



القرية الهندسية  
**ENGINEERING VILLAGE**  
ELECTRONICS & MECHANICA  
للإلكترونيات والميكانيكا

# احترف الأردوينو

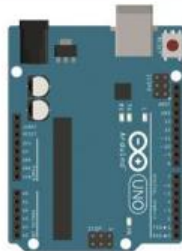
في 10 أيام

[YOUTUB](#)

[INSTAGRAM](#)

[TWITTER](#)

[FACEBOOK](#)



كتاب عملي ، تطبيقي ، مبسط



هذا الكتاب حر ومخصص للتوزيع المجاني في نسخته الإلكترونية ويمكنك أن تنشره بشكل كامل أو جزء منه بشرط ذكر المصدر.

جميع حقوق الطبع محفوظة لدى مؤسسة القرية الهندسية ولا يجوز استخدام الكتاب بأي صورة تجارية إلا بإذن خطي من المؤسسة وكل من يخالف ذلك فإنه يتحمل المساءلة القانونية.

الطبعة الثانية 2015

## فريق العمل

الإشراف العام  
فهد السيابي

فريق الإعداد  
أسعد الرميضي  
علي اللواتي  
مهند الرحيلي

التدقيق والمراجعة  
وائل المغيري

التصميم والإخراج الفني  
فريق توباز  
@TopazTeam

# كلمة شكر

تتقدم إدارة مؤسسة القرية الهندسية بالشكر الجزيل والموصول لكل فرد كان له صلة وبصمة مميزة ساهمت بإثراء هذا العمل وإنجازه على أتم وجه.

إدارة القرية الهندسية





## المقدمة

يتفنن المهندسون والعلماء والمبرمجون حول العالم في طرق تبسيط وتسهيل تدريس العلوم ومبادئ الهندسة من خلال توظيف التكنولوجيا الحديثة في العملية التعليمية التعليمية.

ومع ثورة البرمجيات والتقنيات الحرة والمفتوحة المصدر، أصبح لزاما علينا أن نواكب العالم في هذا المجال وأن نستفيد من هذه الطفرة في تطوير قدرات وإمكانات الشاب العربي لكي يشارك بفاعلية في التنمية والإنتاج.

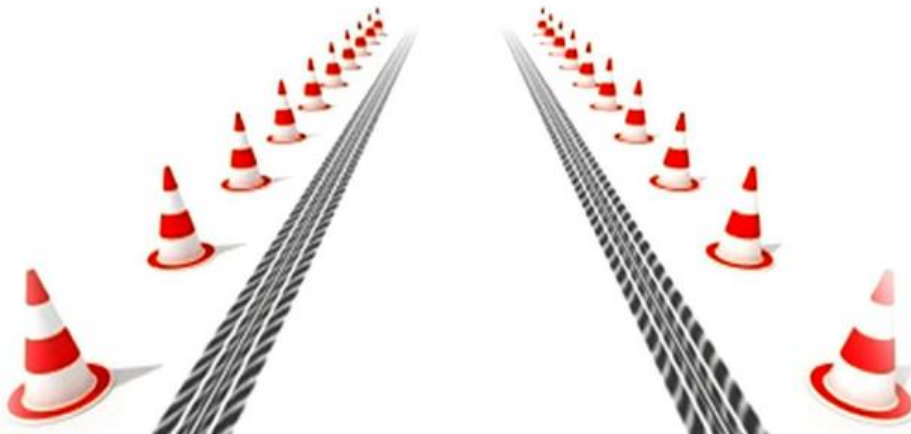
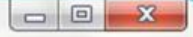
الهدف هو الانتقال من العقلية الاستهلاكية السائدة إلى العقلية الإنتاجية من خلال تسهيل تعلم واكتساب المهارات البرمجية والهندسية إلى أبعد الحدود.

هذا الكتاب هو مدخلك إلى عالم الأردوينو الشيق الذي يسمح لك بتنفيذ العديد من الأفكار والمشاريع الإلكترونية المفيدة. وقد راعينا فيه تبسيط المعلومة والتركيز على الجانب التطبيقي من خلال توضيح كل فكرة بتجربة عملية، وكما قالوا علمني كيف أفعل وسوف أتذكر. نرجو لكم وقتا ممتعا ومفيدا بين طياته !



sketch\_jan23a

- 70 ----- المقاومة الضوئية LDR -
- 72 ----- الإنارة التلقائية -
- 74 ----- حساس الصوت -
- 76 ----- جهاز مستوى الماء -
- 80 ----- مقياس درجة الحرارة -
- 82 ----- Joystick التحكم -
- 84 ----- ما هي خاصية التعديل الرقمي PWM ؟ -
- 85 ----- مكثبات الأردوينو -
- 86 ----- محرك السيرفو Servo -
- 88 ----- تحديد عدد الدورات for -
- 92 ----- تشغيل LED بالتدريج من خلال PWM -
- 94 ----- مجسة الموجات فوق الصوتية Ultrasonic -
- 96 ----- جهاز إنذار المسافة -
- 98 ----- لوحة المفاتيح Keypad 4x4 -
- 101 ----- تحدي رقم 3 و 4 -
- 102 ----- تشغيل وإطفاء LED بواسطة لوحة المفاتيح -
- 106 ----- شاشة العرض LCD 16x2 -
- 108 ----- شاشة الأرقام السباعية 7segments -
- 112 ----- مشروع قياس درجة الحرارة -
- 116 ----- محرك DC -
- 118 ----- تحدي رقم 5 و 6 -
- 120 ----- الملحقات -



# START

هل أنت جاهز للانطلاق إلى عالم الأردوينو الخيالي؟

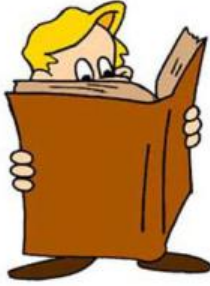


## تمهيد

الأردوينو عبارة عن لوح إلكتروني يسهل عملية برمجة المتحكمات لتسهيل إستخدامها في الابتكارات والمشاريع الإلكترونية المختلفة. وتوجد الكثير من الملحقات المتوفرة في السوق للأردوينو من مجسات صوت ولمس وحرارة وضوء ورطوبة وغيرها. كلها تعمل معا كمنظومة هدفها تسهيل تنفيذ الأفكار والأحلام.

سابقا كان على الطلبة والهواة أن يتعاملوا مع المتحكمات الدقيقة (microcontroller) بشكل مباشر ويقوموا بتركيب دوائر إلكترونية واستخدام لغات برمجية أكثر تعقيدا. أما الآن أصبحت هذه العملية أكثر سهولة عبر لوح برمجي مبسط ولغة برمجة مبسطة بالإضافة إلى مجموعة رائعة من الملحقات والتطبيقات.

ليس من رأى كمن سمع، ولهذا فلن نطيل عليكم. نترككم لتحكموا بأنفسكم عبر كتاب احترف الأردوينو في 10 أيام. فاشمر عن ساعدك أيها المبرمج، وانطلق !



## ما الذي سوف نتعلمه من هذا الكتاب ؟

- 1- البرمجة بلغة Arduino c .
- 2- كيفية دمج الأوامر فيما بينها والخروج بعدة مشاريع .
- 3- كيفية التعامل مع المجسات الكهربائية .
- 4- طرق التعامل مع المحركات الكهربائية .
- 5- معاني المصطلحات الإنجليزية المستخدمة في البرمجة .
- 6- طرق توصيل الأردوينو مع القطع الإلكترونية باستخدام اللوحة التجريبية.



## لمن هذا الكتاب ؟

- 1- طلاب المدارس ( 8 سنوات فما فوق ) .
- 2- معلمي المدارس .
- 3- طلاب الجامعات والكليات الهندسية .
- 4- المدرسين والفنيين للكليات الهندسية .
- 5- هواة الإلكترونيات والبرمجة .
- 6- المهندسين .

## مميزات هذا الكتاب :

- 1- سهولة الإستخدام وتطبيق الدروس والمشاريع .
- 2- شروحات مفصلة ومبسطة.
- 3- شرح جميع الأوامر البرمجية .
- 4- رسومات توضيحية لتركيب الدوائر .
- 5- تسلسل في الدروس والمعلومات .
- 6- مشاريع مختلفة ومتنوعة .
- 7- شرح مجموعة كبيرة من ملحقات الأردوينو .
- 8- استخدام أسلوب الحوار والنقاش .

## نظام تطبيق الدروس :

### النموذج التوضيحي لتركيب الدائرة

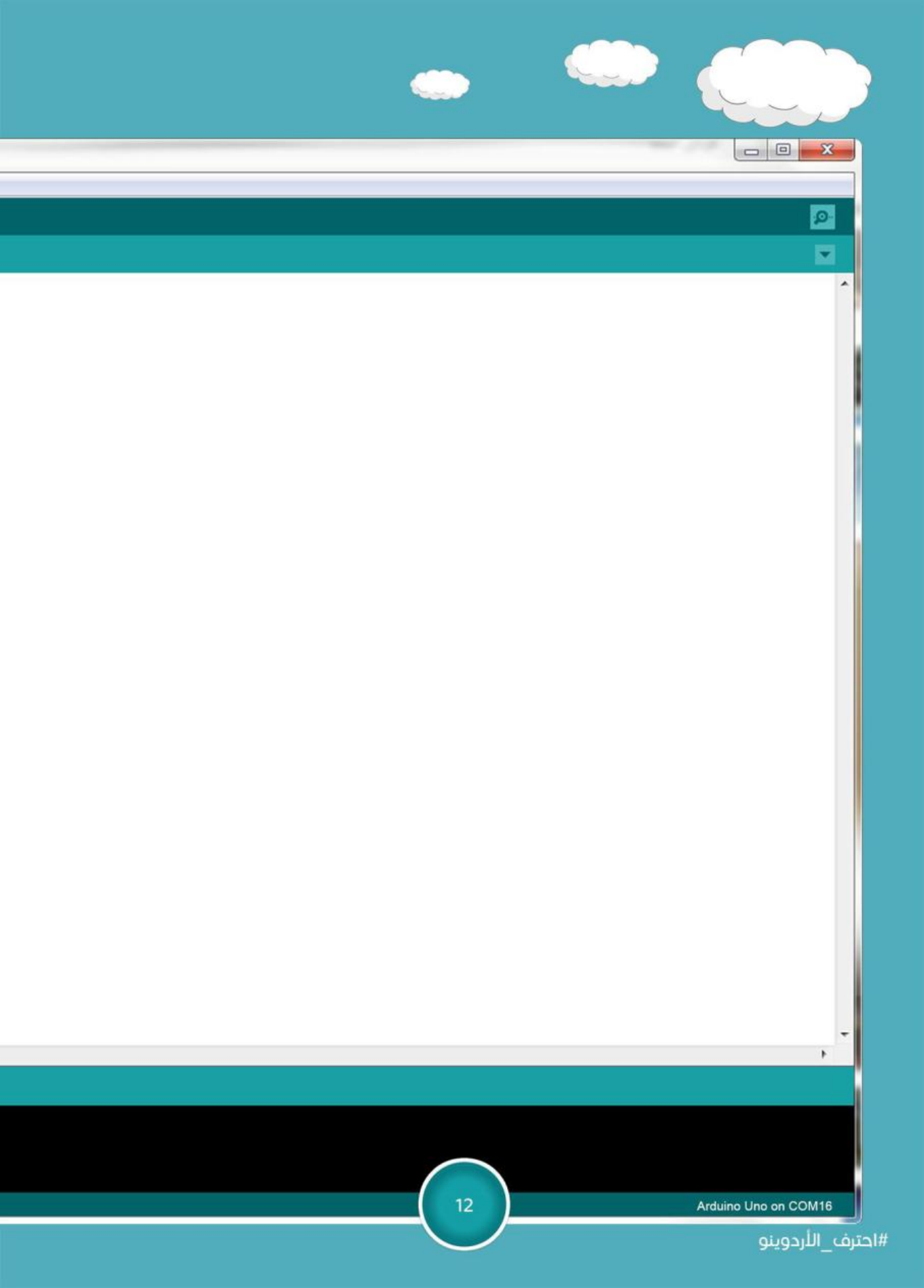
عنوان الدرس

فكرة الدرس

الأوامر

شرح الأوامر

الرسم الفيزيائي





sketch\_jan23a

# مقدمة عن الأردوينو

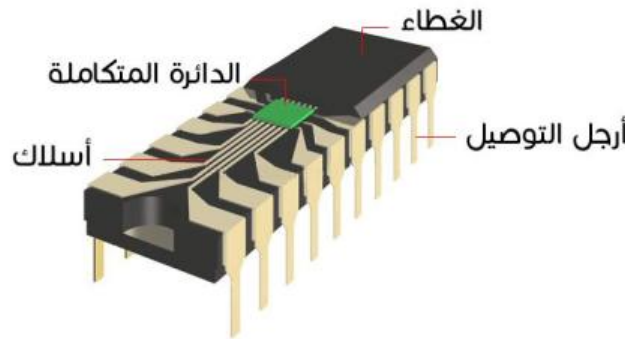


## ما المقصود بالدوائر المتكاملة ؟

هي عبارة عن دائرة إلكترونية مصغرة وهي من ضمن مايعرف بتقنية ( المايكرو ) والتي هي بدورها جزء من الهندسة الإلكترونية، وتعتبر أحدث ثورة في عالم الإلكترونيات.



وتصنع الشريحة الدقيقة من مادة السيلكون تبلغ مساحتها عدة ملليمترات ويطلق عليها (شريحة السيلكون silicon chip) أو رقاقة السيلكون وتحتوي شريحة السيلكون على الآلاف من المكونات الإلكترونية الدقيقة جدا ، مثل الترانزستورات والمقاومات والمكثفات التي تربط معا لتكون دوائر إلكترونية متكاملة، وقد تم إنتاجها لأول مرة بالولايات المتحدة الأمريكية عام 1958.





## ما المقصود بالمتحكمات الدقيقة ؟

كلمة (Microcontroller) تعني المتحكم الدقيق وهي تشمل جميع عوائل شرائح الـ (IC) القابلة للبرمجة وأشهرها نوعين هما الـ (PIC) من شركة Microchip و الـ (AVR) من شركة ATMEL. تتميز هذه الشرائح بقدرتها على تخزين البيانات والبرامج التي يتم كتابتها وتنزيلها في المتحكم من خلال الحاسب الآلي عن طريق جهاز إرسال البيانات للمتحكم للقيام بمهام محددة حسب البرنامج.

أغلب الأجهزة من حولنا تحتوي على المتحكمات الدقيقة، فالمتحكم عبارة عن حاسوب صغير جدا، ويحتوي على كل مكونات الحاسب الاساسي:

**المعالج :** وحدة لتنفيذ الأوامر والعمليات الحسابية.

**الذاكرة:** لتخزين ذاكرة وبيانات البرنامج بشكل دائم او مؤقت.

**Clock:** هامة جدا ولا يعمل المعالج بدونها وهي تحدد سرعة معالجة المعلومات.



## ما هي لوحة الأردوينو ؟

هي عبارة عن لوحة تطوير إلكترونية Development Board تتكون من دائرة إلكترونية مفتوحة المصدر مع متحكم دقيق من شركة ATMELE على لوحة واحدة يتم برمجتها عن طريق الحاسوب .  
تم تصميمها لكي تجعل عملية استخدام الإلكترونيات التفاعلية في مشاريع متعددة التخصصات أكثر سهولة وأقل تكلفة .

من الأشياء التي ساعدت على سرعة إنتشارها أنها مفتوحة المصدر ونقصد هنا أنه قد تم نشر كافة التفاصيل الفنية والتصميمية والبرمجية حول الأردوينو وكل ما تقوم به من مشاريع وتمكين أي شخص من الوصول إليها والتعديل عليها .

تبرمج الأردوينو بلغة Arduino c . وقد تم إشتقاقها من لغة C والتي تعد أساسا إحدى لغات البرمجة الحديثة ومن أقوى لغات البرمجة .



## ما هي مميزات الأردوينو ؟

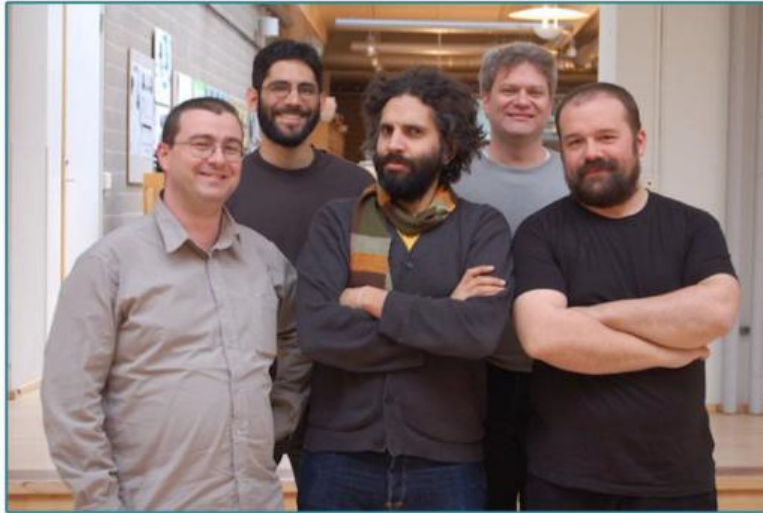
- 1- رخيصة الثمن .
- 2- سهولة التعامل معها .
- 3- بساطة لغة البرمجة .
- 4- توفر ملحقات عديدة لها .
- 5- توفر مكتبات لأغلب ملحقاتها .
- 6- مفتوحة المصدر مما يساعد على سرعة تطويرها .
- 7- يمكنك ربطها بلغات برمجة قوية مثل MATLAB و JAVA و VB.NET .

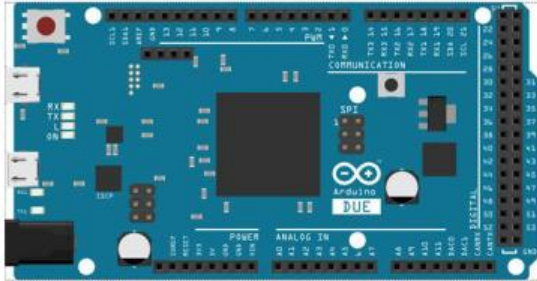


## تاريخ الأردوينو :

تبدأ القصة في عام 2005 في مدينة إيفريا Ivrea الإيطالية حيث قام كل من ماسيمو بانزي بالتعاون مع دايفيد كوارتيليس وجاينلوكا مارتينو بإطلاق مشروع أردوينو إيفريا (Arduino of Ivrea) وتمت تسميته المشروع باسم أشهر شخصية تاريخية في المدينة وكان الهدف الأساسي للمشروع هو عمل بيئة تطوير للمتحكمات الدقيقة بصورة حرة ومفتوحة المصدر مائة في المائة .

يبلغ حالياً سعرها قرابة 27 دولار وهو يعتبر سعراً مقبول جداً لدى الطلاب والهواة ، وحتى عام 2013 تم شحن وبيع أكثر من 700 ألف لوحة أردوينو حول العالم.

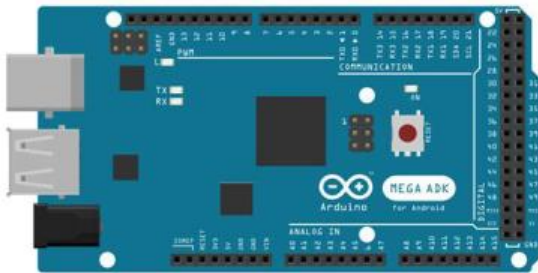




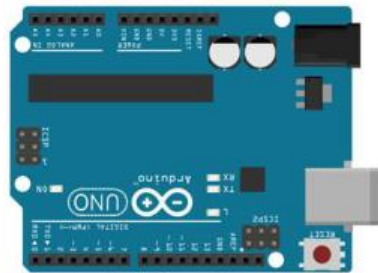
Arduino Due

## أنواع لوحات الأردوينو :

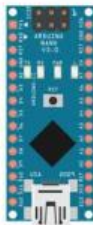
- Arduino Uno -
- Arduino Leonardo -
- Arduino Due -
- Arduino Mega 2560 -
- Arduino Nano -
- Arduino Mini -



Arduino Mega 2560



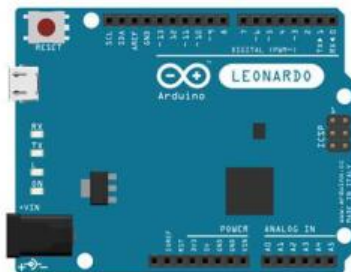
Arduino Uno



Arduino Nano



Arduino Mini



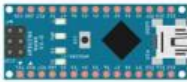
Arduino Leonardo

تختلف لوحات الأردوينو من حيث الحجم وعدد الأرجل التي يمكن إستخدامها .



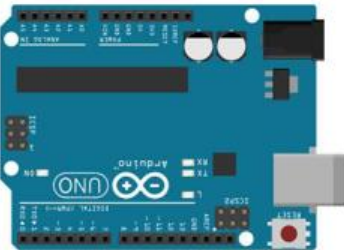
## نظرة في خصائص أهم لوحات الأردوينو :

### 1- Arduino Nano .



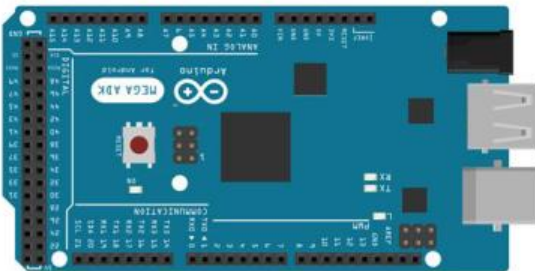
رقم المتحكم المستخدم : ATMEGA328P  
 عدد المداخل والمخارج (Digital I/O) : 14 رجل  
 عدد الأرجل التماثلية (Analog) : 8 أرجل  
 عدد أرجل UART : رجلين فقط  
 عدد أرجل PWM : 6 أرجل

### 2- Arduino Uno .



رقم المتحكم المستخدم : ATMEGA328P  
 عدد المداخل والمخارج (Digital I/O) : 14 رجل  
 عدد الأرجل التماثلية (Analog) : 6 أرجل  
 عدد أرجل UART : رجلين فقط  
 عدد أرجل PWM : 6 أرجل

### 3- Arduino Mega 2560 .



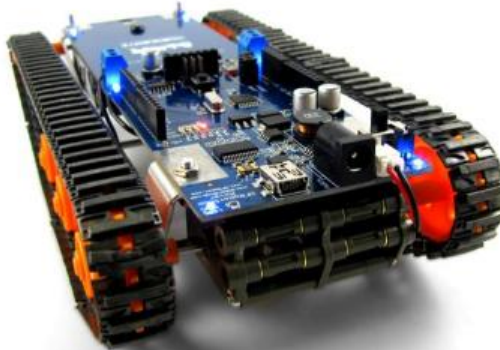
رقم المتحكم المستخدم : ATmega2560  
 عدد المداخل والمخارج (Digital I/O) : 45 رجل  
 عدد الأرجل التماثلية (Analog) : 16 أرجل  
 عدد أرجل UART : 4 أرجل  
 عدد أرجل PWM : 15 أرجل

## ما هي استخدامات الأردوينو ؟

يوجد للأردوينو الكثير من التطبيقات العملية الرائعة، نذكر منها :

- 1- مشاريع قياس درجة الحرارة والرطوبة والضوء...الخ
- 2- مشاريع التحكم بتشغيل الأجهزة وعرض النتائج على شاشة LCD .
- 3- مشاريع قياس المسافة .
- 4- التحكم بأجهزة المنزل عبر الهاتف .
- 5- صنع رجل آلي .
- 6- جهاز انذار الحرائق .
- 7- مشاريع التحكم بالسيارة عن طريق الهاتف .
- 8- سقي المزروعات تلقائياً .
- 9- جهاز تباع الشمس .
- 10- أجهزة تحديد موقع السيارات أو الأشخاص عبر الأقمار الصناعية (GPS) .

لا يمكن حصر استخدامات الأردوينو لكثرتها ، يمكنك إستخدامها في مشاريع كثيرة تخطر في بالك !

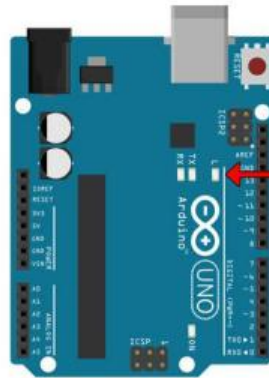


## معلومات عن الأردوينو أونو (Arduino UNO)

تعتبر من أشهر أنواع الأردوينو على نطاق العالم ، وذلك بسبب أنها رخيصة الثمن وأيضا عدد الأرجل مناسب ويفي بالغرض و مناسبة الحجم .

سنقوم خلال الصفحات القادمة بشرح مبسط عنها وذلك لأننا سنقوم باستخدامها في جميع التجارب القادمة . المتحكم الدقيق الموجود بداخلها من شركة ATMEGA328 وهي تعتبر من الشركات الرائدة التي تصنع المتحكمات الدقيقة ، وهو من نوع ATMEGA328 .

يوجد بها 14 رجل يمكنك استخدامها كمدخل أو مخرج للإشارة الرقمية (1 أو 0) و 6 أرجل بها خاصية PWM (ستتعرف عليها لاحقا) . كما يوجد بها أيضا 6 أرجل من النوع التماثلي (ANALOG) ومخرج واحد للنظام للإتصال التسلسلي UART .



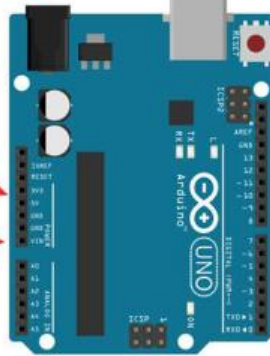
بها متذبذب 16MHz  
Crystal

بها LED متصل بالرجل رقم 13  
يمكنك استخدامه كمخرج .

## مصدر الطاقة للوحة الأردوينو

- 1- يمكن تزويد الأردوينو بالطاقة من خلال منفذ USB .  
2- أو من خلال وصلة الطاقة (7 إلى 20 فولت)

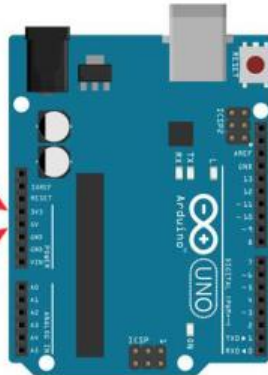
- 3- أو من خلال الرجل 5v (5 فولت فقط)  
4- أو من خلال الرجل Vin (7 إلى 20 فولت)



سؤال: لو كانت عندي قطعة تعمل بـ 3.3 فولت أو 5 فولت هل يمكن أخذ الجهد من لوحة الأردوينو ؟

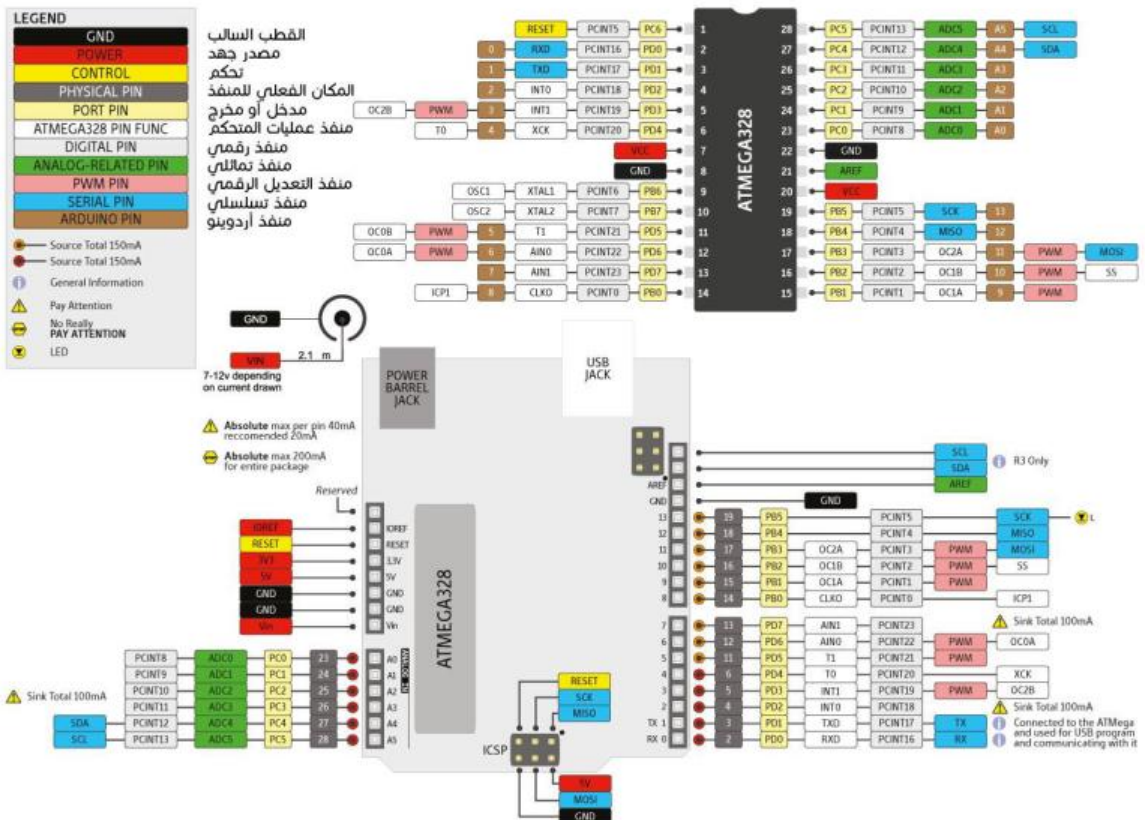
نعم يمكن ذلك .

- 3.3 فولت  
5 فولت

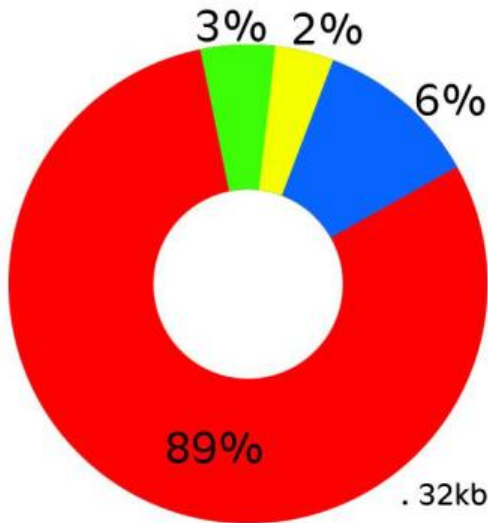


### تشكيلا الأرجل :

هذه الرسمة التوضيحية مفيدة جدا لتحديد منافذ ووظائف أرجل الأردوينو المختلفة، ويمكنك العودة إليها كمرجع عند تنفيذ التجارب لتحديد الأرجل التي تحتاج إليها



## موارد الذاكرة للأردوينو :



- SRAM - 2 Killobyte
- Flash Memory - 29 Killobyte
- EEPROM - 1 Killobyte
- Bootloader 0.5 Killobyte

يعمل هذا المعالج على سرعة 16MHz وذاكرة 32kb .

هناك بعض المصطلحات من المهم معرفتها في عالم الأردوينو:

Boot loader <---- جزء برمجي تم تخزينه في المتحكم الدقيق مسبقاً وهو المسؤول عن كيفية استقبال البرنامج من الحاسوب وتخزينه في المتحكم الدقيق.

SRAM <----- هو الذاكرة المستخدمة في تسجيل أو تخزين المتغيرات لفترة مؤقتة.

Flash Memory <-- هو المكان الذي سنقوم بتخزين البرنامج فيه بشكل دائم.

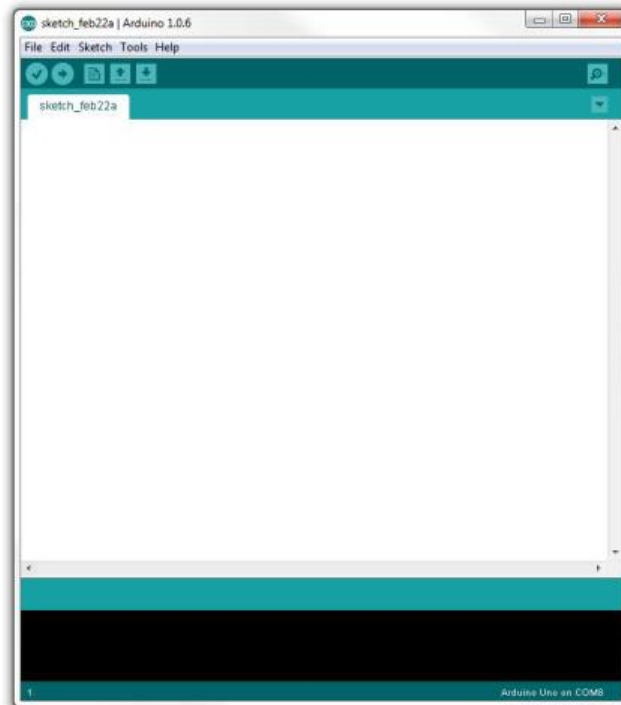
EEPROM <----- هي ذاكرة يتم فيها تخزين المتغيرات بشكل دائم حتى لو تم فصل الطاقة عن المتحكم.



sketch\_jan23a

## برنامج الأردوينو :

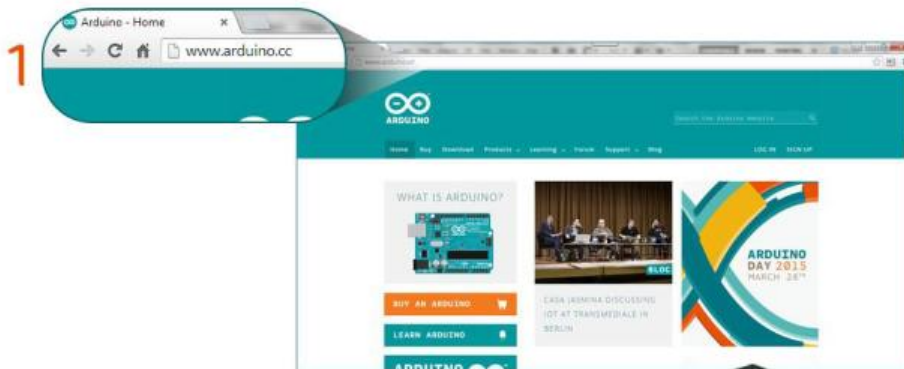
يتم برمجة لوحة الأردوينو من خلال برنامج Arduino IDE (بيئة التطوير المتكاملة) وهو مجاني من شركة أردوينو ، سنقوم في الصفحات القادمة بشرح كيفية تنزيله .



يمكنك الحصول على البرنامج وتنزيله من الموقع الرسمي لشركة أردوينو أو من خلال موقعنا الإلكتروني [www.ev-center.com/Arduino](http://www.ev-center.com/Arduino)

## كيفية الحصول على برنامج Arduino IDE :

1- اذهب إلى الموقع الرسمي للآردوينو وهو [www.arduino.cc](http://www.arduino.cc).



2- ومن ثم اختر كلمة Download .



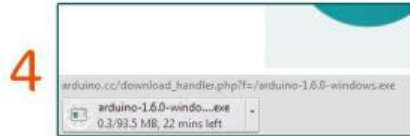


sketch\_jan23a

3- قم بإختيار نظام التشغيل المناسب لجهازك.

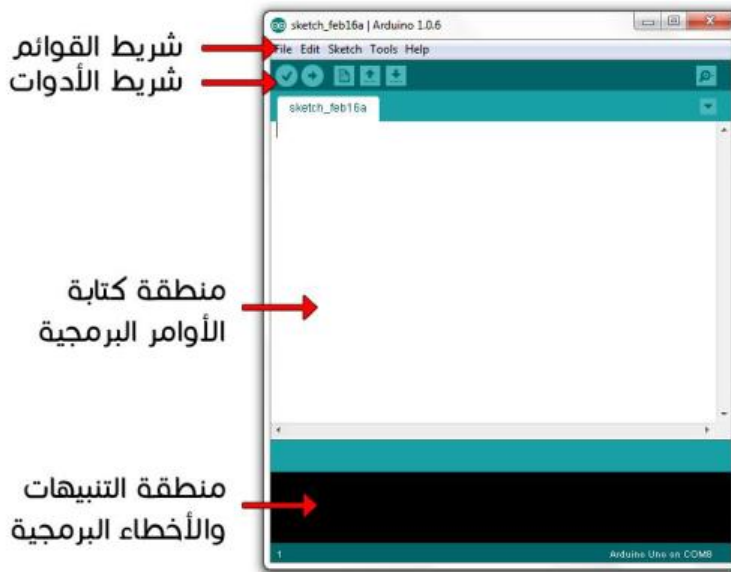


4- بعد الإنتهاء من التحميل قم بتثبيت البرنامج في جهازك .



## بيئة التطوير المتكاملة (Arduino IDE) :

تحتوي واجهة البيئة التطويرية على أربعة أقسام رئيسية كما هو موضح في الصورة:



خلال الصفحات القادمة سنقوم بشرح أهم الأجزاء في بيئة التطوير .

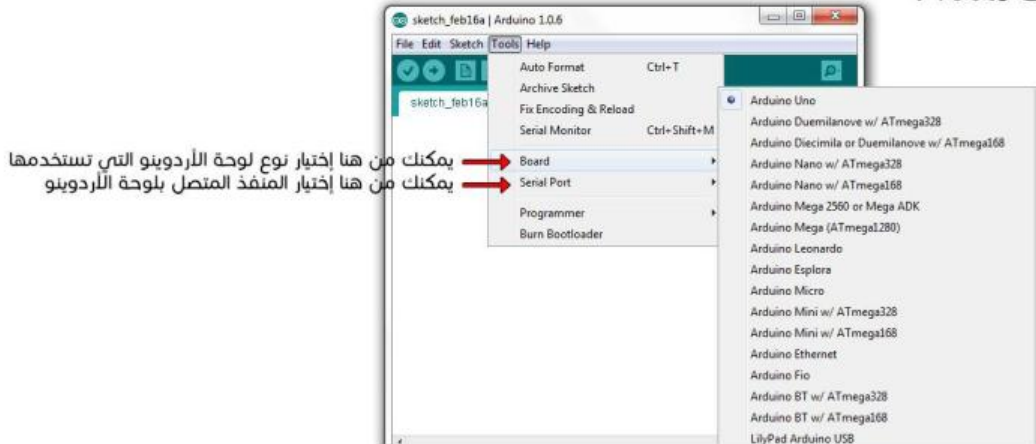
ملاحظة : تتوفر بيئة التطوير بعدة لغات من ضمنها اللغة العربية .

## شريط القوائم :

قائمة File :

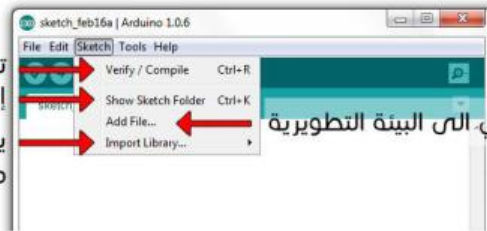


قائمة Tools :



## قائمة Sketch :

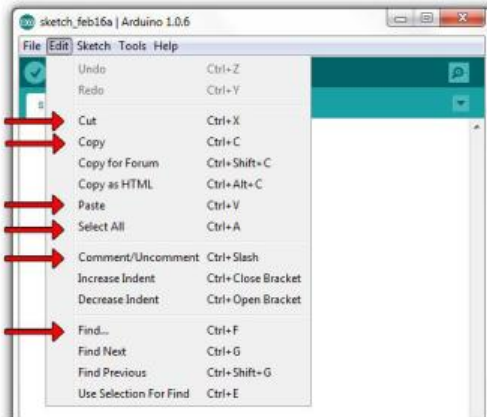
تصحيح الكود  
إظهار مكان حفظ البرنامج  
يمكنك إضافة مكتبة لألي  
ملحق من ملحقات الاردوينو



إضافة ملف برمجي الى البيئة التطويرية

## قائمة Edit :

قص النص المحدد  
نسخ النص المحدد  
لصق النص المنسوخ  
تحديد جميع النص  
تحويل أو إلغاء الملاحظة من النص  
البحث عن كلمة أيضاً يمكن إستبدالها



## شريط الأدوات :

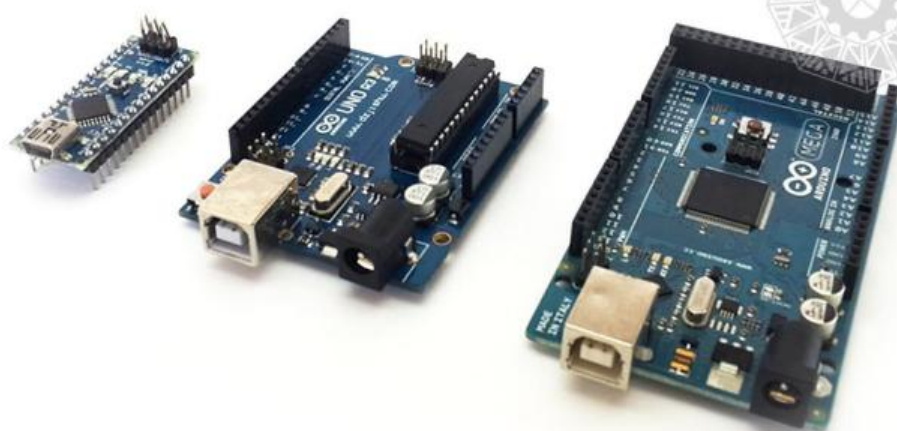


## الأدوات المستخدمة في التجارب :

- 1- لوحة أردوينو أونو Arduino UNO .
  - 2- كابل USB .
  - 3- لوح تجريبي BreadBoard .
  - 4- أسلاك توصيل Wires .
  - 5- وصلة ثنائية ضوئية LEDs .
  - 6- مقاومات ثابتة (220,330,10K) Resistor .
  - 7- مقاومة متغيرة 10k Variable resistor .
  - 8- مقاومة ضوئية LDR .
  - 9- مفتاح ضغط Push Button .
  - 10- محرك السيرفو Servo motor .
  - 11- محرك DC .
  - 12- لوحة مفاتيح Keypad 4x4 .
  - 13- طنان buzzer .
  - 14- شاشة عرض ارقام سباعية 7segments cc .
  - 15- مجسة الأصوات فوق الصوتية UltraSonic sensor .
  - 16- مجسة صوت Sound sensor .
  - 17- مجسة حرارة lm35Dz Temperature sensor .
  - 18- عصا التحكم Joystick .
  - 19- شاشة عرض LCD 16x2 .
- 20- ترانزستور 2N2222 NPN .
- 21- 1N4001 .

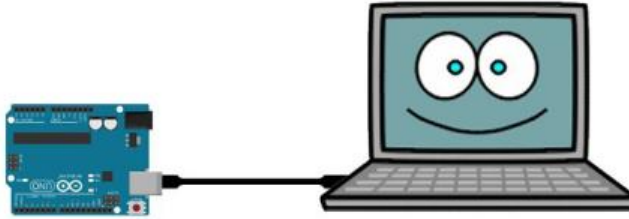


sketch\_jan23a



## كيفية توصيل الأردوينو بجهاز الكمبيوتر

قم بتوصيل لوحة الأردوينو بجهاز الكمبيوتر عن طريق كابل الـ USB.



يجب معرفة رقم المنفذ المتصل باللوحة ، يمكنك ذلك من خلال الطرق التالية :

### الطريقة الأولى :

بعد توصيلك للوحة الأردوينو ستظهر لك رسالة على الجانب الأيمن تخبرك بأن اللوحة متصلة برقم المنفذ كما هو في الصورة



### الطريقة الثانية :

إذا كنت لا تعرف رقم المنفذ المتصل مع لوحة الأردوينو اتبع الخطوات التالية :

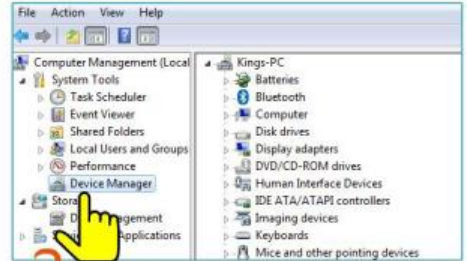




sketch\_jan23a



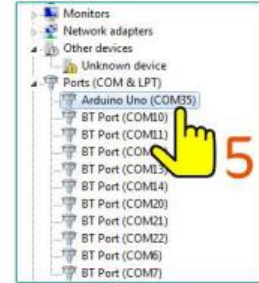
4



3

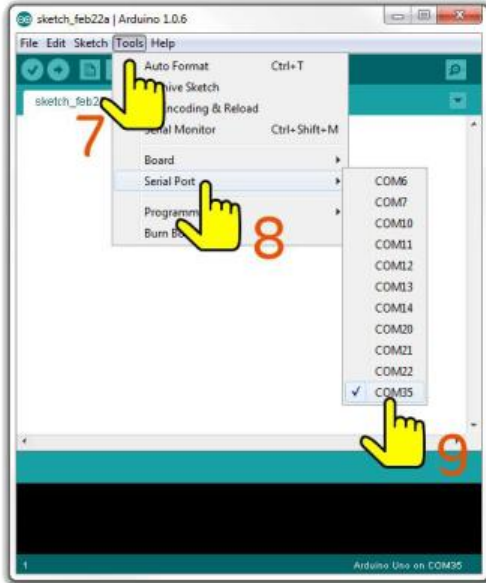


الآن افتح برنامج الأردوينو



5

بهذه الطريقة سيكون رقم المنفذ هو 35



7

8

9

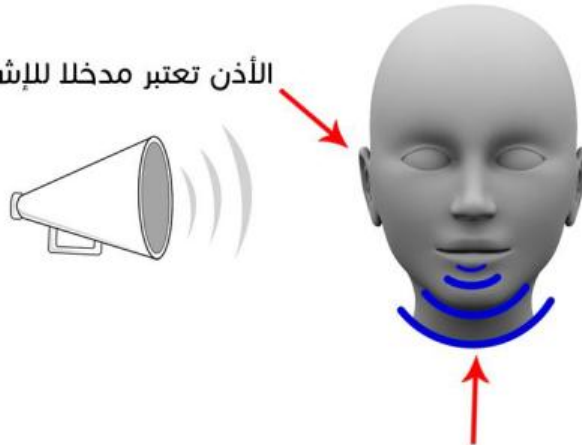
أصبحت لوحة الأردوينو الخاصة بك جاهزة  
أيها المبرمج  
هل أنت متشوق لتجربتها ؟



## مداخل ومخارج الأردوينو ( INPUT & OUTPUT PINS )

هي عبارة عن أرجل موجودة في الأردوينو يتم من خلالها إرسال أو استقبال الإشارات .  
لتوضيح الصورة أكثر يمكننا تشبيه الموضوع بهذه الصورة :

الأذن تعتبر مدخلا للإشارات الصوتية (INPUT)



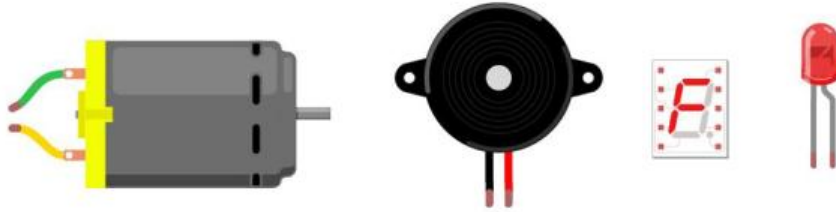
صوت الشخص يعتبر إشارة خارجة (Signal)  
فم الشخص يعتبر مخرجا للإشارة الصوتية (OUTPUT)

الخلايا الحسية الموجودة في الجلد هي عبارة عن  
مجسات ترسل الإشارة (INPUT) إلى الدماغ

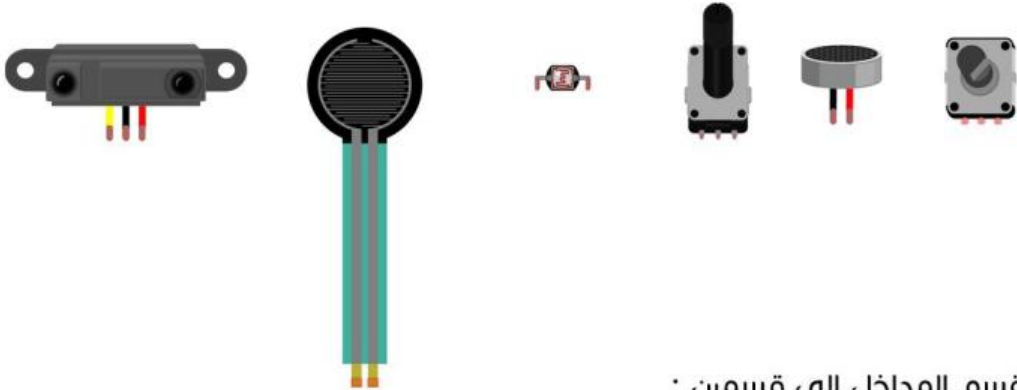


sketch\_jan23a

بعض الأمثلة لمجموعة من القطع التي تستقبل الإشارة ويمكن ربطها بمخارج الأردوينو .



وهنا مجموعة من القطع والمجسات التي تعطي إشارة ويمكن ربطها بمدخلات الأردوينو .



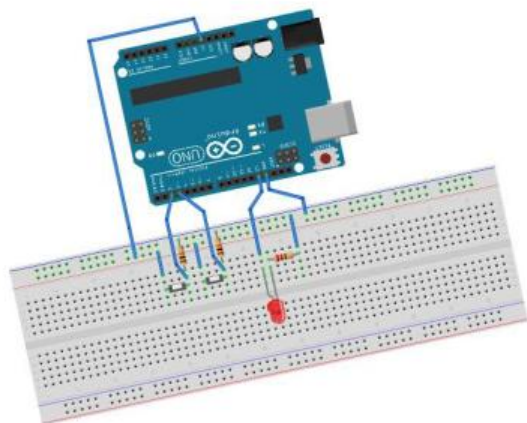
تنقسم المداخل إلى قسمين :

- 1- مداخل رقمية ( يمكن أن نستدل من خلالها على وجود إشارة أو عدمها ) .
- 2- مداخل تماثلية ( يمكن أن نستدل من خلالها على وجود إشارة و أيضا حساب قوتها ) .

```
void setup() {  
  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```



# فصل التجارب



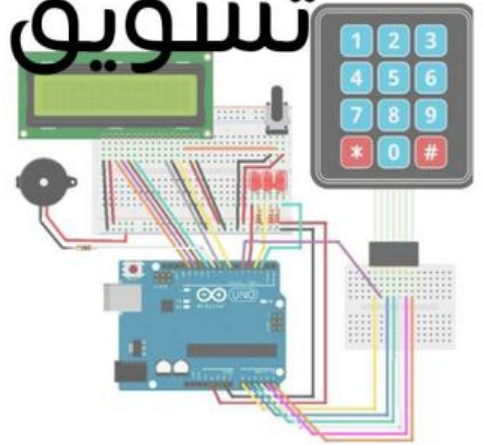
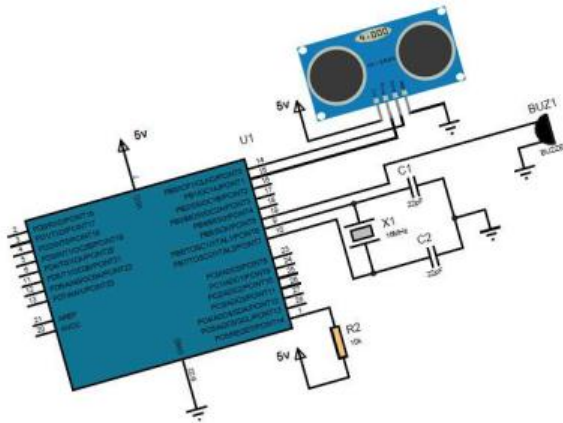
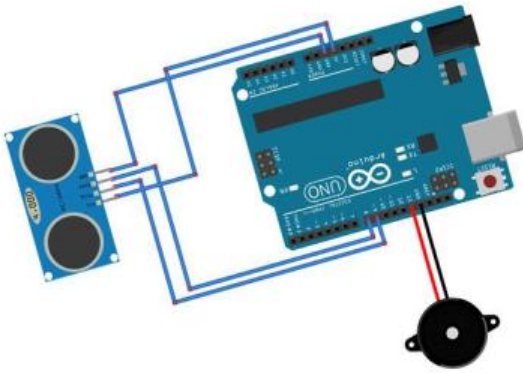


sketch\_jan23a

متعة..

فائدة ...

تشويق ....



## الدرس الأول :

# تشغيل LED

### فكرة الدرس :

هنا ستكون البداية أيها المبرمج في عالم برمجة الأردوينو الرائع . سنتعلم في هذا الدرس كيفية تشغيل LED وإطفائه .

#### الأوامر

```
void setup() {  
  pinMode(13, OUTPUT);  
}
```

➡ الرجل رقم 13 كمخرج <<-----

```
void loop() {
```

♻️ عمل حلقة غير منتهية <<-----

```
  digitalWrite(13, HIGH);
```

➡ إخراج تيار من الرجل رقم 13 <<-----  
(سيضيء LED)

```
  delay(1000);
```

⌚ تأخير لمدة ثانية <<-----

```
  digitalWrite(13, LOW);
```

➡ جعل الرجل 13 تساوي 0 فولت <<-----  
(سينطفئ LED)

```
  delay(1000);
```

⌚ تأخير لمدة ثانية <<-----

```
}
```

Ⓡ <<-----

من الضروري التنبيه إلى أن لغة الـ Arduino حساسة لوضع الحروف . فالحروف الإنجليزية الكبيرة لا يمكن أن تحل محل الحروف الصغيرة والعكس صحيح اليك بعض الأمثلة :

pinMode



pinmode



digitalWrite



digitalwrite



HIGH



high



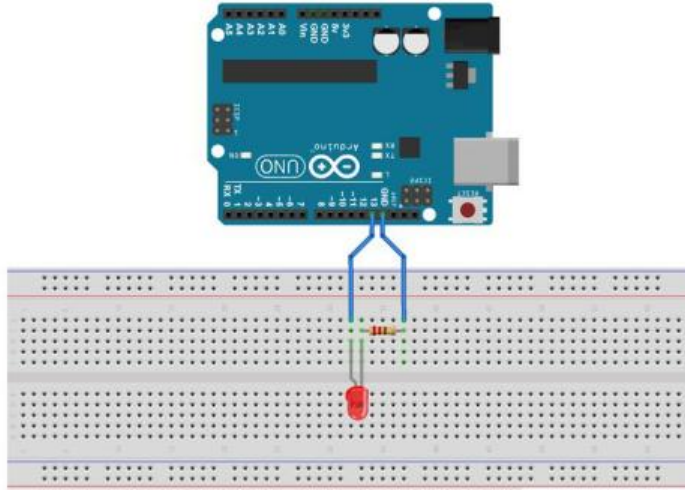


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
LED Red x1  
220 ohm x1

## توصيلة التجربة على اللوح التجريبي:



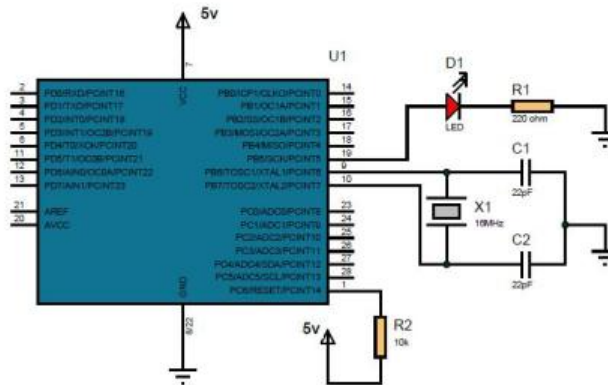
## الرسم الفيزيائي :

### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

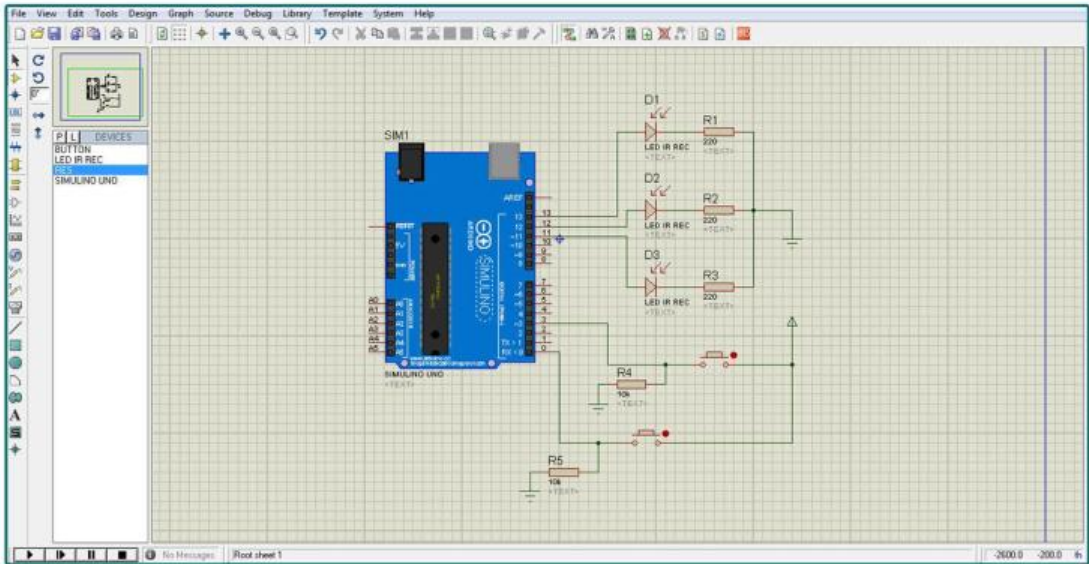
A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28





## برنامج محاكاة الدوائر (Proteus)

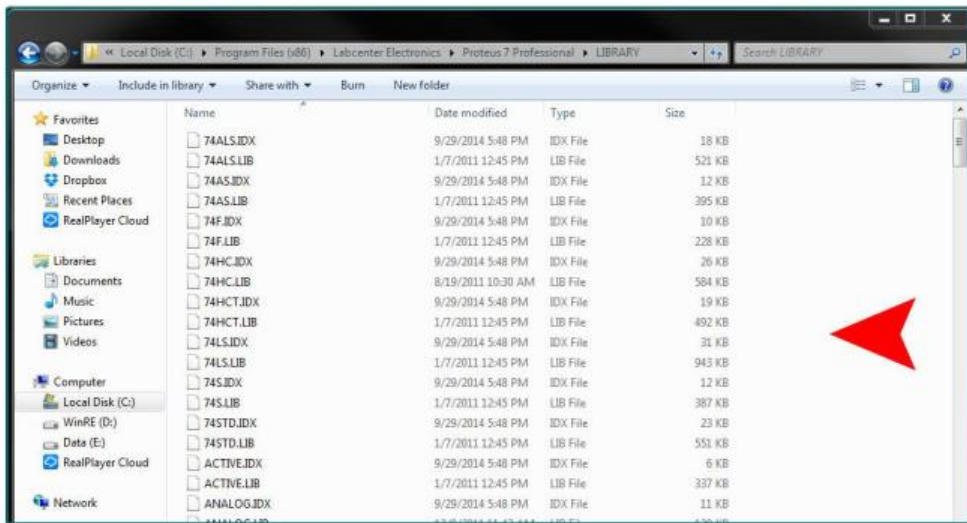
البروتوس (Proteus Simulator) هو عبارة عن برنامج لمحاكاة الدوائر الإلكترونية يسمح لك بتركيب الدوائر الإلكترونية وتشغيلها على جهاز الكمبيوتر قبل تركيبها في الواقع ويستخدم كثيرا من قبل الطلبة والدارسين للإلكترونيات والبرمجة. يتميز البروتوس عن برامج المحاكاة الأخرى بميزتين أساسيتين. الأولى: هي أنه يسمح لك برسم الدائرة الإلكترونية وتجهيزها للطباعة على ألواح طباعة الدوائر (PCB). والثانية هي وجود نماذج محاكاة لمعظم المتحكمات الدقيقة الشائعة الاستخدام ويتم تحديثه بشكل مستمر.





sketch\_jan23a

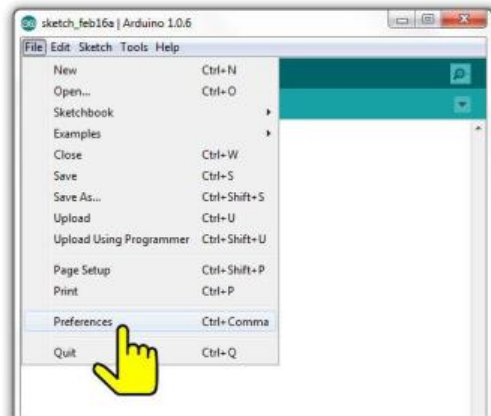
إذا لم تكن تملك لوح أردوينو حالياً أو أنك تريد تجربة الدروس والمشاريع قبل تطبيقها في الواقع فبإمكانك استخدام برنامج البروتوس كبدائية. ولكن سوف تحتاج لإضافة محاكي الأردوينو عبر البحث في شبكة الإنترنت عن مكتبة بروتوس للأردوينو (اكتب في محرك البحث: Proteus Arduino Library) وإضافة هذه المكتبة إلى برنامج البروتوس بوضعه في الملف الموضح في الصورة التالية:



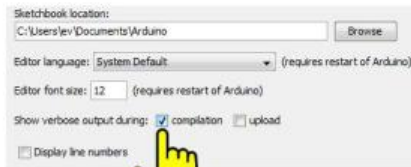
## كيفية ربط الأردوينو مع برنامج isis proteus : ١٦

بعد الانتهاء من كتابة البرنامج في البيئة التطويرية وتصحيحه اذهب إلى قائمة ملف (File)

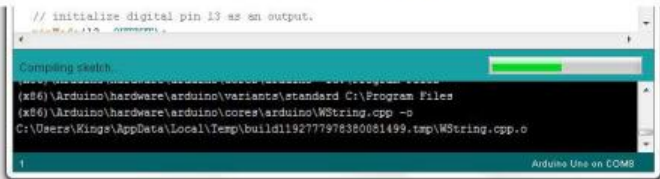
واختار الخيار Preferences .



بعد ذلك ضع علامة صح على الخيار Show verbose output during .



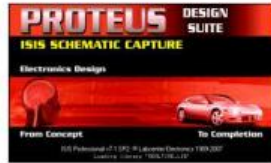
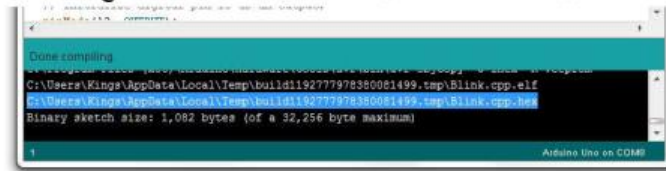
ستلاحظ ظهور كلام كثير في منطقة التنبيهات والأخطاء البرمجية .





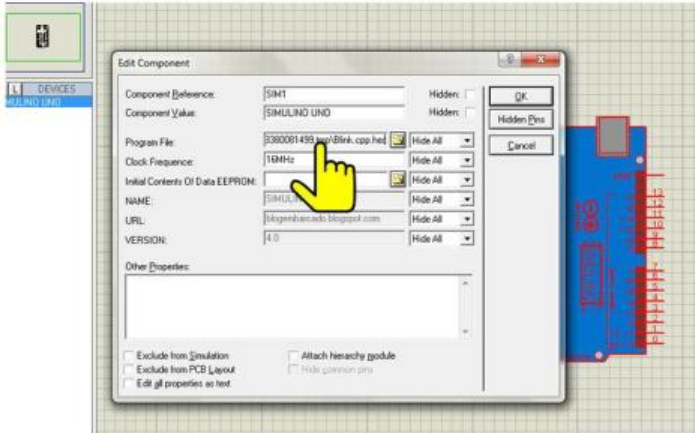
sketch\_jan23a

بعد ذلك قم بنسخ السطر الذي ينتهي بكلمة .hex  
ومن ثم قم بالضغط على زر control + حرف C من لوحة المفاتيح .



الآن افتح برنامج المحاكاة البروتس isis Proteus .

بعد إختيارك للوحة الأردوينو اذهب إلى الخصائص وقم بملصق السطر الذي قمنا بنسخه من البيئة التطويرية.



ملاحظة : يمكنك تطبيق دروس هذا  
الكتاب مباشرة على لوحة الأردوينو  
دون الحاجة إلى برنامج البروتوس.

## الدرس الثاني:

### ترميز الأرجل

#### فكرة الدرس :

يمكنك من خلال هذا الدرس ترميز أو تسمية أرجل الأردوينو كيفما تشاء. سيساعدك ذلك في كتابة الأوامر من جانب تذكر أماكن توصيل القطع مع أرجل الأردوينو.

#### الأوامر

```
int LED=13;

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

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

◆ ترميز الرجل 13 بكلمة LED <<-----

➡ الرجل رقم 13 كمخرج <<-----

♻️ عمل حلقة غير منتهية <<-----

إخراج تيار من الرجل رقم 13 <<-----  
(سيضيء LED)

⌚ تأخير لمدة ثانية <<-----

جعل الرجل 13 تساوي 0 فولت <<-----  
(سينطفئ LED)

⌚ تأخير لمدة ثانية <<-----

⬇️ <<-----

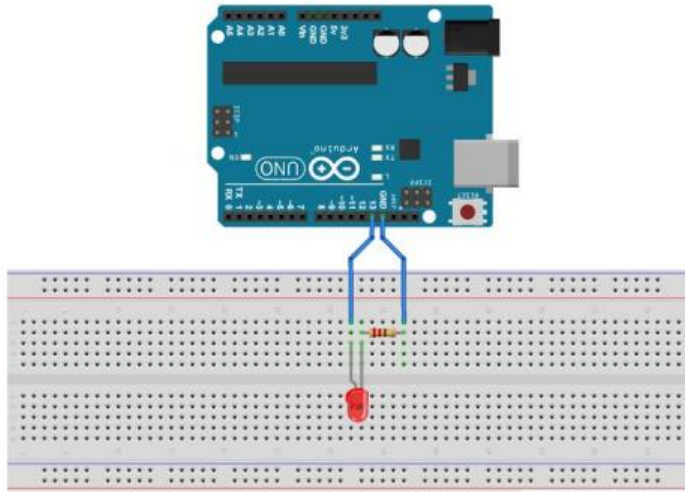


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
LED Red x1  
220 ohm x1

## توصيلة التجربة على اللوح التجريبي:



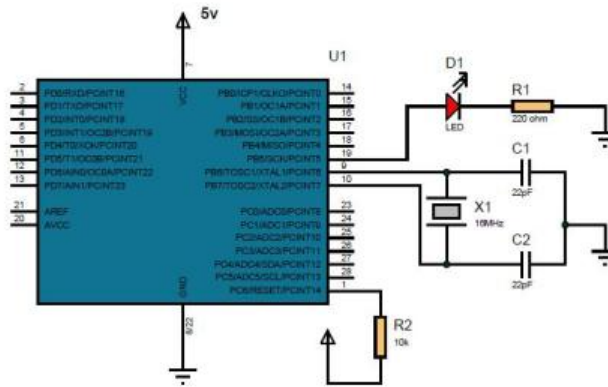
## الرسم الفيزيائي :

### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28



## الدرس الثالث:

# الجملة الشرطية if

### فكرة الدرس :

الجملة الشرطية مهمة جداً في عالم البرمجة وتعمل بالطريقة التالية: في حالة تحقق شرط معين يقوم البرنامج بتنفيذ أمر ما . مثلاً: عند ضغط الزر الأحمر < يعمل جهاز الإنذار

### الأوامر

```
int LED=13;  
int BUTTON=2;
```

◆ ترميز الرجل رقم 13 ب كلمة LED <<-----  
◆ ترميز الرجل رقم 2 ب كلمة BUTTON <<-----

```
void setup() {
```

```
    pinMode(LED, OUTPUT);  
    pinMode(BUTTON, INPUT);  
}
```

➡ الرجل رقم 13 كمخرج <<-----  
➡ الرجل رقم 2 كمدخل <<-----

```
void loop() {
```

♻️ عمل حلقة غير منتهية <<-----

```
if (digitalRead(BUTTON)==HIGH) {  
    digitalWrite(LED, HIGH);  
    delay(1000);  
    digitalWrite(LED, LOW);  
    delay(1000);  
}
```

🔴 الجملة الشرطية في حالة الضغط على المفتاح <<--  
🔴 سيضيء LED <<-----  
🕒 تأخير لمدة ثانية <<-----  
🔴 سينطفئ LED <<-----  
🕒 تأخير لمدة ثانية <<-----  
🔴 نهاية الجملة الشرطية <<-----

🕒 العودة إلى كلمة loop <<-----

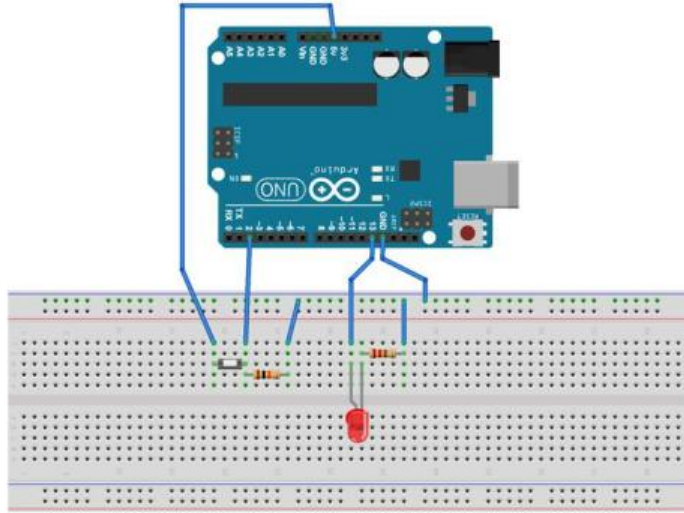


sketch\_jan23a

## Used Parts :

- Arduino UNO x1
- Breadboard x1
- LED Red x1
- 220 ohm x1
- 10k ohm x1
- Push Button x1

## توصيلة التجربة على اللوح التجريبي:



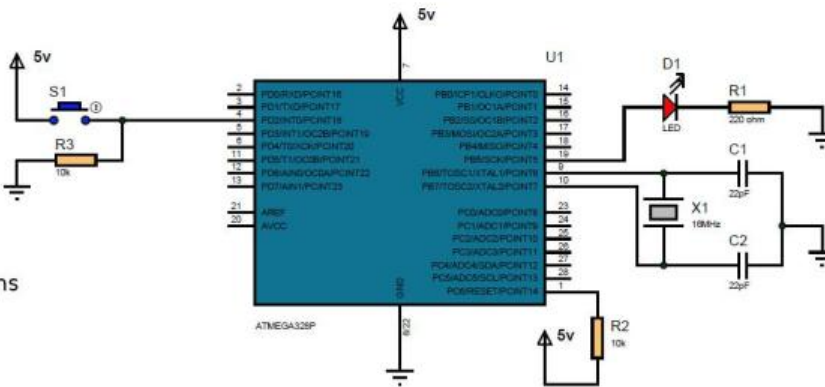
## الرسم الفيزيائي :

### Digital pins

- D0 = 2
- D1 = 3
- D2 = 4
- D3 = 5
- D4 = 6
- D5 = 11
- D6 = 12
- D7 = 13
- D8 = 14
- D9 = 15
- D10 = 16
- D11 = 17
- D12 = 18
- D13 = 19

### Analog pins

- A0 = 23
- A1 = 24
- A2 = 25
- A3 = 26
- A4 = 27
- A5 = 28



## الدرس الرابع:

# جملتين شرطيتين if

### فكرة الدرس :

في هذا الدرس ستقوم بتشغيل LED بمفتاح وإطفائه بمفتاح آخر.

#### الأوامر

```
int LED=13;  
int BUTTON_ON=2;  
int BUTTON_OFF=3;
```

```
void setup() {
```

```
  pinMode(LED, OUTPUT);  
  pinMode(BUTTON_ON, INPUT);  
  pinMode(BUTTON_OFF, INPUT); }
```

```
void loop() {
```

```
  if (digitalRead(BUTTON_ON)==HIGH) {  
    digitalWrite(LED, HIGH); }
```

```
  if (digitalRead(BUTTON_OFF)==HIGH) {  
    digitalWrite(LED, LOW); }
```

```
}
```

ترميز الرجل 13 بكلمة LED <<-----  
ترميز الرجل 2 بكلمة BUTTON\_ON <<-----  
ترميز الرجل 3 بكلمة BUTTON\_OFF <<-----

الرجل رقم 13 كمخرج <<-----  
الرجل رقم 2 كمداخل <<-----  
الرجل رقم 3 كمداخل <<-----

عمل حلقة غير منتهية <<-----

الجملة الشرطية الأولى <<-----  
سيضيء LED <<-----

الجملة الشرطية الثانية <<-----  
سينطفئ LED <<-----

<<-----

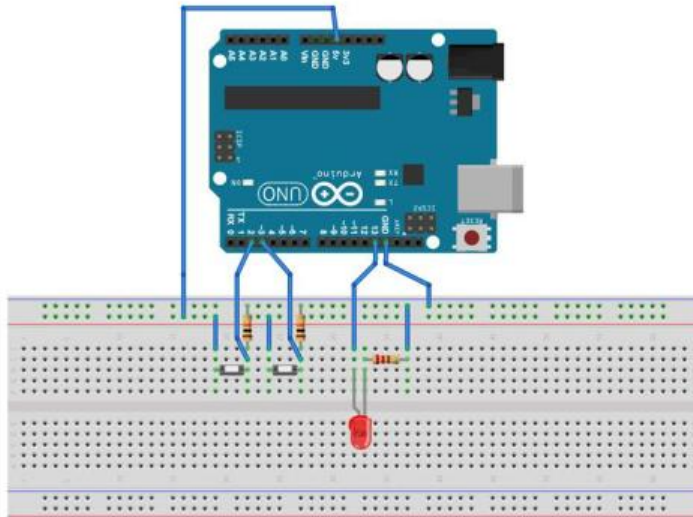


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
LED Red x1  
220 ohm x1  
10k ohm x2  
Push Button x2

## توصيلة التجربة على اللوح التجريبي:



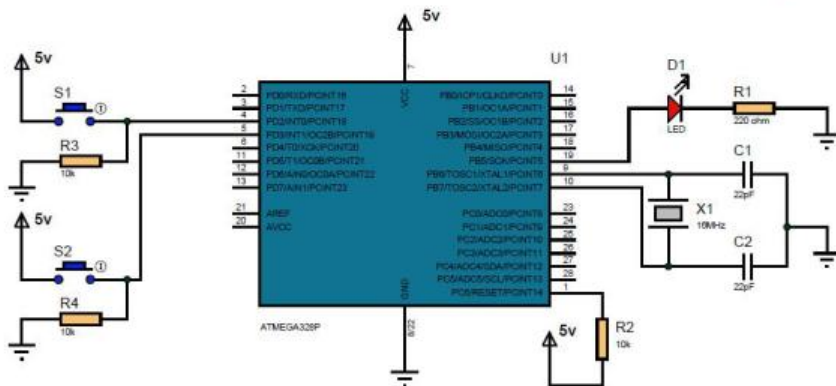
## الرسم الفيزيائي :

### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28



## الدرس الخامس:

# الجملة الشرطية if / else

### فكرة الدرس :

أمر else نستخدمه لتحديد ما يجب على البرنامج فعله إذا لم تتحقق الجملة الشرطية. مثلاً: إذا قمت بحل الواجب سنذهب للحديقة. إذا لم تحل الواجب ستعرض للعقاب.

else

if

### الأوامر

```
int LED=13;  
int BUTTON=2;
```

◆ ترميز الرجل رقم 13 بـ كلمة LED <<-----  
◆ ترميز الرجل رقم 2 بـ كلمة BUTTON <<-----

```
void setup() {
```

```
  pinMode(LED, OUTPUT);  
  pinMode(BUTTON, INPUT);  
}
```

➡ الرجل رقم 13 كمخرج <<-----  
➡ الرجل رقم 2 كمداخل <<-----

```
void loop() {
```

♻️ عمل حلقة غير منتهية <<-----

```
  if (digitalRead(BUTTON)==HIGH) {  
    digitalWrite(LED, HIGH);  
  }  
  else {  
    digitalWrite(LED, LOW);  
  }
```

🔴 الجملة الشرطية في حالة الضغط على المفتاح <<--  
🔴 سيضيء LED <<-----  
🔴 ما عدا (إذا كان الشرط غير متحقق) <<-----  
🔴 سينطفئ LED <<-----

🔄 العودة إلى كلمة loop <<-----

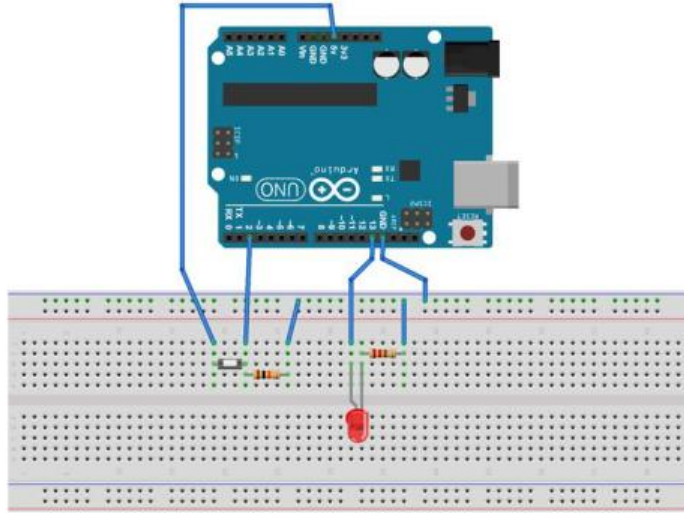


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
LED Red x1  
220 ohm x1  
10k ohm x1  
Push Button x1

توصيلة التجربة على اللوح التجريبي:



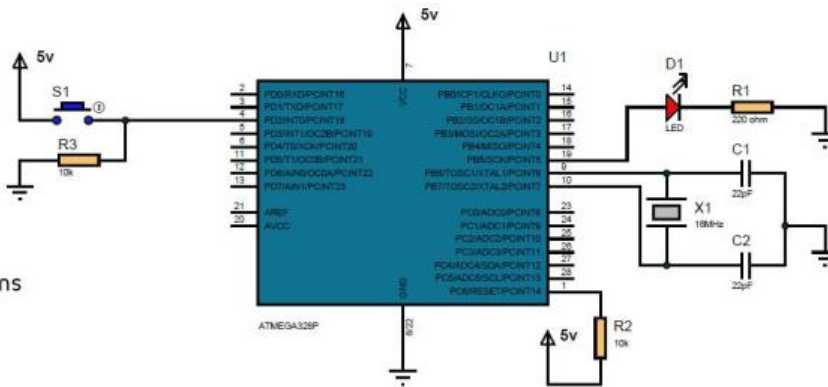
الرسم الفيزيائي :

## Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

## Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28



## الدرس السادس:

# إنذار الطوارئ

### فكرة الدرس :

سنقوم بعمل مشروع إنذار للطوارئ من خلال الأوامر التي تعلمناها إلى الآن أيها المبرمج. نتمنى لك وقتاً ممتعاً.

### الأوامر

```
int LED =13;
int BUZZER =4;
int BUTTON =2;
```

```
void setup() {
  pinMode(LED, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  pinMode(BUTTON, INPUT);
}
```

```
void loop() {
```

```
  if (digitalRead(BUTTON)==HIGH) {
    digitalWrite(LED, HIGH);
    digitalWrite(BUZZER, LOW);
    delay(500);
    digitalWrite(LED, LOW);
    digitalWrite(BUZZER, HIGH);
    delay(500); }
}
```

◆ ترميز الرجل رقم 13 ب كلمة LED <<-----  
◆ ترميز الرجل رقم 4 ب كلمة BUZZER <<-----  
◆ ترميز الرجل رقم 2 ب كلمة BUTTON <<-----

➡ الرجل رقم 13 كمخرج <<-----  
➡ الرجل رقم 12 كمخرج <<-----  
➡ الرجل رقم 2 كمدخل <<-----

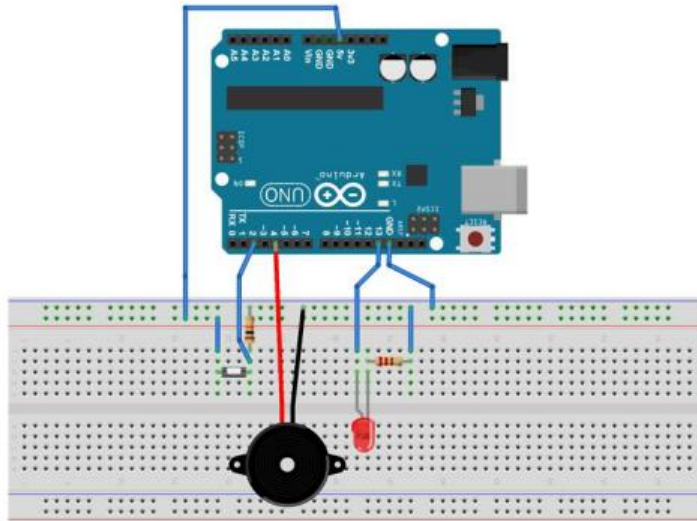
♻️ عمل حلقة غير منتهية <<-----

🔊 الجملة الشرطية في حالة الضغط على المفتاح <<-  
🔴 سيضيء LED <<-----  
🔊 لن يعمل الطنان <<-----  
🔊 سيصدر الطنان صوتاً <<-----  
🔴 سينطفئ LED <<-----  
🔊 سيصدر الطنان صوتاً <<-----  
🕒 تأخير لمدة نصف ثانية <<-----  
🔄 العودة إلى كلمة loop <<-----

### Used Parts :

Arduino UNO x1  
 Breadboard x1  
 LED Red x1  
 220 ohm x1  
 10k ohm x1  
 Push Button x1  
 Buzzer x1

### توصيلة التجربة على اللوح التجريبي:



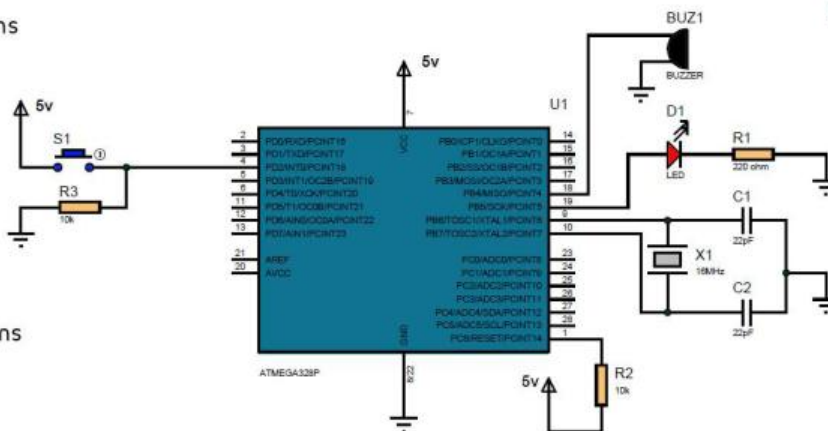
### الرسم الفيزيائي :

### Digital pins

|     |   |    |
|-----|---|----|
| D0  | = | 2  |
| D1  | = | 3  |
| D2  | = | 4  |
| D3  | = | 5  |
| D4  | = | 6  |
| D5  | = | 11 |
| D6  | = | 12 |
| D7  | = | 13 |
| D8  | = | 14 |
| D9  | = | 15 |
| D10 | = | 16 |
| D11 | = | 17 |
| D12 | = | 18 |
| D13 | = | 19 |

## Analog pins

|    |   |    |
|----|---|----|
| A0 | = | 23 |
| A1 | = | 24 |
| A2 | = | 25 |
| A3 | = | 26 |
| A4 | = | 27 |
| A5 | = | 28 |





## الدرس السابع:

# إشارات المرور

## فكرة الدرس :

إشارات المرور تساعدنا في عملية تنظيم سير السيارات . ما رأيك في عمل إشارات مرور خاصة بك أيها المبرمج؟.

### الأوامر

```

int RED_LED = 2;
int YELLOW_LED = 8;
int GREEN_LED = 13;

void setup() {
  pinMode(RED_LED, OUTPUT);
  pinMode(YELLOW_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
}

void loop() {

  digitalWrite(RED_LED, HIGH);
  digitalWrite(YELLOW_LED, LOW);
  digitalWrite(GREEN_LED, LOW);
  delay(2000);

  digitalWrite(RED_LED, LOW);
  digitalWrite(YELLOW_LED, HIGH);
  digitalWrite(GREEN_LED, LOW);
  delay(1000);

```

◆ ترميز الرجل 2 ب كلمة RED\_LED <<-----

◆ ترميز الرجل 8 ب كلمة YELLOW\_LED <<-----

◆ ترميز الرجل 13 ب كلمة GREEN\_LED <<-----

➡ الرجل رقم 2 كمخرج <<-----

➡ الرجل رقم 8 كمخرج <<-----

➡ الرجل رقم 13 كمخرج <<-----

♻️ عمل حلقة غير منتهية <<-----

🚦 يضئ LED الأحمر <<-----

🚦 ينطفئ LED الأصفر <<-----

🚦 ينطفئ LED الأخضر <<-----

🕒 تأخير لمدة ثانيتين <<-----

🚦 ينطفئ LED الأحمر <<-----

🚦 يضئ LED الأصفر <<-----

🚦 ينطفئ LED الأخضر <<-----

🕒 تأخير لمدة ثانية <<-----



sketch\_jan23a

## الأوامر

```
digitalWrite(LED_RED, LOW);
digitalWrite(LED_YELLOW, LOW);
digitalWrite(LED_GREEN, HIGH);
delay(2000);
```

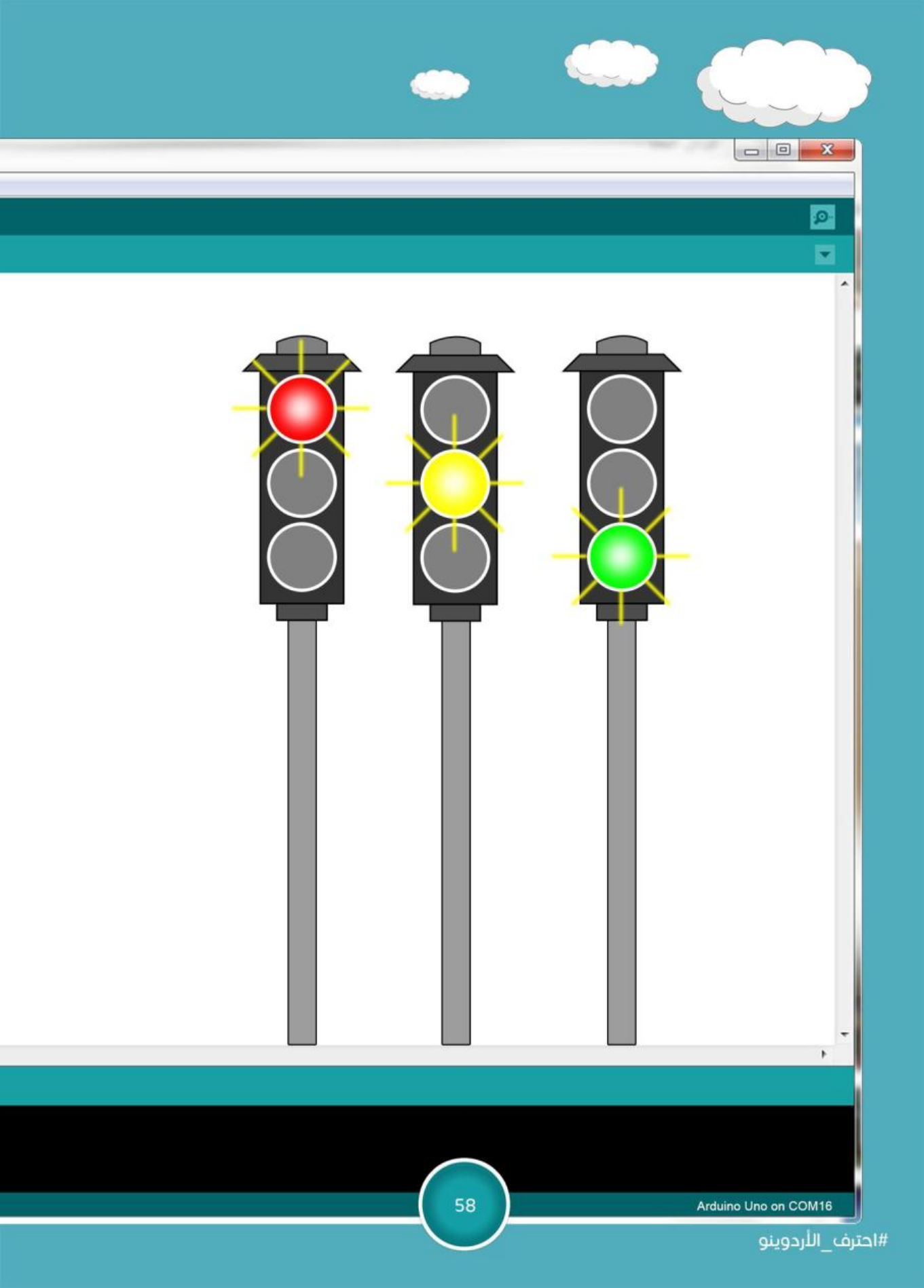
<<----- ينطفئ LED الأحمر   
 <<----- ينطفئ LED الأصفر   
 <<----- يضيء LED الأخضر   
 <<----- تأخير لمدة ثانيتين 

```
digitalWrite(LED_RED, LOW);
digitalWrite(LED_YELLOW, HIGH);
digitalWrite(LED_GREEN, LOW);
delay(1000);
```

<<----- ينطفئ LED الأحمر   
 <<----- يضيء LED الأصفر   
 <<----- ينطفئ LED الأخضر   
 <<----- تأخير لمدة ثانية 

}

<<----- 



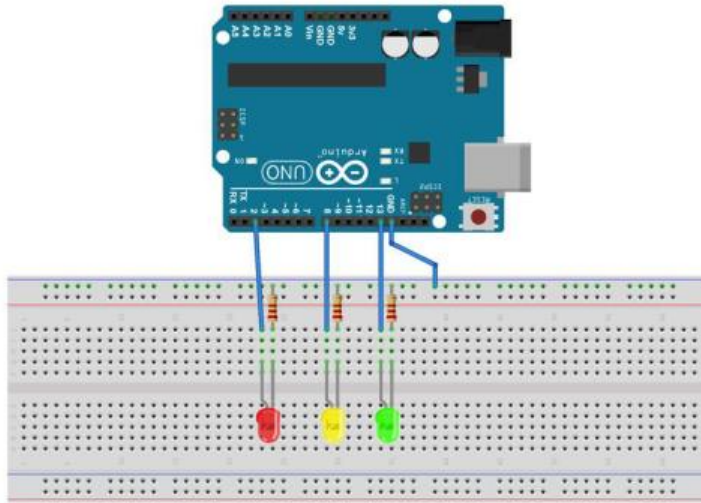


sketch\_jan23a

## Used Parts :

- Arduino UNO x1
- Breadboard x1
- LED Red x1
- LED Yellow x1
- LED Green x1
- 220 ohm x3

## توصيلة التجربة على اللوح التجريبي:



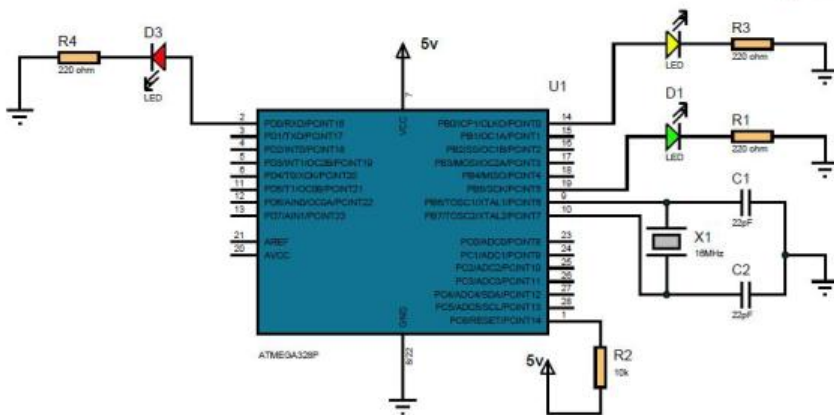
## الرسم الفيزيائي :

### Digital pins

- D0 = 2
- D1 = 3
- D2 = 4
- D3 = 5
- D4 = 6
- D5 = 11
- D6 = 12
- D7 = 13
- D8 = 14
- D9 = 15
- D10 = 16
- D11 = 17
- D12 = 18
- D13 = 19

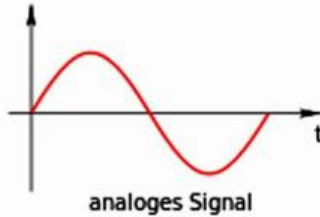
### Analog pins

- A0 = 23
- A1 = 24
- A2 = 25
- A3 = 26
- A4 = 27
- A5 = 28



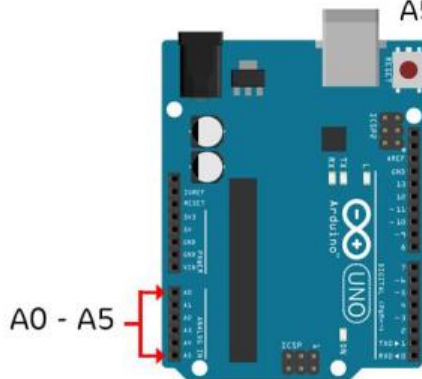
## ما المقصود بالإشارات التماثلية (Analog) ؟

هي إشارة مستمرة تتغير قيمتها مع الزمن . مثال: درجة الحرارة : يمكن أن تحتل عدد لا نهائي من القيم مثلاً 25 درجة ، 30 درجة ، 24.1 درجة وهكذا . ومعظم الظواهر الفيزيائية في الطبيعة كالضوء والصوت والضغط ودرجة الحرارة هي إشارات تماثلية.



تستخدم الإشارات التماثلية بشكل واسع في حياتنا اليومية ومن أمثلتها: صوت المذياع وأجهزة التلفاز وقياس التغيرات في المجسات الإلكترونية .

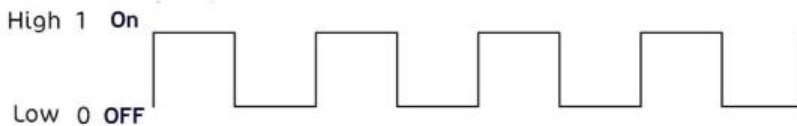
يمكننا من خلال 6 أرجل في لوحة الأردوينو قياس الإشارة التماثلية الصادرة عادة من المجسات عبر المداخل من A0 إلى A5





## ما المقصود بالاشارات الرقمية (Digital) ؟

هي اشارة متقطعة في الوقت ويكون لها عدد محدد من القيم. في أغلب الأحيان تحتل قيمتين فقط (مثلا 0 فولت و 5 فولت). والمثال الأوسع إنتشارا للإشارات الرقمية هو النظام الثنائي في العد المعتمد على القيمتين 1 و 0.



وتعرف موسوعة ويكيبيديا الإختلاف بين النوعين السابقين من الإشارات على أن: الإختلاف بين النظام الرقمي والتماثلي يكمن في نوعية وهئية الإشارة من حيث سعتها أو قيمتها و كذلك من حيث الزمن الذي تشغله. فالإشارة التماثلية يمكن أن تأخذ أي قيمة في زمن مستمر وغير متقطع بينما الرقمية لا تأخذ إلا إحدى القيم المتعارف عليها في النظام ضمن أزمنة مستمرة أو متقطعة.

مصدر الإشارة الرقمية هو دوماً تماثلي حيث أن الحياة حولنا تماثلية ولذلك يتم تحويل أي إشارة تماثلية إلى رقمية بواسطة تقنية وإجراءات معينة وبواسطة مبدل تماثلي رقمي.

## ما هي المتغيرات (Variable) ؟

المتغيرات هي رموز إستدلالية نضعها في المعادلات والعمليات الحسابية،  
مثال على ذلك  $5 = 2 + 3$

س =  $5 - 2 = 3$  . إذا إستدللنا بـ " س " كمتغير لنعوض بها عن قيمتها الحقيقية (3) في المعادلة.

في هذا الجدول توجد كل أنواع الذاكرة التي يمكنك استخدامها في المتغيرات أو المصفوفات :-

| النوع         | حجم الذاكرة       | المدى                                                     |
|---------------|-------------------|-----------------------------------------------------------|
| boolean       | 8-bit(1-byte)     | true or false                                             |
| byte          | 8-bit(1-byte)     | 0 to 255                                                  |
| unsigned char | 8-bit(1-byte)     | 0 to 255                                                  |
| char          | 8-bit(1-byte)     | -128 to 127                                               |
| unsigned int  | 16-bit (2-byte)   | 0 to 65,535                                               |
| int           | 16-bit (2-byte)   | -32,768 to 32,767                                         |
| word          | 16-bit (2-byte)   | 0 to 65535                                                |
| unsigned long | 32 bits (4 bytes) | 0 to 4,294,967,295                                        |
| long          | 32 bits (4 bytes) | -2,147,483,648 to 2,147,483,647                           |
| float         | 32 bits (4 bytes) | -3.4028235*10 <sup>38</sup> to 3.4028235*10 <sup>38</sup> |

لاحظ أن byte و char و word تحتوي على أرقام موجبة فقط وتسمى " unsigned variable "

أما الأنواع الأخرى فتحتوي على أرقام موجبة وسالبة وتسمى " signed variable " .

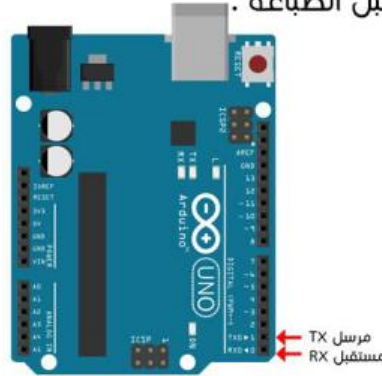


sketch\_jan23a

Serial Monitor

## نظام إرسال المعلومات المتتالي (UART): (universal asynchronous receiver/transmitter)

هو نظام إرسال وإستقبال بين المتحكم الدقيق وجهاز الحاسوب أو أية قطع أخرى . وكل أردوينو يحتوي على واحد على الأقل . لابد من تحديد سرعة إرسال وإستقبال البيانات بين الأردوينو والجانب الآخر قبل الطباعة .



يمكن مراقبة المعلومات الخارجة أو الداخلة للأردوينو من خلال نافذة المراقب التسلسلي (Serial Monitor) .



## الدرس الثامن:

# القيمة التماثلية ANALOG

يتطلب استخدام المراقب التسلسلي

### فكرة الدرس :

مجموعة كبيرة من المجسات الموجودة تعطينا قراءة تماثلية (تناظرية) (Analog).  
في هذا الدرس سنتعلم كيفية أخذ قراءة تماثلية وعرضها في المراقب التسلسلي .

### الأوامر

```
int sensorValue;  
int sensorPin=A5;
```

تعريف متغير باسم sensorValue <<-----  
ترميز الرجل A5 ب كلمة sensorPin <<-----

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

تفعيل شاشة المراقب التسلسلي <<-----

```
void loop () {  
  
  sensorValue=analogRead(sensorPin);  
  Serial.println(sensorValue);  
  delay(500);  
}
```

عمل حلقة غير منتهية <<-----  
قراءة القيمة التناظرية لقيمة المقاومة <<-----  
المتغيرة وتخزينها في متغير <<-----  
عرض المتغير على شاشة المراقب التسلسلي <<--  
تأخير لمدة نصف ثانية <<-----  
<<-----

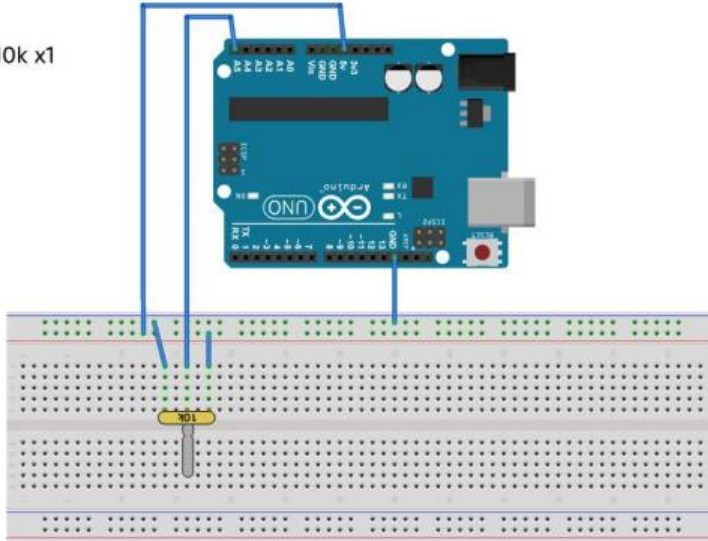


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
Variabel resistor 10k x1

## توصيلة التجربة على اللوح التجريبي:



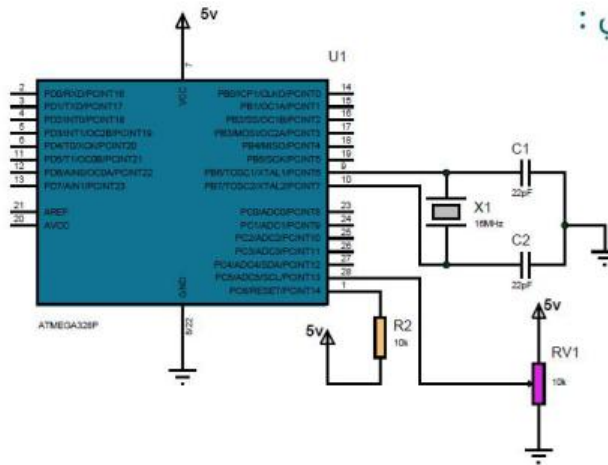
## Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

## Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28

## الرسم الفيزيائي :



## الدرس التاسع:

# ثلاثة أوامر مختلفة

يتطلب استخدام المراقب التسلسلي

Serial Monitor

## فكرة الدرس :

هنا تطبيق للإستخدام القراءة التماثلية في تشغيل ثلاثة أوامر مختلفة.

### الأوامر

```
int RED_LED = 2;
int YELLOW_LED = 8;
int GREEN_LED = 13;
int sensor;
int sensorPin=A5;

void setup() {
  pinMode(RED_LED,OUTPUT);
  pinMode(YELLOW_LED,OUTPUT);
  pinMode(GREEN_LED,OUTPUT);
  Serial.begin(9600);
}
void loop() {

  sensor=analogRead(sensorPin);
  Serial.println(sensor);
  delay(100);

  if((sensor>100) && (sensor<400))
  {
```

◆ تعريف متغير باسم sensor <<-----

◆ ترميز الرجل A5 ب كلمة sensorPin <<-----

--- تفعيل شاشة المراقب التسلسلي <<-----

🖨 قراءة القيمة التناظرية لقيمة المقاومة

المتغيرة وتخزينها في متغير <<-----

عرض المتغير على شاشة المراقب التسلسلي <<---

🚦 الجملة الشرطية إذا كانت القيمة أكبر من 100

وأصغر من 400 <<-----



sketch\_jan23a

## الأوامر

```
digitalWrite(LED_RED, HIGH);
digitalWrite(LED_YELLOW, LOW);
digitalWrite(LED_GREEN, LOW);
}
```

<<----- يضئ LED الأحمر   
 <<----- ينطفئ LED الأصفر   
 <<----- ينطفئ LED الأخضر 

الجملة الشرطية إذا كانت القيمة أكبر من 400 وأصغر من 800 <<----- 

```
else if((sensor >= 400) && (sensor < 800))
{
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_YELLOW, HIGH);
    digitalWrite(LED_GREEN, LOW);
}
```

<<----- ينطفئ LED الأحمر   
 <<----- يضئ LED الأصفر   
 <<----- ينطفئ LED الأخضر 

```
else if(sensor >= 800)
{
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_YELLOW, LOW);
    digitalWrite(LED_GREEN, HIGH);
}
```

الجملة الشرطية إذا كانت القيمة أكبر من 800 <<-- 

<<----- ينطفئ LED الأحمر   
 <<----- ينطفئ LED الأصفر   
 <<----- يضئ LED الأخضر 

```
else
{
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_YELLOW, LOW);
    digitalWrite(LED_GREEN, LOW);
}
```

إذا لم تتحقق أي من الجمل الشرطية <<---- 

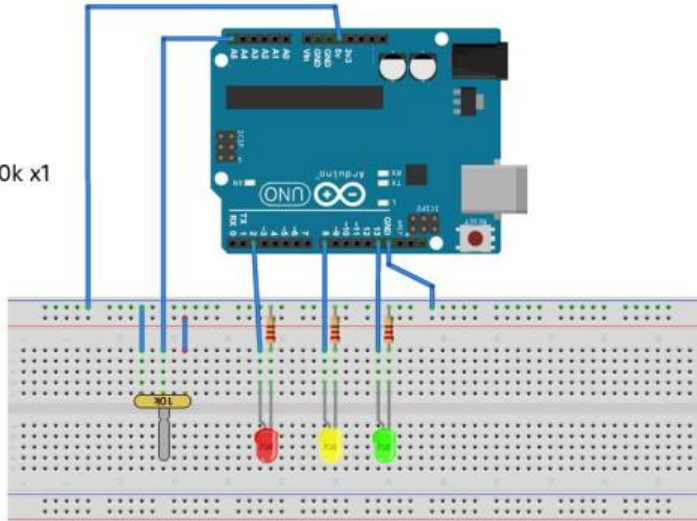
<<----- ينطفئ LED الأحمر   
 <<----- ينطفئ LED الأصفر   
 <<----- ينطفئ LED الأخضر 

<<----- 

## Used Parts :

Arduino UNO x1  
 Breadboard x1  
 LED Red x1  
 LED Yellow x1  
 LED Green x1  
 220 ohm x3  
 Variabel resistor 10k x1

## توصيلة التجربة على اللوح التجريبي:



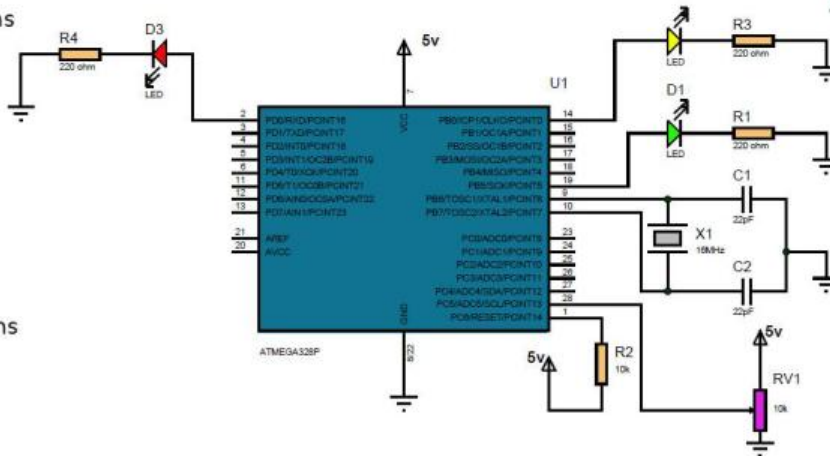
## Digital pins

D0 = 2  
 D1 = 3  
 D2 = 4  
 D3 = 5  
 D4 = 6  
 D5 = 11  
 D6 = 12  
 D7 = 13  
 D8 = 14  
 D9 = 15  
 D10 = 16  
 D11 = 17  
 D12 = 18  
 D13 = 19

## Analog pins

A0 = 23  
 A1 = 24  
 A2 = 25  
 A3 = 26  
 A4 = 27  
 A5 = 28

## الرسم الفيزيائي :





أنا متأكد بأنه يمكنك اجتياز هذه التحديات أيها المبرمج الطموح .

## التحدي رقم 1 :

تشغيل وإطفاء LED واحد بمفتاح واحد .  
بحيث أنه عند الضغط على المفتاح يضيء الـ LED بشكل مستمر حتى لو  
رفعت إصبعك من مفتاح الضغط وعند الضغط مرة أخرى ينطفئ الـ LED  
بشكل مستمر.

## التحدي رقم 2 :

لدينا ثلاثة مفاتيح ، عند الضغط على المفتاح الأول يضيء الـ LED بشكل  
مستمر وعند الضغط على المفتاح الثاني يضيء وينطفئ الـ LED بشكل  
متناوب والمفتاح الأخير يطفئ الـ LED .

#تحدى\_القرية ؟



## الدرس العاشر:

# المقاومة الضوئية LDR

يتطلب استخدام المراقب التسلسلي

### فكرة الدرس :

هل تتذكر المقاومة الضوئية أيها المبرمج ؟ لقد تعلمناها في كتاب المقدمة إلى عالم الإلكترونيات التطبيقية . المقاومة الضوئية تعطينا قراءة تماثلية أيضاً.

### الأوامر

```
int LDR;  
int sensorPin=A5;
```

◆ تعريف متغير باسم LDR <<-----

◆ ترميز الرجل A5 ب كلمة sensorPin <<-----

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

--- تفعيل شاشة المراقب التسلسلي <<-----

```
void loop () {  
  
  LDR=analogRead(sensorPin);  
  Serial.println(LDR);  
  delay(500);  
}
```

▲ قراءة القيمة التناظرية لقيمة المقاومة

الضوئية وتخزينها في متغير <<-----

عرض المتغير على شاشة المراقب التسلسلي <<---

⌚ تأخير لمدة نصف ثانية <<-----

Ⓢ <<-----

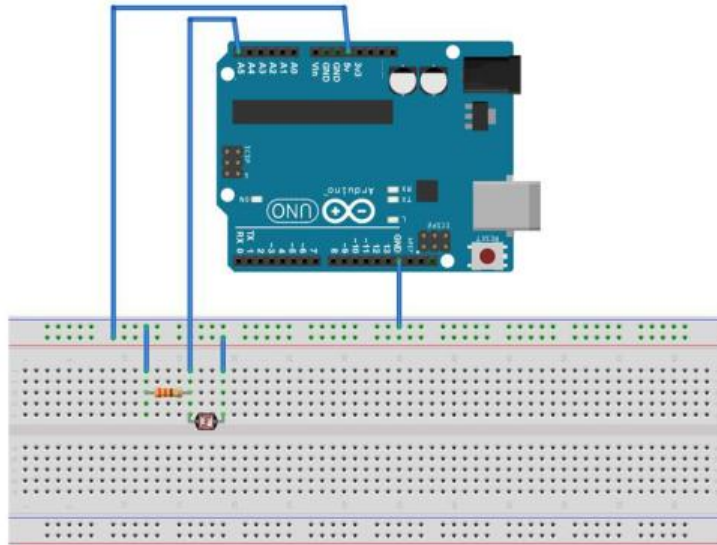


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
LDR x1  
220 ohm x1

## توصيلة التجربة على اللوح التجريبي:



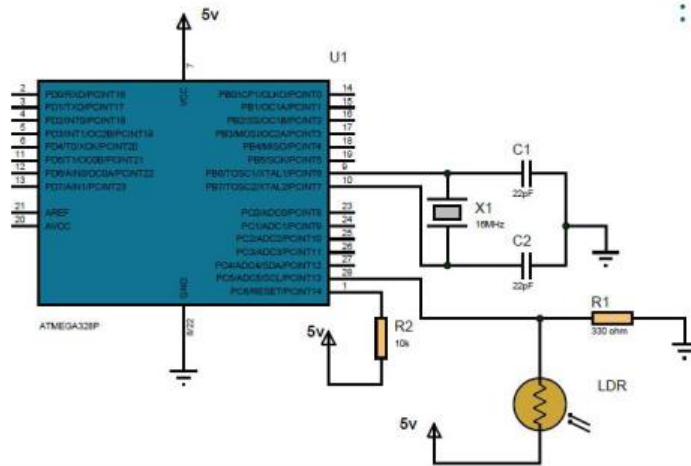
## الرسم الفيزيائي :

### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28



## الدرس الحادي عشر:

# الإشارة التلقائية

يتطلب استخدام المراقب التسلسلي

Serial Monitor

## فكرة الدرس :

ما رأيك في تطوير إنارة شوارع الحي الذي تسكن فيه أيها المبرمج؟  
من خلال هذا المشروع يمكنك ذلك.

## الأوامر

```
int WHITE_LED = 12;  
int LDR;  
int sensorPin=A5;
```

<<----- LDR تعريف متغير باسم  
<<----- sensorPin ترميز الرجل A5 ب كلمة

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

--- تفعيل شاشة المراقب التسلسلي <<-----

```
void loop() {  
  LDR=analogRead(sensorPin);  
  Serial.println(LDR);  
  delay(100);
```

```
if (LDR<140) {  
  digitalWrite(WHITE_LED,HIGH); }  
else {  
  digitalWrite(WHITE_LED,LOW); }  
}
```

الجملة الشرطية إذا كانت القيمة أصغر من 140 <<--

تضيء الإنارة في حالة الإظلام <<-----

وتنطفئ في حالة وجود الضوء <<-----

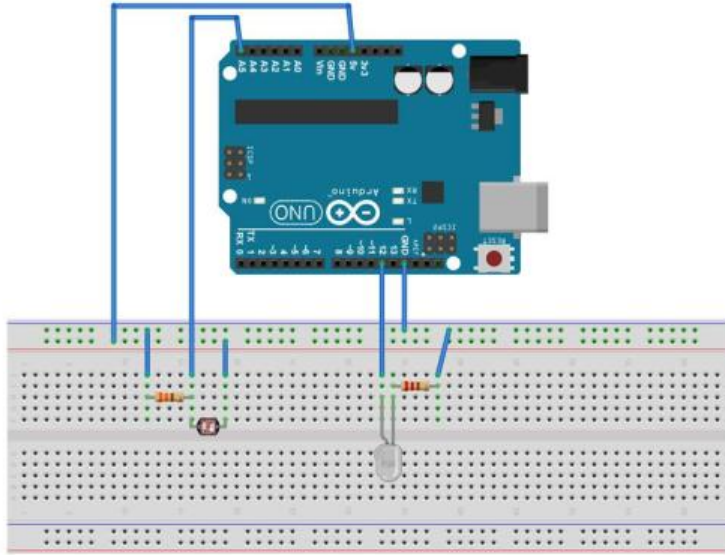


sketch\_jan23a

## Used Parts :

- Arduino UNO x1
- Breadboard x1
- LED White x1
- LDR x1
- 220 ohm x1
- 330 ohm x1

## توصيلة التجربة على اللوح التجريبي:



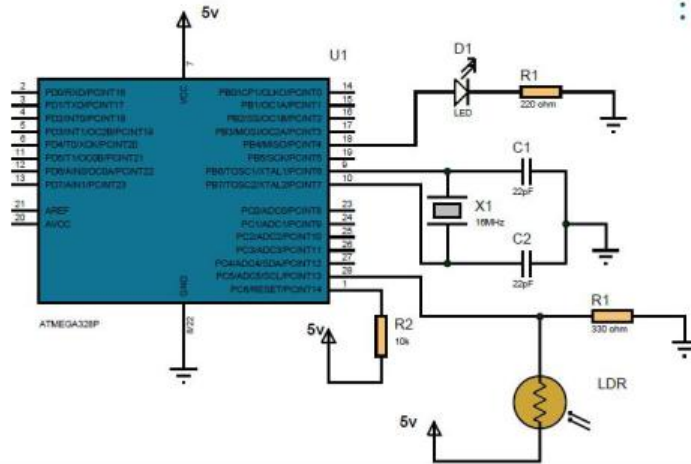
## الرسم الفيزيائي :

### Digital pins

- D0 = 2
- D1 = 3
- D2 = 4
- D3 = 5
- D4 = 6
- D5 = 11
- D6 = 12
- D7 = 13
- D8 = 14
- D9 = 15
- D10 = 16
- D11 = 17
- D12 = 18
- D13 = 19

### Analog pins

- A0 = 23
- A1 = 24
- A2 = 25
- A3 = 26
- A4 = 27
- A5 = 28



## الدرس الثاني عشر:

# حساس الصوت

يتطلب استخدام المراقب التسلسلي

Serial Monitor

### فكرة الدرس :

مجسة الصوت تعطينا أيضا قراءة تماثلية. يمكنك استخدامها في مشاريع كثيرة أيها المبرمج.

### الأوامر

```
int SOUND;  
int sensorPin=A5;
```

تعريف متغير باسم SOUND <<-----

ترميز الرجل A5 ب كلمة sensorPin <<-----

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

تفعيل شاشة المراقب التسلسلي <<-----

```
void loop () {  
  
  SOUND=analogRead(sensorPin);  
  Serial.println(SOUND);  
  delay(500);  
}
```

قراءة القيمة التناظرية لقيمة لحساس الصوت

وتخزينها في متغير <<-----

عرض المتغير على شاشة المراقب التسلسلي <<--

<<-----



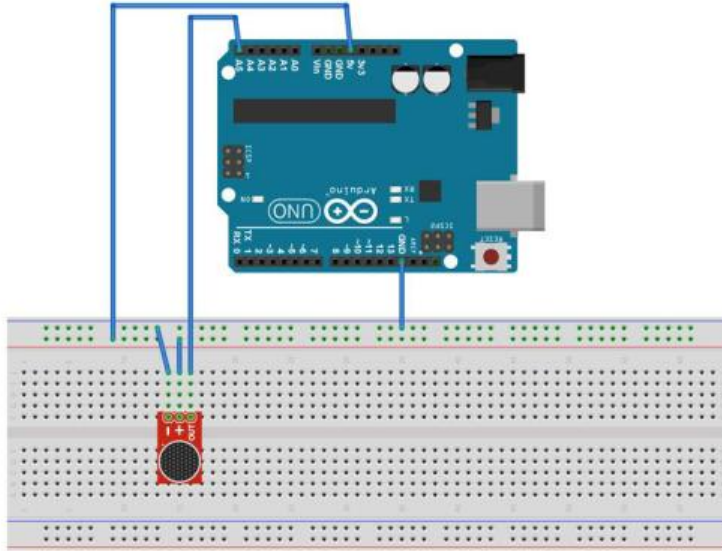


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
Sound sensor x1

## توصيلة التجربة على اللوح التجريبي:



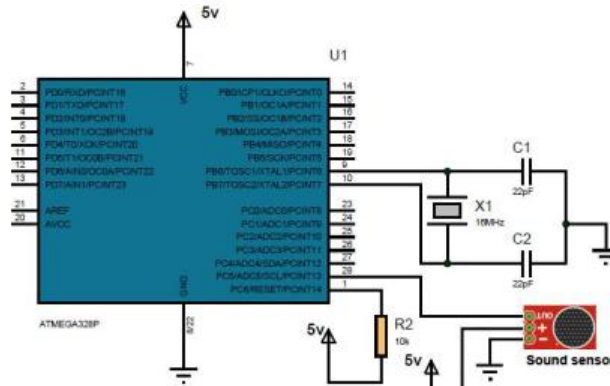
## الرسم الفيزيائي :

### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28



## الدرس الثالث عشر:

# جهاز مستوى الماء

يتطلب استخدام المراقب التسلسلي

### فكرة الدرس :

تتطلب بعض التطبيقات الحياتية معرفة مستوى الماء في الخزان. في هذا التطبيق يمكنك ذلك أيها المبرمج.

### الأوامر

```
int RED_LED = 2;
int YELLOW_LED = 8;
int GREEN_LED = 13;
int WATER;
int sensorPin=A5;

void setup() {
  pinMode(RED_LED,OUTPUT);
  pinMode(YELLOW_LED,OUTPUT);
  pinMode(GREEN_LED,OUTPUT);
  Serial.begin(9600);
}
void loop() {
  WATER=analogRead(sensorPin);
  Serial.println(WATER);
  delay(100);

  if((WATER>60) && (WATER<100))
  {
```

◆ تعريف متغير باسم WATER <<-----

◆ ترميز الرجل A5 بـ كلمة sensorPin <<-----

--- تفعيل شاشة المراقب التسلسلي <<-----

🖨 قراءة القيمة التناظرية لقيمة المقاومة

المتغيرة وتخزينها في متغير <<-----

عرض المتغير على شاشة المراقب التسلسلي <<---

🚦 الجملة الشرطية إذا كانت القيمة أكبر من 60

وأصغر من 100 <<-----



sketch\_jan23a

## الأوامر

```
digitalWrite(LED_RED, HIGH);
digitalWrite(LED_YELLOW, LOW);
digitalWrite(LED_GREEN, LOW);
}
```

<<----- يضئ LED الأحمر   
 <<----- ينطفئ LED الأصفر   
 <<----- ينطفئ LED الأخضر 

الجملة الشرطية إذا كانت القيمة أكبر من 100 

وأصغر من 160 <<----- ((WATER>=100) && (WATER<160))

```
{
digitalWrite(LED_RED, LOW);
digitalWrite(LED_YELLOW, HIGH);
digitalWrite(LED_GREEN, LOW);
}
```

<<----- ينطفئ LED الأحمر   
 <<----- يضئ LED الأصفر   
 <<----- ينطفئ LED الأخضر 

else if (WATER>=160)

<<-- الجملة الشرطية إذا كانت القيمة أكبر من 160 

```
{
digitalWrite(LED_RED, LOW);
digitalWrite(LED_YELLOW, LOW);
digitalWrite(LED_GREEN, HIGH);
}
```

<<----- ينطفئ LED الأحمر   
 <<----- ينطفئ LED الأصفر   
 <<----- يضئ LED الأخضر 

else

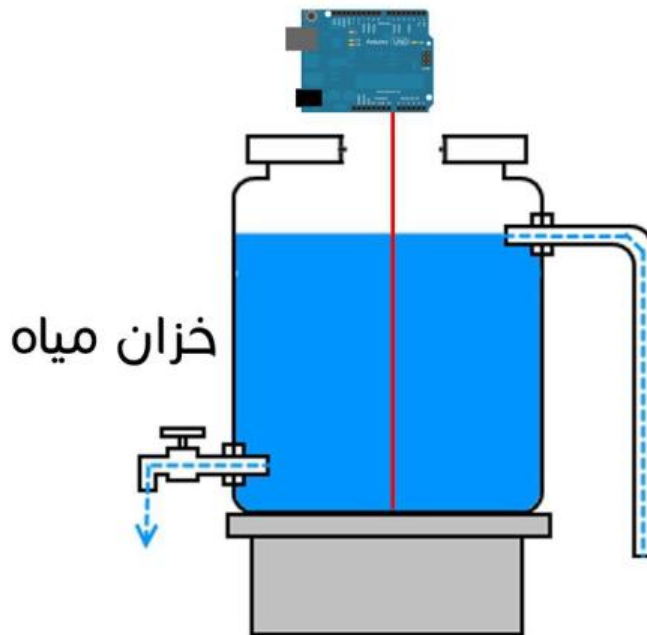
<<---- إذا لم تتحقق أي من الجمل الشرطية 

```
{
digitalWrite(LED_RED, LOW);
digitalWrite(LED_YELLOW, LOW);
digitalWrite(LED_GREEN, LOW);
}
```

<<----- ينطفئ LED الأحمر   
 <<----- ينطفئ LED الأصفر   
 <<----- ينطفئ LED الأخضر 

<<----- 

## قطعة الأردوينو العجيبة



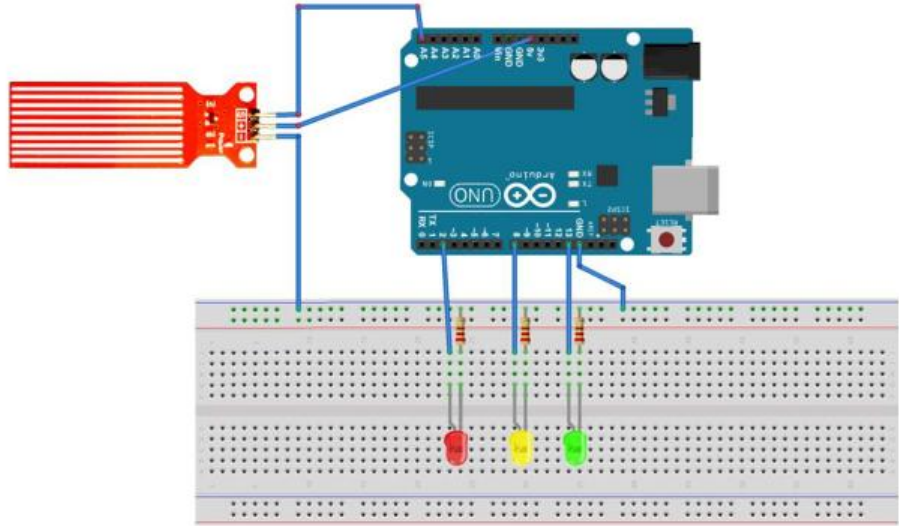


sketch\_jan23a

## Used Parts :

- Arduino UNO x1
- Breadboard x1
- LED Red x1
- LED Yellow x1
- LED Green x1
- 220 ohm x3
- Water sensor x1

## توصيلة التجربة على اللوح التجريبي:



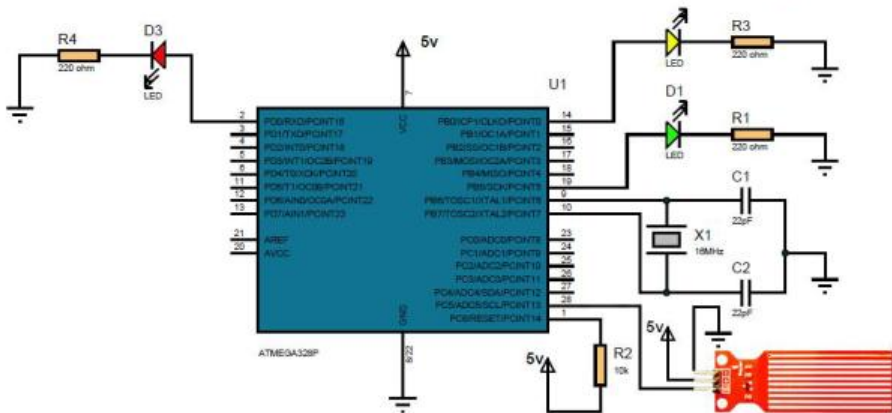
## الرسم الفيزيائي :

### Digital pins

- D0 = 2
- D1 = 3
- D2 = 4
- D3 = 5
- D4 = 6
- D5 = 11
- D6 = 12
- D7 = 13
- D8 = 14
- D9 = 15
- D10 = 16
- D11 = 17
- D12 = 18
- D13 = 19

### Analog pins

- A0 = 23
- A1 = 24
- A2 = 25
- A3 = 26
- A4 = 27
- A5 = 28



## الدرس الرابع عشر:

# مقياس درجة الحرارة

يتطلب استخدام المراقب التسلسلي

### فكرة الدرس :

قياس درجة الحرارة هو أحد التطبيقات الممتعة في برمجة الأردوينو. في هذا الدرس ستقوم بأخذ درجة الحرارة وعرضها في شاشة المراقب التسلسلي.

### الأوامر

```
float temp;  
int tempPin = A0;
```

◆ تعريف متغير باسم temp <<-----  
◆ ترميز الرجل A0 بكلمة tempPin <<-----

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

--- تفعيل شاشة المراقب التسلسلي <<-----

```
void loop()  
{  
  temp = analogRead(tempPin);  
  
  temp = temp * 0.48828125;  
  Serial.print("TEMPRATURE = ");  
  Serial.print(temp);  
  Serial.print(" C");  
  Serial.println();  
  delay(1000);  
}
```

X/= هذه المعادلة لتحويل درجة الحرارة إلى  
درجة سيليزية <<-----  
كتابة كلمة TEMPRATURE على المراقب <<  
عرض درجة الحرارة على المراقب <<-----  
كتابة حرف C <<-----  
هذا الأمر يدل عمل سطر جديد (ENTER) <<

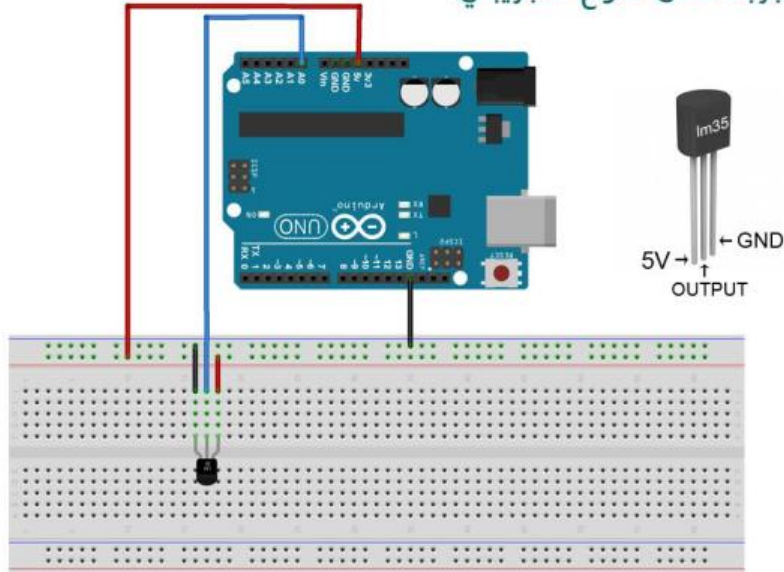


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
LM35Dz x1

## توصيلة التجربة على اللوح التجريبي:



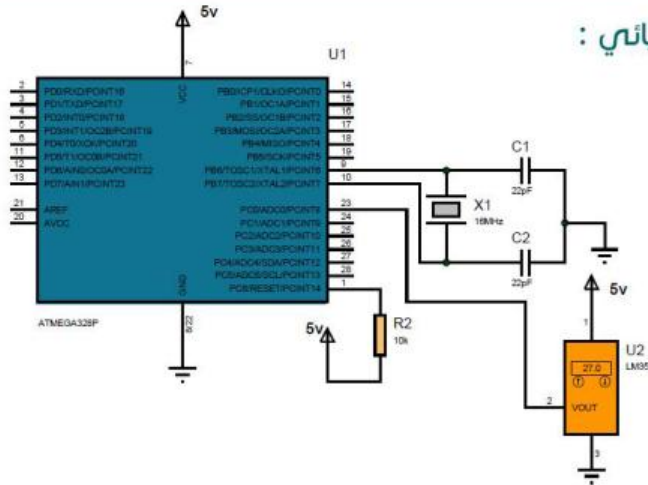
## Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

## Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28

## الرسم الفيزيائي :



## الدرس الخامس عشر:

# عصا التحكم Joystick

Serial Monitor يتطلب استخدام المراقب التسلسلي

### فكرة الدرس :

هل سبق وأن لعبت لعبة بلاي ستيشن أيها المبرمج؟ عصا التحكم (Joystick) موجودة في أجهزة الألعاب. في هذا الدرس سنتعلم كيفية التعامل معها.

### الأوامر

```
int X=0 , Y=0 , KEY=0;
int Z=2;
```

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

```
void loop(){
  X = analogRead(A0);
  Y = analogRead(A1);
  KEY=digitalRead(Z);

  Serial.print("x = ");
  Serial.println(X, DEC);
  Serial.print("y = ");
  Serial.println(Y, DEC);
  Serial.print("key= ");
  Serial.println(KEY, DEC);
  Serial.println();
  delay(500);
}
```

◆ تعريف وتصفير المتغيرات <<-----

◆ ترميز الرجل 12 بحرف Z <<-----

--- تفعيل شاشة المراقب التسلسلي <<-----

⌨ قراءة قيمة A0 وتخزينها في المتغير X <<---

⌨ قراءة قيمة A1 وتخزينها في المتغير Y <<---

⌨ قراءة قيمة Z وتخزينها في المتغير KEY <<---

⌨ كتابة X على المراقب التسلسلي <<-----

⌨ عرض قيمة X على المراقب التسلسلي <<

⌨ كتابة Y على المراقب التسلسلي <<-----

⌨ عرض قيمة Y على المراقب التسلسلي <<

⌨ كتابة Key على المراقب التسلسلي <<---

⌨ عرض قيمة KEY على المراقب التسلسلي <<---

⌨ هذا الأمر يدل عمل سطر جديد (ENTER) <<

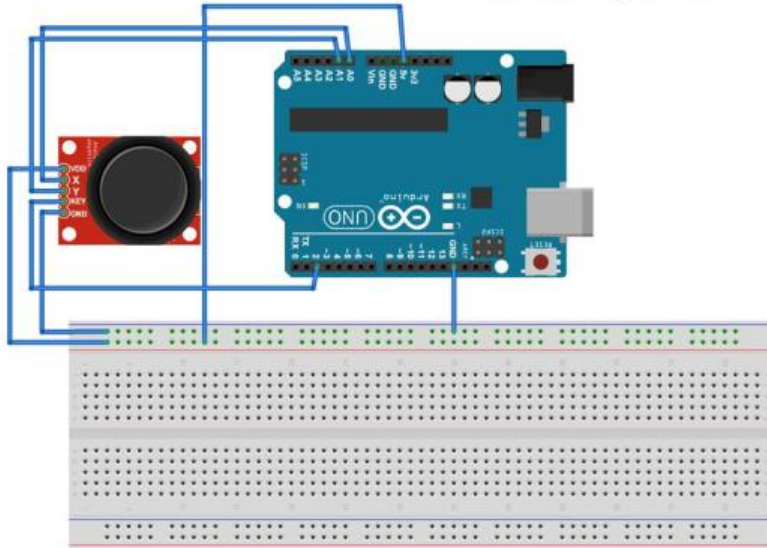


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
Joystick x1

## توصيلة التجربة على اللوح التجريبي:



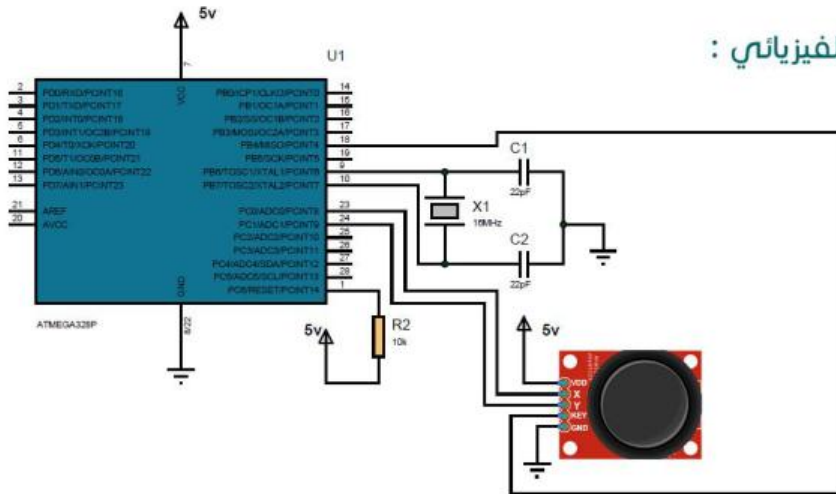
## Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

## Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28

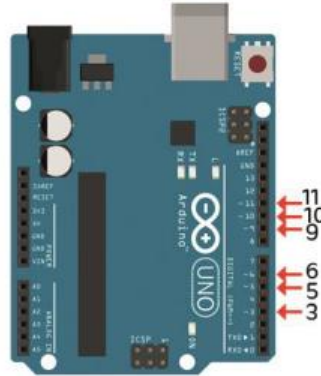
## الرسم الفيزيائي :



## ما هي خاصية التعديل الرقمي PWM ؟ (Pulse Width Modulation)

درسنا سابقاً أن الإشارة الرقمية إما أن تكون قيمتها 0 أو 5 . والإشارة التماثلية تعطينا قيم مختلفة بين 0-5 . الأردوينو يستقبل الإشارات التماثلية ولكن الإشارات التي تخرج منه دائماً تكون رقمية . في بعض التطبيقات نحتاج إلى خرج تماثلي، هنا يأتي دور خاصية التعديل.

سؤال : هل يمكننا أن نحصل على قيم مختلفة من الإشارة الرقمية ؟ الإجابة: نعم يمكننا ذلك عن طريق خاصية PWM . وهذه الخاصية موجودة في 6 منافذ (3,5,6,9,10,11) من أصل 14 منفذ. من هذه الخاصية يمكننا التحكم في درجة إضاءة الوصلة الثنائية الضوئية وأيضاً سرعة محرك السيرفو.



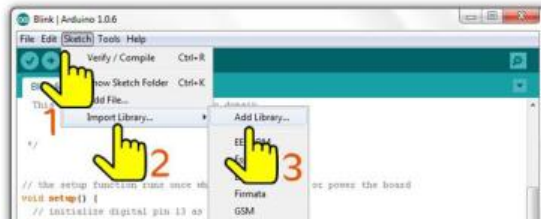
عندما نعطي قيمة التعديل الرقمي PWM قيمة 255 في البرنامج فإن ذلك يدل على خروج 5 فولت من المنفذ وفي حالة كانت القيمة 127 فإن قيمة الجهد الخارج من المنفذ 2.5 فولت و صفر فولت لو كانت القيمة صفر وهكذا . ستتعرف أكثر على هذه الخاصية وتطبيقاتها لاحقاً في تجارب هذا الكتاب.



## مكتبات الأردوينو (Library)

للأردوينو ملحقات تساعدك في تنفيذك للمشاريع ، ولكل ملحق أو تطبيق مكتبة خاصة به. ولكن ما المقصود بالمكتبة ؟ المكتبة في هذا السياق هي عبارة عن مجموعة من الأوامر البرمجية الخاصة بتطبيق معين (مثل محرك السيرفو) التي تم تجهيزها مسبقا وليس عليك سوى كتابة أمر برمجي واحد لإستدعاءها. ونقصد هنا أنه يمكنك تخيل أن لدينا مجموعة كبيرة من الأوامر التي تم دمجها في عدة كلمات فقط ، يكفي أن نقوم بإستدعاء المكتبة والتعامل معها ، أيضاً يمكن استخدام أكثر من ملحق ومكتبة في مشروع واحد .

- هناك نماذج من المكتبات موجودة في البيئة التطويرية للأردوينو منذ البداية مثل :
- (EEPROM) تستخدم هذه المكتبة للقراءة والكتابة في حالة التخزين الدائم .
  - (Ethernet) تستخدم هذه المكتبة لربط الأردوينو بالانترنت عن طريق الكابل .
  - (GSM) تستخدم هذه المكتبة لربط الأردوينو بـ GSM .
  - (SD) تستخدم هذه الخاصة لقراءة وكتابة بطاقة الذاكرة .
  - (SERVO) تستخدم هذه المكتبة للتحكم بمحرك السيرفو .
  - (LiquidCrystal) تستخدم هذه المكتبة للتحكم بشاشات LCD
  - (SPI) تستخدم هذه المكتبة للتواصل مع الأجهزة المختلفة
  - (STEPPER) تستخدم هذه المكتبة للتحكم بمحرك Stepper .



كما يمكنك إضافة مكتبات أخرى من خلال الخطوة التالية :

ملاحظة : يمكن تنزيل جميع المكتبات المستخدمة في هذا الكتاب من موقعنا [www.ev-center.com](http://www.ev-center.com)

## الدرس السادس عشر:

# محرك السيرفو servo motor

يتطلب استخدام مكتبة

### فكرة الدرس :

محركات السيرفو لها تطبيقات عديدة في حياتنا اليومية مثل الأبواب الإلكترونية وأجنحة الطائرات والرجال الآليين ومن المهم أن نتعلم كيفية التعامل معها أيها المبرمج.

### الأوامر

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

◆ إضافة مكتبة Servo إلى البرنامج <<-----  
◆ ربط مكتبة Servo بكلمة myservo <<-----

```
void setup()
{
  myservo.attach(3);
}
```

+ تحديد الرجل رقم 3 كمخرج للإشارة المحرك <<

```
void loop()
{

  myservo.write(0);
  delay(2000);
  myservo.write(90);
  delay(2000);
  myservo.write(180);
  delay(2000);
}
```

✂ أمر أن يتحرك المحرك للزاوية 0 <<-----

✂ أمر أن يتحرك المحرك للزاوية 90 <<-----

✂ أمر أن يتحرك المحرك للزاوية 180 <<-----

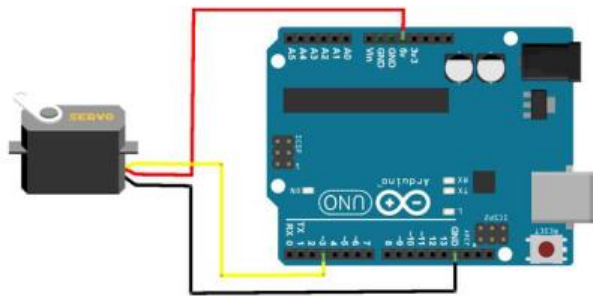


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Servo motor 180° x1

## توصيلة التجربة على اللوح التجريبي:



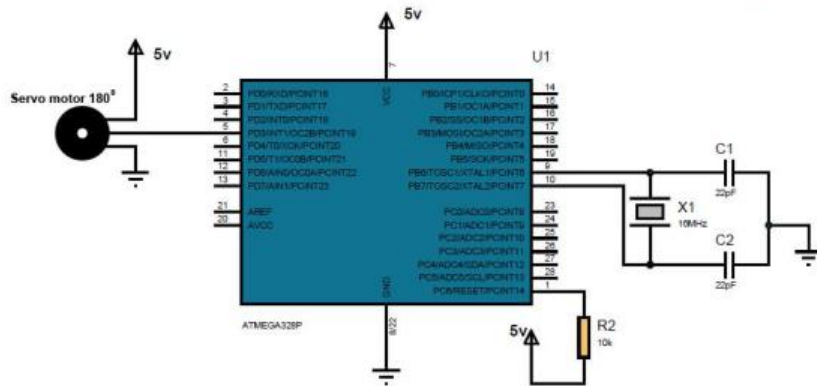
## الرسم الفيزيائي :

### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28



## الدرس السابع عشر:

# تحديد عدد الدورات for

يتطلب استخدام مكتبة

يتطلب استخدام المراقب التسلسلي Serial Monitor

### فكرة الدرس :

والآن سنقوم بتحديد عدد الدورات التي يجب أن يدورها محرك السيرفو . يمكنك استخدام هذا التطبيق مع البكرات بحيث تدور البكرة عددا معينا من الدورات بحسب طول الحبل .

### الأوامر

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

```
Servo myservo;
```

```
int pos = 0;
```

```
void setup() {  
  myservo.attach(3);  
  Serial.begin(9600);  
}
```

```
void loop() {  
  for (pos = 0; pos < 180; pos++)  
  {  
    myservo.write(pos);  
    Serial.print("position: ");  
    Serial.print(pos);  
    Serial.println(" degree");  
    delay(15);  
  }  
}
```

◆ إضافة مكتبة Servo إلى البرنامج <<-----

◆ ربط مكتبة Servo بكلمة myservo <<-----

<<----- تعريف متغير باسم pos

+ تحديد الرجل رقم 3 كمخرج للإشارة المحرك <<

🔄 حلقة تتكرر 180 مرة

يبدأ المتغير من 0 إلى 179 وفي كل مرة

يزداد بواحد <<-----

📏 تتحرك زاوية المحرك بنفس قيمة المتغير <<-

📄 كتابة المتغير على المراقب التسلسلي <<-----



sketch\_jan23a

## الأوامر

```
for (pos = 0; pos < 180; pos--)
{
  myservo.write(pos);
  Serial.print("position: ");
  Serial.print(pos);
  Serial.println(" degree");
  delay(15);
}
}
```

حلقه تكرر 180 مرة يبدأ المتغير من 0 إلى 179 وفي كل مرة يتناقص بواحد <<

تتحرك زاوية المحرك بنفس قيمة المتغير - <<

كتابة المتغير على المراقب التسلسلي <<----

هل تزداد أم تتناقص      متى تتوقف الحلقة      قيمة البداية للمتغير

```
for (pos = 0; pos < 180; pos--)
{
```

الشيء المراد عمله في كل دورة ←

```
}
```





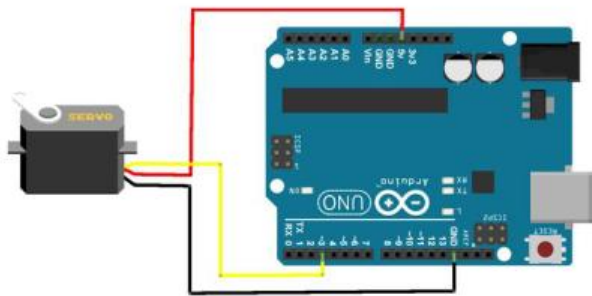
sketch\_jan23a

## Used Parts :

Arduino UNO x1

Servo motor 180° x1

## توصيلة التجربة على اللوح التجريبي:



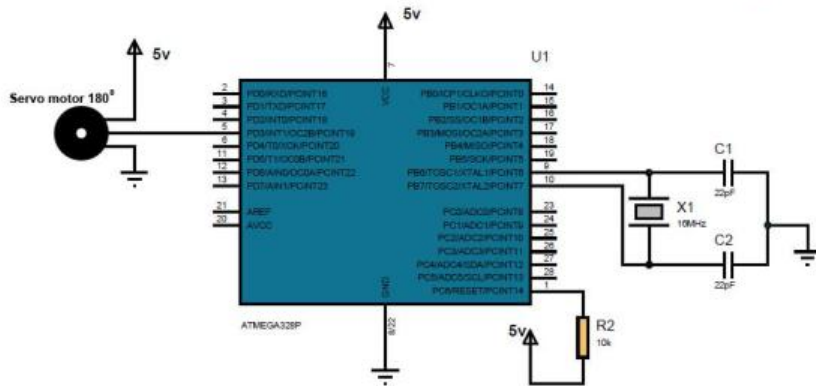
## الرسم الفيزيائي :

### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28



## الدرس الثامن عشر: تشغيل LED بالتدريج من خلال PWM

### فكرة الدرس :

في الصفحة 84 تحدثنا عن خاصية التعديل PWM والتي تسمح بتحويل الإشارة الرقمية إلى إشارة تماثلية. هنا نعطي مثالا عمليا على ذلك. الإشارة الرقمية ستشغل الـ LED مباشرة ولكن مع خاصية التعديل سنتمكن من التحكم بدرجة الإضاءة بالزيادة والنقصان.

### الأوامر

```
int LED = 11;  
int fade = 0;
```

```
void setup() {  
}
```

```
void loop() {  
  
  for (fade = 0; fade <= 255; fade +=3) {  
    analogWrite (LED, fade);  
    delay(20);  
  }  
  
  for (fade = 255; fade >= 0; fade -=3) {  
    analogWrite (LED, fade);  
    delay(20);  
  }  
}
```

<<----- يتطلب اختيار رجل بها خاصية PWM  
تعريف متغير باسم fade <<◆

حلقة تتكرر 255 مرة  
يبدأ المتغير من 0 إلى 255 وفي كل مرة  
يزداد بثلاثة <<-----  
تعتمد قوة إضاءة LED على قيمة fade <<-

حلقة تتكرر 255 مرة  
يبدأ المتغير من 255 إلى 0 وفي كل مرة  
يتناقص بثلاثة <<-----  
تعتمد قوة إضاءة LED على قيمة fade <<-

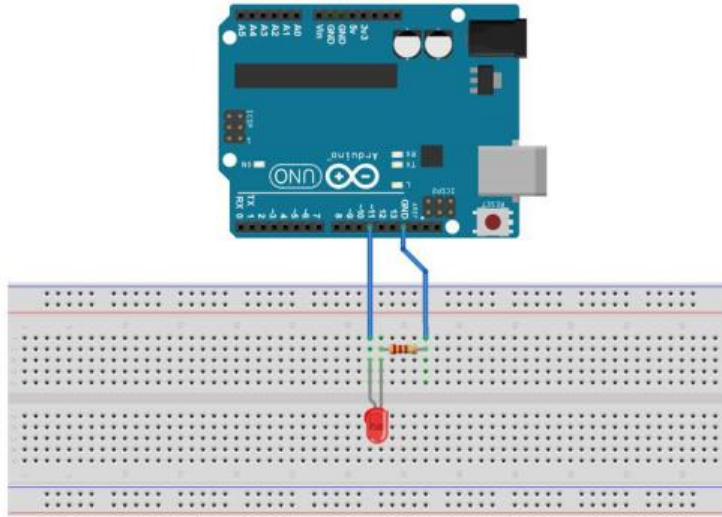


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
LED Red x1  
220 ohm x1

## توصيلة التجربة على اللوح التجريبي:



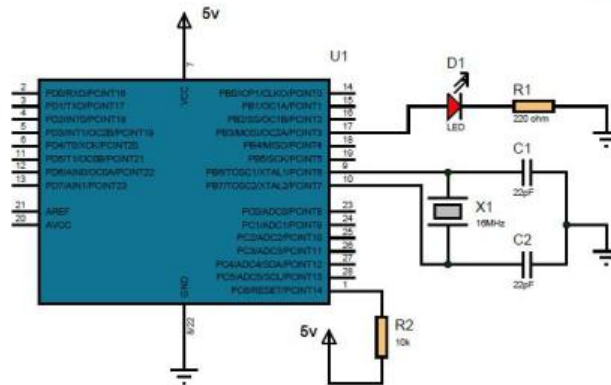
## الرسم الفيزيائي :

### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28



## الدرس التاسع عشر:

### مجسة الموجات فوق الصوتية ULTRASONIC

يتطلب استخدام مكتبة

يتطلب استخدام المراقب التسلسلي Serial Monitor

#### فكرة الدرس :

مجسة الموجات فوق الصوتية لها الكثير من التطبيقات المفيدة وتستخدم في تحسس الحركة مثل البوابات التي تفتح تلقائياً على مداخل المحلات التجارية وتحديد المواقع الفارغة. في هذا الدرس سنقوم بتجربة التعامل معها في عالم الأردوينو .

#### الأوامر

```
#include "Ultrasonic.h"
```

◆ إضافة مكتبة ULTRASONIC إلى البرنامج <<

```
Ultrasonic ultrasonic(9,8);
```

◆ تحديد أرجل ULTRASONIC (Trig,Echo) <<-----

```
int CMM;
```

◆ تعريف متغير باسم CMM <<-----

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

```
void loop()
```

```
{  
  CMM = ultrasonic.Ranging(CM);  
  Serial.print(CMM);  
  Serial.println("cm");  
  delay(50);  
}
```

~> قراءة قيمة المسافة وتخزينها في المتغير CMM <<-

~> عرض المتغير على المراقب <<-----

~> كتابة كلمة CMM على المراقب + ENTER - <<



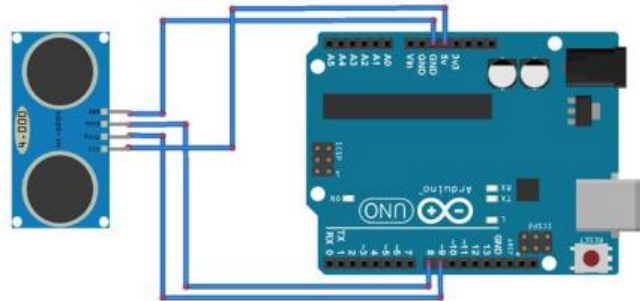
sketch\_jan23a

## Used Parts :

Arduino UNO x1

Ultrasonic sensor x1

## توصيلة التجربة على اللوح التجريبي:



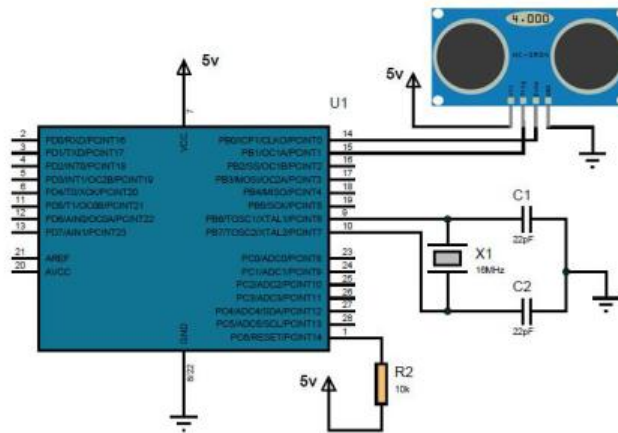
## Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

## Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28

## الرسم الفيزيائي :



## الدرس العشرون:

### جهاز إنذار للمسافة

يتطلب استخدام مكتبة

#### فكرة الدرس :

من تطبيقات الألتراسونيك الشهيرة استخدامها في نظام التنبيه بالرجوع للخلف بالسيارة والذي يصدر تنبيهاً في حالة وجود عائق . يعتمد هذا النظام على قياس المسافة باستخدام مجسة الموجات فوق الصوتية . في هذا الدرس سنقوم بتقليد هذا التطبيق .

#### الأوامر

```
#include "Ultrasonic.h"

Ultrasonic ultrasonic(8,9);
int cm;
int BUZZER =13;

void setup() {
  pinMode(BUZZER , INPUT); }
void loop() {
  cm= ultrasonic.Ranging(cm);

  if (cm>200) {
    digitalWrite (BUZZER,HIGH);
    delay(100);
    digitalWrite (BUZZER,LOW);
    delay(100);
  }
  else {
    digitalWrite (BUZZER,LOW);
  }
}
```

- ◆ إضافة مكتبة ULTRASONIC إلى البرنامج <<
- ◆ تحديد أرجل ULTRASONIC (Trig,Echo) <<-----
- ◆ تعريف متغير باسم CMM <<-----

🔊 قراءة قيمة المسافة وتخزينها في المتغير CMM - <<  
الجملة الشرطية في حالة إقتراب جسم من  
مجسة ULTRASONIC <<-----

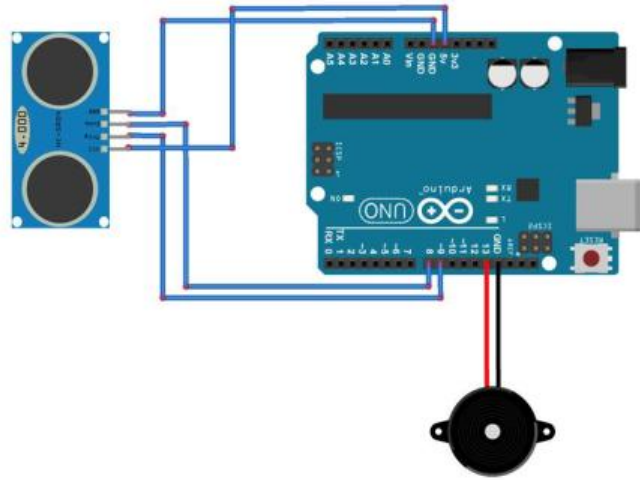


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Ultrasonic sensor x1  
Buzzer x1

## توصيلة التجربة على اللوح التجريبي:



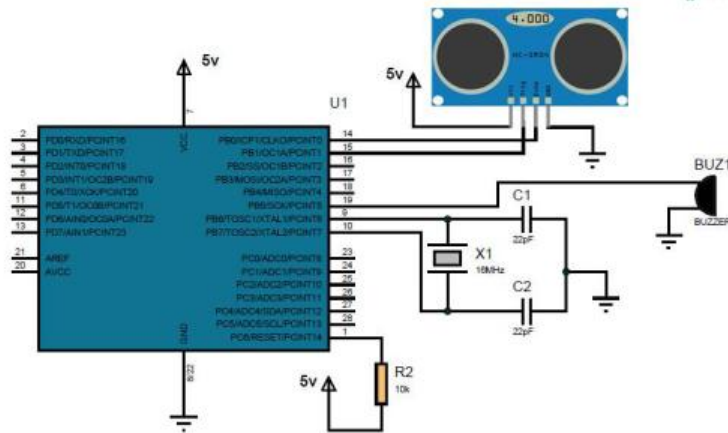
## الرسم الفيزيائي :

### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28



## الدرس الحادي والعشرون:

### لوحة المفاتيح 4x4 Keypad 4x4

يتطلب استخدام مكتبة

يتطلب استخدام المراقب التسلسلي Serial Monitor

#### فكرة الدرس :

لوحة المفاتيح هي أحد القطع الأكثر استخداماً في المشاريع الإلكترونية ، سنقوم بتعلم كيفية التعامل معها في هذا الدرس يا فصيح .

#### الأوامر

```
#include <Keypad.h>
```

◆ إضافة مكتبة Keypad إلى البرنامج <<-----

```
const byte ROWS = 4;
```

◆ أربعة صفوف <<-----

```
const byte COLS = 4;
```

◆ أربعة أعمدة <<-----

```
char hexaKeys[ROWS][COLS] = {
```

◆ تعريف رموز المفاتيح في اللوحة <<-----

```
{ '1','2','3','A'},
```

```
{ '4','5','6','B'},
```

```
{ '7','8','9','C'},
```

```
{ '*', '0', '#', 'D' }
```

```
};
```

```
byte rowPins[ROWS] = {3, 4, 5, 6};
```

◆ أرقام الأرجل التي يتم توصيلها مع الصفوف <<

```
byte colPins[COLS] = {7, 8, 9, 10};
```

◆ أرقام الأرجل التي يتم توصيلها مع الأعمدة <<

```
Keypad customKeypad = Keypad( makeKeymap(hexaKeys),
```

```
rowPins, colPins, ROWS, COLS);
```

```
char customKey;
```

◆ تعريف متغير باسم customKey <<-----

```
void setup() {
```

```
←
```



sketch\_jan23a

## الأوامر

```
Serial.begin(9600);
}

void loop ()
{
  customKey = customKeypad.getKey();

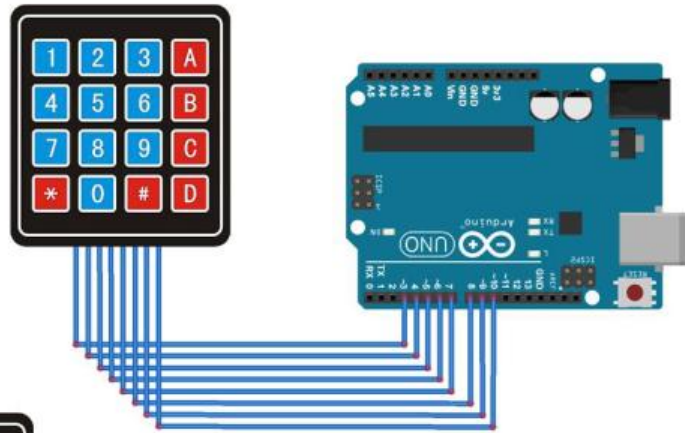
  if (customKey != NO_KEY)
    Serial.println (customKey);
}
```

قراءة رمز المفتاح الذي تم ضغطه وتخزين  
الرمز في متغير <<-----  
الجملة الشرطية المرتبطة بعرض الرقم في  
حالة الضغط على أي مفتاح <<-----  
كتابة المتغير على المراقب التسلسلي <<----

## Used Parts :

Arduino UNO x1  
Keypad 4x4 x1

## توصيلة التجربة على اللوح التجريبي:



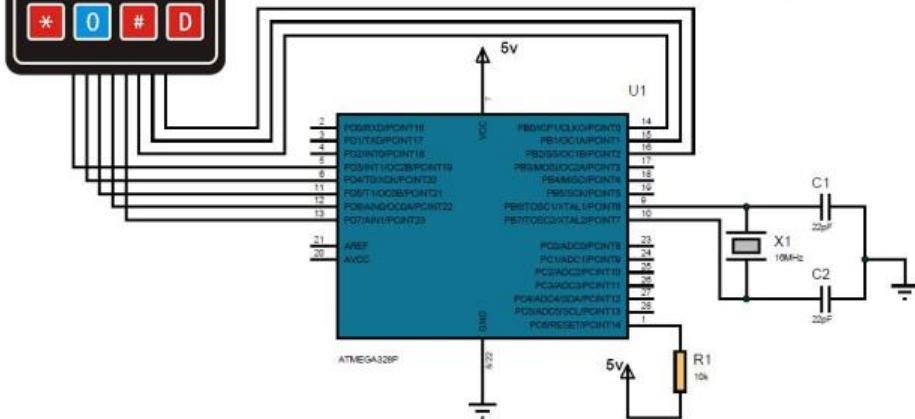
### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28

## الرسم الفيزيائي :





عدنا من جديد أيها المبرمج ، المزيد من التحديات في إنتظارك .

التحدي رقم 3 :

التحكم بشدة إضاءة LED من خلال مفاتيح ، واحد للزيادة والآخر للنقصان .

التحدي رقم 4 :

كلما اقترب الجسم من مجسة الموجات فوق الصوتية (Ultrasonic) يزداد صوت الطنان (Buzzer) .

ما هي الخاصية التي يجب عليك استخدامها هنا ؟

#تتحدى\_القرية ؟



## الدرس الثاني والعشرون:

### تشغيل وإطفاء LED بواسطة لوحة المفاتيح

يتطلب استخدام مكتبة



يتطلب استخدام المراقب التسلسلي Serial Monitor

### فكرة الدرس :

يعتبر هذا الدرس تطبيقاً عملياً للوحة المفاتيح ، يمكن من خلال هذا البرنامج التحكم بمجموعة من الأجهزة الإلكترونية .

#### الأوامر

```
#include <Keypad.h>
```

◆ إضافة مكتبة Keypad إلى البرنامج <<-----

```
const byte ROWS = 4;
```

◆ أربعة صفوف <<-----

```
const byte COLS = 4;
```

◆ أربعة أعمدة <<-----

```
char hexaKeys[ROWS][COLS] = {
```

◆ تعريف رموز المفاتيح في اللوحة <<-----

```
{ '1','2','3','A'},
```

```
{ '4','5','6','B'},
```

```
{ '7','8','9','C'},
```

```
{ '*', '0', '#', 'D' }
```

```
};
```

```
byte rowPins[ROWS] = {3, 4, 5, 6};
```

◆ أرقام الأرجل التي يتم توصيلها مع الصفوف <<

```
byte colPins[COLS] = {7, 8, 9, 10};
```

◆ أرقام الأرجل التي يتم توصيلها مع الأعمدة <<

```
Keypad customKeypad = Keypad( makeKeymap(hexaKeys),
```

```
rowPins, colPins, ROWS, COLS);
```

```
int LED=13;
```

◆ ترميز مجموعة من الأرجل <<-----

```
char customKey;
```

◆ تعريف متغير باسم customKey <<-----

```
void setup() {
```





sketch\_jan23a

## الأوامر

```
pinMode(LED,OUTPUT);
Serial.begin(9600);
}
```

```
void loop()
```

```
{
  customKey = customKeypad.getKey();
```

```
  if (customKey != NO_KEY)
```

```
  {
    Serial.println(customKey);
```

```
    switch (customKey)
```

```
    {
      case '1':
        digitalWrite(LED,HIGH);
        break;
```

```
      case '2':
        digitalWrite(LED,LOW);
        break;
```

```
      default: ;
```

```
    }
  }
}
```

قراءة رمز المفتاح الذي تم ضغطه وتخزين الرمز في متغير <<-----

الجملة الشرطية المرتبطة بعرض الرقم في حالة الضغط على أي مفتاح <<-----

كتابة المتغير على المراقب التسلسلي <<----

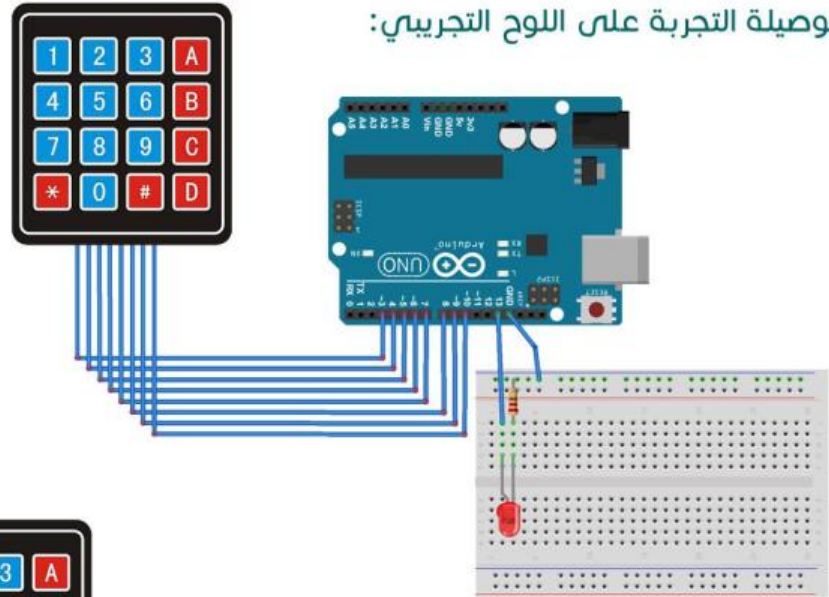
الجملة الشرطية في حالة الضغط على الرقم 1 <<-----  
تشغيل LED

الجملة الشرطية في حالة الضغط على الرقم 2 <<-----  
إطفاء LED

## Used Parts :

Arduino UNO x1  
Breadboard x1  
LED Red x1  
220 ohm x1  
Keypad 4x4 x1

## توصيلة التجربة على اللوح التجريبي:



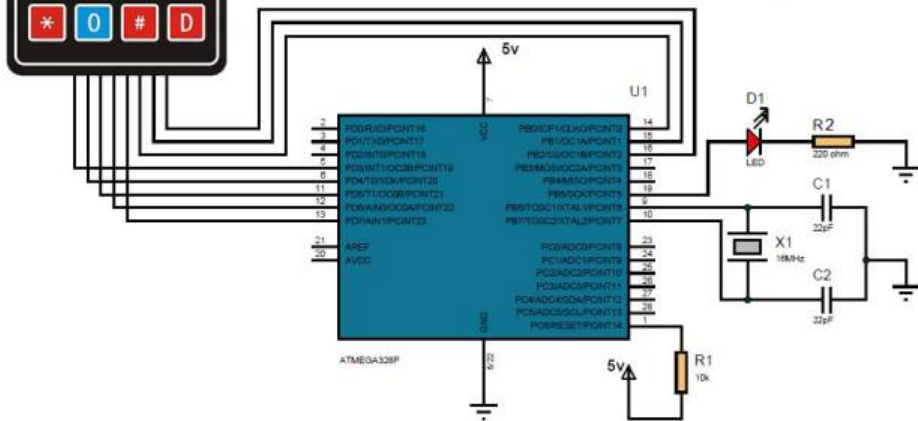
## Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

## Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28

## الرسم الفيزيائي :

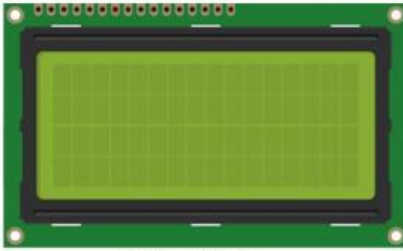




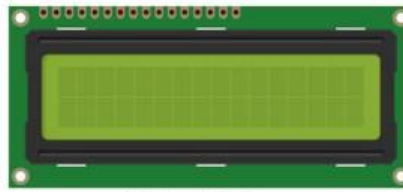
## شاشات العرض LCD :

نستخدم شاشات العرض كثيراً في مشاريعنا الإلكترونية ، مثلاً يمكن استخدامها في عرض درجة الحرارة أو عرض حالة الجهاز أو في الأجهزة التي بها تعليمات للمستخدم وغيرها .

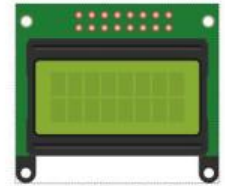
هناك أنواع كثيرة وألوان مختلفة من شاشات العرض . سنتعرف على واحدة منها في الصفحة التالية .



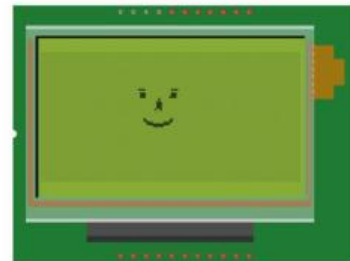
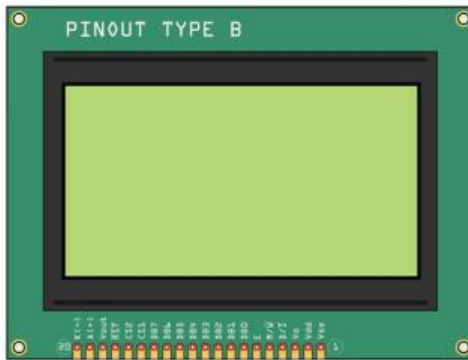
LCD 16x4



LCD 16x2



LCD 8x2



شاشات عرض جرافيك تدعم عرض الصور

## الدرس الثالث والعشرون:

### شاشة العرض LCD 16x2

يتطلب استخدام مكتبة

#### فكرة الدرس :

نرجو أن تستمتع بهذا الدرس ، سنتعلم كيفية برمجة شاشة العرض على كتابة كلمة Hello

#### الأوامر

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
```

◆ إضافة مكتبة LCD إلى البرنامج <<-----  
◆ تهيئة المكتبة مع المنافذ المستخدمة <<-----  
(RS,E,D4,D5,D6,D7)

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

◆ تهيئة شاشة LCD وتحديد أبعادها (الاعمدة والصفوف) <<--  
TEXT طباعة النص على شاشة العرض <<-----

```
void loop() {

  lcd.display();
  delay(1000);
  lcd.noDisplay();
  delay(1000);

}
```

TEXT عرض النص على الشاشة <<-----  
⌚ تأخير لمدة ثانية <<-----  
TEXT إخفاء النص من على الشاشة <<-----  
⌚ تأخير لمدة ثانية <<-----

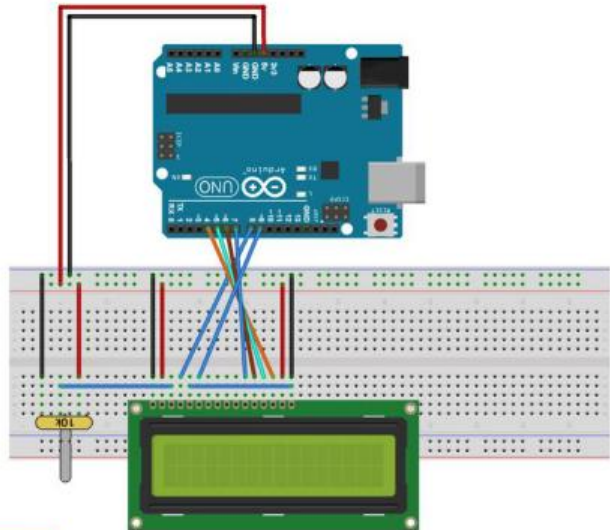


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
LCD 16x2 x1  
Variable resistor 10k x1

## توصيلة التجربة على اللوح التجريبي:



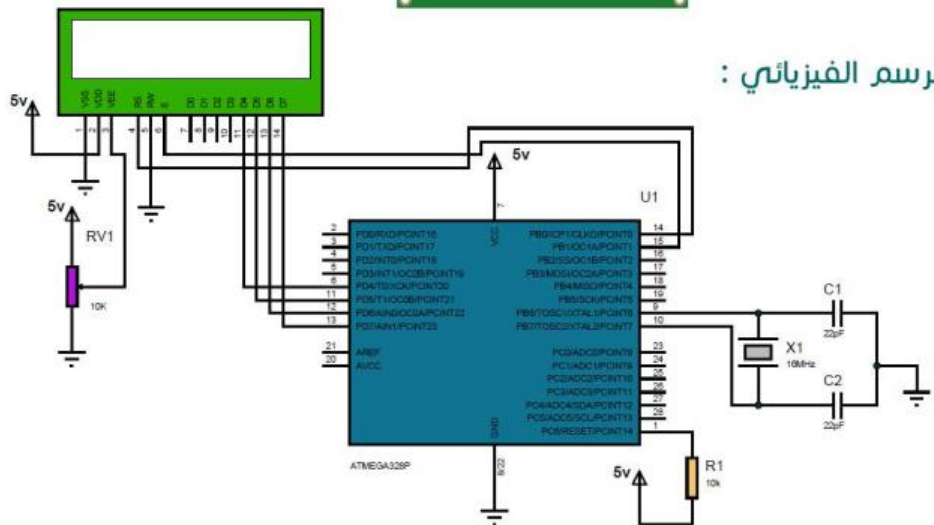
## الرسم الفيزيائي :

### Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

### Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28



## الدرس الرابع والعشرون:

### شاشة الأرقام السباعية 7 SEGMENTS

#### فكرة الدرس :

شاشة الأرقام السباعية كثيرة الاستخدام في حياتنا اليومية وخاصة في الساعات الرقمية وألواح التعليمات . برمجتها ليست صعبة ونترك لك فرصة تجربة ذلك أيها المبرمج .

#### الأوامر

♦ مصفوفة تعريف شاشة الأرقام <<

```
byte seven_seg_digits[10][7] = { { 1,1,1,1,1,1,0 },  
                                  { 0,1,1,0,0,0,0 },  
                                  { 1,1,0,1,1,0,1 },  
                                  { 1,1,1,1,0,0,1 },  
                                  { 0,1,1,0,0,1,1 },  
                                  { 1,0,1,1,0,1,1 },  
                                  { 1,0,1,1,1,1,1 },  
                                  { 1,1,1,0,0,0,0 },  
                                  { 1,1,1,1,1,1,1 },  
                                  { 1,1,1,0,0,1,1 }  
};
```

```
void setup() {  
  pinMode(2, OUTPUT);  
  pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
  pinMode(8, OUTPUT);  
  pinMode(9, OUTPUT);  
}
```



sketch\_jan23a

## الأوامر

```

writeDot(0);
}

void writeDot(byte dot) {
  digitalWrite(9, dot);
}

void sevenSegWrite(byte digit) {
  byte pin = 2;
  for (byte segCount = 0; segCount < 7; ++segCount) {
    digitalWrite(pin, seven_seg_digits[digit][segCount]);
    ++pin;
  }
}

void loop() {

  sevenSegWrite(9);
}

```

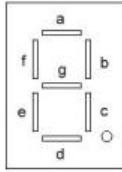
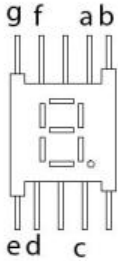
تفعيل النقطة على شاشة الأرقام <<-----

معادلات مرتبطة بالشاشة و المصفوفة <<--

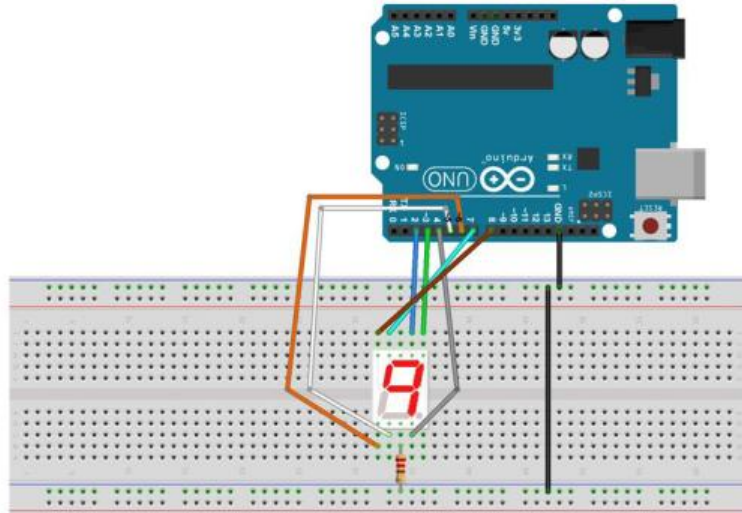
ضع الرقم الذي تريده أن يظهر ما بين القوسين <<--

## Used Parts :

Arduino UNO x1  
Breadboard x1  
220 ohm x1  
7 Segment cc x1



## توصيلة التجربة على اللوح التجريبي:



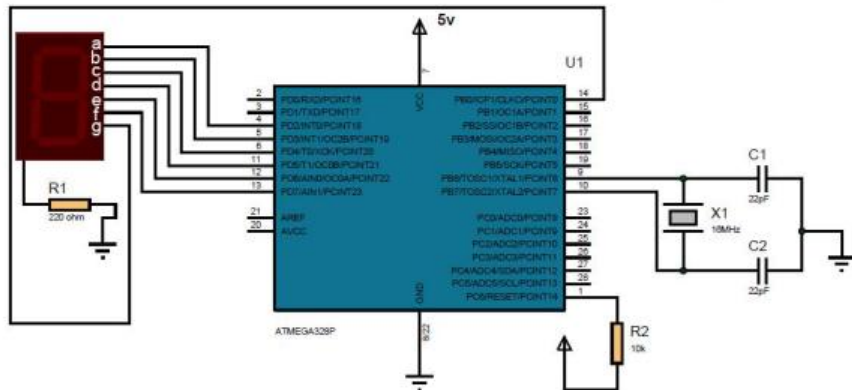
## Digital pins

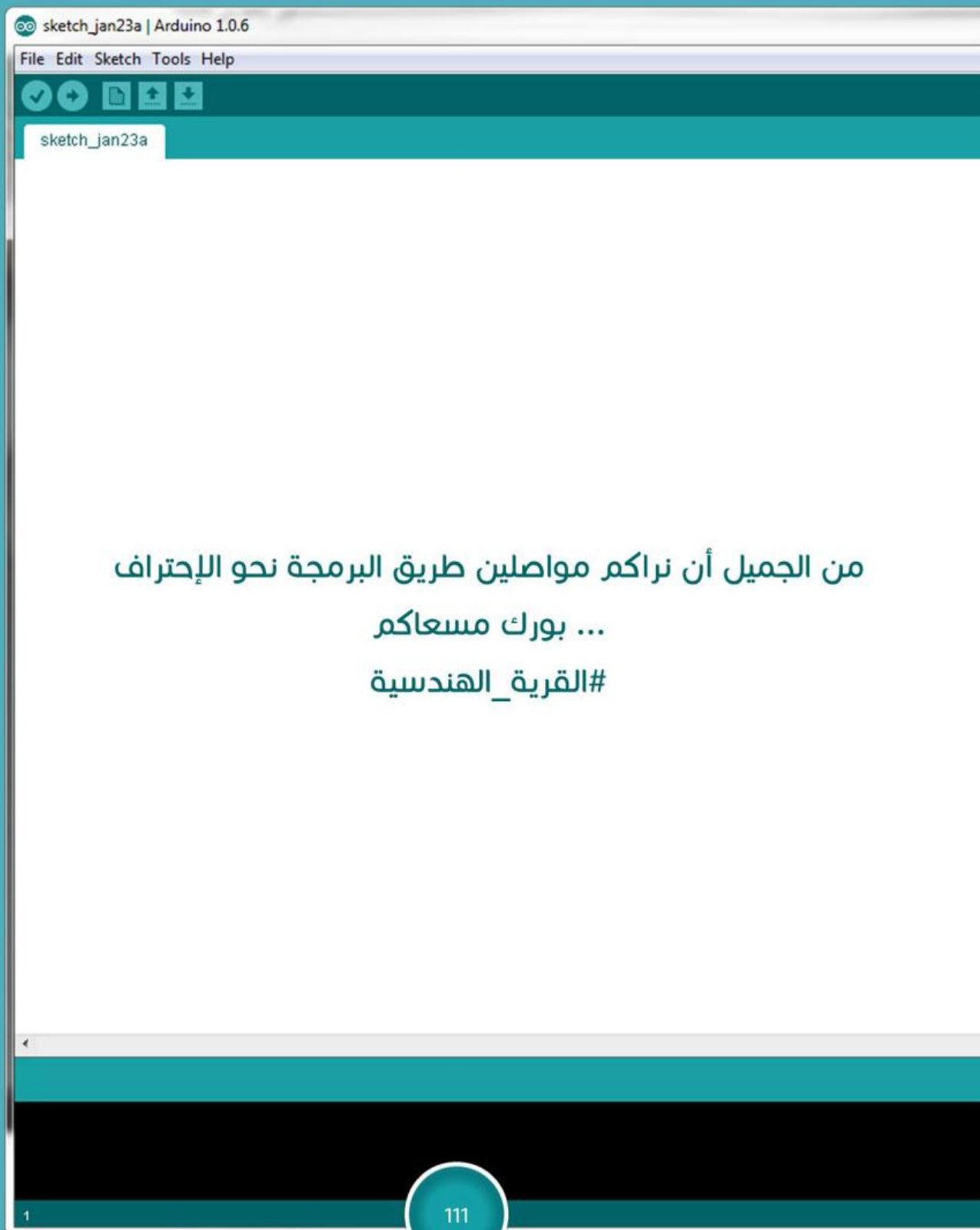
D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

## Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28

## الرسم الفيزيائي :





## الدرس الخامس والعشرون:

### مشروع قياس درجة الحرارة

يتطلب استخدام مكتبة

#### فكرة الدرس :

هذا البرنامج يقوم بقياس درجة الحرارة وعرضها على شاشة العرض ويعطي تنبيه في حالة ارتفاع درجة الحرارة فوق درجة معينة .

#### الأوامر

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
float C;
int tempPin = A0;
int LED= 13;
```

```
void setup() {
{
pinMode(LED,OUTPUT);
Serial.begin(9600);
lcd.begin(16,2);
lcd.setCursor(0,0);
lcd.print("hello");
lcd.setCursor(0,1);
lcd.print("I'm programmer");
delay(2000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("C=");
}
}
```

◆ إضافة مكتبة LCD إلى البرنامج <<-----

◆ تهيئة المكتبة مع المنافذ المستخدمة (RS,E,D4,D5,D6,D7) <<----

◆  
TEXT

TEXT تهيئة شاشة LCD وتحديد ابعادها (الاعمدة والصفوف) <<--

⌘ إعادة مؤشر الكتابة إلى بداية السطر الأول <<--

TEXT طباعة النص على شاشة العرض <<-----

⌘ نقل مؤشر الكتابة إلى بداية السطر الثاني <<--



sketch\_jan23a

## الأوامر

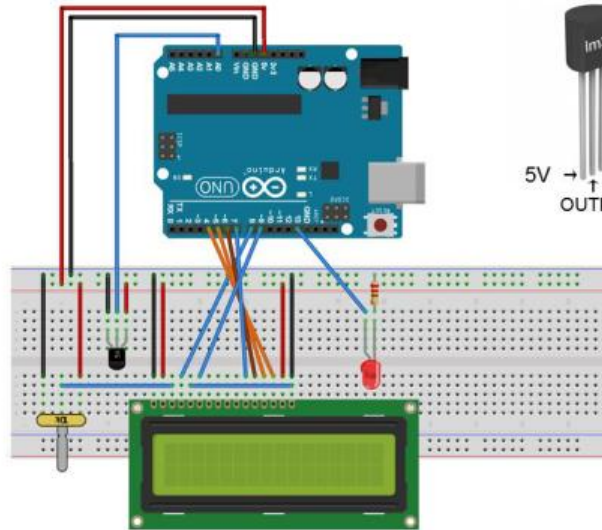
```
void loop()
{
  C=analogRead(tempPin);
  C = C*0.48828125;

  if (C>=30){
    digitalWrite(LED,HIGH);}
  else{
    digitalWrite(LED,LOW);}
    lcd.setCursor(3,0);
    lcd.print(C);
  }
}
```

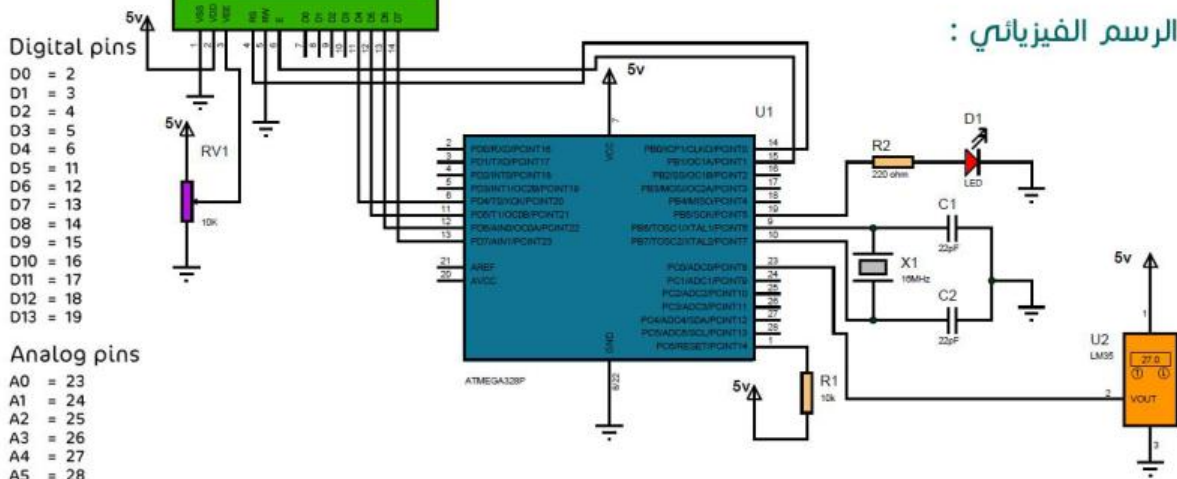
X/= هذه المعادلة لتحويل درجة الحرارة إلى  
درجة سيليزية <<----->>

الجملة الشرطية في حالة إرتفاع درجة الحرارة--><

### توصيلة التجربة على اللوح التجريبي:

[illegible]

### الرسم الفيزيائي :





sketch\_jan23a

لا مستحيل عند أهل العزيمة  
#نعم\_لنصنع\_التغيير

## الدرس السادس والعشرون:

### محركات DC

#### فكرة الدرس :

من المهم أيها المبرمج أن تتعلم كيفية التعامل مع محركات الـ DC (المحركات الموجودة في لعب الأطفال) . الدرس التالي يوضح لك طريقة عمل ذلك يا مبدع .

#### الأوامر

```
int motor = 9 ;
```

يتطلب اختيار رجل بها خاصية PWM <<-----

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

```
void loop () {  
  analogWrite(motor,75);  
  delay(2000);  
  analogWrite(motor,0);  
  delay(2000);
```

إخراج إشارة بسيطة بمقدار 75 <<-----

تأخير لمدة ثانيتين <<-----

إطفاء سرعة المحرك من خلال إعطاء قيمة 0 <<-----

تأخير لمدة ثانيتين <<-----

```
  analogWrite(motor,150);  
  delay(2000);  
  analogWrite(motor,0);  
  delay(2000);
```

إخراج إشارة متوسطة بمقدار 150 <<-----

تأخير لمدة ثانيتين <<-----

إطفاء سرعة المحرك من خلال إعطاء قيمة 0 <<-----

تأخير لمدة ثانيتين <<-----

```
  analogWrite(motor,255);  
  delay(2000);  
  analogWrite(motor,0);  
  delay(2000);  
}
```

إخراج إشارة عالية بمقدار 255 <<-----

تأخير لمدة ثانيتين <<-----

إطفاء سرعة المحرك من خلال إعطاء قيمة 0 <<-----

تأخير لمدة ثانيتين <<-----

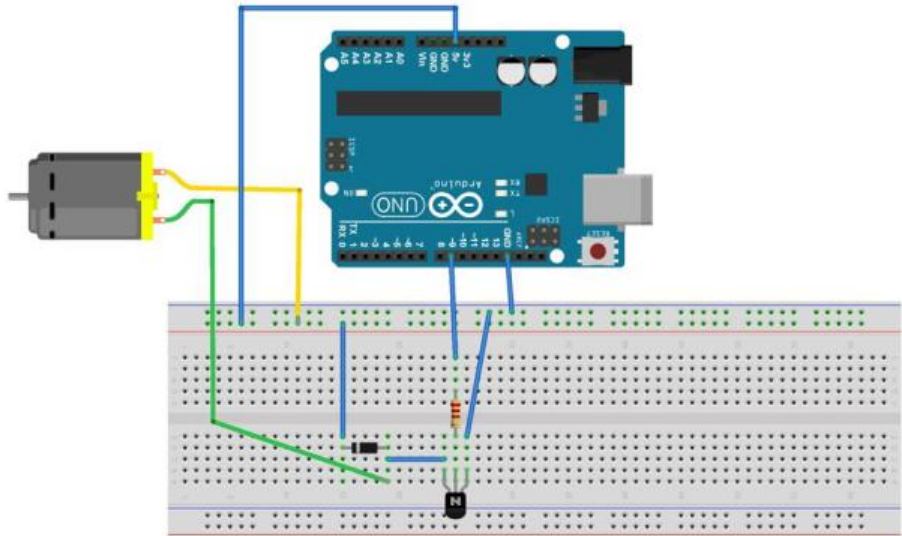


sketch\_jan23a

## Used Parts :

Arduino UNO x1  
Breadboard x1  
220 ohm x1  
DC motor x1  
Diode 1N4001 x1  
2n2222 NPN x1

## توصيلة التجربة على اللوح التجريبي:



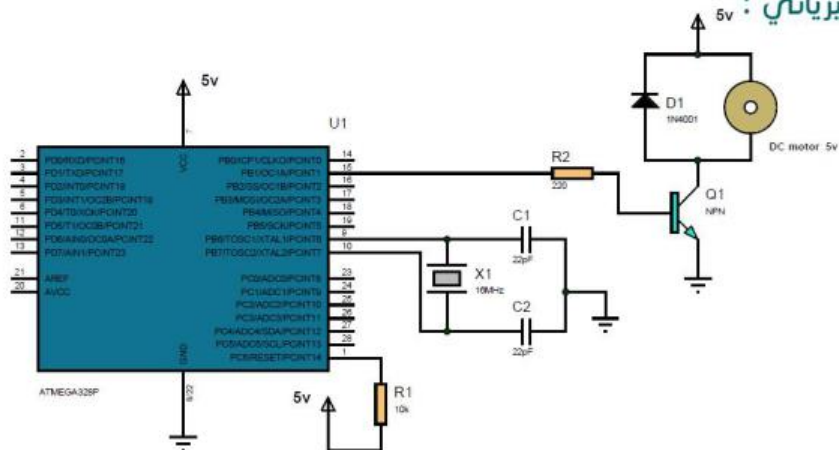
## Digital pins

D0 = 2  
D1 = 3  
D2 = 4  
D3 = 5  
D4 = 6  
D5 = 11  
D6 = 12  
D7 = 13  
D8 = 14  
D9 = 15  
D10 = 16  
D11 = 17  
D12 = 18  
D13 = 19

## Analog pins

A0 = 23  
A1 = 24  
A2 = 25  
A3 = 26  
A4 = 27  
A5 = 28

## الرسم الفيزيائي :



وصلنا إلى تحديات النهاية ، هل أنت مستعد ؟  
انطلق

### التحدي رقم 5 :

في هذا التحدي عليك أن تقوم بكتابة برنامج يقوم بالآتي: في حالة إرتفاع درجة الحرارة تزداد سرعة دوران المروحة .

### التحدي رقم 6 :

التحكم بسرعة دوران المروحة عن طريق لوحة المفاتيح KEYPAD ، بحيث أنه لو ضغطنا على الرقم 1 تدور المروحة بسرعة منخفضة وإذا ضغطنا على الرقم 2 تزداد السرعة وهكذا وفي حالة ضغطنا على الرقم 0 تتوقف عن الدوران.

#تتحدي\_القرية ؟



طاب يومك أيها المبرمج .



## ملحقات وتوابع الأردوينو (Extention Boards)

الأردوينو حر ومفتوح المصدر، أي أنه مفتوح للتطوير والإضافة بشكل مستمر ولهذا توجد له الكثير من النماذج المطورة والإضافات وألواح التطوير الجانبية ما يجعله الأمثل والأنسب لتنفيذ المشاريع التجريبية لكثرة الخيارات المتاحة كتوابع للأردوينو.

في الصفحات القليلة التالية سنتعرف على بعض المجسات وتوابع الأردوينو الكثيرة التي يمكنك الإستعانة بها في تنفيذ مشاريعك وأفكارك الإلكترونية.

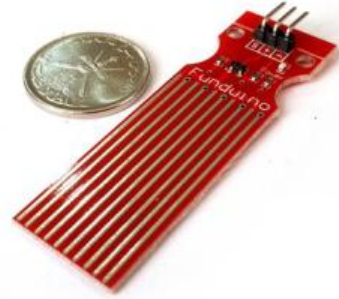
لمعرفة فوائدها التفصيلية يمكنك القراءة عنها في شبكة الإنترنت. كما يمكنك زيارة صفحة الأردوينو في موقعنا الإلكتروني حيث توجد مجموعة من الفيديوات التعليمية المبسطة التي توضح طريقة استخدام بعض هذه المجسات مع أمثلة تطبيقية لها في مشاريع مفيدة



حساس الضوء  
LDR Sensor



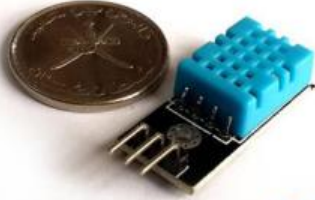
حساس الصوت  
Sound Sensor



حساس مستوى الماء  
Wate Level Sensor



sketch\_jan23a



حساس رطوبة الجو  
Humidity Sensor



حساس الاهتزاز  
Vibration Switch



التعرف على الصوت  
Voice Recognition



حساس المسافة  
IR Distance Sensor



حساس الدخان  
Smoke Sensor



حساس الأltrاسونيك  
Ultrasonic Sensor



نموذج بلوتوث  
Bluetooth Module



حساس تتبع الخط  
Line Tracking Sensor



حساس حركة  
PIR Motion Sensor



قارئ ذاكرة SD  
SD Card Module



الإرسال والإستقبال  
GSM SIM900



تحديد الموقع بالGPS  
GPS Module



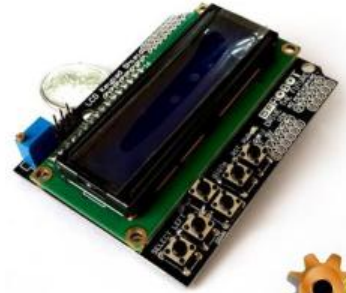
sketch\_jan23a



متحكم الستير موتور  
Stepper Motor Drive



عصا التحكم  
Joystick



شاشة LCD  
LCD Shield



لوح المرحلات  
Relay Board



حساس الإحداثيات  
X Y Z



مسجل صوت  
ISD Recorder



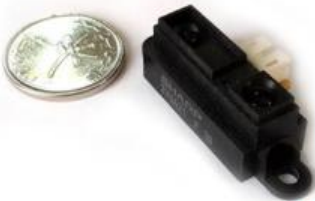
محدد الوقت والتاريخ  
Real Time Clock



حساس التيار  
Current Sensor



حساس الدوران  
Rotary Encoder



حساس المسافة  
IR SHARP Sensor



حساس الغاز  
Gas MQ6



حساس المغناطيس  
Hall Effect Sensor



sketch\_jan23a



حساس الكحول  
Alcohol MQ3 Sensor



مرسل FM  
FM Transmitter



حساس الألوان  
Color Sensor



مؤشر قوة البطارية  
Battery Level  
Indicator



حساس رطوبة التربة  
Grove Moisture  
Sensor



حساس الضغط الجوي  
Atmospheric  
Pressure Sensor



نحن في القرية الهندسية نحلم بجيل من الأطفال والشباب يجد متعته في العمل وإكتساب المعرفة. نحلم بمستقبل مشرق ينافس فيه أبنائنا عباقرة العالم ويتفوقون عليهم بجد وشغف ولهذا نعمل جاهدين على تقديم العلوم بطريقة مسلية تنافس البدائل المتوفرة من ألعاب الفيديو ومشاهدة التلفاز. العالم من حولنا يتطور بسرعة رهيبه في مختلف مجالات التكنولوجيا الرقمية والإلكترونية ومن واجبنا أن نسعى لمواكبة التطور والحدثة ليس بالقليل والقال وإنما بالعلم والعمل. ومعا يمكننا أن نصنع التغيير !

فهد بن سعيد بن محمد السيابي  
الرئيس التنفيذي للقرية الهندسية



## المراجع

موقع الأردوينو الرسمي  
<https://arduino.cc>

دورة برمجة الأردوينو  
القرية الهندسية - صيف 2014

كتاب الأردوينو ببساطة  
للمهندس عبدالله علي عبدالله

نخص بالشكر للمدربين الرائعين على مراجعتهم للكتاب  
سليم الراشدي  
سليمان الحبسي  
اسحاق الراشدي

## تروس في عالم الإلكترونيات تعلم الإلكترونيات في 10 أيام



## مؤلفاتنا الأخرى

### احترف الازبيري باي في 10 أيام



### احترف سكراثش في 10 أيام





sketch\_jan23a

نرجو أن نكون قد استطعنا افادتكم وامتناعكم في عالم  
برمجة الأردوينو. ولمزيد من المعلومات وللتحقق من اجاباتكم زوروا موقعنا  
على الإنترنت [www.ev-center.com](http://www.ev-center.com) ، كما يسعدنا قراءة آرائكم  
وتعليقاتكم في حساباتنا على شبكات التواصل الإجتماعي عبر  
الوسم #مبرمجو\_المستقبل

 @EngineeringOman

 +968 22027662

 Engineering Village

 +968 97377800

 EngineeringVillageEV

 [www.ev-center.com](http://www.ev-center.com)

 @EngineeringVillage

 [school@ev-center.com](mailto:school@ev-center.com)

تم بحمد الله

\* كل شخص عليه أن يتعلم برمجة الحاسوب .. لأنه يعلمك كيف تفكر ،، ” ستيف جوبز“

\* بيل جيتس | مؤسس ميكروسفت | تعلم البرمجة بعمر 13 عام

\* مارك زوكربيرج | مؤسس الفيس بوك | بدأ باستخدام الحاسوب في الصف السادس

\* جاك زورسي | مؤسس شبكة تويتر | أهداه والداه حاسوب وهو بعمر 9 سنوات

لم يكونوا يملكون العبقرية أو الذكاء الخارق وإنما بحثوا عن المرح والمتعة في أماكن الابتكار .

البرمجة هي لغة العصر لخدمة الإنسانية ومساعدة الناس ، هي طريقة التفكير والتنوير لعالم اليوم ويجب أن تبدأ بها منذ صغرك ! سواء أردت أن تغير العالم أو أن تملك الكثير من المال .

في العشر السنوات القادمة سيحتاج العالم إلى 1.4 مليون مبرمج لخدمة البشرية ، ولا يوجد في العالم سوى 400 ألف مبرمج حالياً . العالم يحتاجنا ، فهيا بنا نبرمج .



لنمرح بتعلم مهارات البرمجة مع الأردوينو

#مبرمجوالمستقبل