

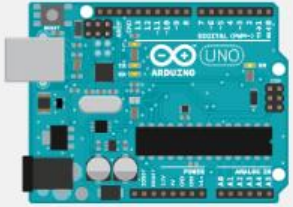
# Les micro contrôleurs Arduino

<http://www.arduino.cc/>



[Home](#) [Buy](#) [Download](#) [Products](#) [Learning](#) [Forum](#) [Support](#) [Blog](#)

## WHAT IS ARDUINO?



BUY AN ARDUINO



LEARN ARDUINO



# ARDUINO ATHEART

Designed for makers and companies wanting to make their products easily recognizable as based on the Arduino technology.

SHOW OFF YOUR



BLOG

ARDUINO WEB  
ARCHITECTURE, BACK TO  
THE FUTURE

ARDUINO LILYPAD USB,  
GREAT RESULTS IN  
A FEW STITCHES!  
SHOP NOW



Arduino Uno



Arduino Leonardo



Arduino GSM Shield



Arduino Due



Arduino Yún



Arduino Ethernet  
Shield



Arduino Tre



Arduino Zero



Arduino WiFi Shield



Arduino Micro



Arduino Esplora



Arduino Wireless SD  
Shield

## BeagleBoard



BeagleBoard rev.B

Développeur	Texas Instruments et Digi-Key
Type	ordinateur monocarte
Processeur	ARM Cortex A8 700 MHz à 1 GHz
Stockage	Carte SD
Mémoire	128 MByte à 512 MByte
Connectivité	USB On-The-Go, DVI-D, PC audio, SDHC, JTAG, HDMI
Dimensions	7,62 cm × 7,62 cm × 1,6 cm
Masse	~37 g <sup>1</sup>

modifier



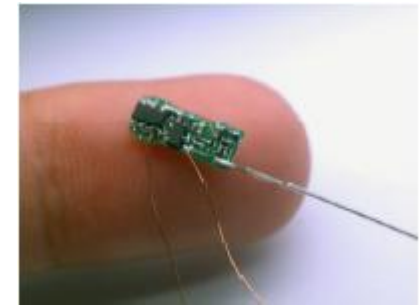
## Raspberry Pi



"Raspberry Pi" Ordinateur modèle B Rev1

Développeur	Raspberry Pi Foundation <a href="#">↗</a>
Type	Ordinateur à carte unique
Date de sortie	29 février 2012
Environnement	Linux (Debian, Fedora et ArchLinux), RISC OS
Système d'exploitation	Linux (Raspbian, Pidora, et Arch Linux ARM), RISC OS, FreeBSD, NetBSD, Plan 9
Alimentation	Micro-B USB 5 V
Processeur	Broadcom BCM2835 - ARM1176JZF-S 700 MHz (modèle 1) Broadcom BCM2836 - Cortex-A7 900 MHz (modèle 2)
Stockage	Carte SD (A, B), Carte microSD (A+, B+, 2)
Mémoire	256 Mo (modèle A et A+) 256 Mo (modèle B rev 1) 512 Mo (modèle B rev 2 et B+) 1 Go (modèle 2)
Carte graphique	Broadcom VideoCore IV <sup>1</sup> ,
Connectivité	USB, Ethernet (modèle B, B+, 2) (RJ45), HDMI, RCA, Jack 3,5 mm
Dimensions	85,60 mm × 53,98 mm × 17 mm (A, B, B+), 65 mm × 53,98 mm × 17 mm (A+)
Masse	44,885 g (A, B, B+), 23 g (A+)

modifier



## Stellaris® LM4F120 LaunchPad Evaluation Kit The new, enhanced Tiva™ C Series [TM4C123G LaunchPad](#) is now available.

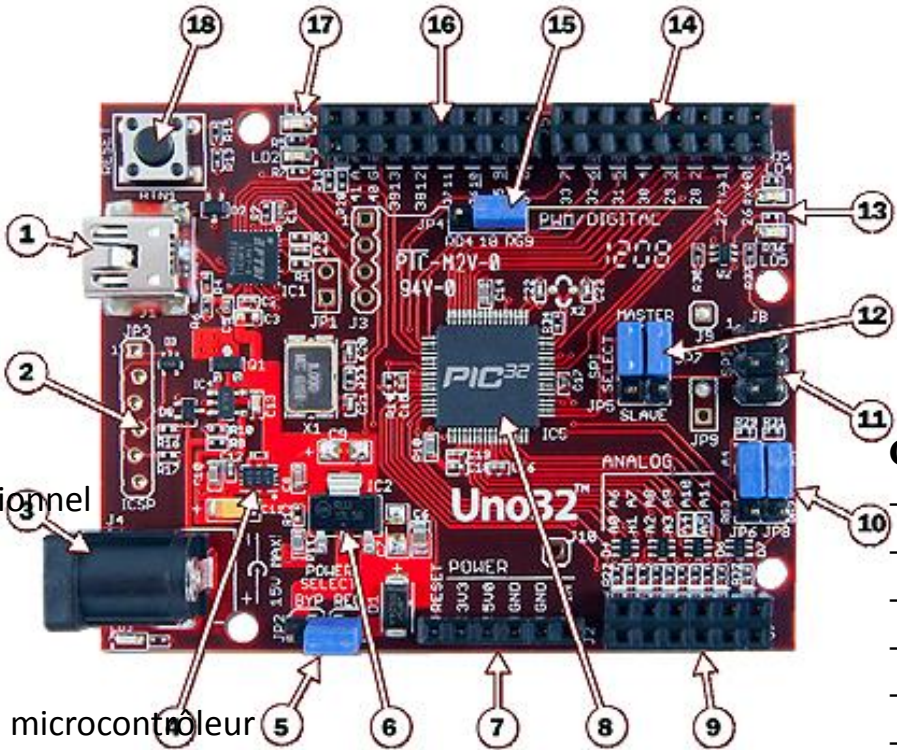
The Stellaris® LM4F120 LaunchPad Evaluation Board is a low-cost evaluation platform for ARM® Cortex™-M4F-based microcontrollers from Texas Instruments. The design of the Stellaris LaunchPad highlights the LM4F120H5QR microcontroller with a USB 2.0 device interface and hibernation module.

The EK-LM4F120XL also features programmable user buttons and an RGB LED for custom applications. The stackable headers of the Stellaris LM4F120 LaunchPad BoosterPack XL Interface make it easy and simple to expand the functionality of the Stellaris LaunchPad when interfacing to other peripherals with Stellaris BoosterPacks and MSP430™ BoosterPacks.



**Module chipKIT Uno32™**

Le module chipKIT Uno32™ est une plate-forme de prototypage microcontrôlée "open-source" **compatible avec le brochage et les applications développées pour les Arduino™**. Ce modèle (bien plus puissant) est achitecturé autour d'un micro-contrôleur Microchip® **32 bits** PIC32MX320F128.

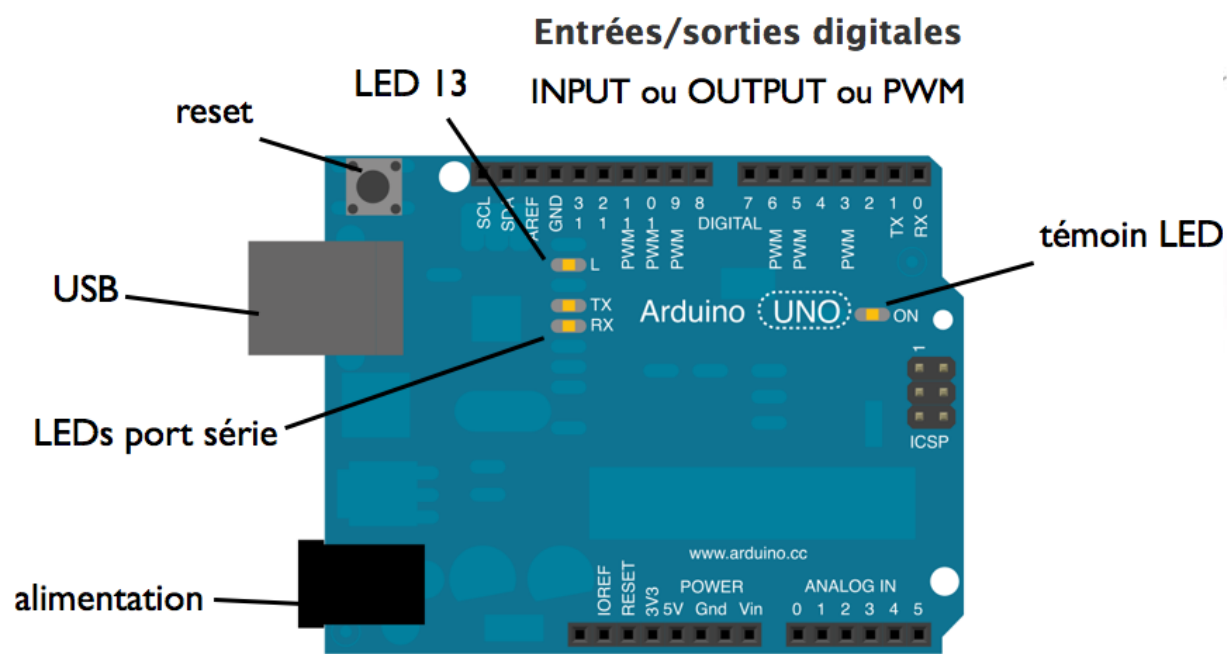


**Détail de la platine:**

- 01 - Connexion USB
- 02 - Connexion pour outil de debug optionnel
- 03 - Connecteur pour alimentation
- 04 - Régulateur 3,3 V
- 05 - Cavalier sélection alimentation
- 06 - Régulateur 5 V
- 07 - Connecteur de reprise des ports du microcontrôleur
- 08 - Microcontrôleur PIC32™
- 09 - Connecteur de reprise des ports du microcontrôleur
- 10 - Cavalier sélection signaux
- 11 - Connexion SPI
- 12 - Sélection SPI mode maître / esclave
- 13 - 2 Leds
- 14 - Connecteur de reprise des ports du microcontrôleur
- 15 - Cavalier sélection signaux
- 16 - Connecteur de reprise des ports du microcontrôleur
- 17 - Leds présence communication USB
- 18 - Bouton-poussoir de Reset

**Caractéristiques de la platine:**

- Microcontrôleur: 32 bits - PIC32MX320F128 (Microchip™)
- Vitesse d'horloge: 80 MHz
- Mémoire Flash: 128 K
- Mémoire RAM: 16 K
- Nombres d'entrée/sorties: 42
- 2 UART (avec mode IrDA)
- 2 ports I2C™
- 2 ports SPI™
- 2 Comparateurs
- Horloge RTC interne
- Timers internes 16/32 bits
- Leds libres d'utilisation
- Compatible avec les programmes développés pour Arduino™
- Compatible avec les platines Shield en 3,3 V pour Arduino™
- Programmable également sous MPLAB™ (avec dongle en option)



Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

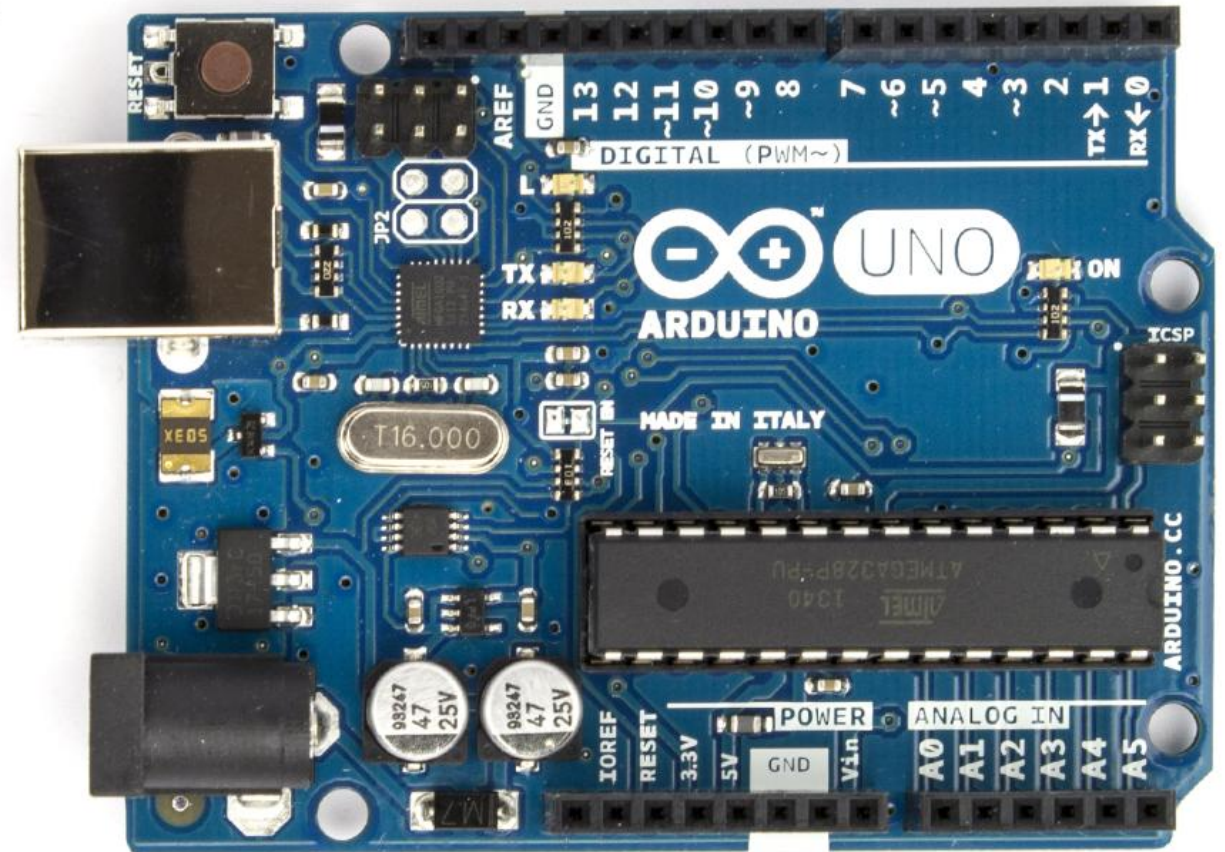
Length 68.6 mm

Width 53.4 mm

Weight 25 g

### Entrées analogiques

6 channels analog-to-digital 10bits converter  
 ou GPIO pins (digital)



		Pr oc ess	Ho rlo ge	Me m flash	RAM	Dig/A nal input	Audio	U S B	D V I	H D M I	Ether	I 2 C	S P I	?	€
Ardui no	UNO	8	16	32	2	14/6		x				x	x		21
	due	32	84	512	96	54/12		x				x	x	3.3V	
	leonardo	8	16	32	2	14/6		x						usb	
	tre	32	1G hz	512		23		x		x	x	2	2		
Dilige nt tech	UNO32	32	80	128	16	42		x			x	2	2	Colab beagle-compatible arduino / shields	32
TI	Beagle board			128/ 512			x	x	x	x				Open source computing	
raspe ry	PI	32	70 0	256/ 1G	Carte sd	8 gpio	x	1 a 4		x	X (b)			Linux/port video	37
TI	stellaris launchpad	32												Info peu buvables ! Soft energia	5 /28 dollars!



# Download the Arduino Software



## ARDUINO 1.6.3

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

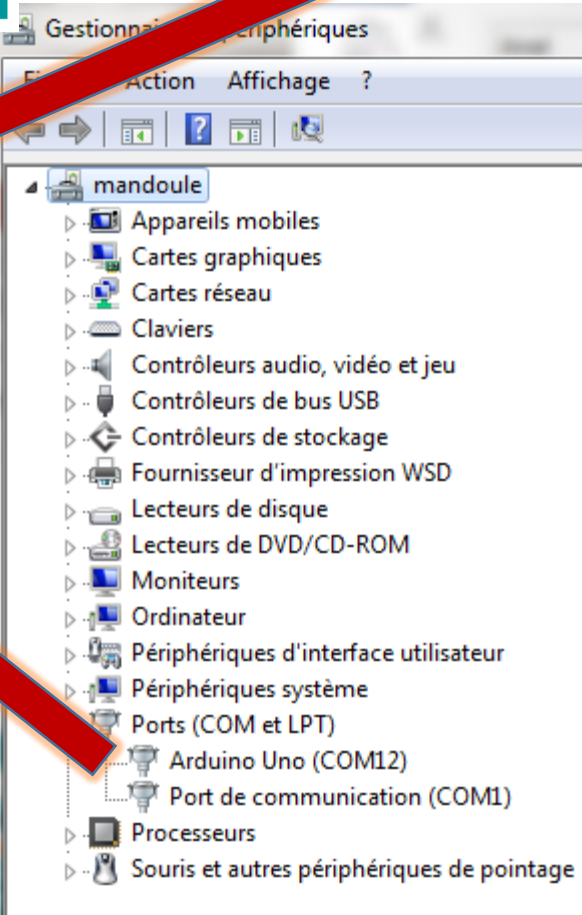
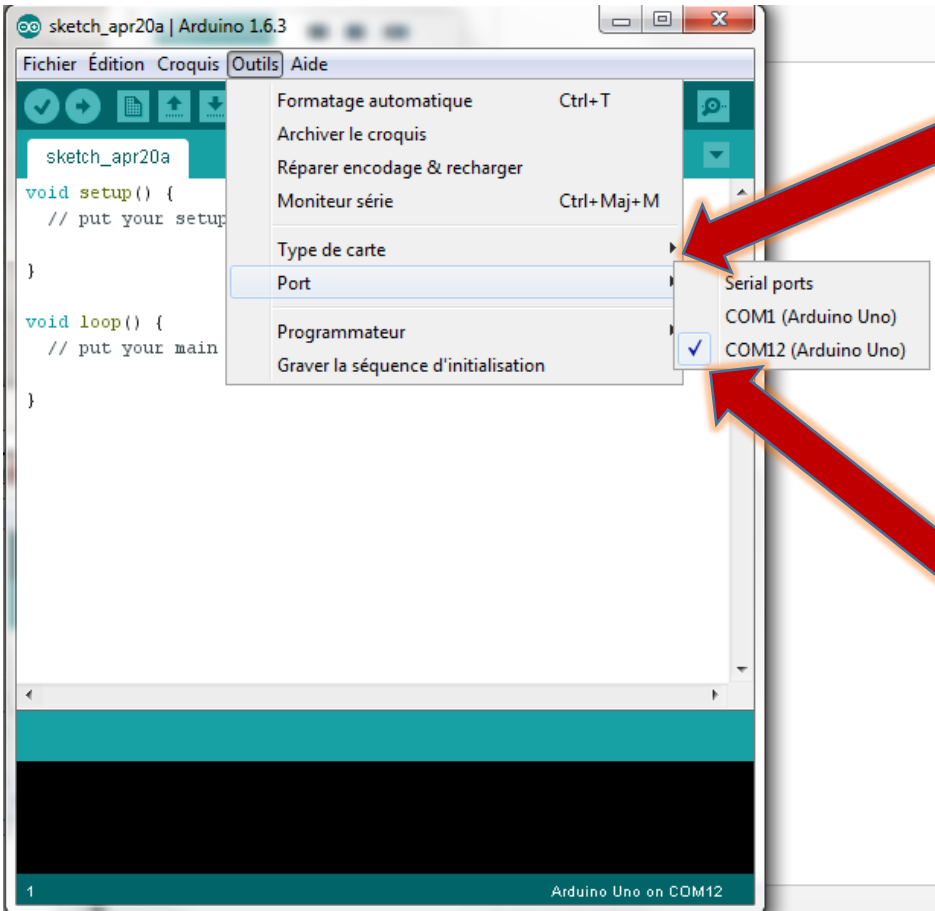
**Windows** Installer  
**Windows** ZIP file for non admin install

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits  
**Linux** 64 bits

[Release Notes](#)  
[Source Code](#)  
[Checksums](#)

- [Arduino BT](#)
- [Arduino Due](#)
- [Arduino Esplora](#)
- [Arduino Fio](#)
- [Arduino ISP](#)
- [Arduino Leonardo and Micro](#)
- [Arduino Mini](#)
- [Arduino Nano](#)
- [Arduino Pro](#)
- [Arduino Pro Mini](#)
- [Arduino Robot](#)
- [Arduino Yún](#)
- [Ethernet shield](#)
- [GSM shield](#)
- [Intel Edison](#)
- [Intel Galileo Gen2](#)
- [LilyPad Arduino](#)
- [LilyPad Arduino USB](#)
- [TFT LCD Screen](#)
- [WiFi shield](#)
- [Wireless shield](#)



# Analog Input Pins

6 channel analog-to-digital (A/D) converter .The `analogRead()` command converts the input voltage range, 0 to 5 volts, to a digital value between 0 and 1023. This is done by a circuit inside the Arduino called an analog-to-digital converter or ADC

the analog pins also have pullup resistors, which work identically to pullup resistors on the digital pins. they are enabled by issuing a command such as `digitalWrite(a0, HIGH);` // set pullup on analog pin 0 while the pin is an input



# Commandes de structure du programme

## Structure générale

- `void setup()` (configuration-préparation)
- `void loop()` (exécution)

## Contrôle et conditions

- `if` (si...)
- `if...else` (si...alors...)
- `for` (pour...)
- `switch case` (dans le cas où...)
- `while` (pendant que ...)

## Opérations de comparaison

- `==` (équivalent à)
- `!=` (différent de)
- `<` (inférieur à)
- `>` (supérieur à)
- `<=` (inférieur ou égal à)
- `>=` (supérieur ou égal à)

## Opérations booléennes

- `&&` (et)
- `||` (ou)
- `!` (et pas)

## Autres commandes

- `//` (commentaire simple ligne)
- `/* */` (commentaire multi-lignes)
- `#define` (donner une valeur à un nom)

## Variables

- `char` (variable 'caractère')
- `int` (variable 'nombre entier')
- `long` (variable 'nombre entier de très grande taille')
- `string` (variable 'chaîne de caractères')
- `array` (tableau de variables)

## Variables

### Niveaux logiques des connecteurs numériques

- `HIGH` (état 1)
- `LOW` (état 0)
- `INPUT` (configuré en entrée)
- `OUTPUT` (configuré en sortie)

## Fonctions

### Entrées-sorties numériques

- `pinMode(broche, état)` (configuration des broches)
- `digitalWrite(broche, état)` (écrire un état sur une broche num.)
- `digitalRead(broche)` (lire un état sur une broche num.)
- `unsigned long pulseIn(broche, état)` (lire une impulsion sur une broche num.)

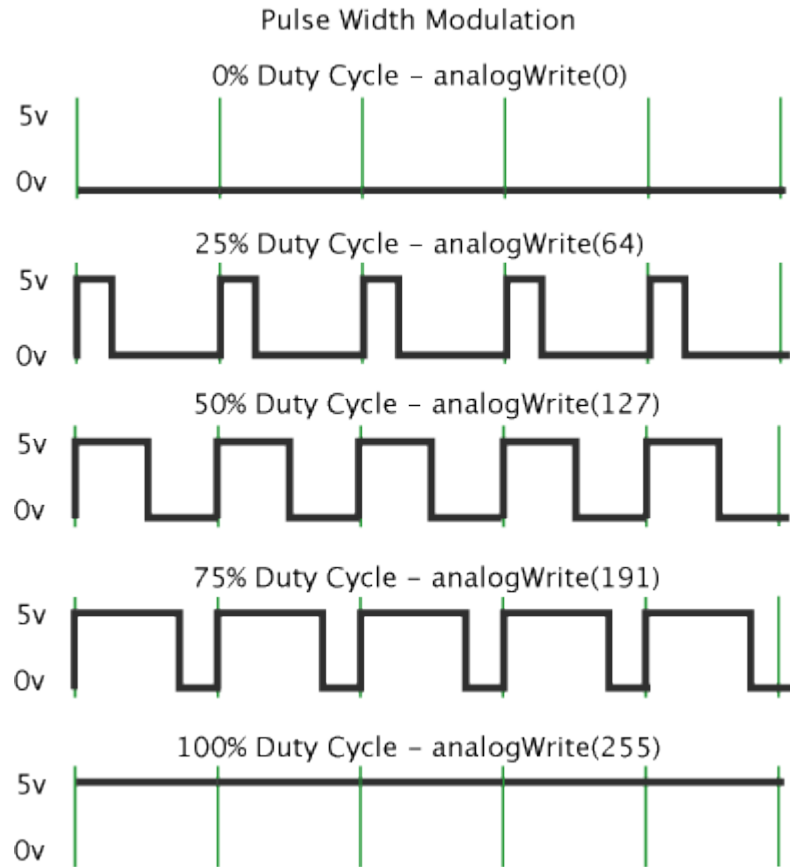
### Entrées analogiques

- `int analogRead(broche)` (lire la valeur d'une broche ana.)
- `analogWrite(broche, valeur)` (PWM : écrire une valeur analogique sur les broches 9, 10 ou 11)

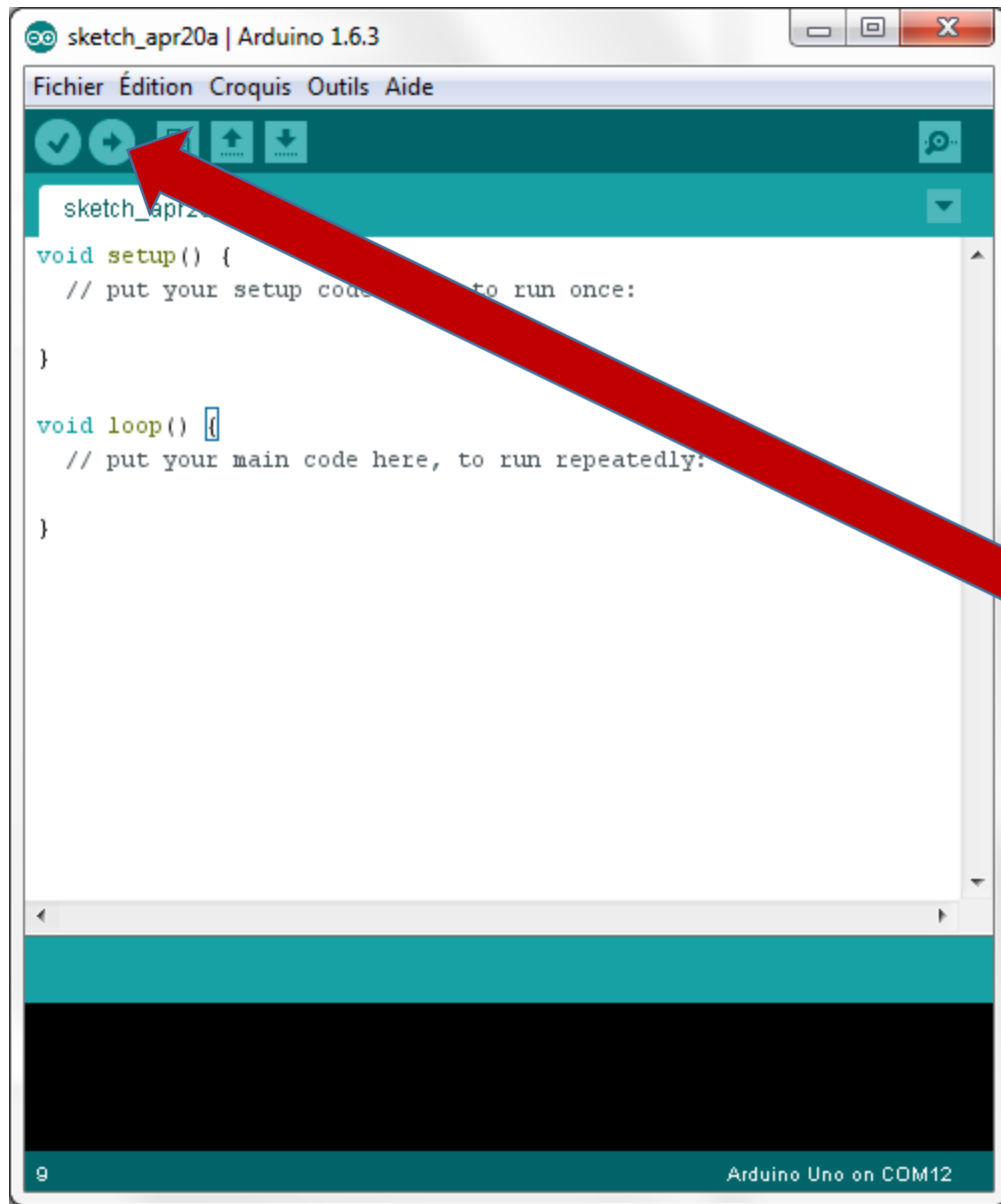
### Gestion du temps

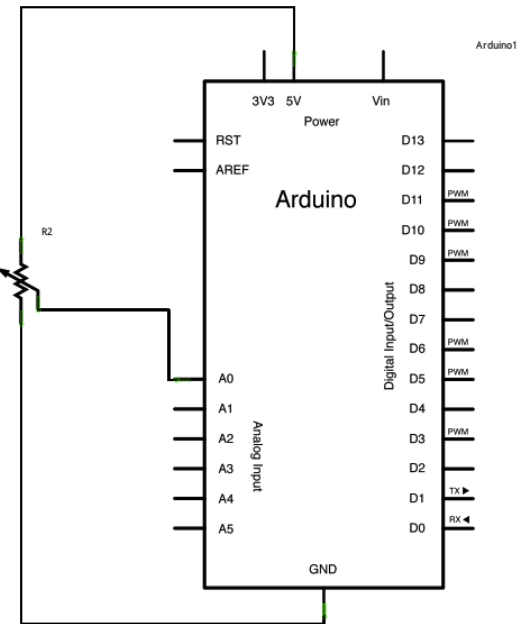
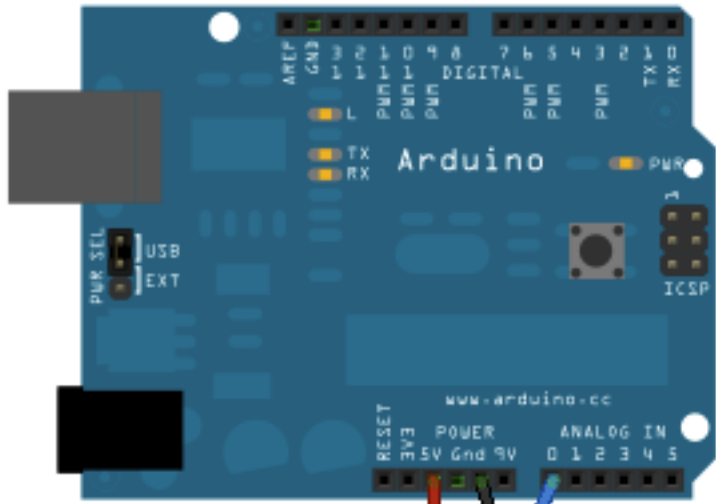
- `unsigned long millis()` (temps de fonctionnement du programme)
- `delay(ms)` (attente, en millisecondes)
- `delayMicroseconds(us)` (attente, en microsecondes)

# Analogwrite on pwm pin



ANALOGWRITE(127) IS A 50%





```
/*
```

```
  ReadAnalogVoltage
```

```
  Reads an analog input on pin 0, converts it to voltage, and
  prints the result to the serial monitor.
```

```
  Attach the center pin of a potentiometer to pin A0, and the
  outside pins to +5V and ground.
```

```
  This example code is in the public domain.
```

```
*/
```

```
// the setup routine runs once when you press reset:
```

```
void setup() {
```

```
  // initialize serial communication at 9600 bits per second:
```

```
  Serial.begin(9600);
```

```
}
```

```
// the loop routine runs over and over again forever:
```

```
void loop() {
```

```
  // read the input on analog pin 0:
```

```
  int sensorValue = analogRead(A0);
```

```
  // Convert the analog reading (which goes from 0 - 1023) to a
  voltage (0 - 5V):
```

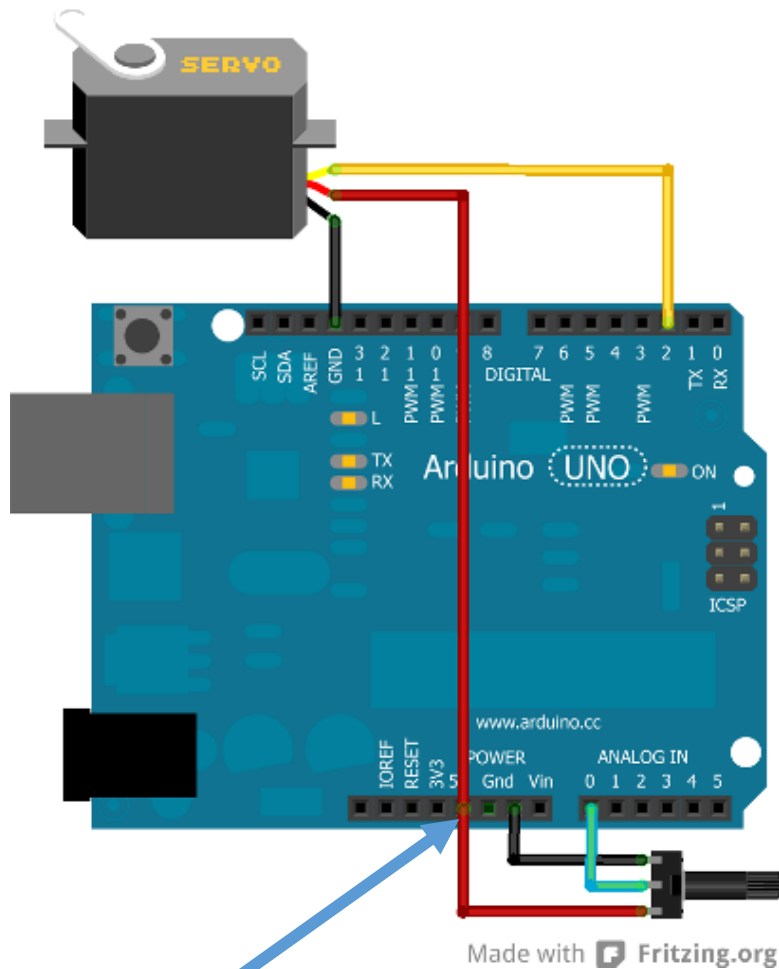
```
  float voltage = sensorValue * (5.0 / 1023.0);
```

```
  // print out the value you read:
```

```
  Serial.println(voltage);
```

```
}
```





Connecté en 5V

Vidéo moteur servo

[https://www.youtube.com/watch?feature=player\\_embedded&v=-XSxfqd1N58](https://www.youtube.com/watch?feature=player_embedded&v=-XSxfqd1N58)

<http://eskimon.fr/287-arduino-602-un-moteur-qui-de-la-tete-le-servo-moteur>

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

```
const int potar = 0; //notre potentiomètre
```

```
//création d'un nouveau servomoteur
Servo monServo;
```

```
void setup()
{
    //on déclare l'entrée du servo connectée sur la broche 2
    monServo.attach(2);

    Serial.begin(9600);
}

void loop()
{
    //on lit la valeur du potentiomètre
    int val = analogRead(potar);
    //mise à l'échelle de la valeur lue vers la plage [0;180]
    int angle = map(val, 0, 1023, 0, 180);
    //on met à jour l'angle sur le servo
    monServo.write(angle);
}
```

# Servo library

This library allows an Arduino board to control RC (hobby) servo motors. Servos have integrated gears and a shaft that can be precisely controlled. Standard servos allow the shaft to be positioned at various angles, usually between 0 and 180 degrees. Continuous rotation servos allow the rotation of the shaft to be set to various speeds.

The Servo library supports up to 12 motors on most Arduino boards and 48 on the Arduino Mega. On boards other than the Mega, use of the library disables `analogWrite()` (PWM) functionality on pins 9 and 10, whether or not there is a Servo on those pins. On the Mega, up to 12 servos can be used without interfering with PWM functionality; use of 12 to 23 motors will disable PWM on pins 11 and 12.

## Circuit

Servo motors have three wires: power, ground, and signal. The power wire is typically red, and should be connected to the 5V pin on the

## Functions

- `attach()`
- `write()`
- `writeMicroseconds()`
- `read()`
- `attached()`
- `detach()`

## Examples

- **Knob**: control the shaft of a servo motor by turning a potentiometer.
- **Sweep**: sweeps the shaft of a servo motor back and forth.