



innoFAB

CASTRES-MAZAMET

Arduino – Premier contact

ou

comment faire de l'électronique en utilisant un langage de programmation ?

InnoFab



Applications possibles

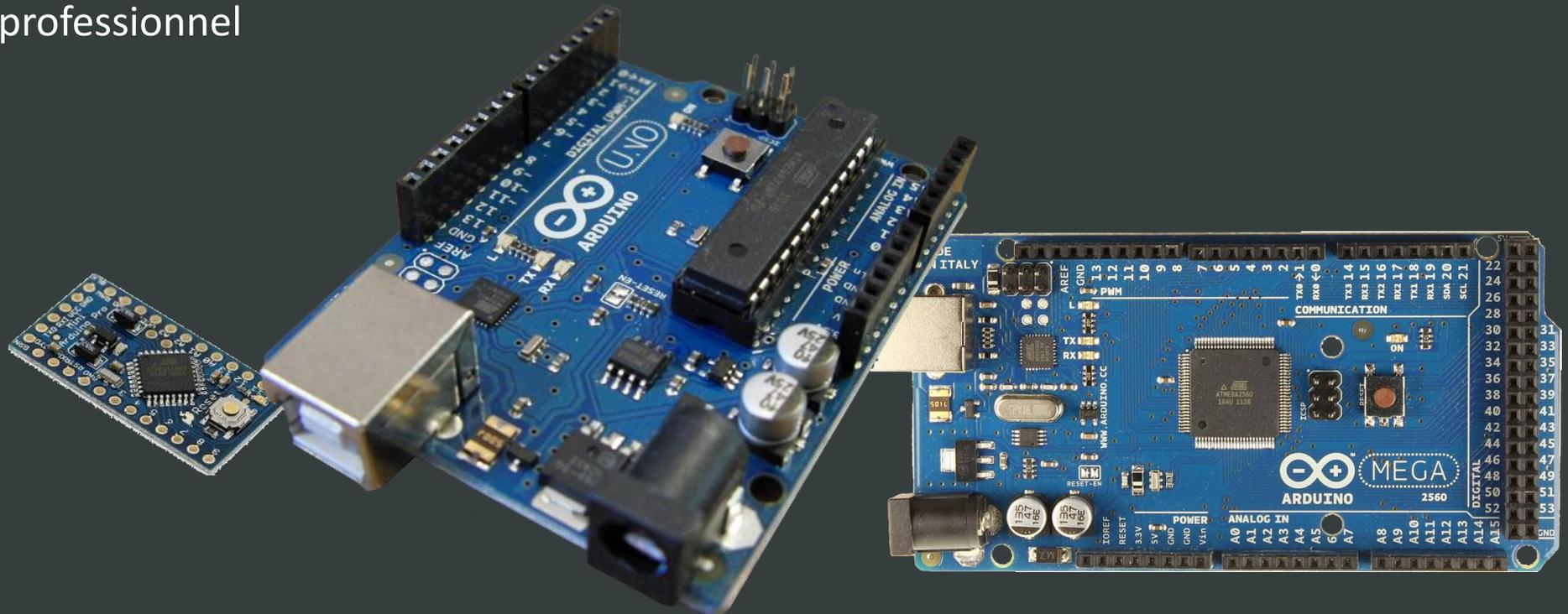
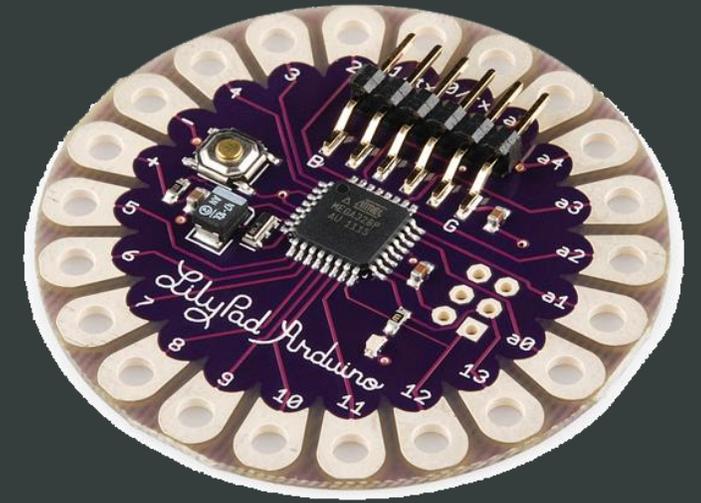
- contrôler des appareils domestiques
- donner une "intelligence" à un robot
- réaliser des jeux de lumières
- permettre à un ordinateur de communiquer avec une carte électronique et différents capteurs
- télécommander un appareil mobile (modélisme)
- etc...

-> « arduino » sur www.instructables.com



Introduction à l'Arduino

- Faire de l'électronique en utilisant un langage de programmation
- Accessible à tous par sa simplicité
- Monde professionnel

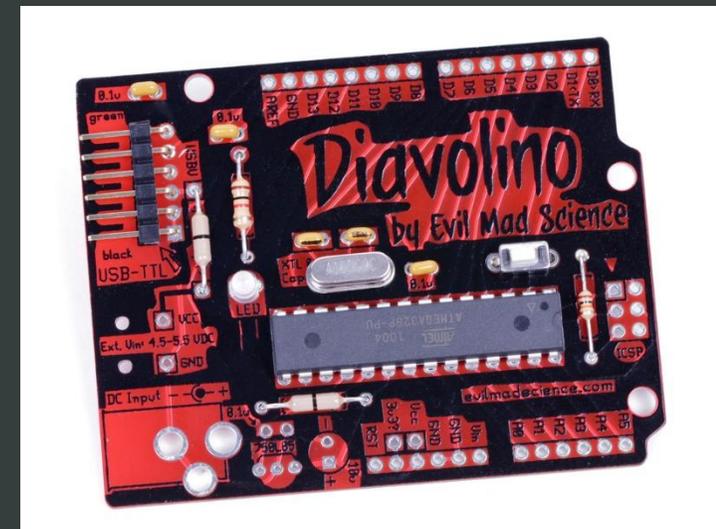




Arduino, Genuino, Freeduino, Seeduino, ...

Il existe trois types de cartes :

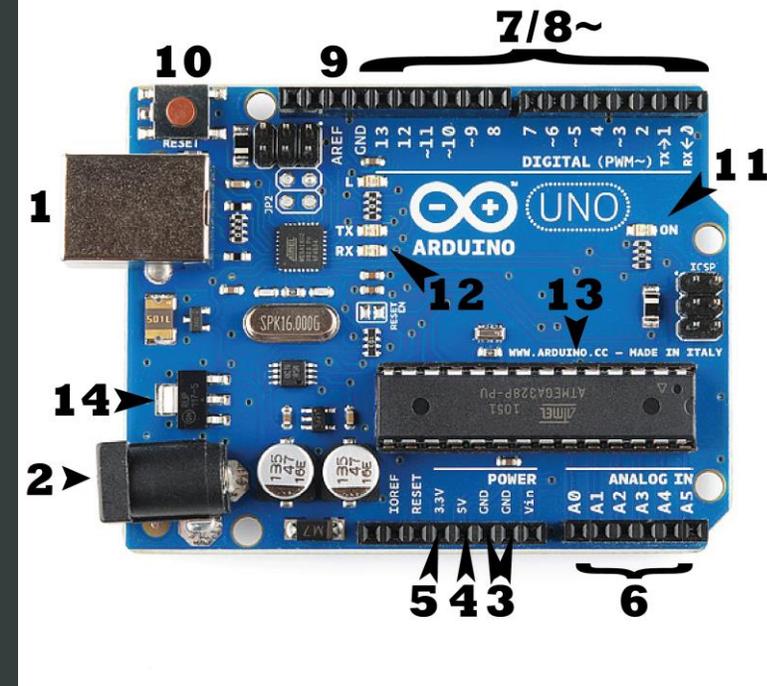
- « officielles » : fabriquées en Italie par le fabricant officiel : Smart Projects
- « compatibles » : compatibles avec les Arduino officielles.
- « autres » : commercialisées sous un nom différent (Freeduino, Seeduino, Femtoduino, ...).



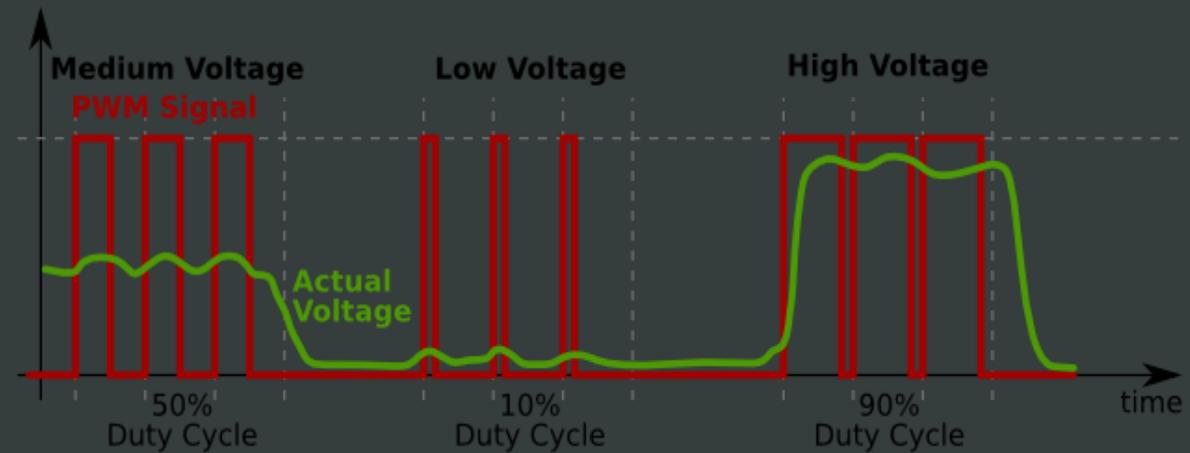
Arduino UNO

COTÉ MATÉRIEL

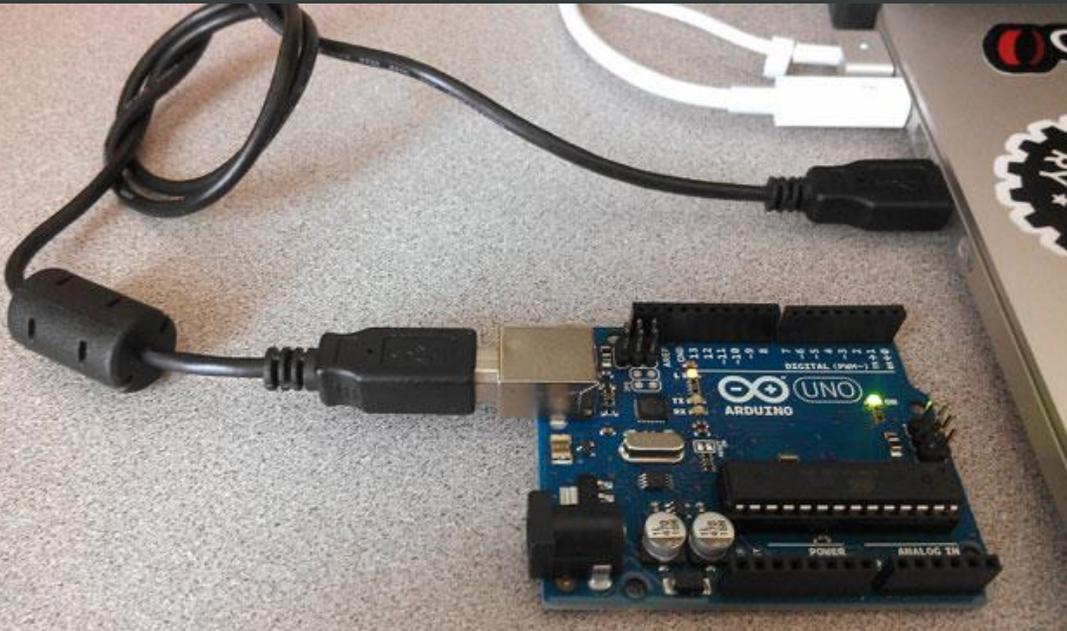
- Micro-contrôleur ATmega328P
- 14 connecteurs pour toutes les entrées et sorties numériques (dont 6 avec modulation de largeur d'impulsions)
- 6 entrées analogiques
- Port USB type B (carré) pour la programmation et/ou la communication
- Un connecteur d'alimentation
- Tension de fonctionnement : 5V
Tension d'entrée : 7-12V (limites: 6-20V)



- 1) USB
- 2) Jack
- 3) GND
- 4-5) Sortie d'alimentation
- 6) Entrée Analog (>=0 et <=1023)
- 7) Entrée / Sortie Digital (0 ou 1)
- 8) PWM (Pulse-Width Modulation) (3, 5, 6, 9, 10 et 11)
- 9) AREF (Analog Reference)
- 10) Reset
- 11) Power LED
- 12) TX & RX LED
- 13) Micro-contrôleur



Arduino UNO



COTÉ INFORMATIQUE

- Un environnement de programmation regroupant éditeur de code, compilation, upload et débogueur
- Multi-plateforme
- Un langage C/C++ avec des simplifications pour les débutants
- Des bibliothèques open source

```
Blink | Arduino 1.5.3-Intel.1.0.4

/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

Done uploading.
moving downloaded file to /sketch/sketch.elf on target
echo "Moving downloaded file to /sketch/sketch.elf on target"
"$fixed_path/lrz" --escape -c "mv $target_download_name /sketch/sketch.elf;
chmod +x /sketch/sketch.elf" < $tty_port_id > $tty_port_id

Transfer complete
#
#
1 Intel® Edison on /dev/cu.usbmodem1413
```



Bibliothèques

- EEPROM : lecture et écriture de données dans la mémoire permanente.
- Ethernet : pour se connecter à Internet en utilisant le Shield Ethernet.
- LiquidCrystal : pour contrôler les afficheurs à cristaux liquides (LCD).
- SD : pour la lecture et l'écriture de données sur des cartes SD.
- Servo : pour contrôler les servomoteurs.
- SPI : pour communiquer avec les appareils qui utilisent le protocole de communication SPI (Serial Peripheral Interface).
- Stepper : pour commander des moteurs « pas à pas ».
- Wire : pour interfacer plusieurs modules électroniques sur un bus de données utilisant le protocole de communication TWI/I2C.
- ...

-> <http://www.arduino.cc/en/Reference/Libraries>



IDE / Outils

```
1 #include <Arduino.h>
2
3 /*
4  Turns on an LED on for one second, then off for one second, repeatedly.
5
6  This example code is in the public domain.
7  */
8
9 void setup()
10 {
11   Serial.begin(9600);
12
13   // initialize the digital pin as an output.
14   // Pin 13 has an LED connected on most Arduino boards:
15   pinMode(13, OUTPUT);
16
17   Serial.println("Hello");
18 }
19
20 void loop()
21 {
22
23   delay(1000);
24   digitalWrite(13, HIGH);
25   delay(1000);
26   digitalWrite(13, LOW);
27 }
```

Arduino Uploader (Ver. 0.7.0) Powered by ArduinoDev.com

COM1 COM10 USBASP ArduinoISP

Choose a serial port or programmer to start

Memory Type	Used	Free
FLASH Utilization	2718 bytes	30050 bytes
SRAM Utilization	213 bytes	1835 bytes
EEPROM Utilization	0 bytes	1024 bytes

ArduBlock untitled *

Nouveau Enregistrer Enregistrer sous Ouvrir Téléverser vers l'Arduino Moniteur série

Contrôle
Broches
Tests
Opérateurs mathématiques
Variables/constantes
Generic Hardware
Communication
SCOOP (Multitask)
Stockage
Mise en réseau

Faire

set integer variable Nom de la variable: joyx Valeur: Broche#: 1 Mini joystick Grove

set integer variable Nom de la variable: joyy Valeur: Broche#: 2 Mini joystick Grove

boucle

Servo Plus Broche#: 3 Angle: joyx Vitesse: 1000

Servo Plus Broche#: 4 Angle: joyy Vitesse: 1000

Oled I2C Angle X= joyx de ligne: 1 Luminosité: 255

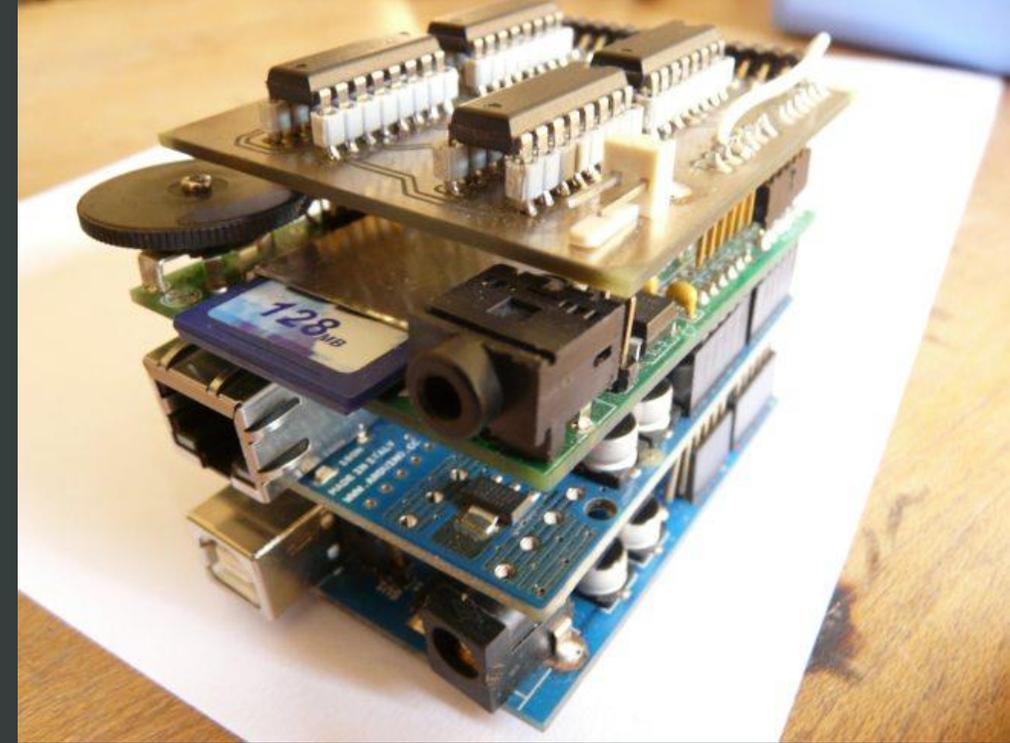
Oled I2C Angle Y= joyy

Enregistrer comme image... Accéder au Site Web v 21040826 (beta)



Programmation

```
void  setup() {  
    //contenu de l'initialisation  
}  
  
void  loop() {  
    //contenu de la fonction principale  
}
```



Les principales instructions

```
pinMode(pin, INPUT);
```

Configure la broche spécifiée pour qu'elle se comporte soit en entrée, soit en sortie OUTPUT, INPUT, INPUT_PULLUP
Par défaut: INPUT

Une entrée est généralement utilisée par les capteurs qui envoient des informations à l'Arduino et les sorties servent à communiquer et/ou à alimenter les composants.

Les principales instructions

<code>pinMode(pin, INPUT);</code>	Configure la broche spécifiée pour qu'elle se comporte soit en entrée, soit en sortie OUTPUT, INPUT, INPUT_PULLUP Par défaut: INPUT
<code>digitalWrite(pin, HIGH);</code>	Envoie du courant positif ou négatif (+5v ou GND) dans un pin de l'Arduino. HIGH pour +5v ou LOW pour GND
<code>digitalRead(pin);</code>	Permet de connaître le courant circulant au niveau du pin concerné HIGH ou LOW

Les principales instructions

<code>pinMode(pin, INPUT);</code>	Configure la broche spécifiée pour qu'elle se comporte soit en entrée, soit en sortie OUTPUT, INPUT, INPUT_PULLUP Par défaut: INPUT
<code>digitalWrite(pin, HIGH);</code>	Envoie du courant positif ou négatif (+5v ou GND) dans un pin de l'Arduino. HIGH pour +5v ou LOW pour GND
<code>digitalRead(pin);</code>	Permet de connaître le courant circulant au niveau du pin concerné HIGH ou LOW

```
void setup() {
  pinMode(7, INPUT); // on déclare notre entrée
  pinMode(8, OUTPUT); // on déclare notre sortie
}

void loop() {
  if(digitalRead(7) == HIGH) { // on regarde si notre entrée capte du courant
    digitalWrite(8, LOW); // on éteint la DEL
  } else {
    digitalWrite(8, HIGH); // on allume la DEL
  }
}
```

Les principales instructions

<code>pinMode(pin, INPUT);</code>	Configure la broche spécifiée pour qu'elle se comporte soit en entrée, soit en sortie OUTPUT, INPUT, INPUT_PULLUP Par défaut: INPUT
<code>digitalWrite(pin, HIGH);</code>	Envoie du courant positif ou négatif (+5v ou GND) dans un pin de l'Arduino. HIGH pour +5v ou LOW pour GND
<code>digitalRead(pin);</code>	Permet de connaître le courant circulant au niveau du pin concerné HIGH ou LOW
<code>analogWrite(pin, 123);</code>	Permet de faire passer un courant plus ou moins puissant, compris entre 0 et 255 dans le pin concerné
<code>analogRead(pin);</code>	Permet de connaître le courant circulant au niveau du pin concerné Entre 0 et 1023

Les principales instructions

<code>pinMode(pin, INPUT);</code>	Configure la broche spécifiée pour qu'elle se comporte soit en entrée, soit en sortie OUTPUT, INPUT, INPUT_PULLUP Par défaut: INPUT
<code>digitalWrite(pin, HIGH);</code>	Envoie du courant positif ou négatif (+5v ou GND) dans un pin de l'Arduino. HIGH pour +5v ou LOW pour GND
<code>digitalRead(pin);</code>	Permet de connaître le courant circulant au niveau du pin concerné HIGH ou LOW
<code>analogWrite(pin, 123);</code>	Permet de faire passer un courant plus ou moins puissant, compris entre 0 et 255 dans le pin concerné
<code>analogRead(pin);</code>	Permet de connaître le courant circulant au niveau du pin concerné Entre 0 et 1023
<code>delay(1000);</code>	Mettre en pause l'Arduino pendant un temps indiqué, en millisecondes
<code>millis();</code>	Retourne le nombre de millisecondes depuis que le programme a démarré Le compteur est réinitialisé à zéro après environ 50 jours.

DÉMONSTRATION



Questions ?

Merci pour votre attention

innofab.fr