

## المتحكمات الصغيرة و النظم المضمنة

محاضرة 9

# Arduino

د. فادي متوج

## ما هو أردوينو؟

- Arduino عبارة عن منصة تطوير متحكم صغري مقترنة بلغة برمجة نقوم بتطويرها باستخدام بيئة التطوير المتكاملة (IDE) Arduino .
- من خلال تزويد Arduino بحساسات ومشغلات ولبادات ومكبرات صوت و أغطية ودارات متكاملة أخرى ، يمكننا تحويل Arduino إلى "عقل" قابل للبرمجة لأي نظام تحكم تقريبًا.

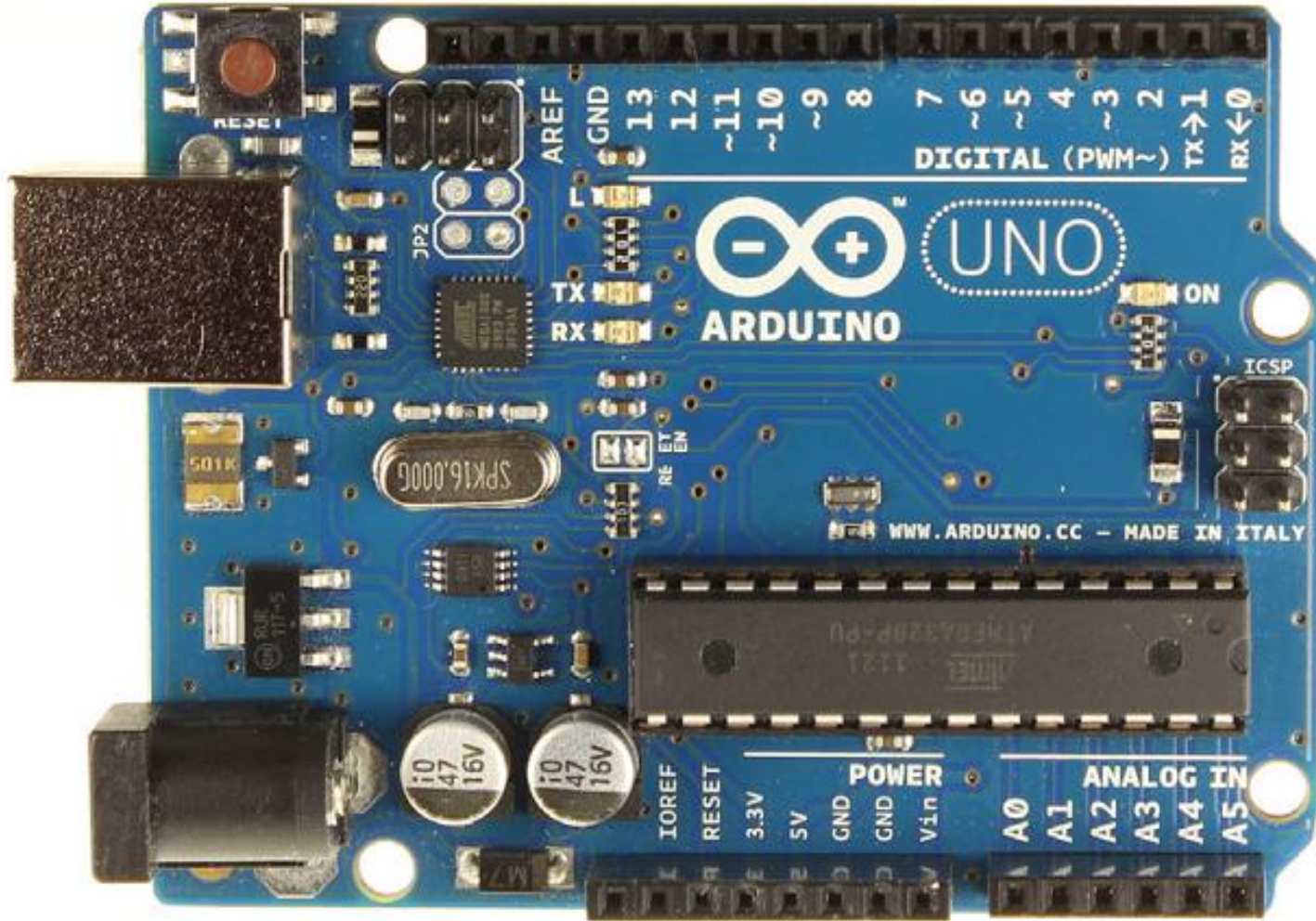


## أرجل الدخل و الخرج

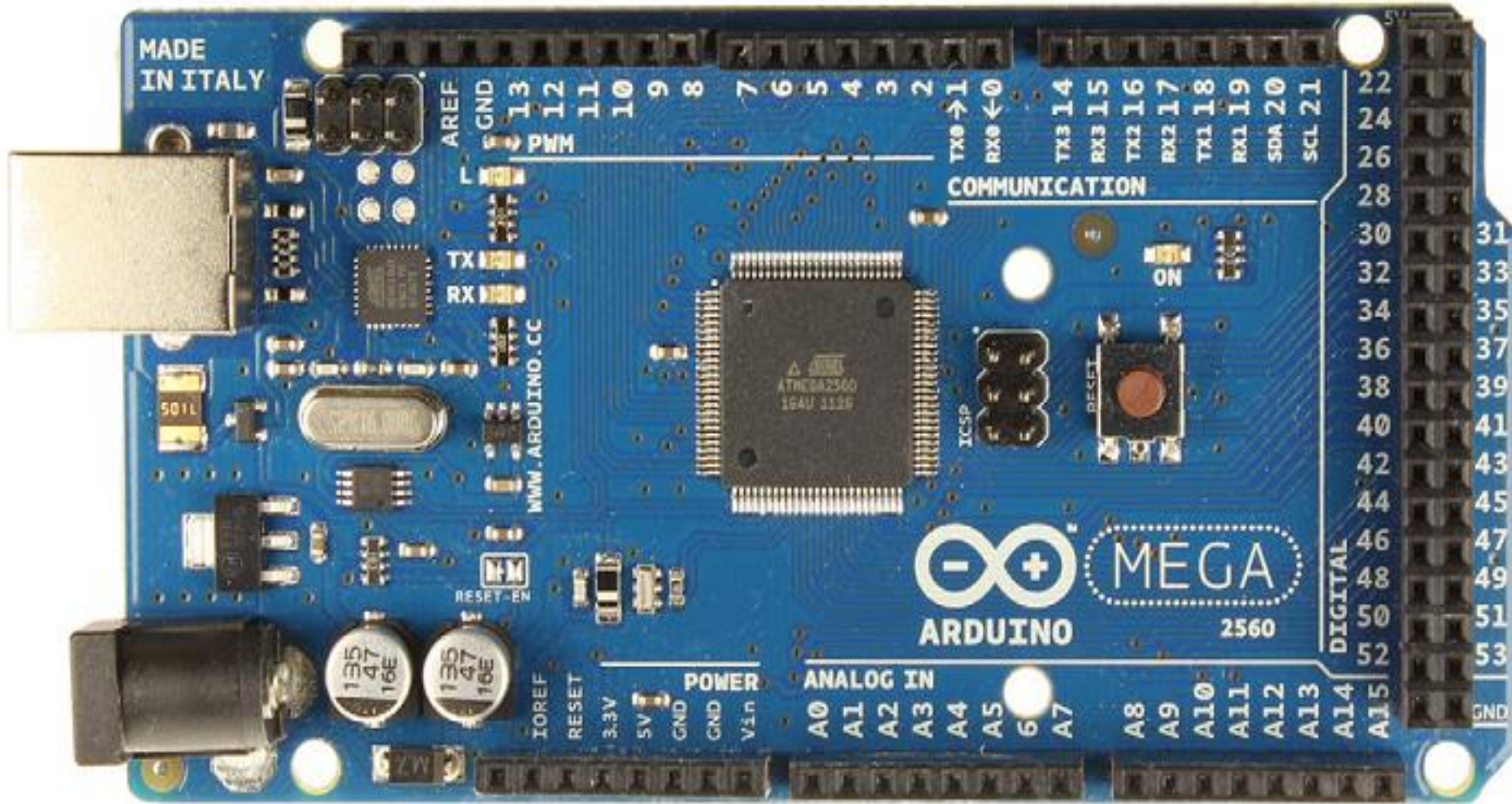
- تعمل أرجل الإدخال / الإخراج (الأرجل من 0 إلى 13) كمدخلات ومخرجات رقمية.
- تعمل الأرجل (من A0 إلى A5) كمدخلات تشابهيّة يمكنها قراءة الجهد بين 0 و 5 فولت
- تلعب العديد من هذه الأرجل وظائف إضافية. تتضمن هذه الوظائف الخاصة الاتصال التسلسلي، و إخراج إشارات تعديل عرض النبضة PWM ، ومقاطعات خارجية.

## تزويد بورد أردوينو بالطاقة

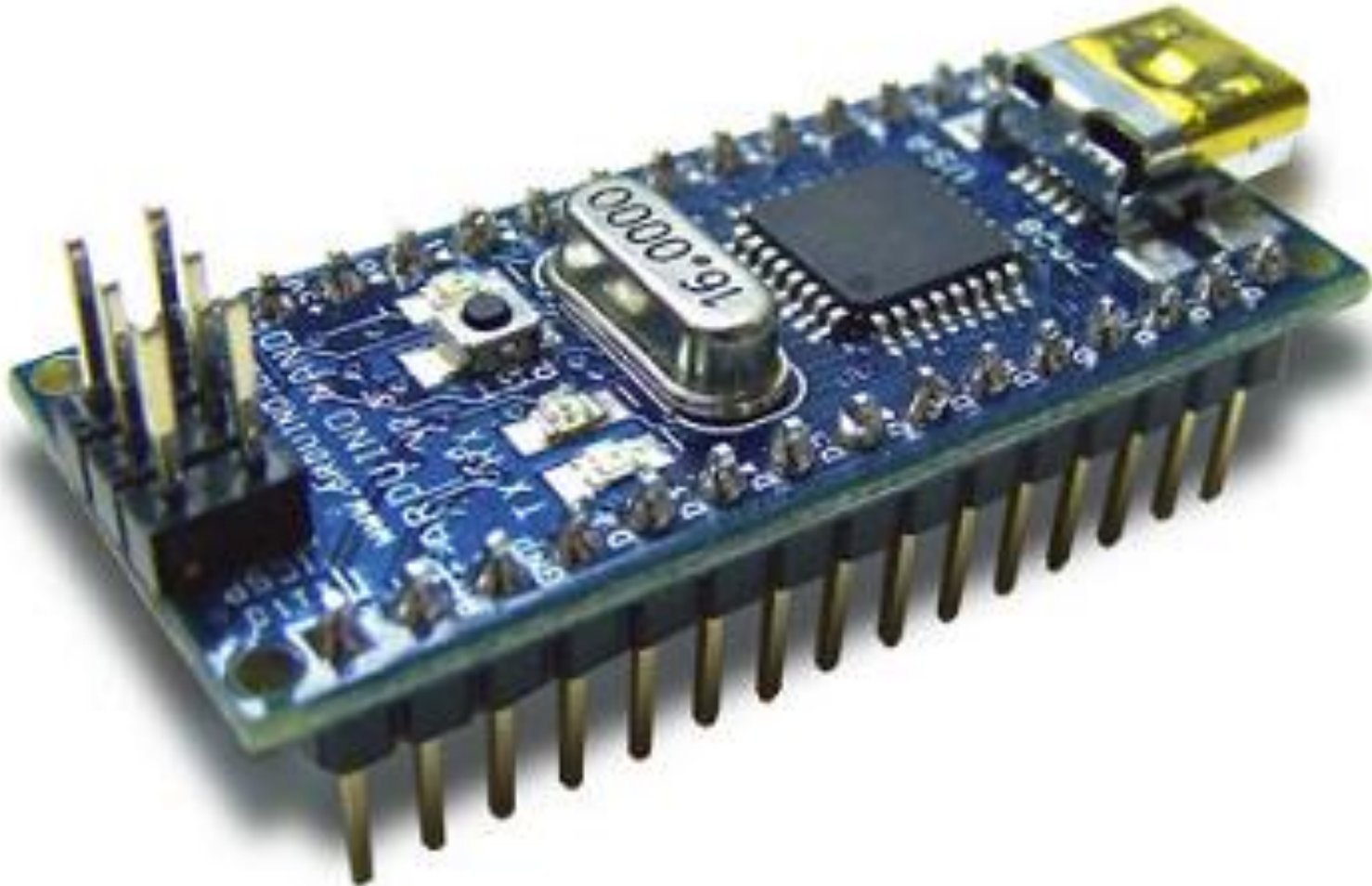
- بالنسبة لغالبية المشاريع، سنستخدم ببساطة مصدر 5 فولت التي يتم توفيرها عبر كابل USB .
- عند الحاجة لفصل المشروع عن جهاز الكمبيوتر، يكون لدينا خيارات طاقة أخرى:
- يمكن أن يقبل Arduino ما بين 6 فولت و 20 فولت (7-12 فولت موصى به) عبر موصل مقبس تيار مستمر (DC)، أو بتوصيل بطارية خارجية من خلال الرجل Vin
- يحتوي Arduino على منظمات جهد 5 فولت و 3.3 فولت مدمجة مع البورد



Arduino Uno

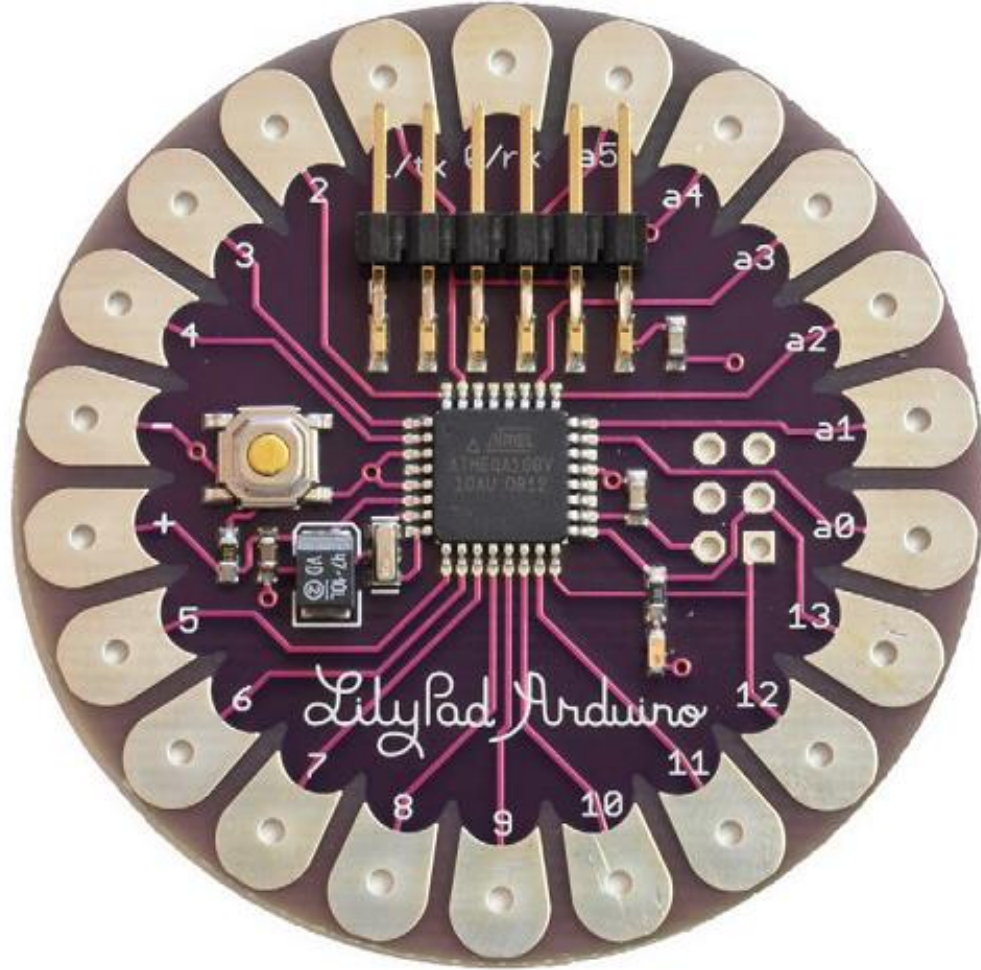


Arduino Mega 2560



Arduino Nano



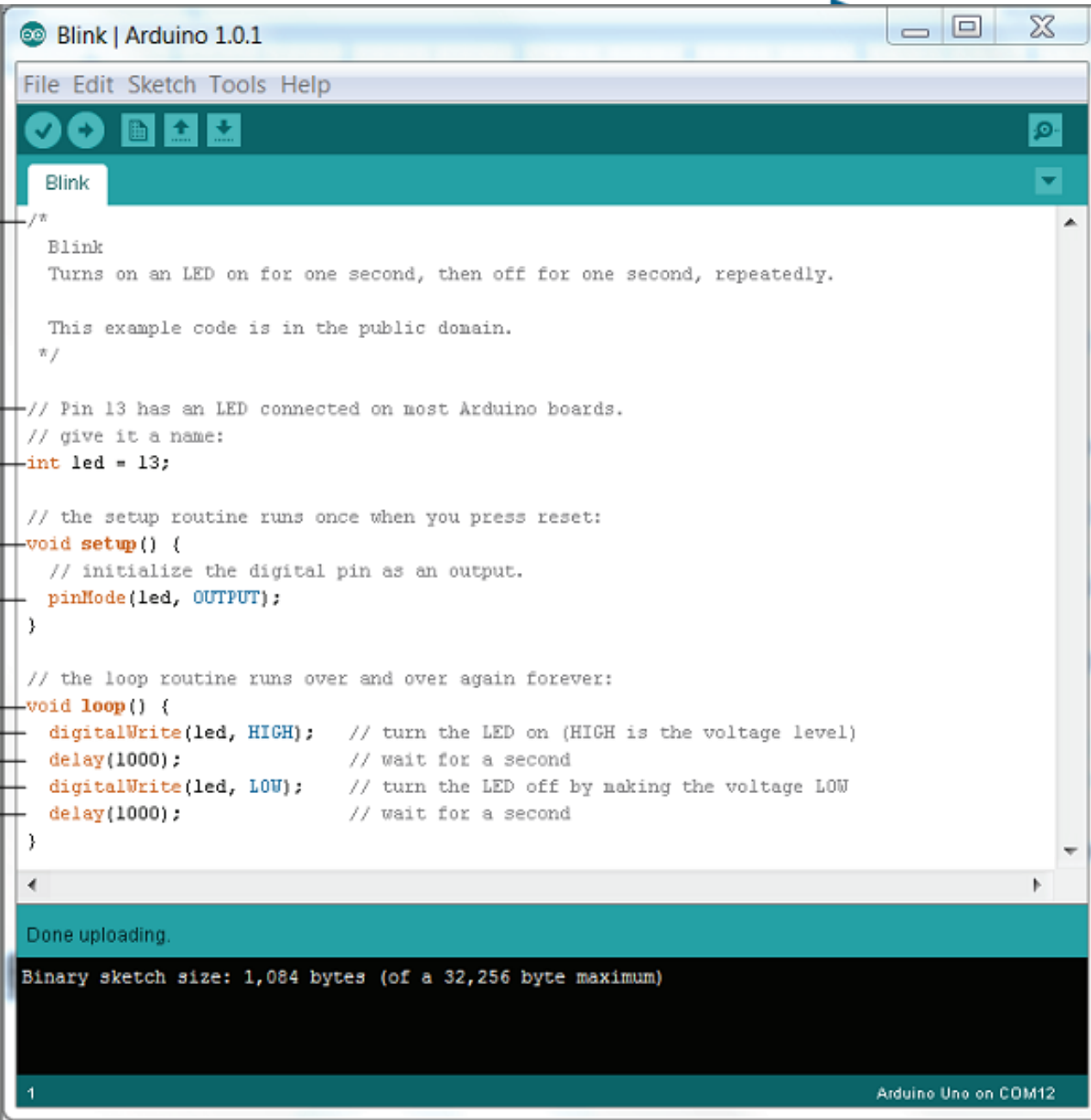


Arduino LilyPad



**ArduPilot**

# برنامجنا الأول - برنامج Blink



The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The code is as follows:

```
1 /*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
2 // Pin 13 has an LED connected on most Arduino boards.  
  // give it a name:  
3 int led = 13;  
  
  // the setup routine runs once when you press reset:  
4 void setup() {  
  // initialize the digital pin as an output.  
5  pinMode(led, OUTPUT);  
  }  
  
  // the loop routine runs over and over again forever:  
6 void loop() {  
7  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
8  delay(1000);             // wait for a second  
9  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
10 delay(1000);             // wait for a second  
  }  
}
```

Below the code editor, the status bar shows: Done uploading. Binary sketch size: 1,084 bytes (of a 32,256 byte maximum). The board is identified as Arduino Uno on COM12.

## 1- هذا تعليق متعدد الأسطر:

- التعليقات مهمة لتوثيق التعليمات البرمجية الخاصة بنا. كل ما نكتبه بين هذه الرموز لن يتم ترجمته أو حتى رؤيته بواسطة Arduino

- تبدأ التعليقات متعددة الأسطر ب `/*` وتنتهي ب `*/`

- تُستخدم التعليقات متعددة الأسطر بشكل عام عندما تضطر إلى قول الكثير.

## 2- هذا تعليق من سطر واحد.

- عندما نضع `//` على أي سطر، يتجاهل المترجم كل النص بعد هذا الرمز في نفس السطر.

- يعد هذا أمرًا رائعًا للتعليق على سطور معينة من التعليمات البرمجية أو "للتعليق" على سطر برمجي معين يعتقد أنه قد يتسبب في حدوث مشكلات.

3- هذا السطر يتضمن التصريح عن متغير.

- المتغير هو مكان في ذاكرة الـ Arduino يحتوي على معلومات.

- المتغيرات لها أنواع مختلفة. في حالتنا هذه، يكون من النوع **int**، مما يعني أنه سيحتوي على عدد صحيح. في هذه الحالة، يتم إسناد القيمة 13 لمتغير صحيح يسمى `led`، وهي الرجل التي يتصل بها LED موجود على بورد Arduino Uno

- خلال بقية البرنامج، يمكننا ببساطة استخدام `led` كلما أردنا التحكم في الرجل 13.

- يعد استخدام المتغيرات مفيداً لأنه يمكننا فقط تغيير هذا السطر البرمجي إذا قمنا بتوصيل الـ LED الخاص بنا برجل أخرى لاحقاً، حيث ستظل بقية التعليمات البرمجية تعمل كما هو متوقع بدون تغيير.

4- **void setup()** هو إحدى تابعين يجب تضمينهما في كل برنامج من برامج Arduino

- التابع بشكل عام هو جزء من الكود يقوم بمهمة محددة.

- يتم تنفيذ الكود داخل الأقواس المتعرجة للتابع ( ) setup مرة واحدة في بداية البرنامج.

- هذا التابع مفيد للإعدادات، مثل ضبط اتجاه الأرجل وتهيئة منافذ الاتصال وما إلى ذلك.

5- يمكن أن تعمل أرجل Arduino الرقمية كدخل أو خرج.

■ لتحديد اتجاه الأرجل، نستخدم الأمر ( ) **pinMode**

■ يأخذ هذا الأمر معاملين:

- المعامل الأول لـ **pinMode** يحدد رقم الرجل التي يتم تعيين اتجاهها.

- يحدد المعامل الثاني اتجاه الرجل: **INPUT** أو **OUTPUT**

- الأرجل هي دخل بشكل افتراضي، لذلك نحتاج إلى تعيينها بشكل صريح على أنها خرج إذا أردنا أن تعمل كمخرجات.

## 6- التابع الثاني المطلوب في جميع برامج Arduino هو التابع (**void loop()**)

- تتكرر محتويات التابع loop إلى الأبد طالما أن Arduino قيد التشغيل.
- إذا أردنا أن يقوم Arduino الخاص بنا بعمل شيء مرة واحدة فقط عند التشغيل، فلا بد من تضمين التابع loop، لكن يمكننا تركه فارغ بدون تعليمات.

## 7- يتم استخدام (**digitalWrite()**) لتعيين حالة رجل الخرج.

- يمكنه ضبط الرجل على 5 فولت أو 0 فولت
- عند توصيل مؤشر LED ومقاوم بدبوس، فإن ضبطه على 5 فولت سيمكنك من إضاءة مؤشر LED.
- المعامل الأول لـ (**digitalWrite()**) هو رقم الرجل التي نريد التحكم فيها.
- المعامل الثاني هو القيمة التي نريد ضبط الرجل عليها، إما HIGH (5V) أو LOW (0V)
- تبقى الرجل في هذه الحالة حتى يتم تغييرها في الكود البرمجي

**8-** يقبل التابع (**delay ()**) معامل واحد هو زمن تأخير بالملي ثانية.

- عند استدعاء (**delay ()**)، يتوقف Arduino عن فعل أي شيء خلال الفترة الزمنية المحددة.

- في هذه الحالة ، نقوم بتأخير البرنامج لمدة 1000 ملي ثانية ، أو ثانية واحدة.

- ينتج عن هذا بقاء ال LED مضاء لمدة ثانية واحدة قبل تنفيذ الأمر التالي.

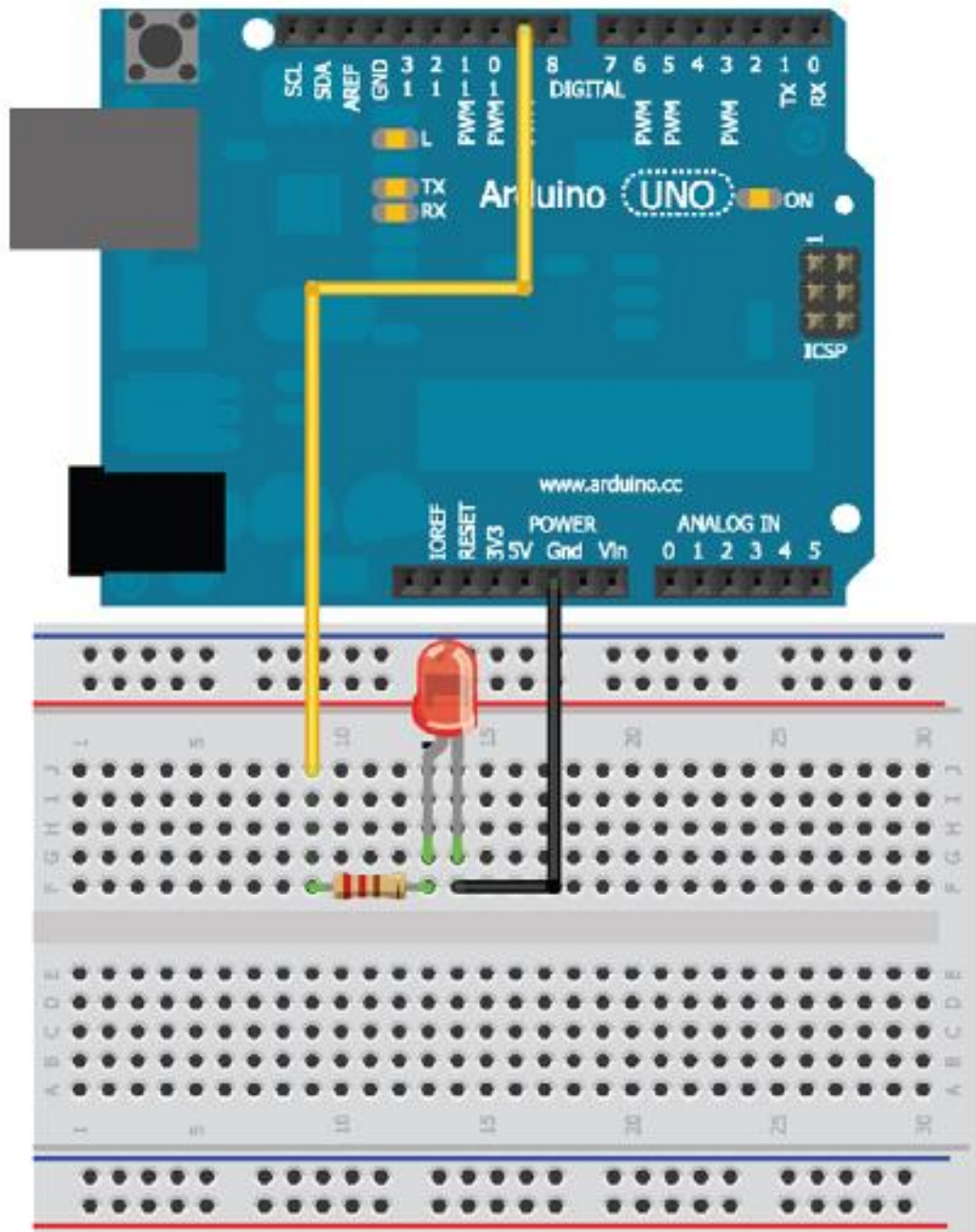
**9-** يتم استخدام (**digitalWrite ()**) لإطفاء ال LED، عن طريق ضبط حالة الرجل على **LOW**

**10-** مرة أخرى ، نقوم بعمل تأخير لمدة ثانية واحدة لإبقاء ال LED في حالة إطفاء قبل أن تتكرر الحلقة و

ينتقل ال LED للإضاءة مرة أخرى.



# الإخراج الرقمي: إضاءة LED



```
const int LED=9;           //define LED for pin 9
void setup()
{
  pinMode (LED, OUTPUT); //Set the LED pin as an output
  digitalWrite(LED, HIGH); //Set the LED pin high
}
void loop()
{
  //we are not doing anything in the loop
}
```

```
const int LED=9;    //define LED for Pin 9
void setup()
{
  pinMode (LED, OUTPUT);    //Set the LED pin as an output
}
void loop()
{
  for (int i=100; i<=1000; i=i+100)
  {
    digitalWrite(LED, HIGH);
    delay(i);
    digitalWrite(LED, LOW);
    delay(i);
  }
}
```



جامعة  
المنارة  
MANARA UNIVERSITY

# تعديل عرض النبضة

## Pulse Width Modulation

### analogWrite()

- يعد التحكم الرقمي في الأرجل أمرًا رائعًا لإضاءة الليدات، والتحكم في الريليهات ، و دوران المحركات بسرعة ثابتة.
- ولكن ماذا لو أردنا إخراج جهد غير 0 فولت أو 5 فولت؟  
لا يمكننا ذلك - ما لم نستخدم دائرة خارجية DAC للتحويل من رقمي إلى تشابهي.
- ومع ذلك، يمكننا الاقتراب جدًا من توليد قيم خرج تشابهية باستخدام خدعة تسمى تعديل عرض النبضة . PWM

# تعديل عرض النبضة

## Pulse Width Modulation

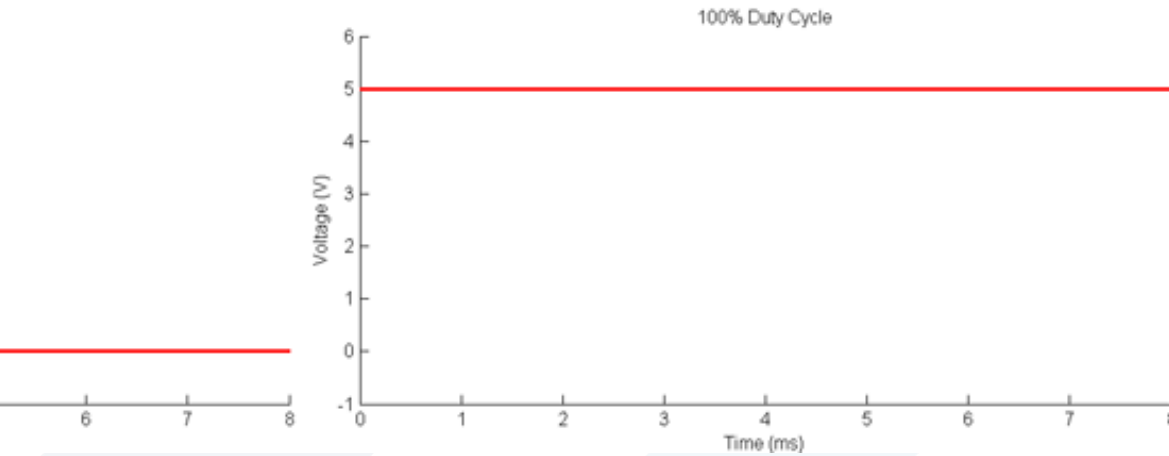
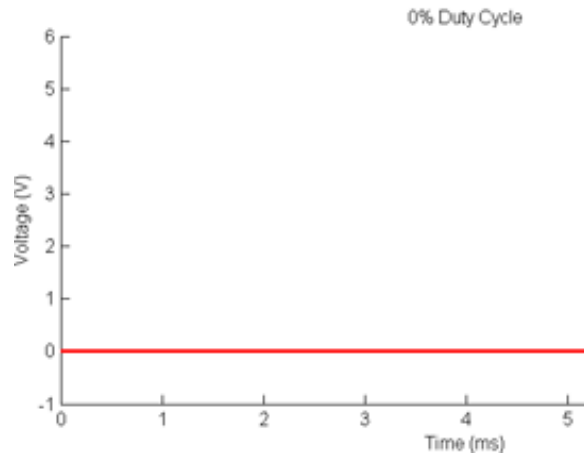
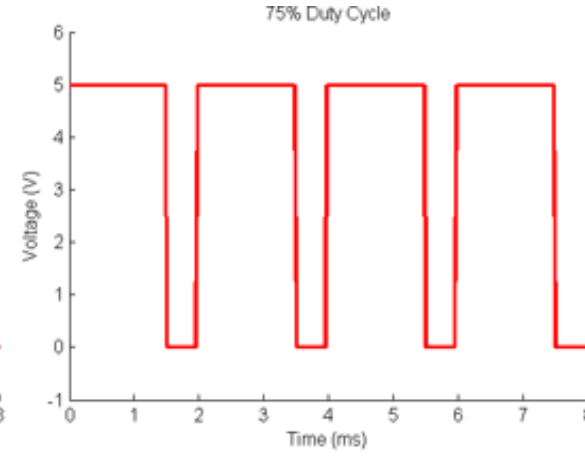
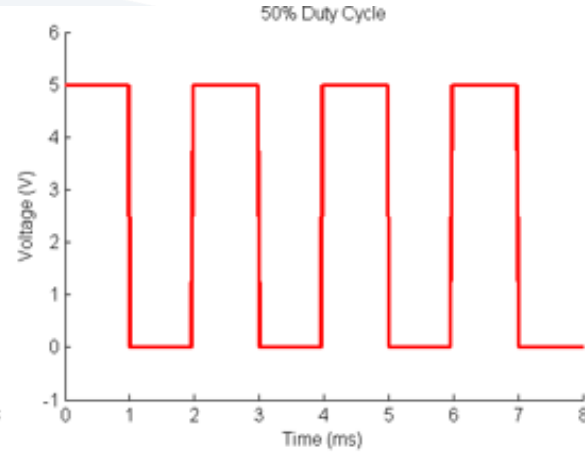
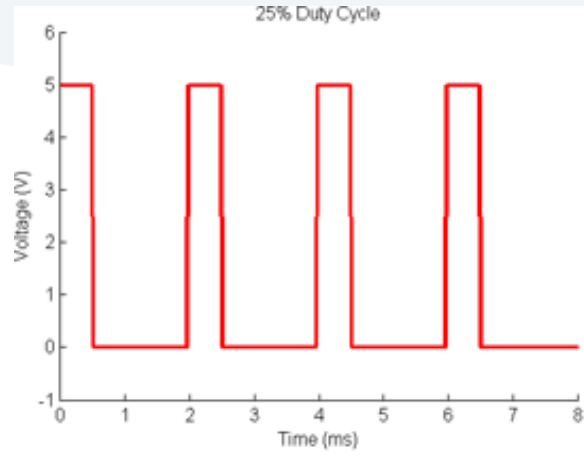
### analogWrite()

- في كل بورد Arduino، توجد أرجل يمكنها استخدام الأمر `analogWrite ()` لتوليد إشارات PWM التي يمكنها محاكاة إشارة تشابهيية نقية عند استخدامها مع أجهزة طرفية معينة.
- يتم تمييز هذه الأرجل بعلامة ~ على البورد.
- على Arduino Uno، الأرجل 3 و 5 و 6 و 9 و 10 و 11 هي أرجل PWM
- يقبل الأمر `analogWrite ()` معاملين: الرجل التي يجب التحكم فيها والقيمة المراد إخراجها عليها.
- الخرج PWM هو قيمة تقع في المجال من 0 إلى 255.

```
const int LED=9; //define LED for Pin 9
void setup() {
  pinMode (LED, OUTPUT); //Set the LED pin as an output
}
void loop() {
  for (int i=0; i<256; i++)
  {
    analogWrite(LED, i);
    delay(10);
  }
  for (int i=255; i>=0; i--)
  {
    analogWrite(LED, i);
    delay(10);
  }
}
```

## ماذا يفعل الـ LED عند تنفيذ الكود السابق؟

- يجب أن نلاحظ ازدياد تدريجي في شدة سطوع الليد من كونه مطفأ في البداية وصولاً إلى السطوع الكامل، ثم بالعكس انخفاض تدريجي في شدة سطوع الليد من حالة السطوع الكامل وصولاً إلى انطفائه بشكل كامل.
- بالطبع ، لأن التعليمات كلها موجودة في الحلقة الرئيسية loop، فإن هذا السيناريو يتكرر.



- تعمل تقنية PWM عن طريق تعديل دورة عمل الموجة المربعة.
- تشير دورة العمل (Duty cycle) إلى النسبة المئوية من الزمن التي تكون فيها الموجة المربعة بحالة HIGH بالنسبة لكامل دور الإشارة.
- مثلاً الموجة المربعة التي لها دورة عمل بنسبة 50% تكون في حالة HIGH خلال نصف الدور، وفي حالة LOW في نصف الدور.



- يحدد الأمر **analogWrite ()** دورة عمل الموجة المربعة اعتمادًا على القيمة التي نمررها إليه:
  - كتابة قيمة 0 باستخدام **analogWrite ()** تشير إلى موجة مربعة مع دورة عمل بنسبة 0 بالمائة (دائمًا LOW).
  - تشير كتابة 255 إلى موجة مربعة مع دورة عمل بنسبة 100 بالمائة (دائمًا HIGH).
  - تشير كتابة 127 إلى موجة مربعة مع دورة عمل بنسبة 50 بالمائة (نصف الزمن HIGH و نصف الزمن LOW).
- بالنسبة للإشارة ذات دورة العمل بنسبة 25 بالمائة ، تكون بحالة HIGH بنسبة 25 بالمائة من الزمن، وبحالة LOW بنسبة 75 بالمائة من الزمن.
- تردد هذه الموجة المربعة ، في حالة Arduino ، يبلغ حوالي **490 هرتز**. بمعنى آخر ، تتناوب الإشارة بين HIGH (5 فولت) وLOW (0 فولت) حوالي 490 مرة كل ثانية.
- لنطرح السؤال التالي : طالما لم نقم بالفعل بتغيير الجهد الذي يتم توصيله إلى LED ، فلماذا نرى أن الـ LED يصبح باهتًا عندما نخفض دورة العمل؟ في الحقيقة يحصل هذا نتيجة لعب أعيننا خدعة علينا! إذا كان الـ LED يضيء ويطفئ كل 1 ميلي ثانية (وهذا هو الحال مع دورة العمل بنسبة 50 بالمائة) ، فيبدو أنه يعمل بنصف سطوع تقريبًا لأنه يومض بشكل أسرع مما يمكن أن تدركه أعيننا. لذلك ، يقوم دماغنا في الواقع بتقدير متوسط الإشارة ويخدعنا للاعتقاد بأن الـ LED يعمل بنصف سطوع.