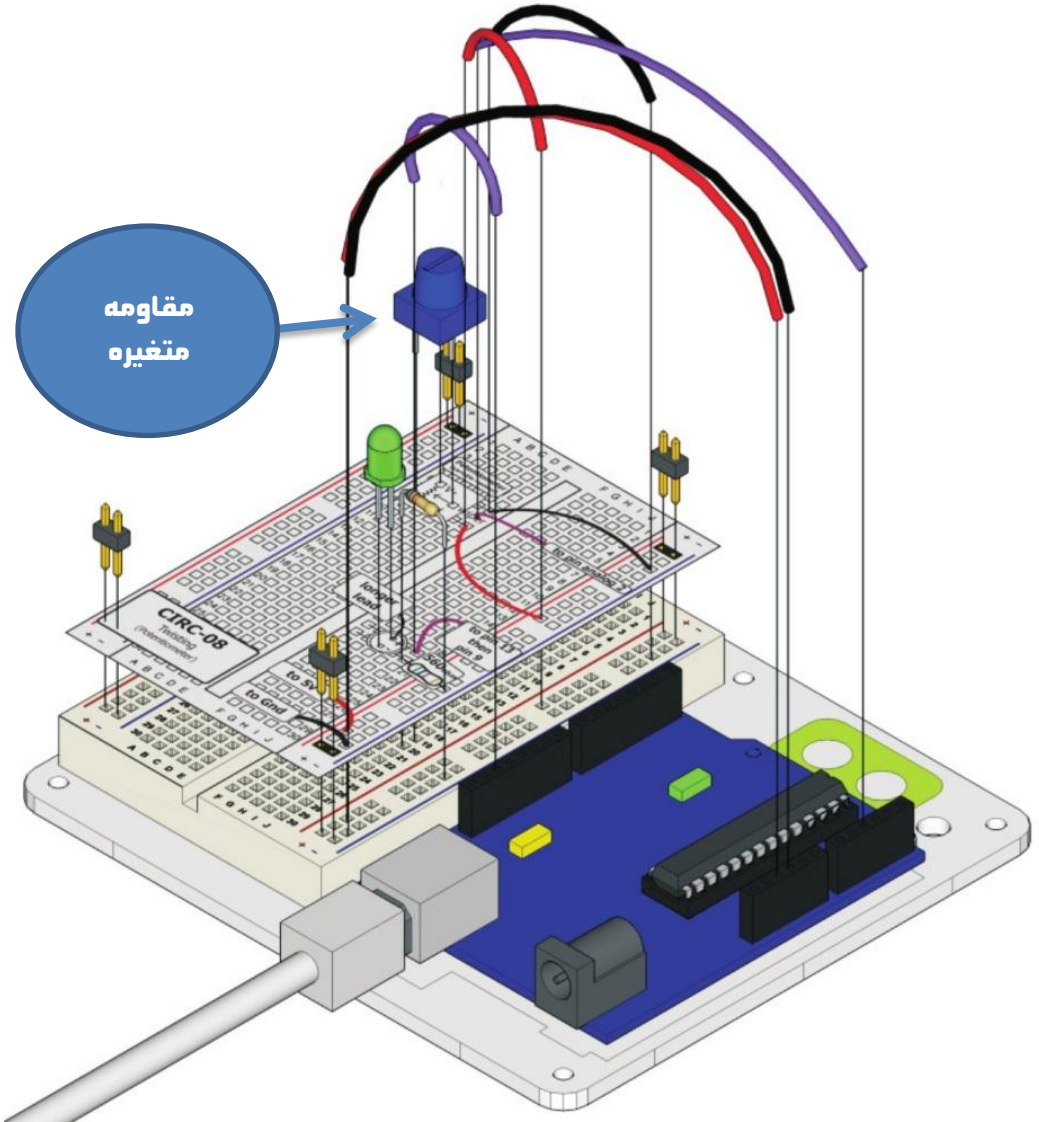
### مكونات المثال (4):

- ✓ بورده اردوينو Arduino Uno
- ✓ لوحه تجارب Breadboard
- ✓ دايود ضوئي 5mm led
- ✓ مقاومه 560 اوم
- ✓ مقاومه متغيره 3 اطراف 10 كيلو اوم
- ✓ أسلاك توصيل
- ✓ ورقه المساعدة المطبوعه



**قم** بطباعه الورقه المساعدة ثم وصل المكونات الالكترونيه على لوحه التجارب

**ملاحظه:** لا يشترط ان تستخدم مقاومه متغيره صغيره الحجم مثل التى فى الصوره ولك الحريره 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 التماثلئ 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 اهم الدوال البرمجيه الموجوده في لغة اردوينو **`analogRead(pin number)`** تقوم هذه الداله بقراء فرق الجهد بصوره تماثليه 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 sensorVlaue

```
sensorValue = analogRead(sensorPin);
```

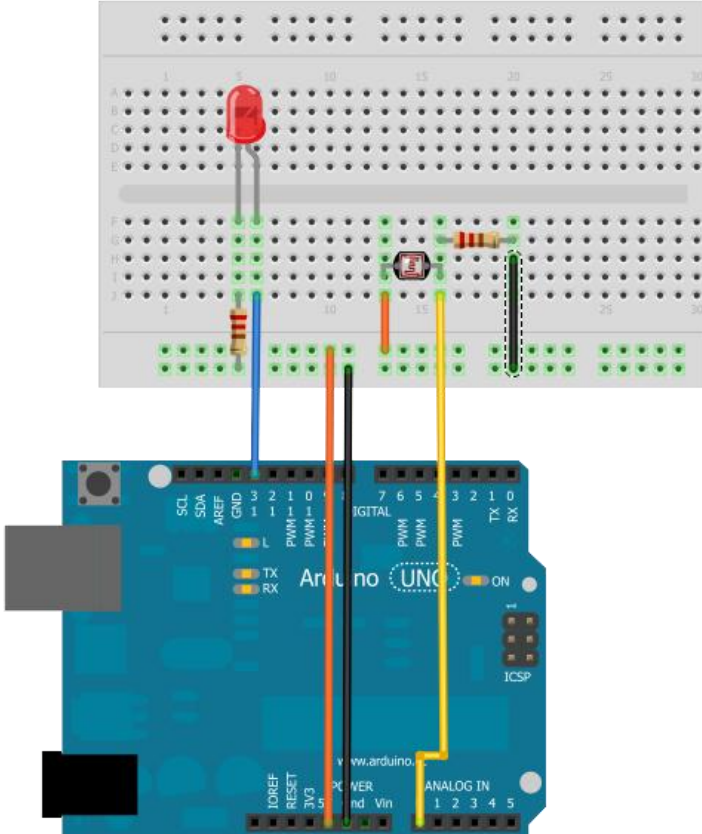
**بعد ذلك** تقوم المتحكمه بتشغيل الدايود الضوئى فتره زمنيّه تساوى قيمه هذا المتغير و تقوم بأطفائه بنفس الزمن مثل المثال الاول blinking led ولكن الزمن فى هذا المثال متغير تبعاً لقراءه الجهد التماثلّى على المدخل A0.

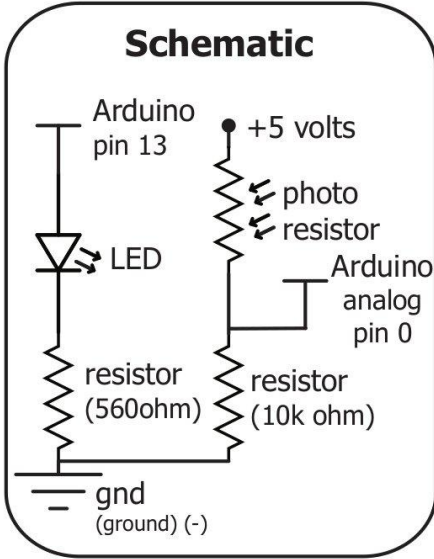
فى هذا المثال استخدمنا مقاومه متغيره يمكننا ان نغير قيمتها عن طريق لف (تدوير) عمود الدوران الخاص بها و بذلك نحصل على اشاره تماثلّيّه (تسمى تلك العمليه تغير متحكم به من الانسان) ، فى المثال القادم سنتعلم نوع جديد من المقاومات المتغيره و هى المقاومه الضوئيه التى تتغير قيمتها بتغير كثافه الضوء الساقط عليها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

## المثال الخامس: المقاومه الضويه (حساس الضوء التماثلي)

### Photo resistor as light sensor (التماثلي)

**تنصنع** المقاومات الضوئية عادة من نوع خاص من السيلكون المُعالج، و يعتبر السيلكون المُعالج مادة 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

✓ دايدو ضوئى led 5mm

✓ مقاومه 560 اوم

✓ مقاومه ضوئيه

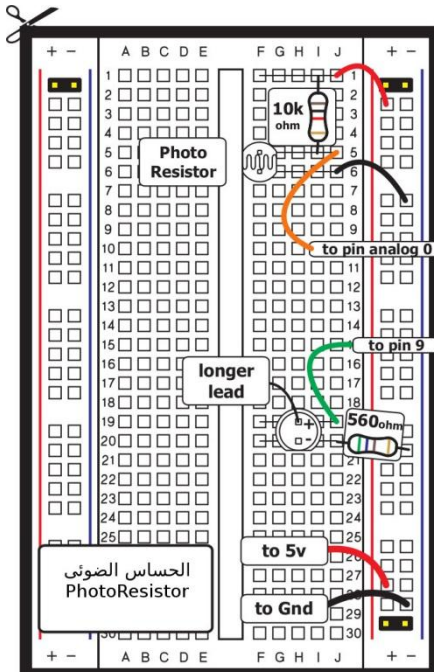
✓ اسلاك توصيل

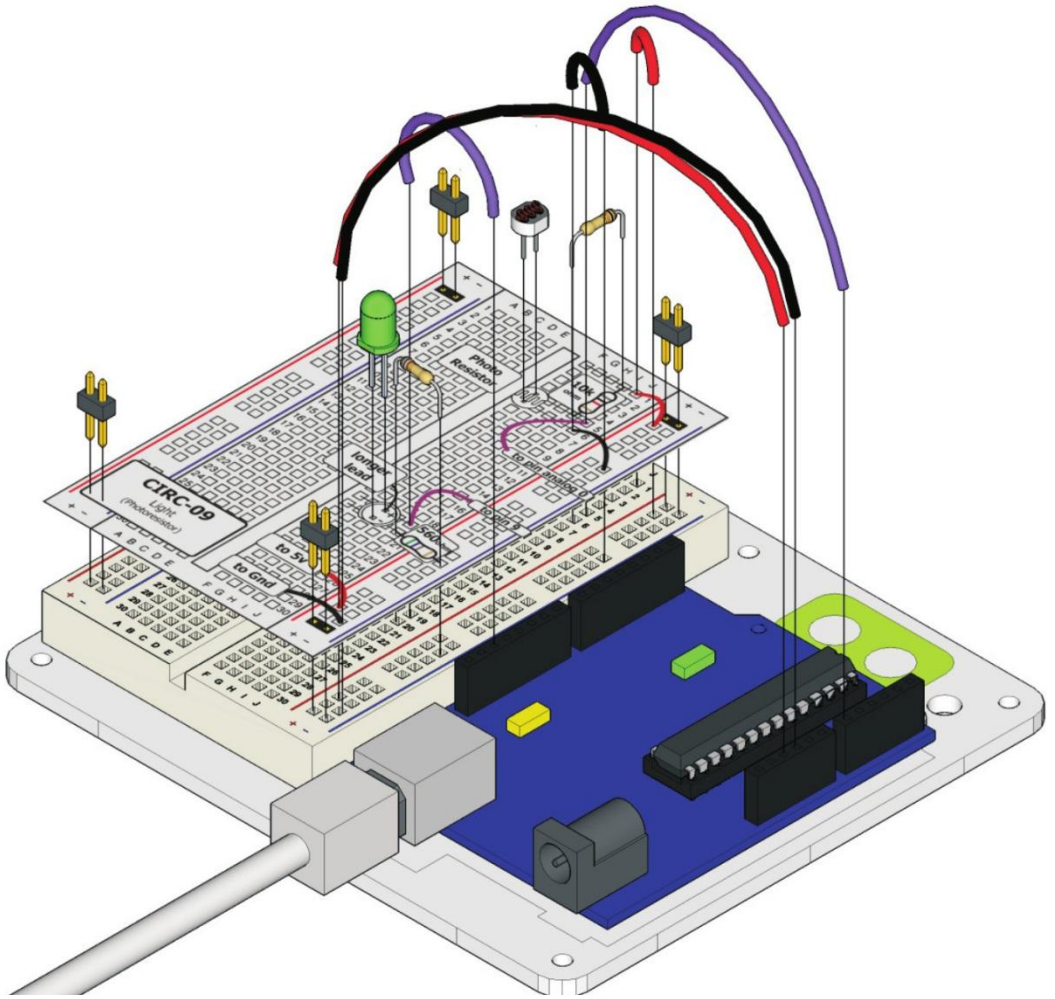
✓ ورقه المساعده المطبوعه

تتوفر المقاومات الضوئية في احجام مختلفه تبعاً

لمقدار حساسيتها للضوء ويمكن استخدام اى

حجم





```
//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 الضوئي 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 الضوئي.

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 النبضه (`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 اشارة رقميه، وبذلك تستطيع أن تحول الأوامر الرقميه 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:

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

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

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

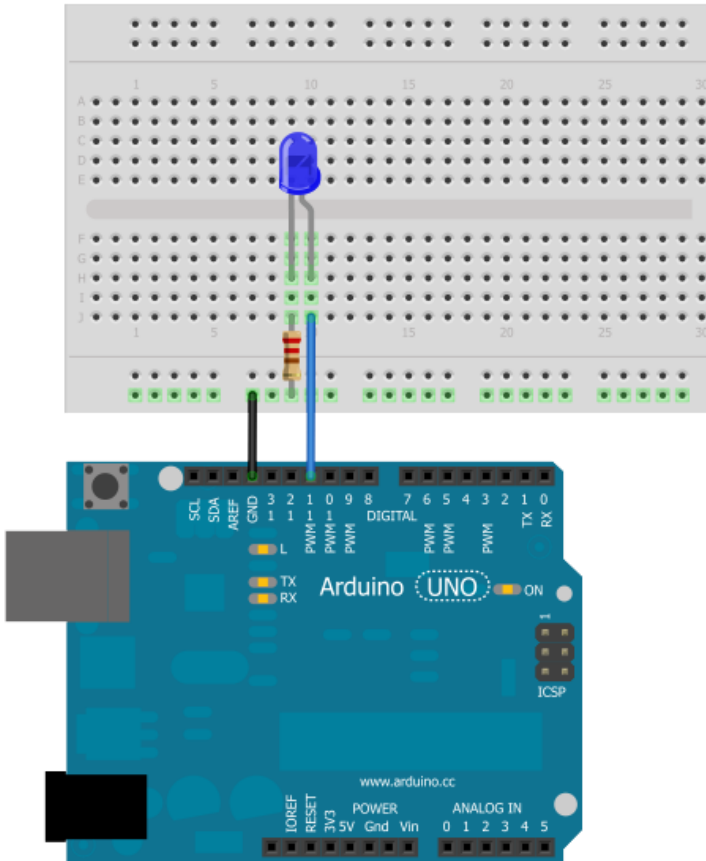
### كيف نستفيد من هذه الخاصيه؟؟

الكثير من المكونات الالكترونيّه والكهريبه تتفاعل مع القيم المختلفه من فرق الجهد بصور مختلفه مثل الدايود الضوئى، مثلاً عندما تطبق 3 فولت على الدايود الضوئى ستجد ان اضائته ضعيفه و اذا رفعت الجهد الى 4 فولت ستجد الاضائه قد ازدادت و اذا وصلت الى 5 فولت ستجد الاضائه وصلت الى اعلى معدل سطوع.. وهناك مكونات كهريبه مثل

المحرك الكهربى (الموتور) و الذى تزداد سرعته بزيادة الفولت الكهربى.  
**في هذا المثال** وصلنا الدايدود الضوئى على المخرج رقم 9 (الذى يدعم PWM) و  
استخدمنا مقاومه ضوئيه حتى تستشعر كميه الضوء الموجود حولها ثم ترسل الاشاره  
التمائليه الى اردوينو فيقوم بتشغيل الليد بنائاً 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 إضاءته عالية مقارنة بالألوان الأخرى و يمكنك اختيار اللون الأبيض كذلك) إلى المنفذ رقم 11 و الذي يدعم خاصية PWM و سنستغل تلك الخاصية في عمل بعض المؤثرات الضوئية الجميلة

و الآن 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

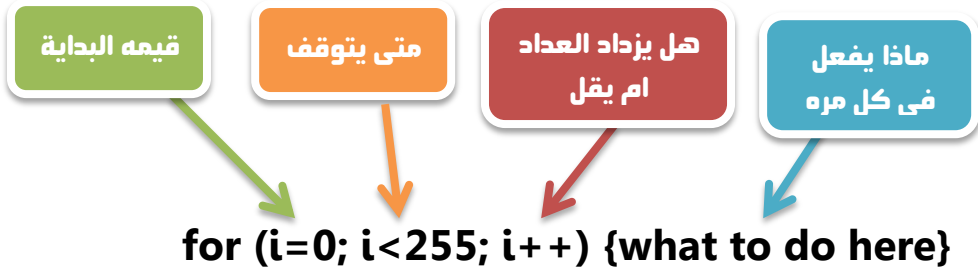
بعد الانتهاء من كتابه الكود و رفعه إلى المتحكم الدقيقة.. لاحظ ماذا يحدث للدايود

الضوئي و كيف تزداد إضاءته بهدوء و تنخفض بهدوء ☺



## ستعلم الآن احدى اهم الأوامر المعبية 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 تكرر 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 كالتالي:



**for (i=0; i<255; i++) {what to do here}**

في المثال السابق استخدمنا عداد يقوم بعد الأرقام ابتدائياً من صفر حتى 255 و في 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)` و بذلك 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 و يظل الرقم يتناقص بمقدار واحد حتى يصل إلى الصفر و في كل مره يسجل العداد قيمه المتغير (i) و يطبق جهد تماثلي على الدايمود الضوئي بقيمه (i) و هذا ما يجعل الدايمود الضوئي يبدأ في الوميض بشده ثم تنخفض شده إضاءةه 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 و لكي 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.

## ملاحظات شخصية :

هذه الصفحة مخصصة لكتابه ملاحظتك الشخصية عن الفصل الرابع :

اكتب ملاحظتك  
هنا

## من أقوال المشاهير:

العلم مغرس كل فخر فافتخر  
واحذر يفوتك فخر ذاك المغرس

واعلم بأن العلم ليس يناله  
من همه في مطعم أو ملبس

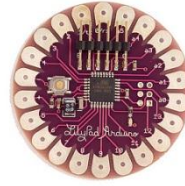
الإمام الشافعي

# الفصل الخامس

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



**جميع** بوردرات اردوينو (بخلاف lilypad, mini) 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 تحتاج إلى ما يسمى بالـ FTDI interface وهي شريحة صغيرة تستخدم لتبادل



Liliypad

البيانات بين الحاسب الآلي و المتحكمات الدقيقة

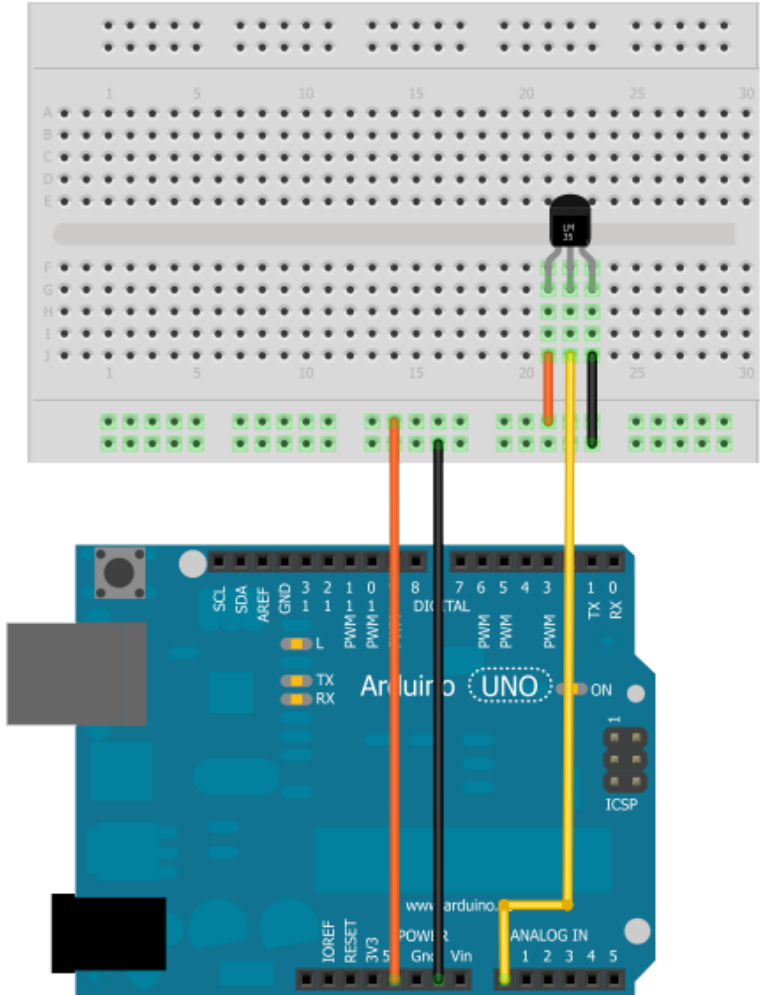


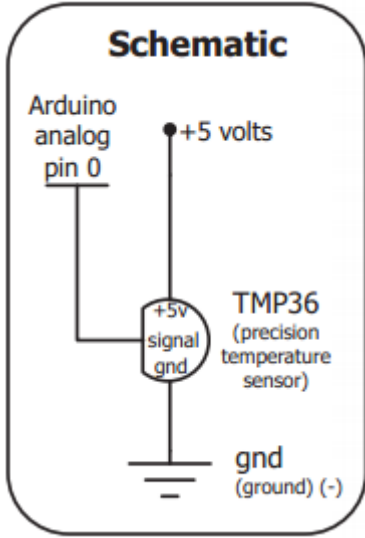
FTDI chip

**يتحدث** الفصل الخامس عن إمكانية استخدام اردوينو في الحصول على معلومات من الحساسات المختلفة و إرسال 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 و إظهار ناتج الحساس على دايود ضوئي ليوضح مقدار هذه القراءة أما الآن فسنستخدم خاصية serial interface لأرسال نتائج القراءة الحساسات إلى الحاسب الآلي حيث يمكن قرائه النواتج بالأرقام و إجراء بعض العمليات الحسابية عليها.

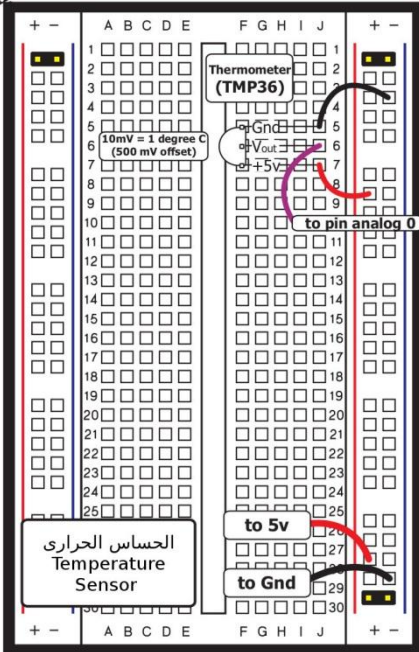
## المثال السابع: استخدام حساس الحرارة و إرسال درجة الحرارة إلى الحاسب الآلي





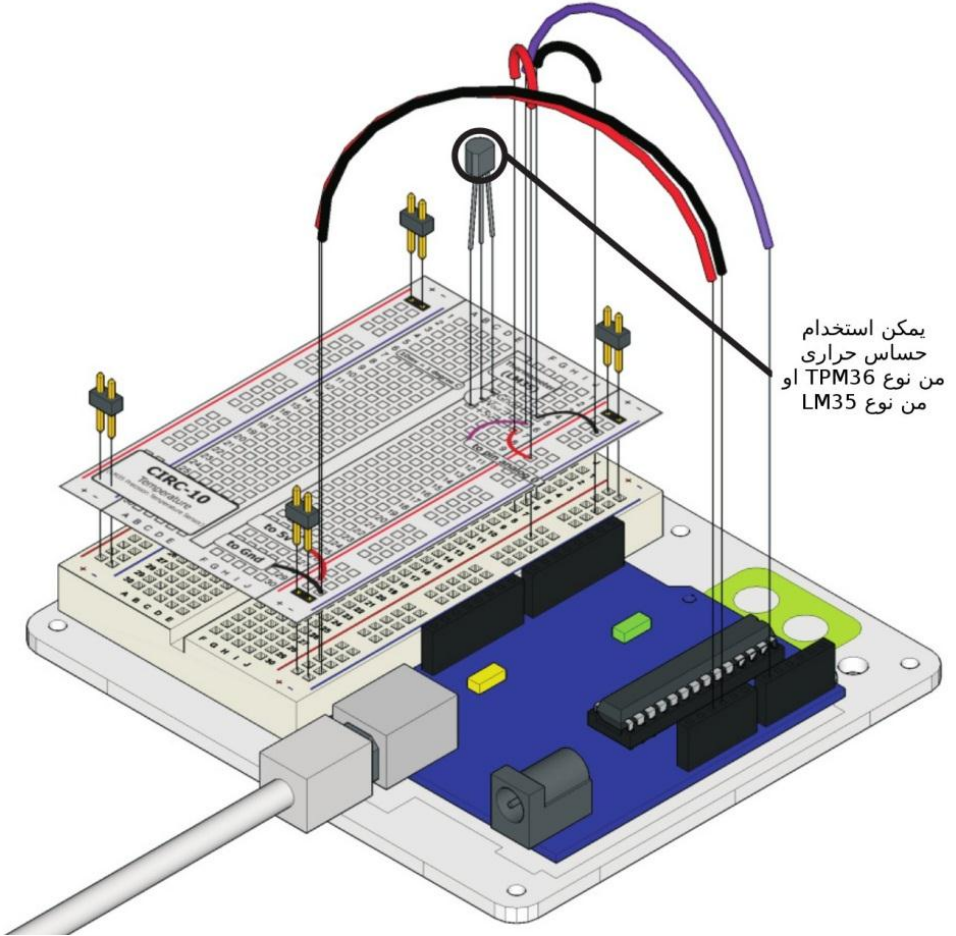
### مكونات المثال (7):

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



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





يمكن استخدام  
حساس حراري  
من نوع TPM36 او  
من نوع LM35

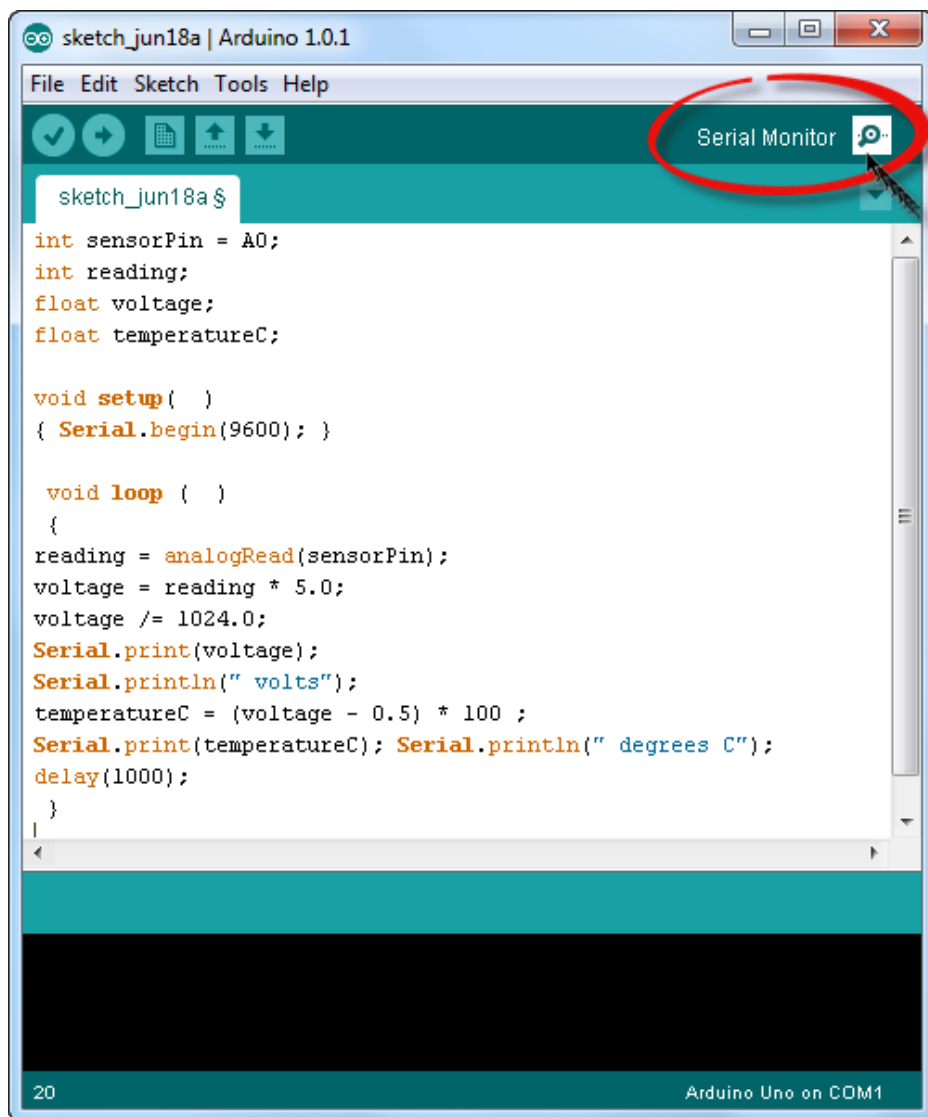
بعد الانتهاء من توصيل المكونات على لوحه التجارب قم بكتابه الأكواد التالية و تأكد من صحتها عن طريق زر **Verify** ثم ارفعها إلى بورده اردوينو

```
// 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 المتحكممة نقوم بالضغط على زر Serial Monitor الموجود في شريط الأوامر السريعة كما في الصورة التالية:



ستظهر صفحة خاصة تستقبل البيانات من بورده اردوينو و تقوم بعرض درجة الحرارة كل ثانية كما سنلاحظ أن الدايدود 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 درجة 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 درجة مئوية تقريبا)

ملحوظه: علامه الشرطه ( - ) التي تكتب قبل درجة الحرارة لا تعنى سالب درجة 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 اردوينو تبدأ التواصل مع الحاسب الآلي عن طريق منفذ الـ USB و بهذا يمكن للاردوينو 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 الحرارى يقيس درجة الحرارة بدقه عالية تصل إلى 0.1 درجة مئوية و من المتوقع أن تكون قيمه الجهد الكهربى الناتج عنه و درجة الحرار بالكسور العشرية و ليس الأعداد الصحيحة لذلك استخدامنا الأمر `float` لجعل هذه المتغيرات تقبل قيمه تحتوى على كسور عشريه .

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

كما ذكرنا سابقا ان المتحكمه 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 الألى لعرضها على بيئة التطوير باستخدام الـ `Serial` Monitor ثم نستخدم الأمر `Serial.println("voltage");` ليقوم بطابعه كلمه `voltage` بعد كل قيمه فولت يعرضها على شاشه الحاسب الآلي

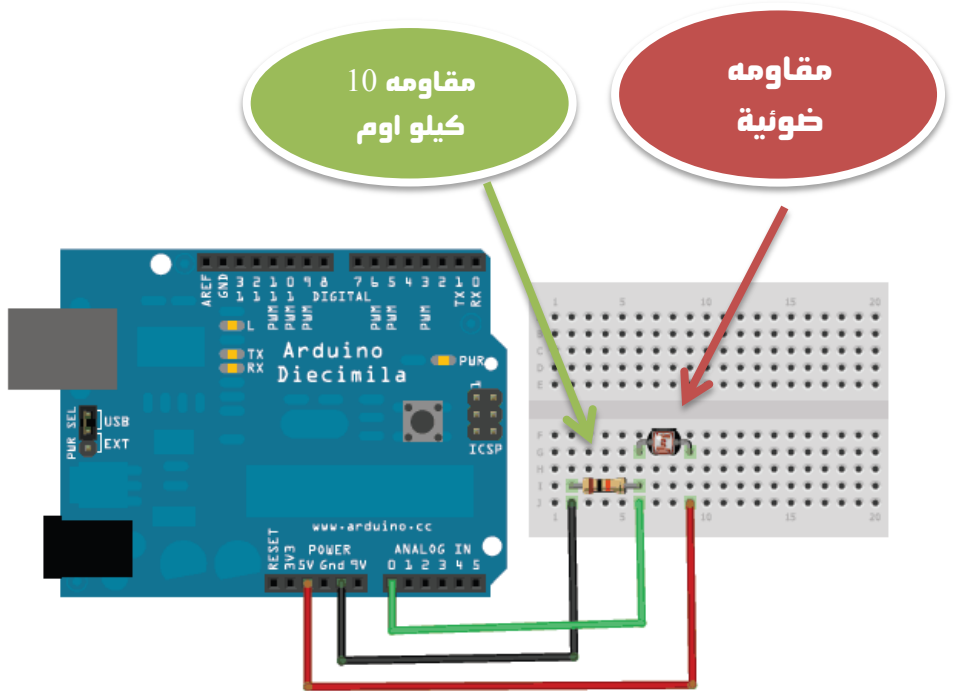
نستخدم الأمر `temperatureC = (voltage - 0.5) * 100;` لتحويل قيمه الفولت إلى درجه حراره بوحدده سي لزيوس Celsius ثم نكرر نفس أوامر `Serial.print` مع درجه الحرارة لعرضها على شاشه الحاسب الآلي

```
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 تذكر المثال الخامس، سنقوم بأعاده تنفيذ 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); }

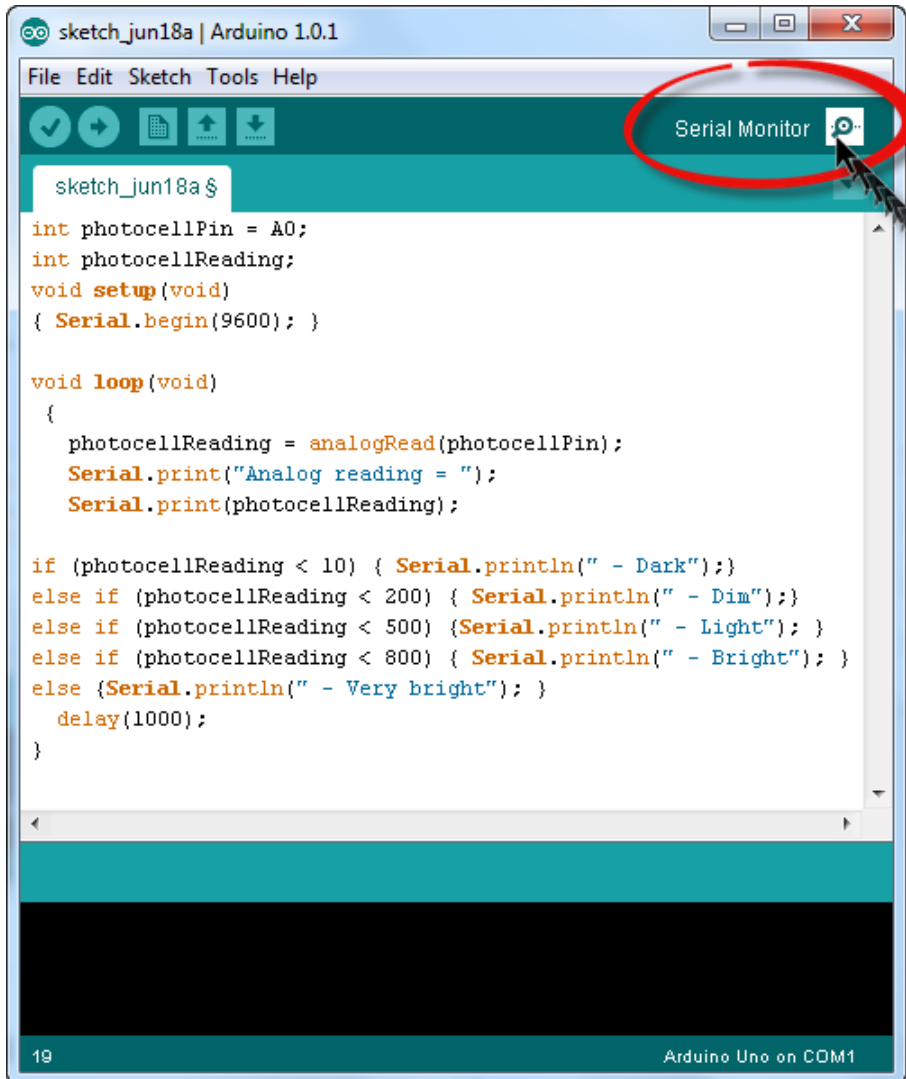
void loop(void)
{
  photocellReading = analogRead(photocellPin);
  Serial.print("Analog reading = ");
  Serial.print(photocellReading);

  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 المتحكممة 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 لتتجنب عنها الضوء تماما

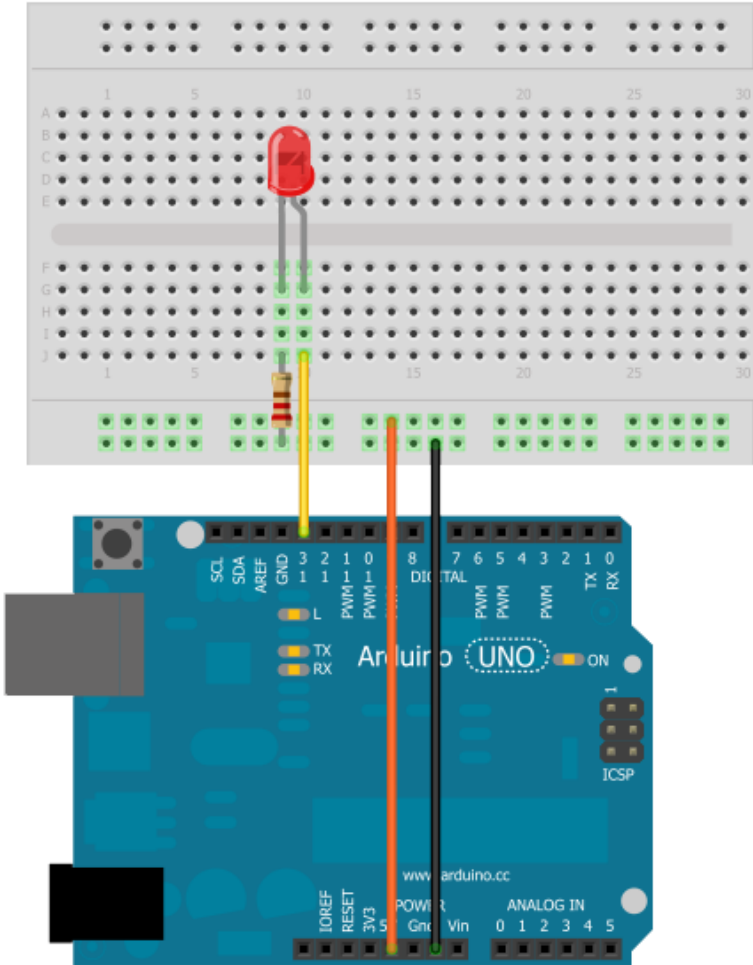
```

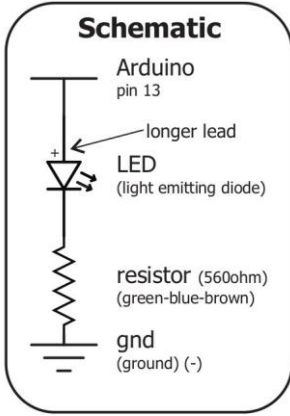
Analog reading = 326 - Light
Analog reading = 325 - Light
Analog reading = 326 - Light
Analog reading = 118 - Dim
Analog reading = 76 - Dim
Analog reading = 66 - Dim
Analog reading = 75 - Dim
Analog reading = 64 - Dim
Analog reading = 100 - Dim
Analog reading = 11 - Dim
Analog reading = 0 - Dark
Analog 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 ورفعه إلى أردوينو اضغط على أيقونة serial Monitor ستجد شريط كتابي في أعلى 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

Autoscroll    No line ending     9600 baud

0

Autoscroll    No line ending     9600 baud

في هذا المثال استخدمنا الأمر `Serial.read()` وهو الأمر المستخدم في قراءة البيانات المرسلّة من الحاسب الآلي إلى اردوينو عبر منفذ الـ USB و قمنا بإضافة `value=` قبل هذا الأمر و ذلك حتى تقوم المتحكممة بقراءة ما يرسل من الـ USB، تخزن هذه القيمة في المتغير `value`

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

- إذا كانت قيمه المتغير `value == 1` تقوم المتحكممة بتشغيل الـ دايدود الضوئي
- أما إذا كانت قيمه المتغير `value == 0` تقوم المتحكممة بإطفاء الـ دايدود الضوئي

**الآن حاول أن تعدل الكود بنفسك و تزيد 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>

## ملاحظات شخصية :

هذه الصفحة مخصصة لكتابه ملاحظاتك الشخصية عن الفصل الخامس :

اكتب ملاحظاتك  
هنا



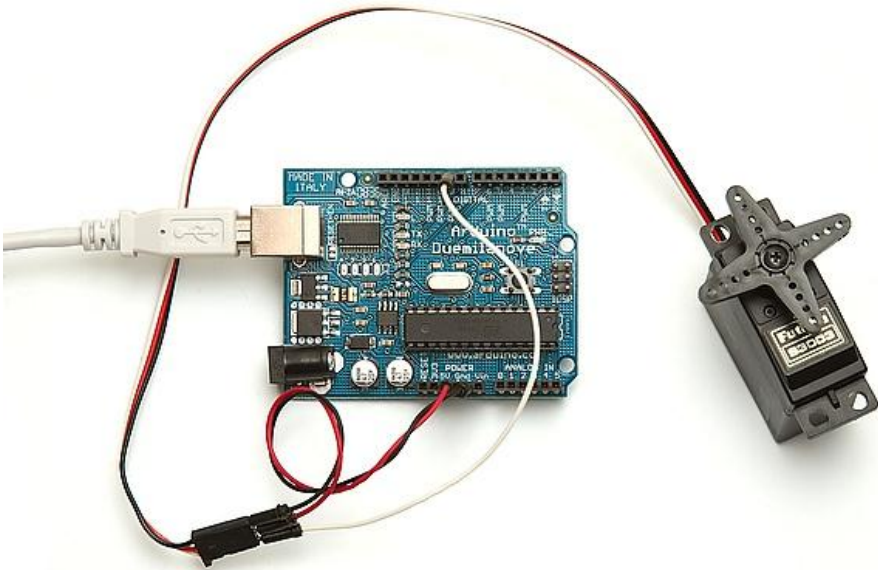
## من أقوال المشاهير:

المعرفة قوة في يد من يعلمها

ألبرت أينشتاين

# الفصل السادس

## أستخدام المحركات The Motors



تعتبر المحركات من أهم العناصر الداخلة في المشاريع الألكترونيه و هي العنصر المسؤل 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 الـ DC Motor و الـ Servo Motor وهما أشهر أنواع المحركات المستخدمه فى مشاريع المتحكمات الدقيقه بأنواعها المختلفه.

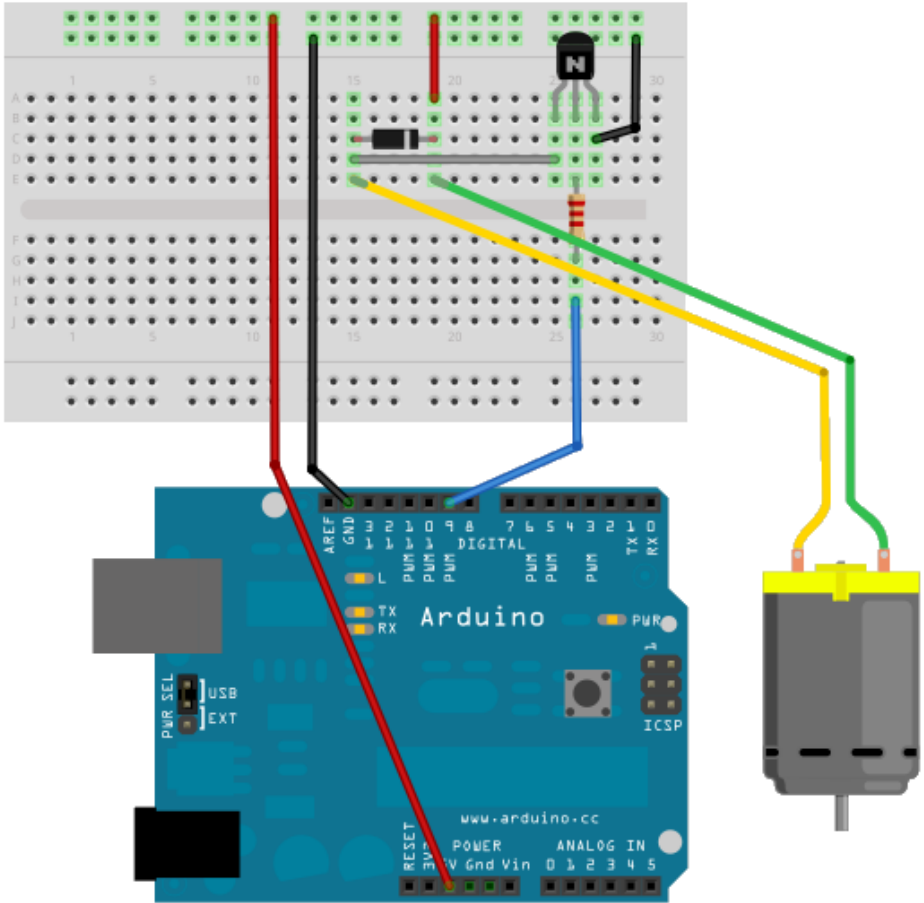


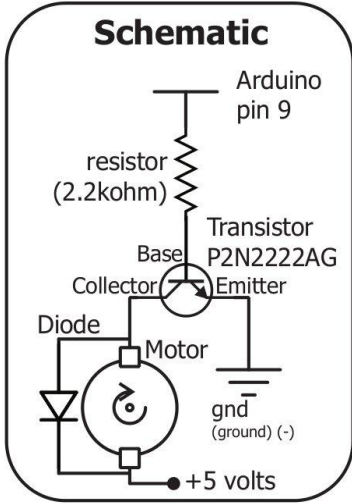
Servo Motor



DC Motor

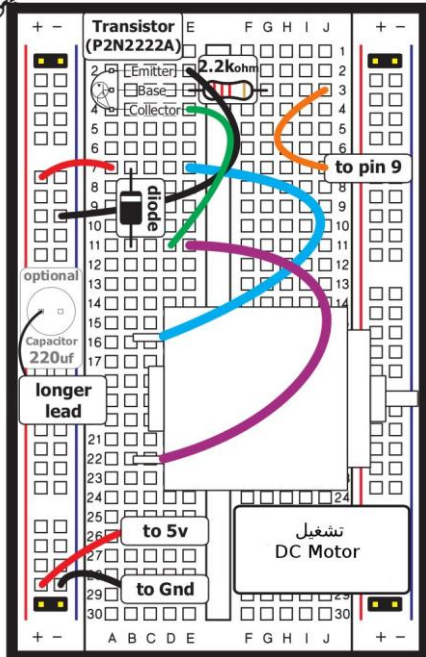
## المثال العاشر: استخدام محرك التيار المستمر DC





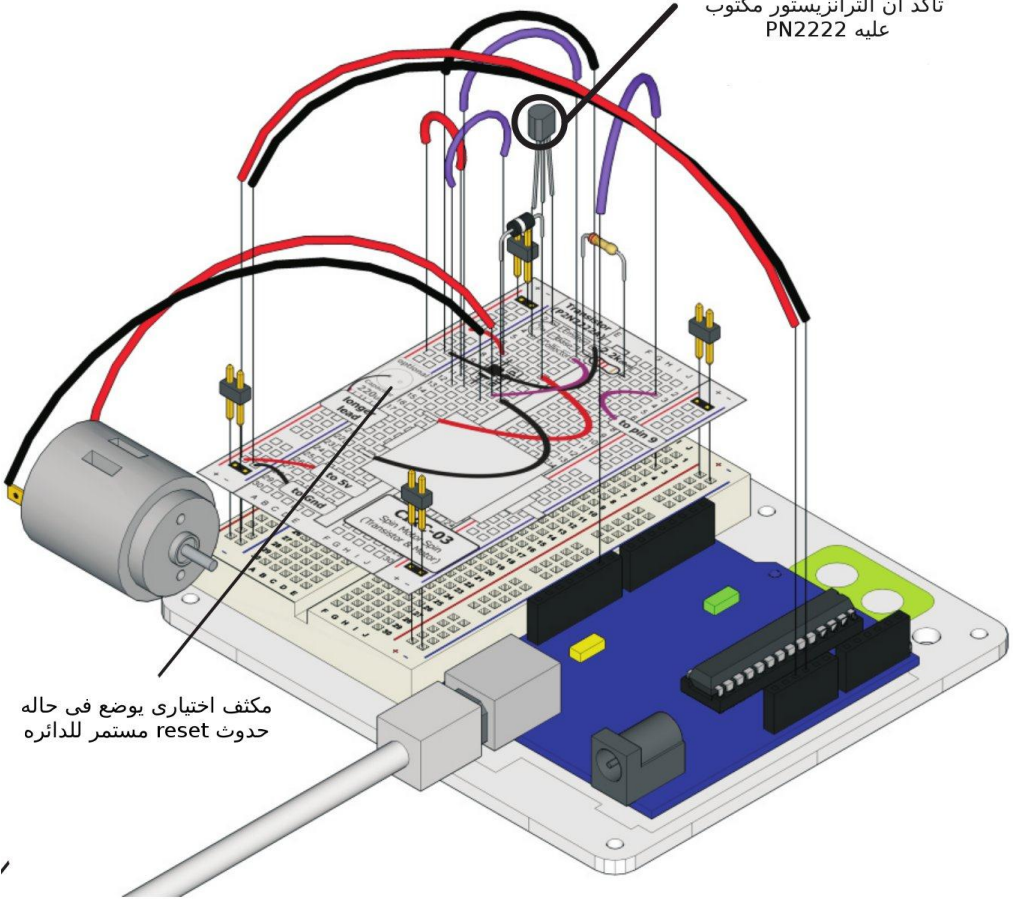
### مكونات المثال (7):

- ✓ 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 بالUSB
- ✓ ورقه المساعدة المطبوعه



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

تأكد ان الترانزيستور مكتوب  
عليه PN2222



مكثف اختياري يوضع فى حاله  
حدوث reset مستمر للدائره

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 ;

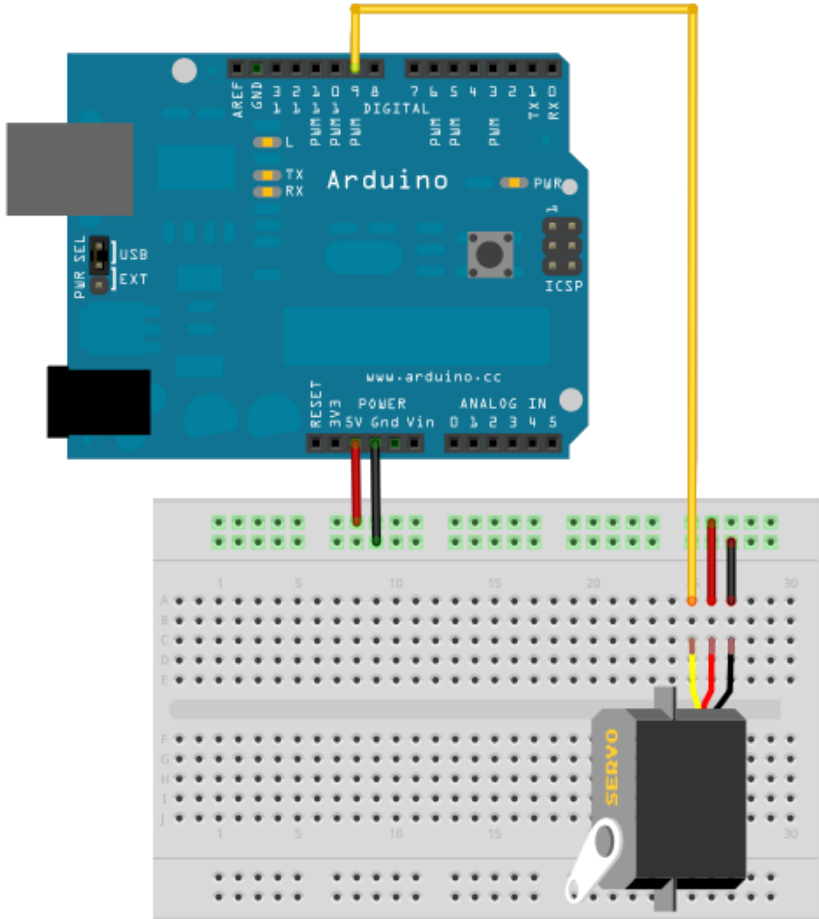
void setup ( )
{pinMode(motorPin, OUTPUT); }

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

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

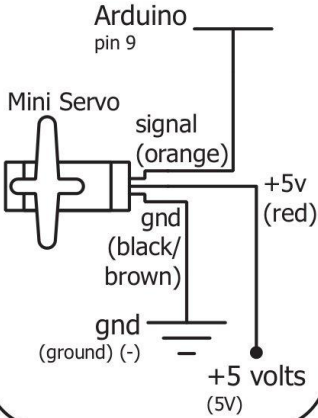
analogWrite(motorPin,255);
delay(onTime);
digitalWrite(motorPin, LOW);
delay(offTime);
}
```

## المثال الحادي عشر: استخدام محرك سيرفو

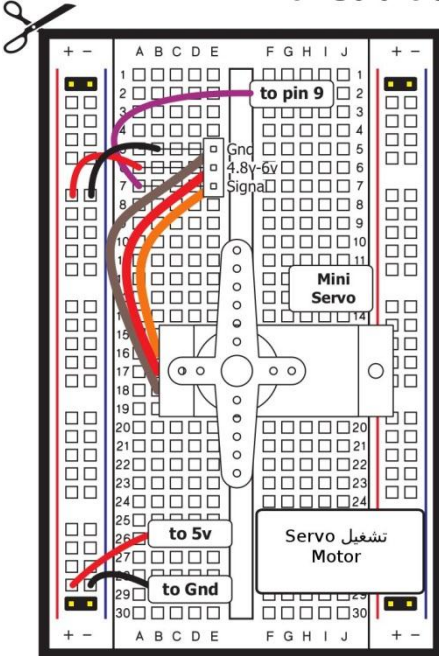




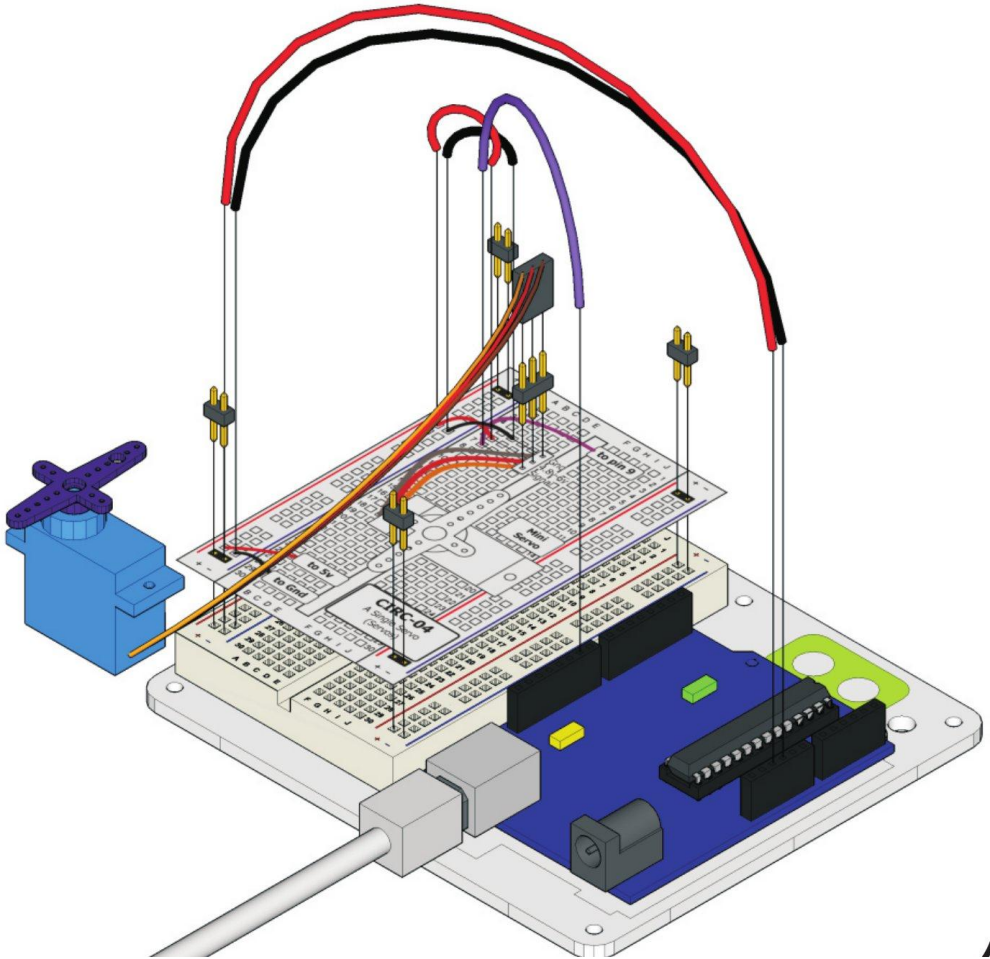
## Schematic

**مكونات المثال (7):**

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



في هذا المثال سوف نستخدم محرك تيار مستمر من نوع سيرفو موتور و الذي 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 (ملحوظه: يمكنك ان تجد الكود البرمجي في قائمه الأمثله الجاهزه في برنامج اردوينو examples - servo - sweep)

```
//Example_11_Servo_Motor
#include <Servo.h>
Servo myservo;
int pos = 0;
```

مكتبة اضافيه

```
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);
  }
```

```
}
```

## ملاحظات شخصية :

هذه الصفحة مخصصة لكتابه ملاحظاتك الشخصية عن الفصل السادس :

اكتب ملاحظاتك  
هنا

## من أقوال العظماء:

الجبال قد تصمد امام الزلازل لكنها لن تقاوم  
قَطرات الماء التي تهطلُ بانتظام في هدوء و  
تكرار ، فأجعل عمك مثل قطرات المياه حتى  
تتغلب على جميع العوائق

كلمه صينيه قديمه

# الفصل السابع

وسائل الادخال و الاخراج المتطورة

Advanced Inputs/Outputs



في الفصول السابقة اعتمدنا بشكل اساسي على ابسط وسائل الادخال و الاخراج للتعامل مع اردوينو مثل السويتش (ادخال Input) و الدايود الضوئي (اخراج output)



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

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



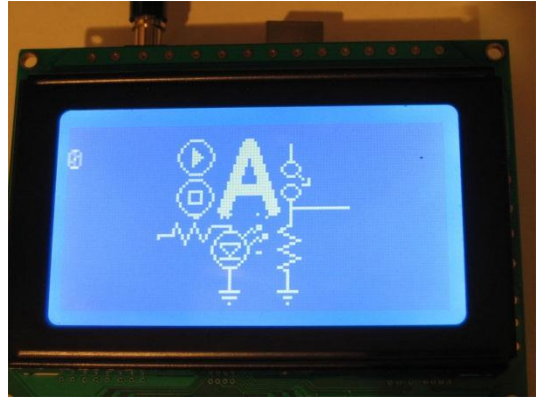
شاشات عرض الكريستال السائل : تتكون هذا الشاشات من زجاج الكريستال المعالج و تتوفر هذه الشاشات بأحجام و انواع مختلفه و سوف نستعرض منها نوعين اساسيين و هما:

- 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 الوان مختلفه مثل:

Green 16x2 lcd

Blue 16x2 lcd

Greens 20x4 lcd

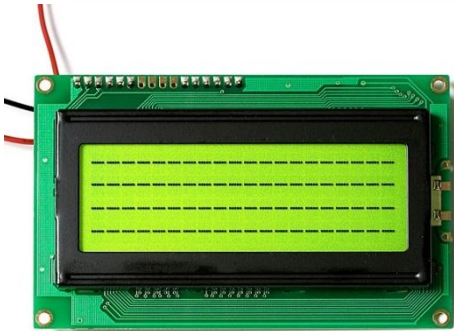
و يمثل الرقم 16x2 عدد السطور (2) الحروف التى يمكن كتابتها فى كل سطر (16) حرف كما تتوفر بألوان مختلفه كما فى الصور التاليه:



Green 16x2 LCD



Blue 16x2 LCD

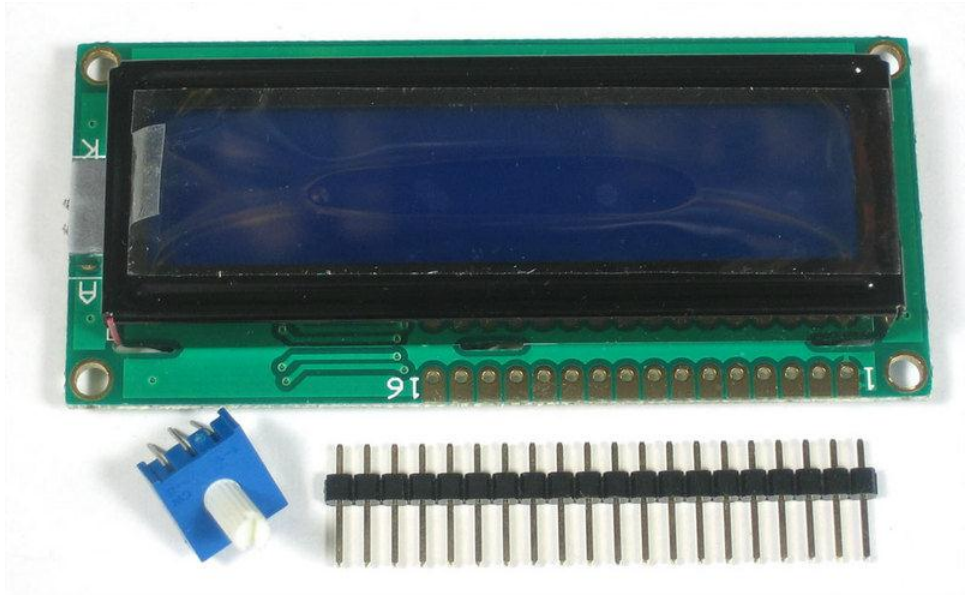


Green 20x4 LCD



Black 16x2 LCD

## المثال العاشر: توصيل شاشة Character LCD بمقاس 16x2



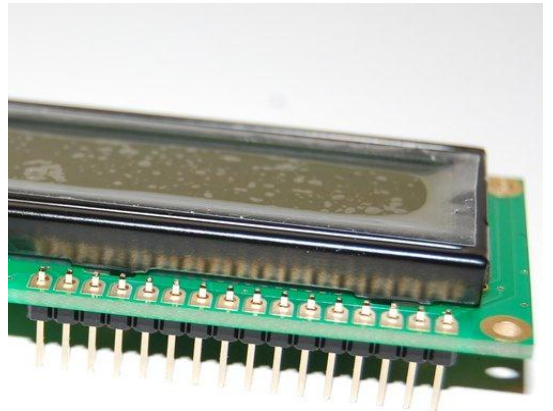
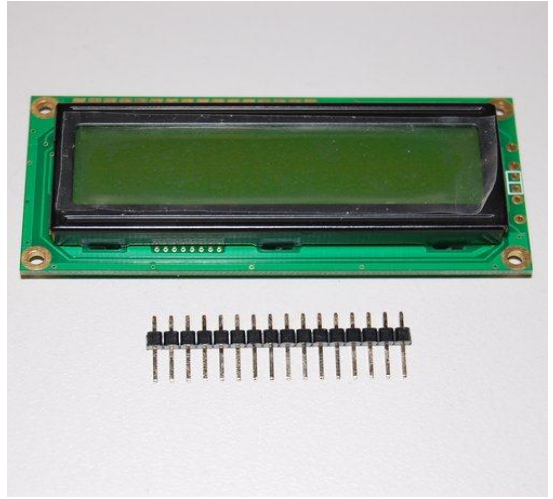
### سنحتاج المكونات التاليه:

- 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 العمليه:

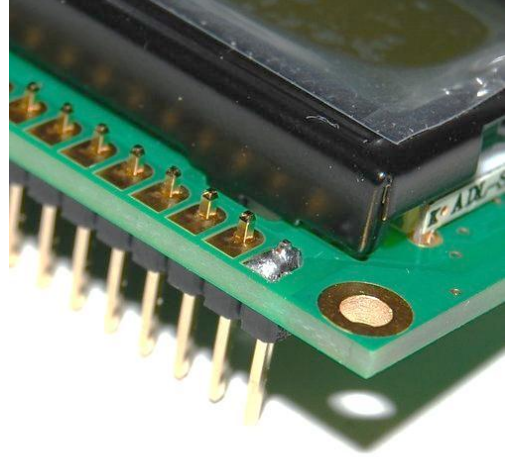
إذا اردت تعلم لحام المكونات الإلكترونية توجهه إلى اللينك التالي

<http://www.aaroncake.net/electronics/solder.htm>



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

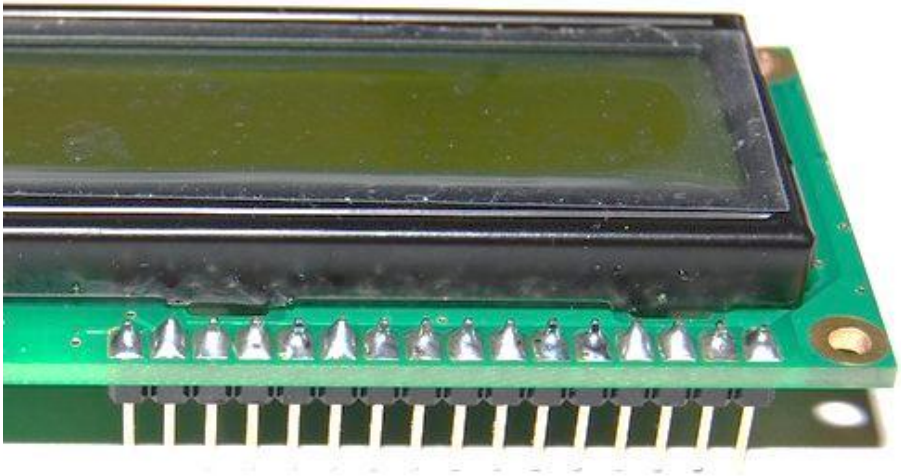
تنبيه: لا تحاول لمس نقطه اللحام بيدك بعد اللحام مباشره فقد يؤدى ذلك الى حدوث حروق للجلد من حراره اللحام



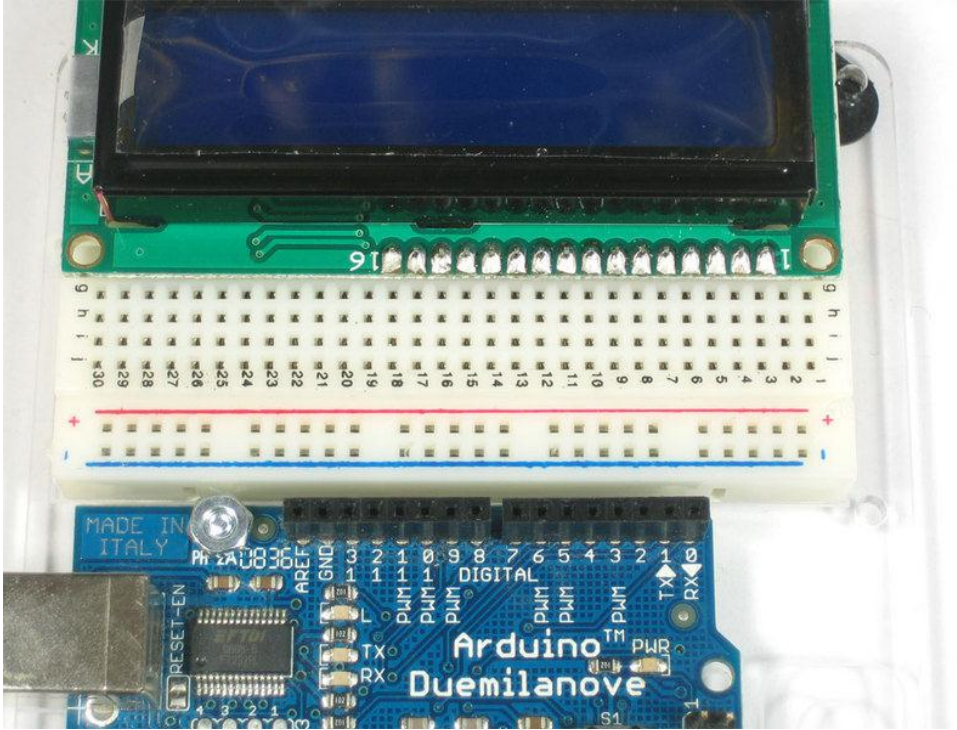
بعد ذلك قم بلحام نقطه التوصيل الاخيره وذلك حتى تقوم بتثبيت pin headers من كلا الجانبين



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



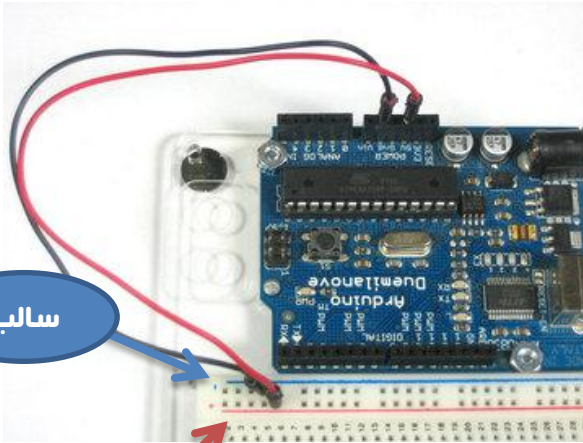
بعد الانتهاء من اللحام قم بتركيب الـ pin header على لوحة التجارب كما في الصورة



يفضل استخدام قاعده تثبيت بلاستيكيه تستخدم في تثبيت اردوينو و لوحة التجارب معا  
المثل التي تستخدم في الصورة التاليه



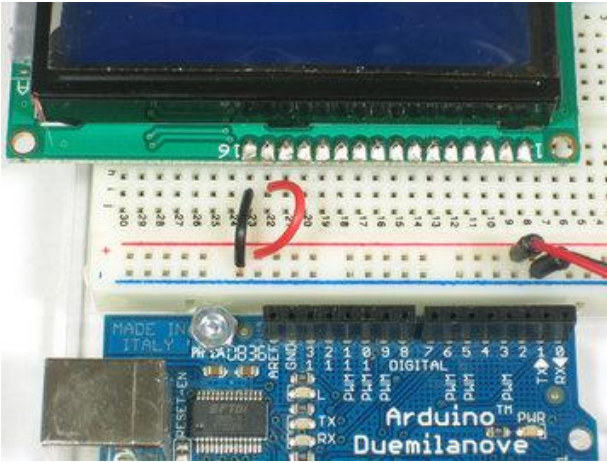
قاعده التثبيت  
البلاستيكيه تحمل  
اردوينو و لوحه  
التجارب و الشاشه  
معا



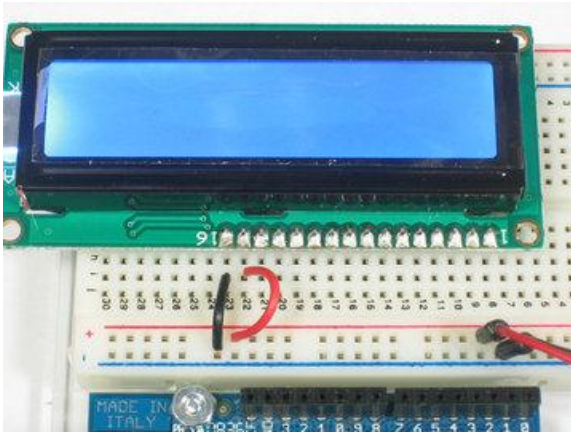
- الان نقوم  
بتوصيل المخرج 5v على  
خط الموجب الاحمر و  
الطرف GND على الخط  
الارضى فى لوحه  
التجارب

سالب

موجب



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



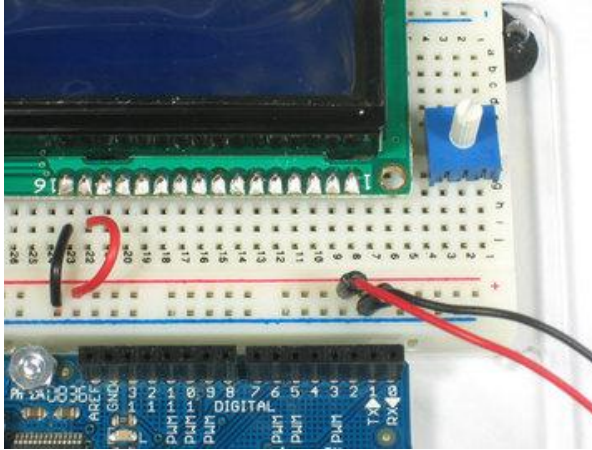
- قم بتوصيل اردوينو بالبطاريه او كابل الـ USB ثم لاحظ ان الضوء الخلفى للشاشه بدأ في العمل كما في الصورة التاليه :

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

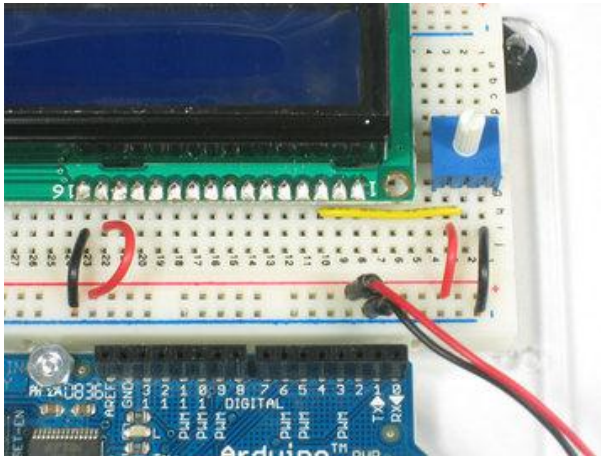
- الاحمر - الأخضر -
- الأزرق - الأبيض -



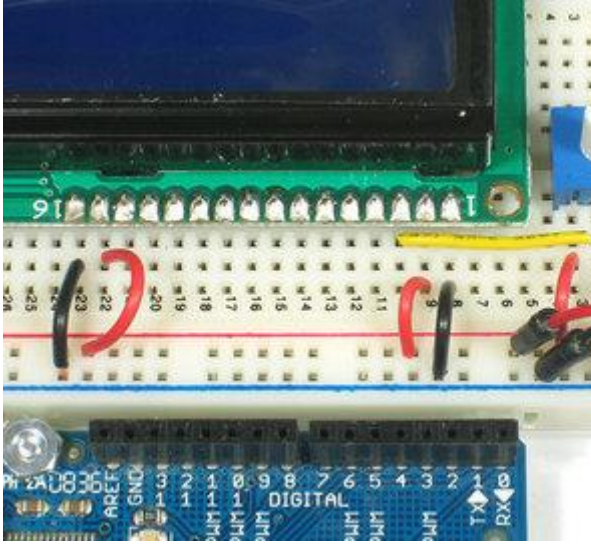
## تركيب المقاومه المتغيرة للتحكم فى شدة الاضاءة



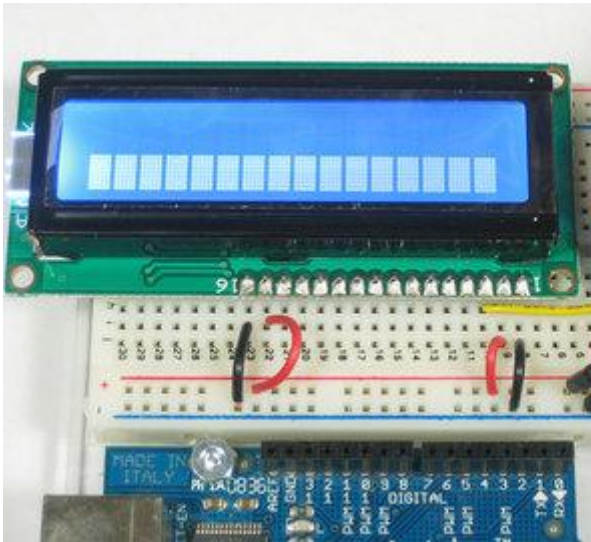
تعتبر تلك الخطوه اختياريه و يمكنك ان تهملها اذا اردت. الهدف من تركيب المقاومه المتغيره هو التحكم فى التيار الكهربى الداخلى الى الشاشه و بالتالى التحكم فى شدة السطوع (الاضاءه) الناتجه



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



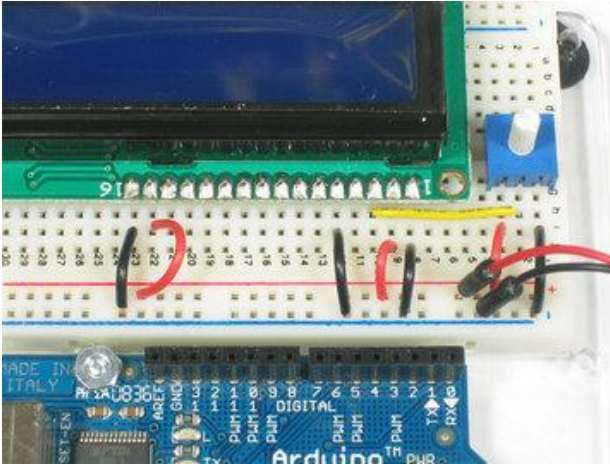
- قم بتوصيل المدخل رقم 1 على شاشة العرض بالخط الارض والمدخل رقم 2 بالخط الموجب على لوحة التجارب



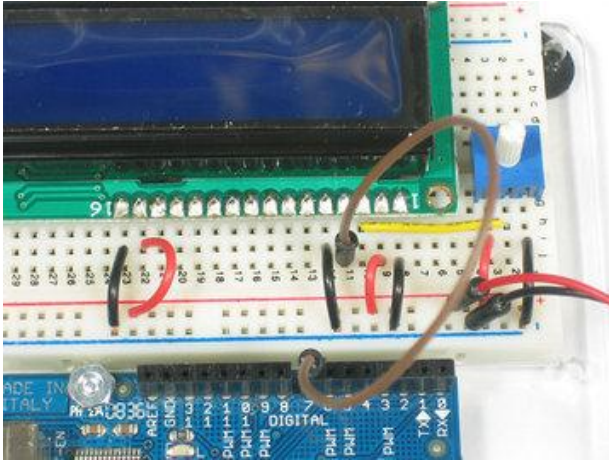
- الان قم بوصيل البطاريه مره اخرى و قم بتدوير عصا الدوران فى المقاومه المتغيره ولاحظ ماذا سيحدث للاضاءه الخلفيه لشاشه العرض



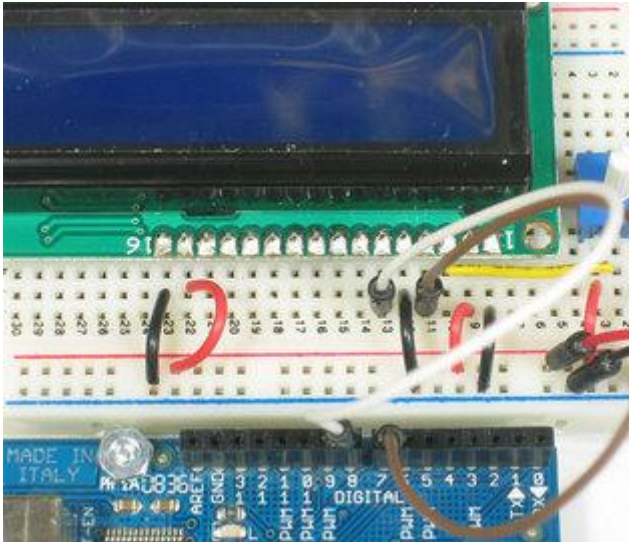
جميع الخطوات السابقة كانت تهدف لتوصيل الشاشة بالطاقة و بالمقاومه الضوئيه التي تتحكم في مدى سطوع الشاشة و شده الاضائه الخلفيه ، الخطوات التاليه ستوضح كيف توصل منافذ نقل البيانات من اردوينو الى الشاشة.



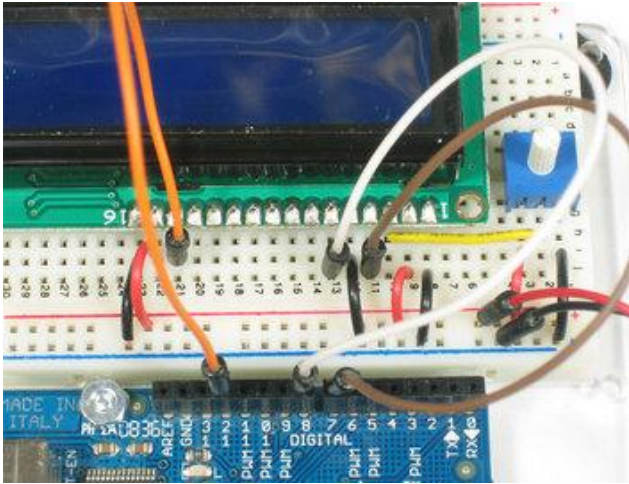
في بعض المشاريع قد تستخدم المدخل رقم 5 و الذي يسمى RW لكن في حالتنا هذا سنقوم بتوصيله بالخط الارضى



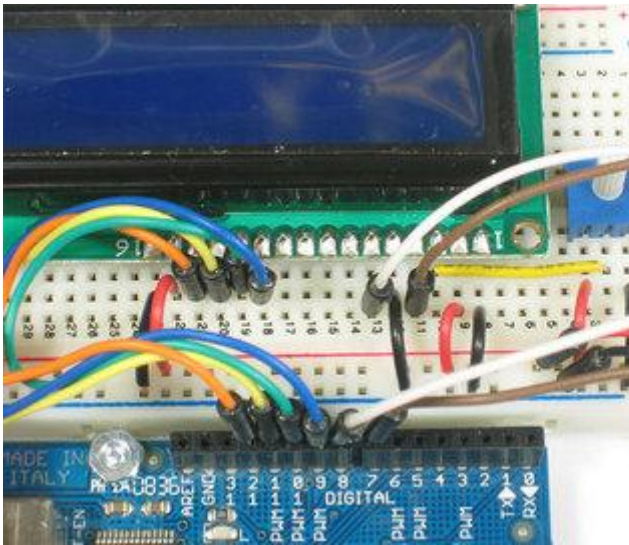
قم بتوصيل المدخل  
رقم 4 في شاشة العرض  
على المخرج رقم 7  
في بورده اردوينو



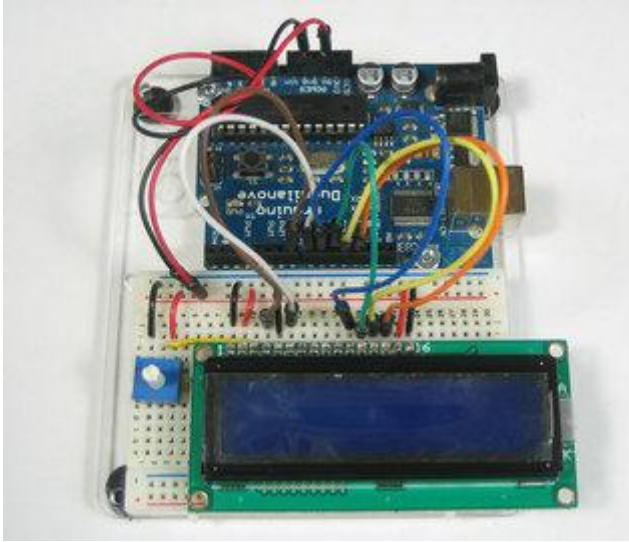
بعد ذلك نقوم بتوصيل  
المدخل رقم 6 في  
شاشة العرض على  
المخرج رقم 8 في  
بورده اردوينو



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  
الموجوده على  
اردوينو بنفس الترتيب  
كما في الصوره التاليه



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

### الان حان وقت كتابه الكود البرمجي

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

**File→Examples→LiquidCrystal→HelloWorld**

سنحتاج ان نعدل السطر البرمجي التالي :

```
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 في الامثله السابقه

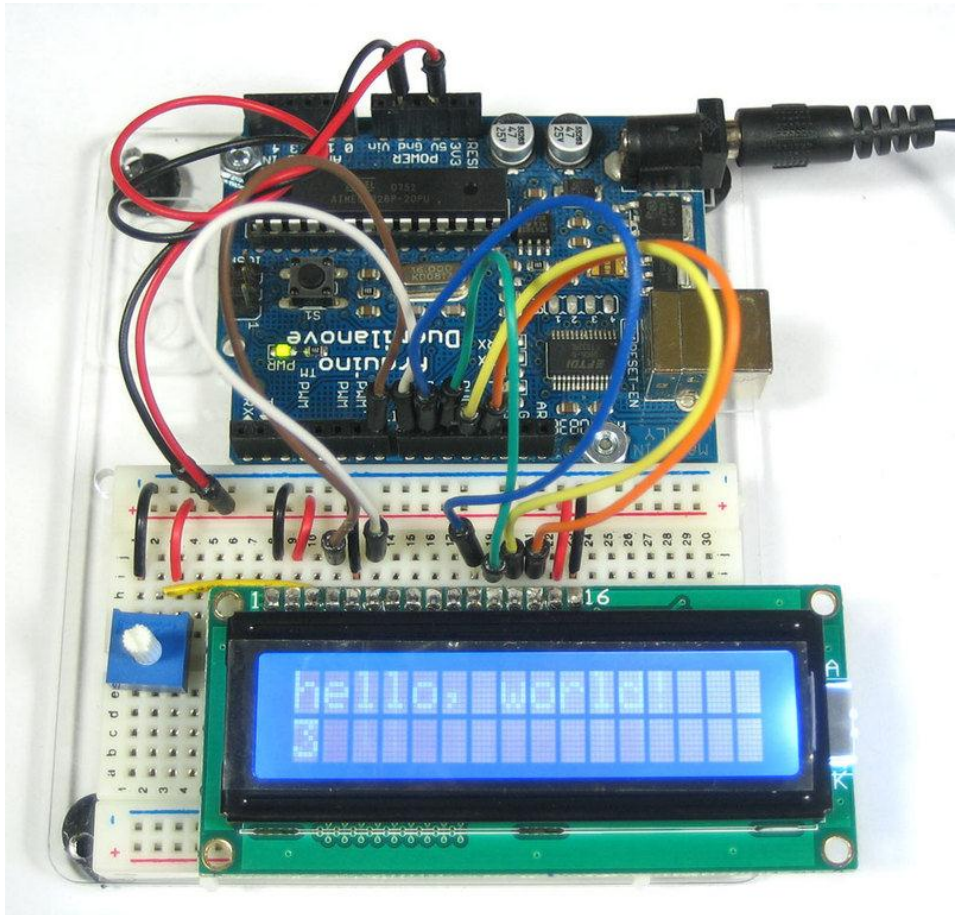
الكود البرمجي بعد التعديل:

```
//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 (^\_^)

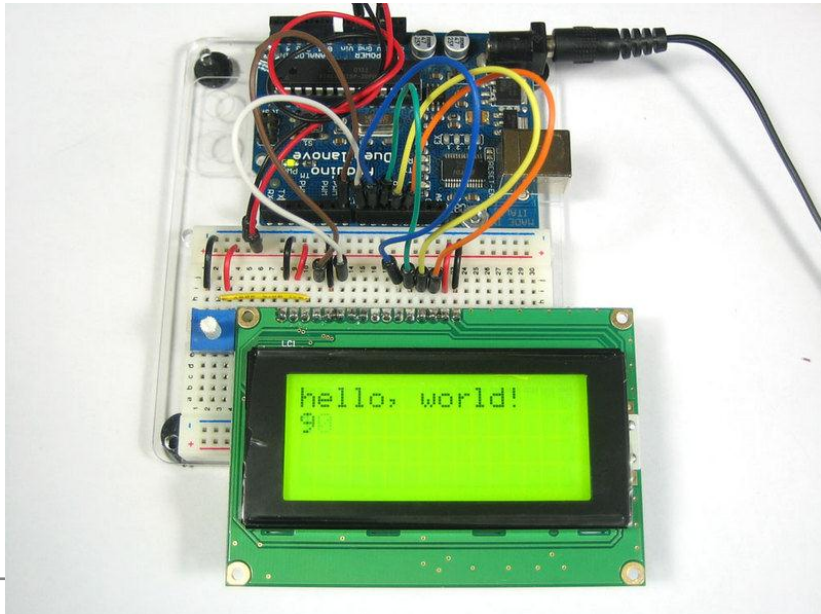


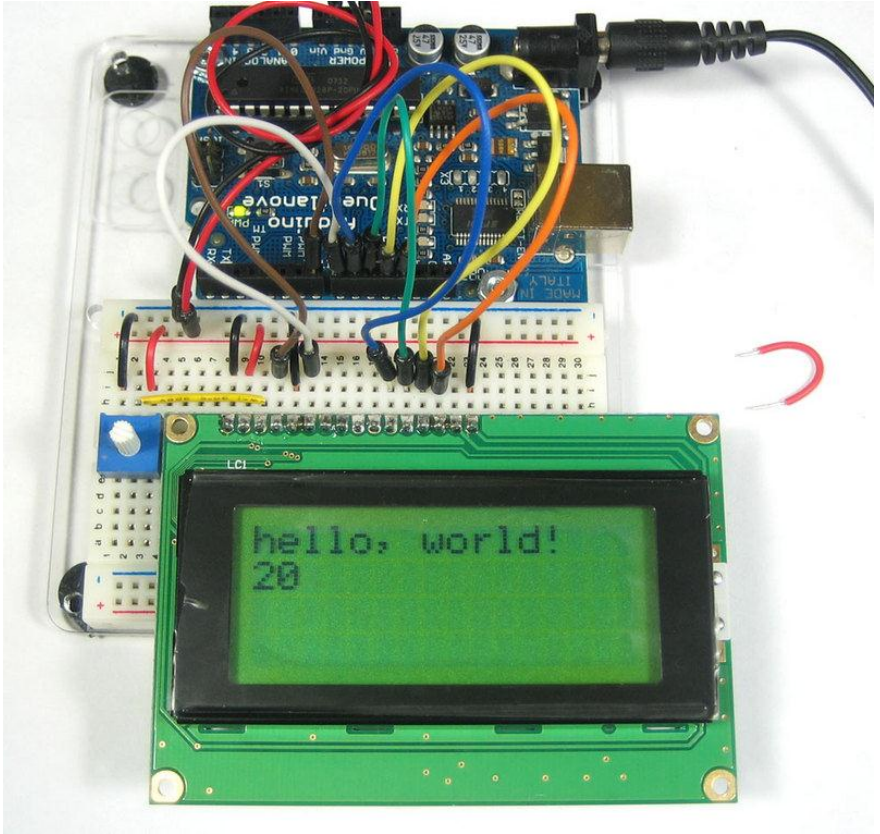


تستطيع تعديل مستوى سطوع الشاشة عن طريق تغير قيمه المقاومه المتغيره



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 او Nokia 2600

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

## استخدام لوحة الأرقام Keypad مع اردوينو



تعتبر لوحة الارقام من اهم وسائل الادخال المستخدمه بكثره فى مشاريع المتحكمات الدقيقة و التى ستجدها حولك فى العديد من الاجهزه الالكترونيه مثل الهاتف المنزلى و لوحة المفاتيح و لوحة التحكم فى المصاعد الموجوده فى البيوت متعددده الطوابق ... الخ

و تختلف اشكال لوحات الارقام تبعاً لحجمها و عدد الارقام المتاحة ، فى بعض اللوحات ستجد رموز اضافيه مثل علامه النجمه (\*) و علامه الشباك (#) او حتى بعض الحروف الانجليزيه مثل A,D,F و تعتبر لوحة الارقام بحجم 4x3 و 4x4 هي اشهر لوحات الكتابه



3x4



4x4



4x4



كما تتوفر بعض اللوحات المرنة والقابلة للطي بسهولة والتي تصنع من نوع خاص من البلاستيك المرن و تتميز بالنعافه حتى تبلغ ثخانه اللوحه بضعه مليمترات فقط و تتميز بالسعر المنخفض (حوالي 2 دولار فقط)



سوف نشرح لوحه الارقام المرنة بمقاس 3x4 والتي تتميز بالموصفات التاليه:

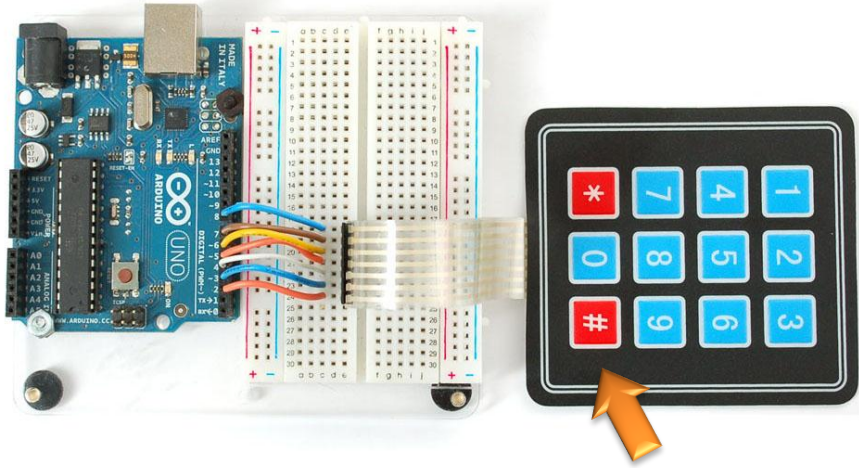
- Weight: 7.5 grams
- Keypad dimensions: 70mm x 77mm x 1mm (2.75" x 3" x 0.035")
- Length of cable + connector: 85mm
- 7-pin 0.1" pitch connector

## المثال الحادي عشر استخدام لوحة ارقام



### مكونات المثال الحادي عشر:

- لوحة ارقام (مرنه) بمقاس 3x4
- أعمده توصيل نحاسيه 7 نقاط ( 7 ) Pin-Headers
- اردوينو Arduino Uno
- لوحة تجارب Breadboard



قم بتوصيل مخارج اردوينو بدءاً من المخرج رقم 2 حتى المخرج رقم 8 بلوحة الأرقام مع ملاحظة ان يكون رمز # هو الرمز المتواجد امام المخرج رقم 2 في بورده اردوينو كما في الصورة بالظبط و بذلك نكون قد انتهينا من تجهيز لوحة الأرقام.

قبل ان نبدأ كتابة الكود البرمجي سيتوجب تحميل مكتبه لوحة الارقام من موقع اردوينو حيث لا تتوفر هذه المكتبه بشكل افتراضى فى برنامج Arduino IDE على عكس مكتبه استخدام الشاشات من نوع Character LCD و التى تتوفر بشكل افتراضى داخل برنامج Arduino IDE

أولاً: توجهه الى اللينك التالى على موقع اردوينو الرسمى:

<http://www.arduino.cc/playground/Code/Keypad>

ثم قم بتحميل مكتبته لوحه الارقام كما فى الشكل التالى:



Version 3.0 has just been posted (19 July 2012) and was rewritten to support multi-keypresses by default. But for those who still need the original single-keypress functionality, the library is fully backwards compatible.

You won't need external resistors or diodes because the library uses the internal pullup resistors and additionally ensures that all unused column pins are high-impedance.

Support was added to allow other hardware to be used with the keypad. When you download the Keypad library you will also get the Keypad\_I2C library. That library adds support for using a matrix keypad with I2C port expanders. You can find more information inside the docs folder under the Keypad\_I2C library folder. Start with the noHeaders example sketch to learn which libraries are needed.

---

## Download, install and import

Download here: [keypad.zip](#)

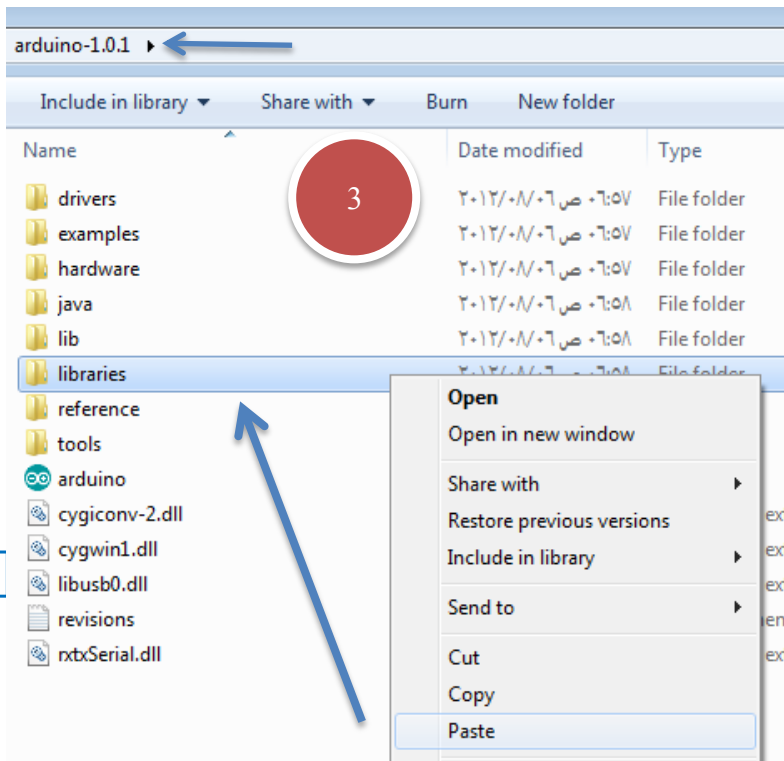
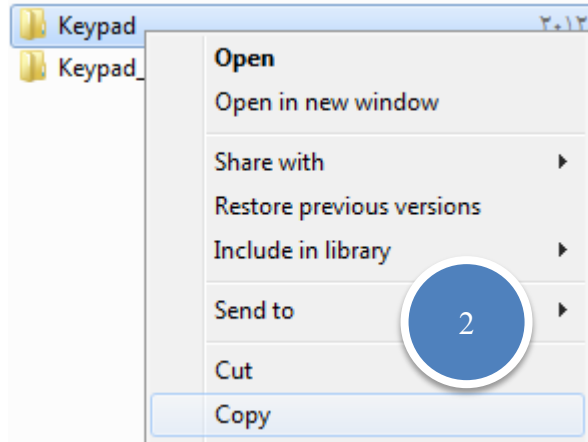
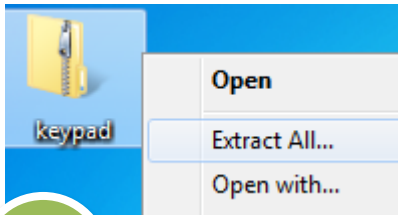
Put the Keypad folder in "*arduino*\libraries\".

In the Arduino IDE, create a new sketch (or open one) and select from the menubar "Sketch -> Import Library -> Keypad".

Once the library is imported, an "#include <Keypad.h>" line will appear at the top of your Sketch.

بعدها سيبدأ المتصفح بتنزيل مكتبته لوحه الأرقام و هى عبارته عن ملف مضغوط اسمه keypad.zip ، الخطوه التاليه هى ان تقوم بفتح الملف و نسخ الفولدر المسمى keypad و توجهه الى مكان تواجد برنامج اردوينو **Arduino IDE** حيث ستجد فولدر اسمه libraries قم بفتح الفولدر و الصق keypad داخله كما فى الصوره التاليه:





بعد الانتهاء من تجهيز المكتبة البرمجية نبدأ في كتابه الكود:

```
//Example_13_Keypad_Input
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] =
{
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'#','0','*'}
};
byte rowPins[ROWS] = {5, 4, 3, 2};
byte colPins[COLS] = {8, 7, 6};

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS,
COLS );

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

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

  if (key != NO_KEY) { Serial.println(key); }
}
```

عدد الصفوف

الحروف و الرموز

أمر يقوم بتفعيل  
استقبال الأرقام من  
لوحة الأرقام

بعد الانتهاء من كتابه الكود قم بتشغيل serial monitor واضغط على الأرقام في الKeypad وانظر ماذا سيحدث !!

## مصادر إضافية للمعلومات:

مكتبه لوحة الأرقام و الأوامر البرمجيّه :

<http://www.arduino.cc/playground/Code/Keypad>

كيف تصنع قفل الكتروني بأستخدام اردوينو و السيرفو موتور:

<http://www.instructables.com/id/Access-control-with-Arduino-Keypad-4x4-Servo>

كيف توصل لوحة الأرقام بأردوينو عن طريق 3 أسلاك فقط بدلا من 7 أسلاك:

<http://www.instructables.com/id/Arduino-3-wire-Matrix-Keypad>

عمل نظام امنى (مفتاح الكتروني شامل) مكون من اردوينو ميجا و شاشه و لوحة ارقام و

قفل الكتروني (سيرفو موتور):

<http://www.instructables.com/id/Password-Lock-with-Arduino>

## أستخدام المرّجِل Relay مع اردوينو



يعتبر الريلاى من اهم العناصر المستخدمه فى التحكم الالكترونيى بأنواعه المختلفه

## ما هو ذلك العنصر؟ وكيف يمكن استخدامه؟ وماهي تطبيقاته؟ إيجابياته؟ سلبياته؟

### مم يتكون؟

الريلاي هو عنصر ميكانيكي/إلكتروني، ويمكننا تخيله على شكل مفتاح أو زر كهربائي، داخلياً يتكون من جزئين رئيسيين:

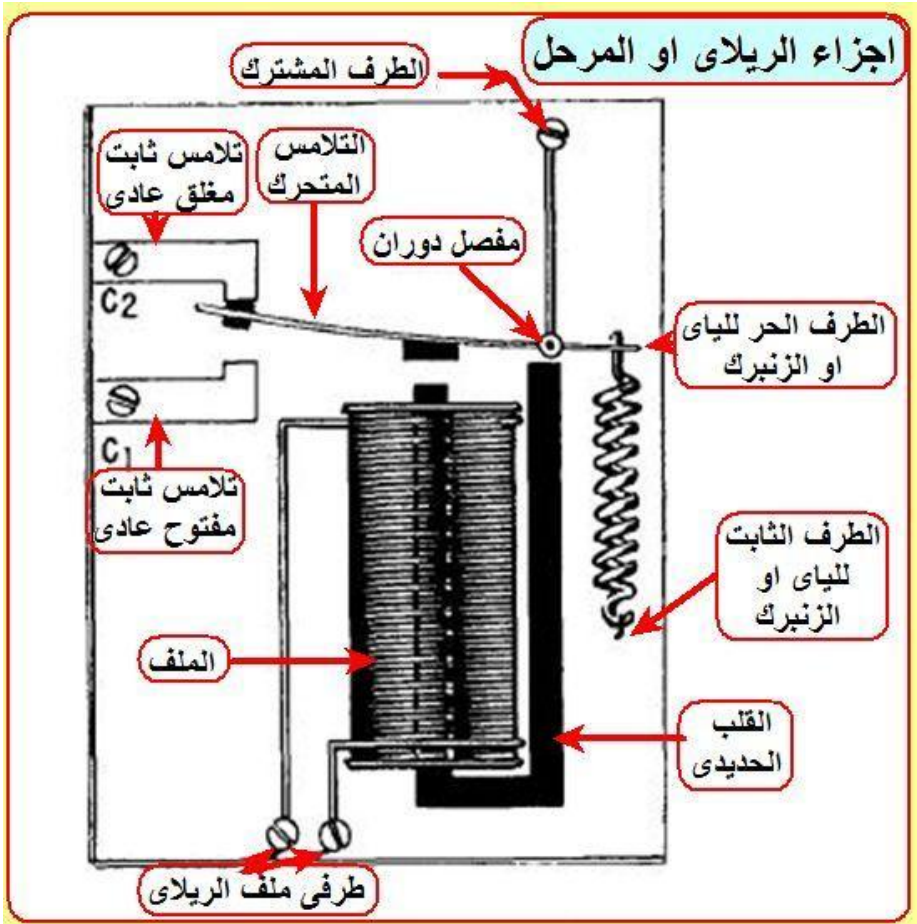
**الأول:** سلك ملفوف حول قالب حديدي، وفي مقدمة ذلك القالب يتوضع **الجزء الثاني:** وهو لسان أو قطعة معدنية مستطيلة الشكل تقع أمام القالب/الملف وهي بمثابة المفتاح فهي تحتوي على تماسات معدنية يتم من خلالها توصيل حمل كهربائي ليعمل الريلاي على فصله أو تشغيله بحسب وضع الملف في الريلاي.

### ماهو الرمز الإلكتروني للريلاي في الدارات الإلكترونية؟

يرمز للريلاي بالشكل التالي: حيث الجزء الأيسر هو الملف والأيمن هو رمز المفتاح الكهربائي



## التركيب من الداخل



## كيف يعمل؟

عندما يتم تطبيق فرق جهد على طرفي الملف الداخلي للريلاي، سيمر تيار في الملف ليتحول بواسطته الملف لمغناطيس كهربائي مولداً مجالاً مغناطيسياً سيقوم بدوره بجذب

ذلك اللسان أو القطعة المواجهة للملف بحيث تغلق التماسات الكهربائية، وعند فصل الجهد المطبق سيتلاشى التيار تدريجياً ليختفي ذلك المجال المغناطيسي، وهناك زنبك سيقوم بإعادة اللسان لوضعه الطبيعي وفصل التماسات وفتح المفتاح الذي أغلق.

إذا بكل بساطة فالريلاي هو مفتاح كهربائي ميكانيكي، ما إن يتم تطبيق جهد على ملفه، يبدأ بالعمل لجذب تماسات معدنية ستعمل على غلق أو فتح دائرة خارجية متصلة بها.

### ومافادته إذا؟ ألا يمكننا استخدام مفتاح كهربائي مباشرة؟

تخيل معي أننا نحتاج ان نوصل اردوينو بحمل كهربائي بعمل على فرق جهد متردد قيمته 110 فولت او 220 فولت. منطقياً، لا يمكننا تشغيل ذلك الحمل من خلال اردوينو لأن أقصى جهد يمكن توليده من اردوينو هو 5 فولت فقط،



وهنا يأتي دور الريلاي فهو سيقوم بعملية تشغيل لأي حمل ذا استهلاك كبير للطاقة من خلال فرق جهد صغير، والسر يمكن وراء تشغيل الملف بجهد صغير مثل **5 فولت** لنقوم بتشغيل المصباح الذي سيتم توصيله على أطراف التماسات التي سيجذبها الملف داخل الريلاي.

**إذا للريلاي** فائدة عظيمة كونه سيعمل على ترحيل جهد وتيار صغير لقيادة حمل كبير. وأيضاً ميزة الريلاي تكمن في قدرته على العزل التام ما بين دائرة المصدر والحمل “الخرج”، فهما معزولان تماماً ولكل واحد منهما نقطة أرضية مرجعية مختلفة عن الأخرى،

وهذا له فائدة كبيرة في منع انتقال التشويش والجهود العابرة لدارتنا وبمعنى أصح حمايتها. أخيراً، يمكن استخدام الريلاي في التحكم عن بعد، أي وضع الريلاي بالقرب من الجهاز ومد سلك للدراة التي تصدر الأوامر.

## عيوبه:

لكل عنصر إلكتروني فوائد وسلبيات ، وكذلك الأمر بالنسبة للريلاي، فأهم سلبياته هي:

- بما أنه يعتمد على الملف -وهو حثي- فلذلك ستتولد فيه قوة دافعة كهربائية عكسية عند مرور تيار به ( تعرف بقاعدة لنز ) عند التوصيل وعند الفصل مما قد يؤدي لعطب الدارة المتصلة به. إلا أنه يمكننا تجاوز هذه المشكلة البسيطة بواسطة وضع عنصر الدايدود بين طرفي الملف وذلك لمنع عودة تلك القوة الدافعة العكسية .
- حدوث ارتدادات ميكانيكية عند كل تحويل من وضع الفصل الى وضع التوصيل أو العكس. مما قد يؤدي لعطب الحمل
- يحتاج إلى دائرة موائمة لكي يعمل جيداً مع الأنظمة الإلكترونية . وعادة تكون هذه الدارة هي مكونة من ترانزستور
- العمر الافتراضي للريلاي صغير نسبياً خصوصاً في الدوائر التي تتطلب عدداً كبيراً من مرات الوصل والفصل وعادة ما تذكر قيمة عدد مرات الفصل والوصل القصوى لكل ريلاي في ملف المواصفات الفنية



## دارة عملية لاستخدام الريلاي مع الدارات الإلكترونية

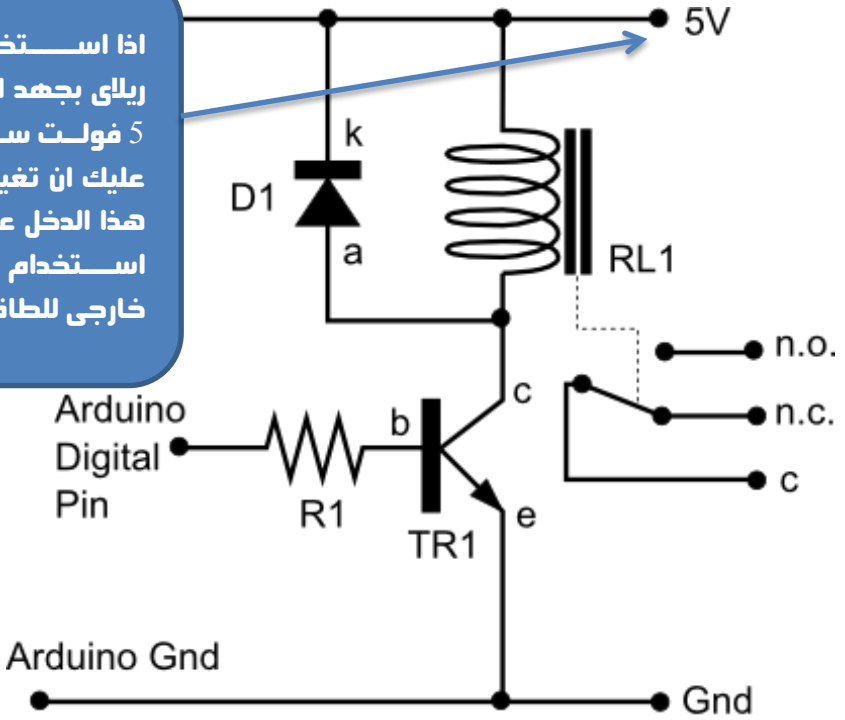
عند اختيار أي ريلاي فلا بد من الانتباه للأمور التالية كي تتمكن من الاختيار العلمي الصحيح ومعرفة البدائل التي يمكن استخدامها في حالة عدم توفر الريلاي المناسب

- فرق الجهد اللازم لتشغيل الملف الداخلي للريلاي: فمثلا لو كان لدينا دارة تعمل على فرق جهد 5 فولت فيجب اختيار ريلاي له فرق جهد لملفه هو 5 فولت، وفي حالة تعذر ذلك يمكننا استخدام جهد أكبر بعد استخدام دارة مواعمة سبق أن طرحتها في تدوينة الترانزستور.
- أقصى تيار لازم لتشغيل ملف الريلاي: وغالبا هذه القيمة لا تعطى بشكل واضح بل يتم ذكر مقاومة الملف بدلا منها، ومن خلال قسمة فرق جهد الملف على مقاومته يتم حساب التيار الأعظم. وهي قيمة مهمة جدا كي نعرف هل يمكننا تشغيل ووصل الريلاي بدارتنا مباشرة أو لا بد من وضع دارة مواعمة
- جهد وتيار مرحلة الخرج أو التماسات، وهنا يجب معرفة ما يحتاجه الحمل ومقارنته بالريلاي وينبغي دائما أن يتم اختيار قيمة تيار عظمى تزيد عن القيمة المطلوبة بنسبة 5 إلى 10%
- العمر الافتراضي لتماسات الريلاي وعادة ما تعطى بملايين المرات

## نتقل الآن للدارة العملية سنحتاج لهذه الدائره المكونات التاليه:

- ريلاي يعمل على جهد 5 فولت (يمكن استخدام 9 او 12 فولت)
- دايود
- مقاومه 1 كيلو اوم
- ترانزستور من نوع 2N2222

اذا استخدمت اي  
ريلاي بجهد اعلى من  
5 فولت سيتوجب  
عليك ان تغير قيمه  
هذا الدخل عن طريق  
استخدام مصدر  
خارجي للطاقة



TR1 - BC548, 2N2222, 2N3704 transistor

D1 - 1N4004, 1N4007 diode

R1 - 1k  $\Omega$  resistor

RL1 - 5V relay

c - collector

b - base

e - emitter

n.c. - normally closed

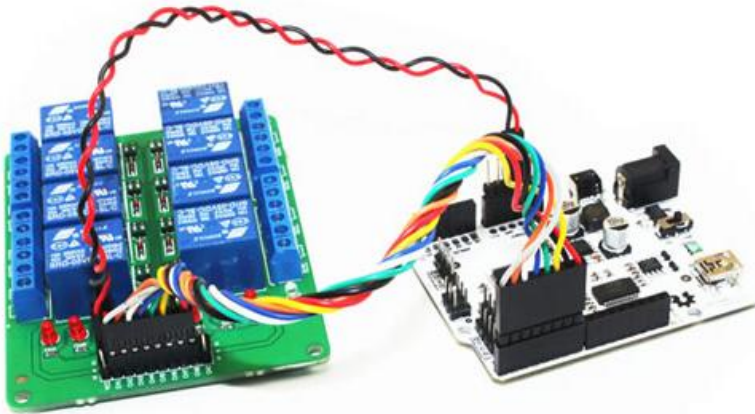
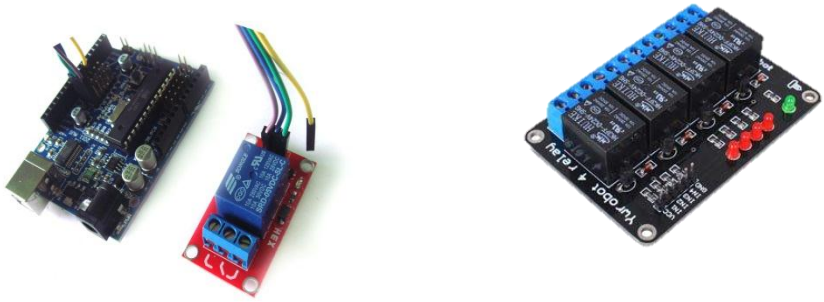
n.o. - normally open

c - common

a - anode

k - cathode

تتوفر في الأسواق العديد من دوائر المُرَحِلَات الجاهزه للأستخدام مع Relay Boards  
 اردوينو كما تتوفر بأحجام و اعداد مُرَحِلَات مختلفه مثل الصور التاليه:



في حال كان مشروعك يحتاج تشغيل احمال بفرق جهد كبير مثل 220 فولت فاني  
انصح بشده بأن تشتري بورده جاهزه بالحجم المناسب للمشروع الخاص بك وذلك لأن  
هذه البورده تصنع بجوده ودقه عاليه وهو ما تحتاجه في المشاريع التي تعمل بجهد  
كبير تجنباً لحدوث اي اخطاء قد لا يُحمد عقباها.

### مصادر اضافيه للمعلومات:

هناك مشروع جميل يشرح كيفيه استخدام الريلاي مع اردوينو لتجهيز منفذ للطاقه بجهد  
220 فولت يعمل من خلال اشارات التحكم الصاده من اردوينو

<http://www.sparkfun.com/tutorials/119>

ملحوظه: مرجع الصور و المعلومات المذكوره عن المُرجل **relay** في هذا الفصل

<http://muslimlead.com/?p=2121>

و تحتوى المدونه ايضا على العديد من الدروس الرائعه حول نظريات عمل العديد من  
المكونات الالكترونيه مثل الدايود، الثنائي المشع للضوء، المكثف، الملف، الترانزستور

يمكنك الضغط على اي اسم من اسماء العناصر المذكوره ليتم نقلك الى موضوع يشرح

اساسيات العنصر بالتفصيل ☺

## ملاحظات شخصية :

هذه الصفحة مخصصة لكتابه ملاحظتك الشخصية عن الفصل السابع :

اكتب ملاحظتك  
هنا

## من أقوال العظماء :

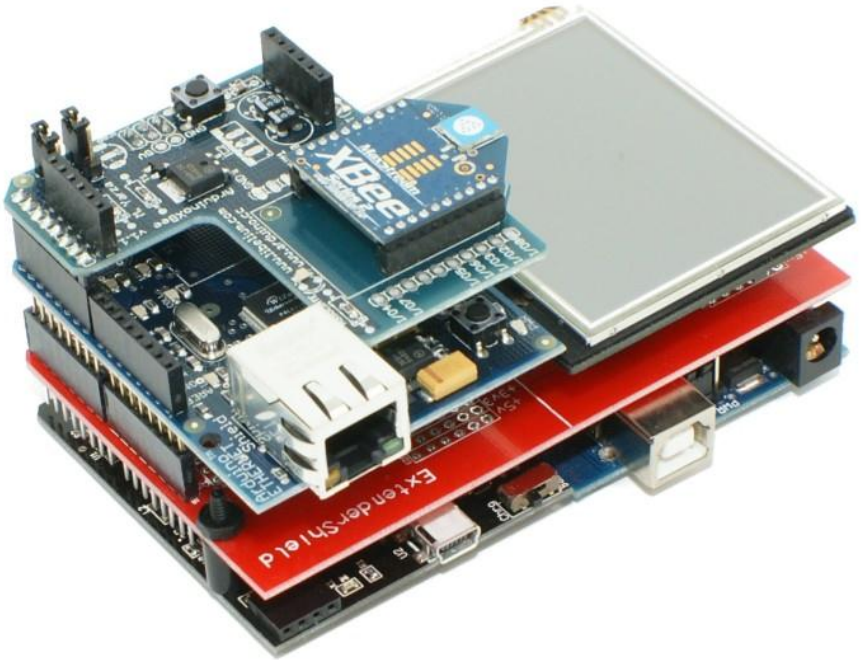
سُبْحَانَكَ اللَّهُمَّ خَيْرَ مَعْلَمٍ  
عَلَّمْتَ بِالْقَلَمِ الْقُرُونَ الْأُولَى

أَخْرَجْتَ هَذَا الْعَقْلَ مِنْ ظُلُمَاتِهِ  
وَهَدَيْتَهُ النُّورَ الْمُبِينَ سَبِيلًا

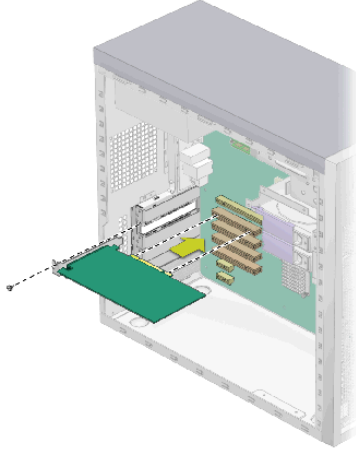
أمير الشعراء أحمد شوقي

# الفصل الثامن

## أغويه اردوينو الجاهزة Arduino Shields

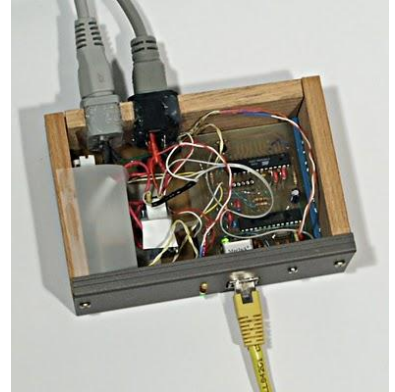


**من اكثر المميزات المثيرة في اردوينو هي توافر عدد كبير من الاغطية التي يمكن تركيبها مباشرة على البورداة المختلفة مثل Arduino uno والإصدارة الأعلى في الإمكانيات وقوه المعالجة Arduino Mega**



تعمل الاغطية بنفس فكره كروت PCI الموجوده في الحاسب الآلى ، على سبيل المثال اذا لم توفر اللوحه الام Motherboard كارت شبكه للدخول على الانترنت فأنت تستطيع بكل بساطه ان تشتري كارت شبكه network interface card و تقوم بتركيبه و الدخول على الانترنت فوراً .. هكذا تعمل اغطية اردوينو

اذا كان لديك خبره سابقه مع المتحكمات الدقيقه فلا بد انك تدرك مدى الصعوبه و المجهود المبذول حتى تقوم بتوصيل المتحكم الدقيق بالانترنت او شبكه محليه (حيث ستقوم ببناء وحده Ethernet من الصفر بنفسك) و سيستهلك ذلك الكثير من الوقت و المجهود ، اما في حاله اردوينو كل ما عليك فعله ان تشتري الغطاء جاهز و تركيبه مباشرة على البورده الخاصه بك و تبدأ في برمجه مشروعك بكل سهوله



**تتوفر العديد من الاغطية و سنستعرض منها التالي:**

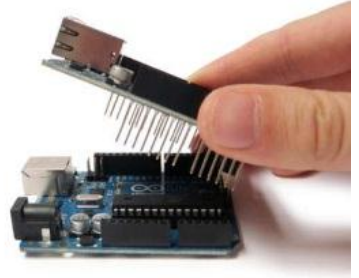


## • غطاء اردوينو للشبكة السلكية Arduino Ethernet Shield

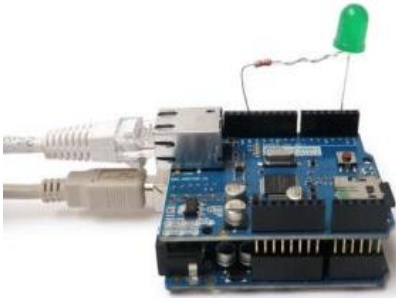


**يوفر** ذلك الغطاء امكانيه توصيل اردوينو بشبكات الحاسب الآلى و الانترنت عن طريق توصيل كابل شبكه من نوع CAT5 مثل المستخدم فى الحاسب الآلى و يمكنك استغلال ذلك الغطاء فى عمل مشاريع التحكم عن بعد بأستخدام الشبكه المحليه او

الانترنت فيمكنك مثلا توصيل عدد من الاجهزه الالكترونيه بالاردوينو و التحكم بها من اى مكان فى العالم عن طريق الانترنت ، وهناك بعض الاصدارات الحديثه من هذا الغطاء توفر ايضا امكانيه تركيب كارت ذاكره لتخزين الملفات و يمكن استغلال ذلك فى

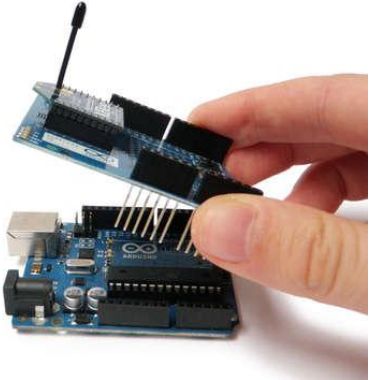


بعض المشاريع التى تحتاج تسجيل قيم بصوره دوريه مثل مشروع لتسجيل درجات الحراره كل فتره زمنيّه معينه و تخزين القيم على كرت ذاكره ثم ارسالها عن طريق الانترنت الى صاحب المشروع



## • غطاء الشبكات اللاسلكية Xbee Shield

يوفر هذا الغطاء نفس الوظائف الموجوده في Ethernet Shield لكن بصوره لاسلكيه حيث يمكنك من توصيل الاردوينو باى شبكه لاسلكيه فى نطاق يقارب 100 متر (يختلف النطاق بناء على قوه الاشاره اللاسلكيه)

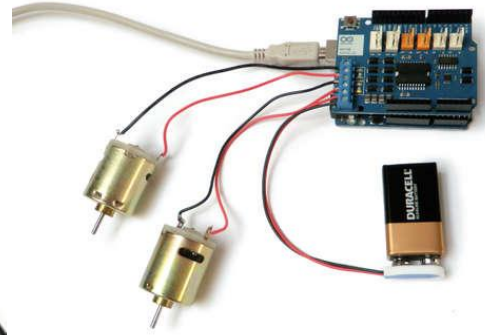
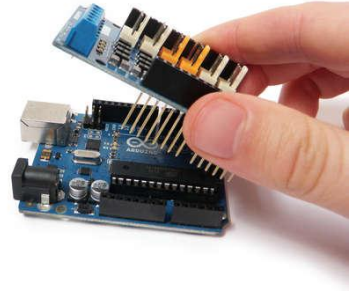


## • غطاء المحركات (الموتور) Arduino Motor Shield

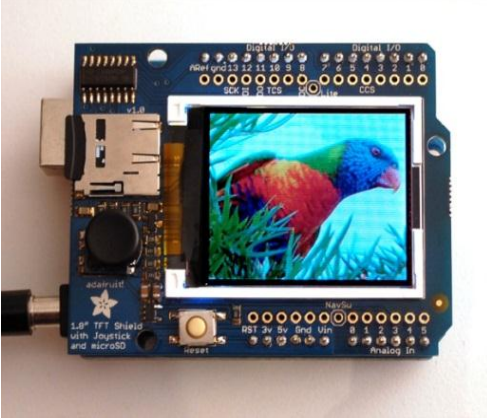


يستخدم هذا الغطاء في توصيل المحركات من نوع DC Motor, Servo Motors, Stepper Motors حيث يمكنك توصيل 3 محركات في نفس الوقت (ملحوظه: بعض

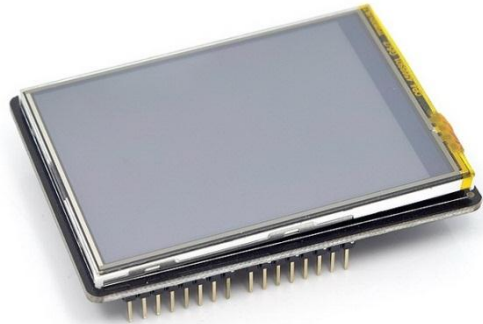
اصدارت هذا الغطاء تسمح بتوصيل محركين فقط) و يمكنك استغلال هذا الغطاء في المشاريع التي تحتاج محركات مثل مشاريع الإنسان الآلي Robots و ماكينات CNC



## • غطاء الشاشة الملونه التي تعمل باللمس Arduino colored Touch screen



يستخدم ذلك الغطاء في توفير شاشة ملونه تفاعليه تعمل بالاستجابة للمسات الأصابع و يمكن استخدامها في المشاريع التي تحتاج إلى وسيله لعرض صوره او بيانات معقده و التي لا تكفي الشاشات lcd ذات اللون الواحد لعرضها، و تتوفر ذلك الغطاء بأحجام و مقاسات مختلفه تبدأ من مساحه 2 انش مربع إلى 4 انش مربع (الإنش هو وحده قياس تساوى 2.5 سنتي متر)



هناك قائمه كبيره بأسماء و مواصفات أغطيه اردوينو المختلفه تجدها على الموقع التالي:

<http://shieldlist.org>

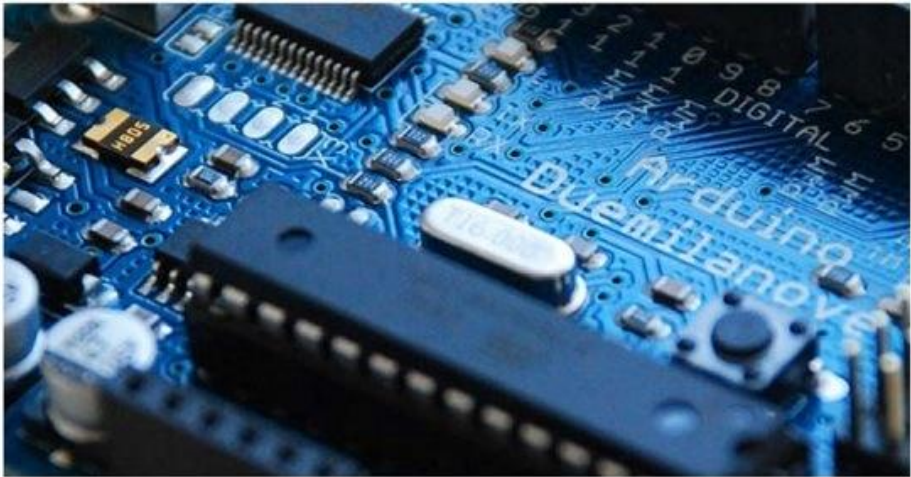
## ملاحظات شخصية :

هذه الصفحة مخصصة لكتابه ملاحظتك الشخصية عن الفصل الثالث :

اكتب ملاحظتك  
هنا

# الفصل التاسع

اردينو و تطور ثورة المتحكمات الدقيقة مفتوحة المصدر من  
عائلته AVR الى ARM Cortex انتقائاً بفPGA



**تبدأ** القصة في عام 2005 في مدينة ايفريا **Ivrea** الايطاليه عندما أجمع فريق من مهندسي الألكترونيات ليناقدشوا سبل تعليم طلاب الجامعه و المعاهد التقنيه علوم المتحكمات الدقيقه او ما يعرف بأسم الـ **Microcontrollers** ، كان الهدف من اللقاء ايجاد أسلوب سهل و غير مكلف لأدخال هذا النوع من العلوم في الدراسه الجامعيه و يكون متاح للهواه في ذات الوقت دون ان تكون هناك قيود أو تراخيص مكلفه يحتاجون لشرائها لأستخدام البرمجيات التي كانت تستخدم في برمجته المتحكمات الدقيقه في هذا الوقت .. من هنا تبدأ رحله نشأه اردوينو **Arduino**

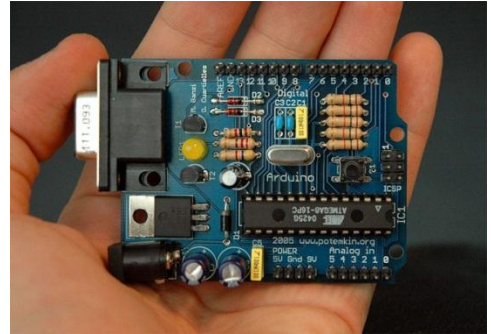


قام كل من " ماسيمو بانزي Massimo Banzi " بالتعاون مع " دايفيد كوارتيليس David Cuartielles " و جاينلوكا مارتينو Gianluca Martino " بأطلاق مشروع أردوين ايفريا **Arduin of Ivrea** و تمت تسميه المشروع بأسم اشهر شخصيه تاريخيه في المدينه و كان الهدف الأساسى للمشروع هو

عمل بيئه تطوير للمتحكمات دقيقه بصوره مفتوحه المصدر 100 في المئه و تضمن هذا المشروع عمل بيئه تطوير برمجيه للمتحكمات الدقيقه **Integrated Development Environment** و تكون مجانيه في ذات الوقت كما تضمن عمل لوحات تطوير **Development Boards** صغيره الحجم بتكلفه بسيطه يمكن للطلاب و الهواه التقنين تحمل سعرها.

**بدءاً تطوير** المشروع اعتماداً على مشروع سابق اسمه Wiring Platform والذي قد نجح في تحقيق بعض أهداف مشروع اردوينو الأساسيه مثل المصدر المفتوح لكن لم يصل الى المستوى الذي تمناه فريق التطوير من الناحيه البرمجيه والتصميم العتادي Hardware Design لذلك تم البدء في العمل على تطوير اللغه البرمجيه المستخدمه في مشروع Wiring وجعلها أسهل وأفضل كما تم دمج بعض التقنيات المستخدمه في لغه [Processing](#) وسميت هذه اللغه الجديده بأسم **Arduino C** و هي مشابهه جداً للغه البرمجيه C++ من ناحيه طريقه كتابه الأوامر وتنسيق الأكواد.

تم إطلاق أول لوحه تطويره لأردوينو في اواخر عام 2005 وأتمدت على شريحه ATmega168 من العائله المشهوره AVR والتي تنتجها شركه Atmel للشرائح الألكترونيه، وسميت هذه اللوحه بأسم **Arduino Serial V.1**



بعد إطلاق اللوحه قام مؤسسى المشروع بنشر كل ما يتعلق بأردوينو برخصه مفتوحه المصدر على موقع [arduino.cc](http://arduino.cc) والذي كان يعنى أن أى فرد فى العالم يمكنه الاطلاع والتعديل على التصميمات الهندسيه والشفرات المصدرية Source Codes لكل من بوردرات اردوينو المختلفه Arduino Boards والبرمجيات Arduino IDE وتم نشر كل هذا مجاناً.



وما ان تم نشر المشروع حتى توالى الكثير من التطويرات على التصميمات الهندسيه للوحات الألكترونيه و بيئه التطوير من مهندسين و هواه مختلفين حول العالم و تم إصدار العديد من اللوحات الأحدث مثل:

**Arduino UNO , Arduino Mega, Arduino lilyPad, Arduino Duemilanove, Arduino Fio**

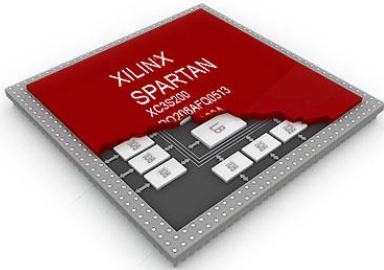
وكان القاسم المشترك فى كل الأصدارات السابقه هى انها تعتمد على شرائح المتحكمات الدقيقه من عائله AVR التى تنتجها شركه Atmel و حقق المشروع نجاح باهر حتى وصل عدد اللوحات الألكترونيه المباعه الى 300,000 قطعه فى مختلف دول العالم.

**فى عام 2009** تعرض أردوينو لنقله نوعيه جديده جائت على يد فريق LeafLabs و الذى قام بتطوير لوحه أردوينو جديده تسمى The Maple board تعمل بشرائح ألكترونيه من عائله ARM Cortex و التى نقلت أردوينو الى افق جديد من السرعه و القوه حيث تتميز جميع شرائح ARM باحتوائها على معالجات عاليه السرعه تمتلك قوه معالجه تبدأ من 72 ميجا هرتز و تصل فى بعض الإصدارات الى 1.5 جيجا هرتز .



وبذلك أستطاع فريق LeafLab الخروج من حدود امكانيات شرائح AVR و استخدام الجيل الجديد من ARM Cortex مع الاحتفاظ ببلغه أردوينو البرمجيّه التى تتميز بالسهوله و خلوها من التعقيدات و فى ذات الوقت الاحتفاظ بعامل السعر حيث بلغ سعر Maple Board حوالى 34 دولار مع تكلفه الشحن .

**و بالرغم من كل** التطورات السريعة في لوحات أردوينو المختلفة الا ان التطوير لم يتوقف عن حدود ARM Cortex بل وصل الى شرائح " مصفوفات البوابات المنطقية القابلة للبرمجة " (Field-Programmable Gate Arrays (FPGA) في عام 2010 تم نشر مشروع مفتوح المصدر اسمه Papilio Arduino والهدف منه هو صناعه لوحه اردوينو تعمل بشرائح من نوع FPGA ويمكن برمجتها بلغه Arduino C



**تتميز شرائح** FPGA بالسرعة الفائقة في اداء عمليات

المعالجه بصوره متوازيه وفك تشفير البيانات Decoding لذلك تستخدم في الكثير من التطبيقات الصناعيه التي يكون فيها عامل الوقت امرا حاسماً في تصميم المنتج النهائي لكن يعيب هذه الشرائح عدده أمور تجعلها مقتصره على المحترفين و الشركات فقط،

الأول: تكلفه لوحات التطوير الخاصه بها و التي يبدأ سعرها غالبا من 100 دولار و اكثر،  
ثانيا: صعوبه برمجته الشرائح حيث تستخدم لغات برمجيه معقدته في تشغيل هذه الشرائح مثل VHDL و لغه Verilog .

جاء مشروع Papilio Arduino ليحل هذه المشكلات و يقدم لوحه تطوير جديده الى عائله لوحات اردوينو و تم اصدار Papilio Arduino IDE و هي نسخه معدله من برنامج Arduino IDE لجعل لغه اردوينو



تناسب مع شرائح FPGA كما تم اصدار لوحات papilio boards للبيع من منتصف عام 2010 بأسعار تبدأ من 40 دولار فقط و يمكنك الاطلاع على المزيد من المعلومات عن لوحات papilio من الموقع التالي <http://papilio.cc> بعد مشروع Papilio Arduino اصبحت عائله اردوينو تحتوى على لوحات تطويرات تعمل تقريبا بكافه التقنيات التى تم ابتكارها مثل AVR و شرائح ARM Cortex و حتى شرائح FPGA

وفى النهايه يمكننا القول أن مشروع اردوينو صنع أكبر مجتمع تقنى من المهندسين و الفنيين و الهواه يعملون على تطوير الافكار و المشاريع المتعلقة بالتحكم الآلى بصوره تشاركيه و تفاعليه حول العالم اعتماداً على استخدام لوحات الكترونيه بتقنيات مختلفه و مع ذلك يتم برمجتها جميعا بلغه برمجيّه واحده و مجانيه و متاحه للجميع.

انه مجتمع جديد و مختلف .. حقاً انها ثوره الكترونيه ....



# FABLAB EGYPT

□ فاب لاب مصر او ما يعرف بأسم ورشه التصنيع الشخصي

هو مكان مُصمم لتحويل الافكار من مجرد خيال الى واقع ملموس و بناء مجتمع من مصنعي الافكار و المنتجات من الهواه و المتحرفين فى مجالات مختلفه مثل: (الالكترونيات - الهندسه الميكانيكيه - هندسه الحاسب - التصنيع الرقمى - الفنون)

و يوفر المكان العديد من الادوات التقنيه التى تساعدك على تصنيع افكارك مثل:

- ماكينه القطع بالليزر LASER cutter machine
- طابعه ثلاثيه الابعاد 3D printer
- ماكينه حفر بالتحكم الرقمى CNC machine
- بورداات اردوينو المختلفه و مكونات الكترونيه منوعه

و العديد من الادوات الاخرى ....

يهدف الفاب لاب الى توفير بيئه تشاركيه و تعليميه فى ذات الوقت تساعد الافراد على الابداع و اطلاق العنان لافكارهم و خيالهم و مشاركه هذه الافكار مع المجتمع.

كما يوفر وورش و دورات تدريبيه للافراد مثل:

- ورش تدريب عمليه لاستخدام اردوينو Arduino workshops
- ورش تدريبيه للاطفال لتعلم الالكترونيات Young Fab.Academy

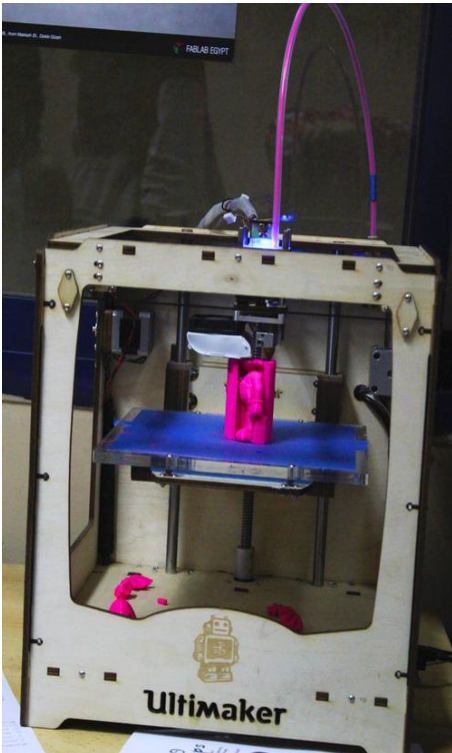
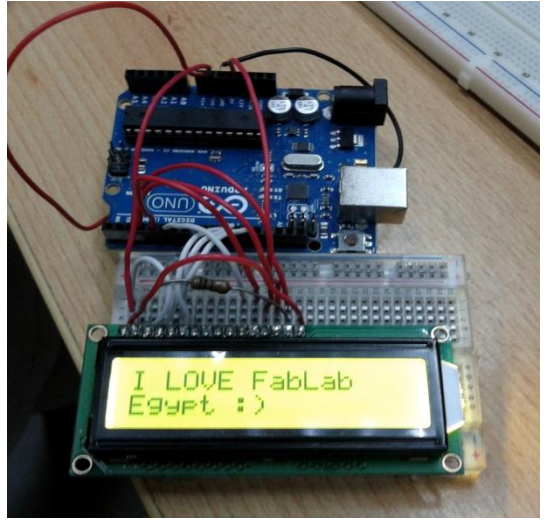


اذا كانت لديك فكره و تريد ان تنفذها و تحتاج الى بعض المساعدة او تعاون من اشخاص اخرين او تحتاج الى دورات تدريبيه فى اردوينو فيمكنك زياره فاب لاب مصر يوميا من الساعه 12 ظهرا حتى 8 مساءً جميع الايام

<http://fablab-egypt.com>

<https://www.facebook.com/fablab.egypt>





# ملحق المراجع

## المصادر التعليميه التي اعتمدت عليها في كتابه محتوى الكتاب:

- <http://www.instructables.com/id/Arduino-Projects>
- <http://www.oomlout.com>
- <http://ladyada.net>
- <http://bildr.org>
- <http://hlt.media.mit.edu/?cat=5>
- <http://circuit-projects.com/microcontroller>
- <http://www.eeweb.com>
- <http://www.lvl1.org>
- <http://www.coolcircuit.com>
- <http://www.instructables.com/id/Breadboard-How-To>
- <http://blog.makezine.com/2009/12/11/arduino-shields-open-source-hardwar/>



## لمن يريد استخدام اردوينو مع الماتلاب MATLAB

- <http://www.mathworks.com/academia/arduino-software/arduino-matlab.html>
- <http://arduino.cc/playground/Interfacing/Matlab>

## مشاريع رائعة باستخدام اردوينو يمكنك ان تصنعها بنفسك:

- <http://fritzing.org/projects/>
- <http://www.instructables.com/id/Arduino-Projects>
- <http://www.ladyada.net/make/boarduino/index.html>
- <http://www.ladyada.net/make/mshield/index.html>
- <http://www.shapeoko.com/wiki/index.php/About>
- <http://www.ladyada.net/make/monochron/index.html>
- <http://www.ladyada.net/make/bedazzler/index.html>
- <http://www.ladyada.net/make/mintyboost/index.html>
- <http://dangerousprototypes.com/forum/viewtopic.php?f=56&t=2892#p28410>
- <http://www.ladyada.net/make/logshield/>
- <http://www.wayneandlayne.com/projects/video-game-shield/>
- <http://excamera.com/sphinx/gameduino/>
- <http://dangerousprototypes.com/2012/03/24/arduino-voice-control-with-easyvr-shield/>
- <http://blog.miniblog.org/p/documentation.html>
- <http://www.doctormonk.com/2011/09/arduino-solar-radio.html>
- <http://j4mie.org/blog/how-to-make-a-physical-gmail-notifier/>
- <http://blog.tinyenormous.com/2008/11/25/gmail-notifier-project-for-dummies/>
- [http://reprap.org/wiki/Arduino Mega Pololu Shield](http://reprap.org/wiki/Arduino_Mega_Pololu_Shield)

## مراجع أخرى: كتب

- 30 Arduino Evil Geniis projects
- Arduino.Internals
- Arduino Projects to Save the World
- Arduino robotics
- Beginning Android ADK with Arduino
- Environmental Monitoring Arduino
- Getting Started with Arduino (Second Edition)
- Shrinkify Your Arduino Projects -burn arduino to ATiny (Video)
- Make a mind-controlled arduino robot
- Make\_Magazine\_Volume\_25
- Making Things Talk Using Sensors (Second Edition)
- Making Things see hear and feel your world (Second Edition)
- O'Reilly - Arduino Cookbook
- O'Reilly-Make - Arduino Bots and Gadgets
- Oreilly Verlag - Arduino Physical Computing
- Practical Arduino Cool Projects For Open Source Hardware
- Programming Interactivity
- Programming Your Home Automate with Arduino, Android, and Your Computer
- The Art of Electronics 2nd edition Complete - Horowitz and Hill