

# Parcours Arduino débutant

## Que peut on faire avec un Arduino

[Domotique](#)

[Autres Idées 001](#)

[Idées 002](#)

[Modélisme Ferroviaire](#)

[Fabriquer un Drone avec un Arduino](#)

[Modelisme 001](#)

[Programmes pour modelisme](#)

[Arduino et Raspberry 001](#)

[Arduino et Raspberry 002](#)

[Arduino et Raspberry 003](#)

[Un Livre qui étudie des projets avec le Raspberry et/ou L'Arduino](#)

[On fini en beauté : avec un Feu d'artifice](#)

## Toutes les cartes arduino

[les cartes arduino actuelles](#)

## Les cartes Arduino que l'on va utiliser





Arduino Nano Connect RP2040

<https://store.arduino.cc/products/arduino-nano-rp2040-connect-with-headers>

## Les logiciels utiliser

**A télécharger sur votre ordinateur :**

La version **IDE Arduino** au 01/09/2024 est la 2.3.2 , mais on peut bien sur utiliser une version plus ancienne 1.8.19 ou [une version en ligne](#)

\* [IDE Arduino](#)

Interface graphique d'utilisation de l'utilitaire **"avrdude"** de Atmel (celui qui est utilisé, notamment, par Arduino IDE pour téléverser le firmware ou fichier .hex sur la carte mère de votre imprimante préférée, dès lors qu'elle est à base de microcontrôleur Atmel (comme l'ATmega 2560, UNO, ou ATiny ). Cet outil est développé et entretenu par Zak Kemble (cf.

<http://blog.zakkemle.net/avrdudess-a-gui-for-avrdude/>). \*[AVRDUDESS](#)

**PlatformIO** est un éco système open source conçu pour le développement IOS supportant le framework Arduino. Celui-ci est accessible depuis tout type de systèmes d'exploitation puis qu'il est programmé en python. \*[platformio pour programmer un arduino](#)

**Pour tester en ligne**

[Tinkercad](#)

Après avoir créer un compte gratuit ou utiliser votre compte Google , vous pourrez tester vos montage Arduino et aussi faire de modélisation ( Simplifiée ) pour des pieces 3D .

[Un exemple de montage arduino](#)

## Le materiel dont on va avoir besoin

### Comparaison des caractéristiques des principales cartes Arduino et autres cartes

| Cartes                     | MControleurs  | (bits) | Mhz              | V                     | Flash(KB) | Boot(KB) | SRam(KB) | EEprom     | DigitalI/O | PWM  | AnalogI/O | int |
|----------------------------|---|--------|------------------|-----------------------|-----------|----------|----------|------------|------------|------|-----------|-----|
| <b>Uno</b>                 | ATmega328P  | 8      | 16               | 5                     | 32        | 0.5      | 2        | 1          | 14         | 6    | 6         | 2   |
| <b>Nano</b>                | ATmega328P  | 8      | 16               | 5                     | 32        | 2        | 2        | 1          | 14         | 6    | 8         | 2   |
| <b>Mega</b>                | ATmega2560  | 8      | 16               | 5                     | 256       | 8        | 8        | 4          | 54         | 15   | 16        | 6   |
| <b>Giga</b>                | STM32H747XI<br>2Coeurs                                      | 32     | M7=480<br>M4=240 | 3.3                   | 2MB       |          | 1MB      |            | 76         | 12   | 12        |     |
| Micro                      | ATmega32U4  | 8      | 16               | 5                     | 32        | 4        | 2.5      | 1          | 20         | 7    | 12        | 5   |
| Leonardo                   | ATmega32U4  | 8      | 16               | 5                     | 32        | 4        | 2.5      | 1          | 20         | 7    | 12        | 5   |
| Nano<br>RP2040             | ARM Cortex<br>M0+   | 32     | 133              | 3.3                   | 16M       | -        | 264      | -          | 20         | 20   | 8         | 20  |
| MKR ZERO                   | SAMD21 Cortex-<br>M0  | 32     | 48               | 3.3                   | 256       | -        | 32       | no         | 22         | 12   | 8         | 10  |
| <b>DUE</b>                 | AT91SAM3X8E   | 32     | 84               | 3.3                   | 512       | -        | 96       | -          | 54         | 12   | 14        | 54  |
| MKR WIFI<br>1010           | SAMD21 Cortex-<br>M0  | 32     | 48               | 3.3                   | 256       | -        | 32       | no         | 8          | 12   | 8         | 10  |
| <b>MKR Vidor<br/>4000]</b> | Intel®<br>Cyclone®<br>10CL016 FPGA-<br>SAMD21<br>Cortex®-M0 | 32     | 48/200           | 3.3                   | 2000      | -        | 8000     | no         | 22         | tout | 8         | 8   |
| <b>ESP32</b>               | Xtensa® single-<br>/dual-core                               | 32     | 240              | 2.7<br>~<br>3.6       | 64000     | -        | 520      | 448<br>ROM | 28         | tout | 18        | -   |
| <b>PYBStick26</b>          | STM32F411   | 32     | 100              | 1.7<br>V<br>to<br>3.6 | 512       | -        | 128      | -          | 17         | 17   | 6         | -   |
| <b>DocAtiny85</b>          | ATINY85   | 8      | 1-20             | 5                     | 8         | -        | 512o     | 512o       | 6          | 2    | 2         | -   |

### Materiels

#### Des liens pour des kits arduino

[Arduino France : Kits 001 Arduino](#)

[Cdiscount : Kits 002 Arduino](#)

[Kubii : Kits 003 Arduino](#)

[Lextronic : Kits 004 Arduino](#)

[Semageek :Kits 005 Arduino](#)

[Arduino cc :Kits 006 Arduino](#)

[Gotronic :Kits 007 Arduino](#)

[eBay : Kits 008 Arduino](#)

[Adafruit : Kits 009 Arduino](#)

[SainSmart : Kits 010 Arduino](#)

.....

## **Pour apprendre à utiliser un Arduino**

[Un MOOC pour s initier à l'Arduino \(fin inscription 12/2020\)](#)

[Des cours donner au Funlab de Tours il y a quelques temps ...](#)

[Des Quiz pour mieux connaître l'Arduino](#)

[Le langage C](#)

[Le Langage C++ Le C++ en Ligne](#)

[Reference Langage Arduino FR](#)

[ATTiny 85](#)

Doc : SRAM Arduino

[Des programmes Arduino ...](#)

[Arduino : le blog d'Eskimon](#)

## **Des Livres**

\*

[Arduino : le guide complet pdf FR](#)

\*

[Arduino pour les nuls](#)

[Arduino le grand Livre](#)

[Arduino Reference EN](#)

## **On apprend de nos erreurs**

[les variables globales](#) et la fonction void setup [Les variables](#)

[le sens des LEDS](#) : cathode = -, anode = +, le meplat = -

[la valeur d une résistance](#) : 220 ohms n'est pas 10k ohms [mesure des resistances et anneaux Calcul resistances Tension de seuil](#)

la connexion à la carte Arduino : [les ports USB sur windows 10](#)

La syntaxe du langage Arduino : [les ; et les { et }](#)

### Non détection des cartes Arduino :

Nano : soit mettre ancien bootloader, soit mettre Uno comme carte dans -> outils -> type de cartes  
-> UNO

Mega ou nano : le port COM non reconnu sous W10 , mettre à jour pilote dans gestion des périphériques, clic droit sur port COM et MAJ pilotes

## Les programmes pour commencer

### Pour tester la carte : Blink

[Voir dans les programmes exemples de l IDE arduino](#)

Arduino le grand Livre p221  
Programmes des montages N°1

[Blink.ino](#)

```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);      // Définit la pin 13 comme pin de sortie
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // Allume la LED
  delay(1000);               // Attend une seconde
  digitalWrite(led, LOW);    // Eteint la LED
  delay(1000);               // Attend une seconde
}
```

### Le chenillard avec des LEDS Programme 1



### Chenillard001.ino

```
int led1 = 3; // Déclaration de la variable
int led2 = 5; // Déclaration de la variable
int led3 = 6; // Déclaration de la variable
int led4 = 9; // Déclaration de la variable
int led5 = 10; // Déclaration de la variable
int led6 = 11; // Déclaration de la variable

void setup() {
  pinMode(led1,OUTPUT); // Défini la variable en tant que sortie
  (OUTPUT)
  pinMode(led2,OUTPUT); // Défini la variable en tant que sortie
  (OUTPUT)
  pinMode(led3,OUTPUT); // Défini la variable en tant que sortie
  (OUTPUT)
  pinMode(led4,OUTPUT); // Défini la variable en tant que sortie
  (OUTPUT)
  pinMode(led5,OUTPUT); // Défini la variable en tant que sortie
  (OUTPUT)
  pinMode(led6,OUTPUT); // Défini la variable en tant que sortie
  (OUTPUT)
}

void loop() {

  digitalWrite(led1, HIGH);
  delay(200);
```

```
digitalWrite(led1, LOW);  
delay(200);  
  
digitalWrite(led2, HIGH);  
delay(200);  
digitalWrite(led2, LOW);  
delay(200);  
  
digitalWrite(led3, HIGH);  
delay(200);  
digitalWrite(led3, LOW);  
delay(200);  
  
digitalWrite(led4, HIGH);  
delay(200);  
digitalWrite(led4, LOW);  
delay(200);  
  
digitalWrite(led5, HIGH);  
delay(200);  
digitalWrite(led5, LOW);  
delay(200);  
  
digitalWrite(led6, HIGH);  
delay(200);  
digitalWrite(led6, LOW);  
delay(200);  
  
}
```

## Chenillard programme2

### Chenillard002.ino

```
// ce programme allume les leds une par une  
int pin[6]={3,5,6,9,10,11};  
int temp0=50; //tempo pour allumer les leds  
int temp1=50; // tempo pour eteindre  
  
void setup(){  
  for (int i = 0 ; i<=5 ; i++){  
    pinMode(pin[i], OUTPUT);  
  }  
}  
  
void loop(){  
  for (int i = 0 ; i<=5 ; i++){ // boucle for pour allumer les leds une
```

```
par une
digitalWrite(pin[i], HIGH);
delay(temp0);
digitalWrite(pin[i], LOW);
delay(temp0);
}
for (int i = 5 ; i>=0 ; i--){
digitalWrite(pin[i], LOW);
delay(temp1);
digitalWrite(pin[i], HIGH);
delay(temp0);

}

}
```

## Chenillard Programme 3 en utilisant une fonction sinus

[Chenillard003.ino](#)

```
// les sorties avec PWM
#define LED1 11
#define LED2 10
#define LED3 9
#define LED4 6
#define LED5 5
#define LED6 3

// Valeur pour decaler le sinus
#define DECAL PI*2/6

// configuration des sorties
void setup(){
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED5, OUTPUT);
  pinMode(LED6, OUTPUT);
}

void loop() {
  for (float i=0; i<PI*2; i=i+PI*2/1000) {
    analogWrite(LED1, sin(i)*127.5+127.5);
    analogWrite(LED2, sin(i+DECAL)*127.5+127.5);
    analogWrite(LED3, sin(i+DECAL*2)*127.5+127.5);
    analogWrite(LED4, sin(i+DECAL*3)*127.5+127.5);
    analogWrite(LED5, sin(i+DECAL*4)*127.5+127.5);
```

```
analogWrite(LED6, sin(i+DECAL*5)*127.5+127.5);  
}  
}
```

## Chenillard Programme 4 avec moins de ligne de code toujours en utilisant une fonction sinus

### ChenillardSinus4

```
// ce programme allume les leds une par une  
int pin[6]={3,5,6,9,10,11};  
  
// Valeur pour decaler le sinus  
#define DECAL PI*2/6  
  
void setup(){  
  for (int i = 0 ; i<=5 ; i++){  
    pinMode(pin[i], OUTPUT);  
  }  
}  
  
void loop() {  
  for (float i=0; i<PI*2; i=i+PI*2/300) {  
    analogWrite(LED6, sin(i+DECAL*5)*127.5+127.5);  
    // ATTENTION ce programme ne fonctionne pas il faut ajouter une  
    // boucle for ...  
    // A vous de chercher une solution pour remplacer 6 lignes par une  
    // seule....  
    // Indices : boucle for imbriquée dans la première avec une valeur  
    // de 0 à 5 ....  
  }  
}  
}
```

Correction  
programme correction

## Des Électronique 1 avec Bouton poussoir

### Des électronique 2

### La condition ternaire en langage C



```
#define WAITTIME 20
int pips[6][7] = {{0, 0, 0, 1, 0, 0, 0}, //Nombre sorti 1
                 {1, 0, 0, 0, 0, 0, 1}, //Nombre sorti 2
                 {1, 0, 0, 1, 0, 0, 1}, //Nombre sorti 3
                 {1, 0, 1, 0, 1, 0, 1}, //Nombre sorti 4
                 {1, 0, 1, 1, 1, 0, 1}, //Nombre sorti 5
                 {1, 1, 1, 0, 1, 1, 1}}; //Nombre sorti 6

int pin[] = {2, 3, 4, 5, 6, 7, 8};
int pinOffset = 2; //Première LED sur broche 2
int buttonPin = 13; //Bouton-poussoir sur broche 13

void setup(){
  for(int i = 0; i < 7; i++)
    pinMode(pin[i], OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop(){
  if (digitalRead (buttonPin) == HIGH)
    displayPips(random (1, 7)); //Générer un nombre entre 1 et 6
}

void displayPips(int value){
  for(int i = 0; i < 7; i++)
    digitalWrite(i + pinOffset,(pips[value - 1][i] == 1)?HIGH:LOW);
  delay(WAITTIME); //Ajouter une courte pause
}
```

## Le des électronique 2 avec capteur Lumière



Des électronique

Des\_electronique.ino

```

const int ECART=100;
const int TEMPO=2000;

void setup(){

  randomSeed(analogRead(1));

  for(int i=2;i<=8;i++){
    pinMode(i,OUTPUT);
  }
}

void loop(){

  int luminosite1=analogRead(0);
  delay(100);
  int luminosite2=analogRead(0);

  if(luminosite2-luminosite1>ECART){

    afficheDe(random(1,7));

    delay(TEMPO);
    eteintDe();
  }
}

```

```
void afficheDe(int nombre){  
  
    if(nombre==1){  
        digitalWrite(5,HIGH);  
    }  
  
    if(nombre==2){  
        digitalWrite(2,HIGH);  
        digitalWrite(8,HIGH);  
    }  
  
    if(nombre==3){  
        digitalWrite(2,HIGH);  
        digitalWrite(5,HIGH);  
        digitalWrite(8,HIGH);  
    }  
  
    if(nombre==4){  
        digitalWrite(2,HIGH);  
        digitalWrite(4,HIGH);  
        digitalWrite(6,HIGH);  
        digitalWrite(8,HIGH);  
    }  
    if(nombre==5){  
        digitalWrite(2,HIGH);  
        digitalWrite(4,HIGH);  
        digitalWrite(6,HIGH);  
        digitalWrite(8,HIGH);  
        digitalWrite(5,HIGH);  
    }  
  
    if(nombre==6){  
        digitalWrite(2,HIGH);  
        digitalWrite(3,HIGH);  
        digitalWrite(4,HIGH);  
        digitalWrite(6,HIGH);  
        digitalWrite(7,HIGH);  
        digitalWrite(8,HIGH);  
    }  
}  
  
void eteintDe(){  
    for(int i=2;i<=8;i++){  
        digitalWrite(i,LOW);  
    }  
}
```

## TP

TP et corrections cours Arduino

[Correction3](#)

[Correction\\_2 Corrections LCD I2C](#)[Créer des caractères sur un LCD](#)---page doc LCD

[Correction Corrections et Doc](#)---[Doc 74HC595](#)---[Autre Alternative MCP23017](#)

## Autres Cours

Autres Cours sur arduino

[Des programmes Arduino ...](#)

From:

<https://chanterie37.fr/fablab37110/> - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:

<https://chanterie37.fr/fablab37110/doku.php?id=start:arduino:debutant&rev=1724603901>

Last update: **2024/08/25 18:38**

