

```
sketch_feb08a

void setup() {
   // put your setup code here, to run once:
}

void loop() {
   // put your main code here, to run repeatedly:
```

Introduction à Arduino

http://www.arduino.cc

Décembre 2020

<u>F</u>ichier Édition Croqui<u>s</u> Ou<u>t</u>ils Aide

Une histoire d'Arduino ...



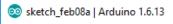
```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

https://youtu.be/D4D1WhA_mi8

https://arduinohistory.github.io

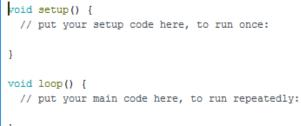




Historique sketch_feb08a







Design by Numbers

http://dbn.media.mit.edu

Date: 1999-2001

Lieu: MIT Media Lab

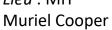
John Maeda



Visible Language Workshop

http://museum.mit.edu/150/115

Date: 1975 Lieu: MIT



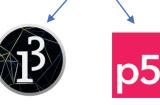




http://www.processing.org

Date: Printemps 2001

Lieu: MIT Media Lab Ben Fry / Casey Reas



Processing 3

p5.js





Date: 2003 Lieu: IDII

Hernando Barragán



http://www.arduino.cc

Date: 2005 Lieu: IDII

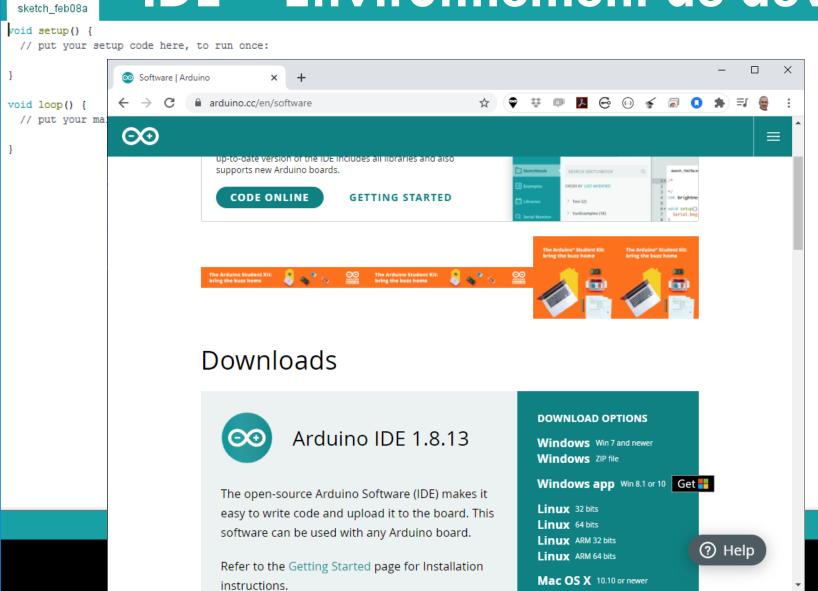
Massimo Banzi





IDE – Environnement de dév.



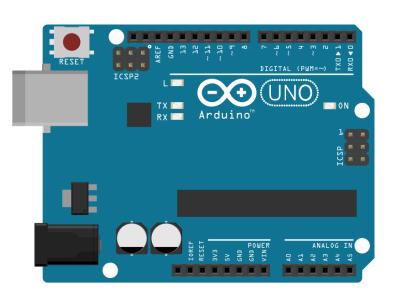


La carte générale ...

```
O
```

- Des entrées/sorties numériques
 - Des entrées analogiques (A)

•



Avantages sketch_feb08a

```
:
:
```

```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Les ((+))

- Prototypage rapide et simple d'objets physiques interactifs!
- Peu cher (suivant les cartes), logiciel et matériel open-source (et donc possibilité de clones!)
- Environnement de programmation simple

Avantages

```
sketch feb08a
```

```
void setup() {
  // put your setup code here, to run once:
  // put your main code here, to run repeatedly:
```

- Multiplateforme (Windows, MacOS, linux, RPi)
- Nombreuses librairies
- Des « shields » connectables pour augmenter les possibilités (ethernet, GPS, afficheur graphique, ...)

Qu'est ce qu'Arduino?





```
void setup() {
  // put your setup code here, to run once:
void loop() {
```

De multiples versions disponibles





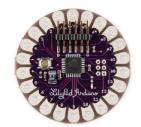








... plein d'autres!



Qu'est ce qu'Arduino ?

```
<u>0</u>
```

```
sketch_feb08a

void setup() {
    // put your setup code here, to run once:
}

void loop() {
    // put your main code here, to run repeated!
    Arduino est (( )
}
```

Arduino est « un langage commun » indépendant des langages bas-niveau permettant de prototyper rapidement des applications physiques.

La base du programme Arduino est le « *sketch* » (programme, prototype)
L'extension est le « .ino »

<u>Fichier Édition Croquis</u> Ou<u>t</u>ils Aide

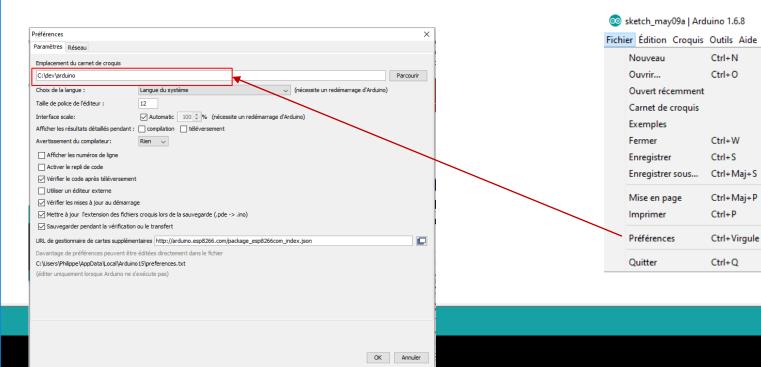
Structure Structure

```
₽
P
```

```
void setup() {
   // put your setup code here, to run once:
}

void loop() {
   // put your main code here, to run repeated!
```

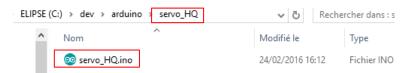
• Les « sketchs » (programmes) sont localisés dans le répertoire « préférences »



```
sketch_feb08a
void setup() {
  // put your setup code here, to run once:
void loop() {
 // put your main code here, to run repeatedly:
```

un sketch est composé de :

 Au moins un fichier (.ino)> (cela peut être plus — un par classe objet). Le fichier principal doit avoir le même nom que le répertoire du sketch



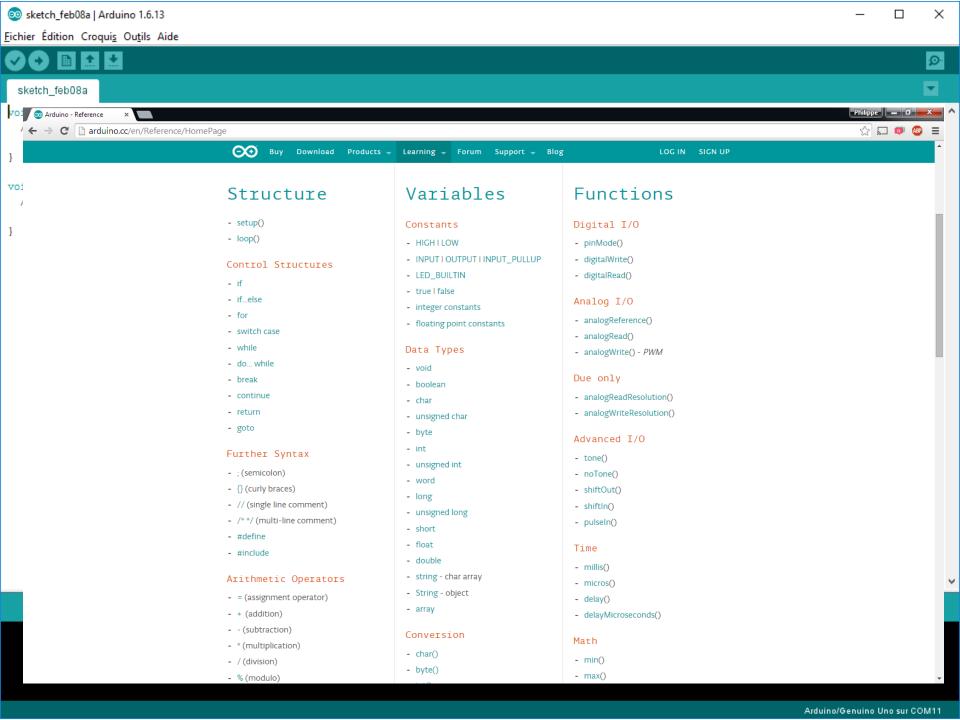
Deux fonctions basiques sketch feb08a

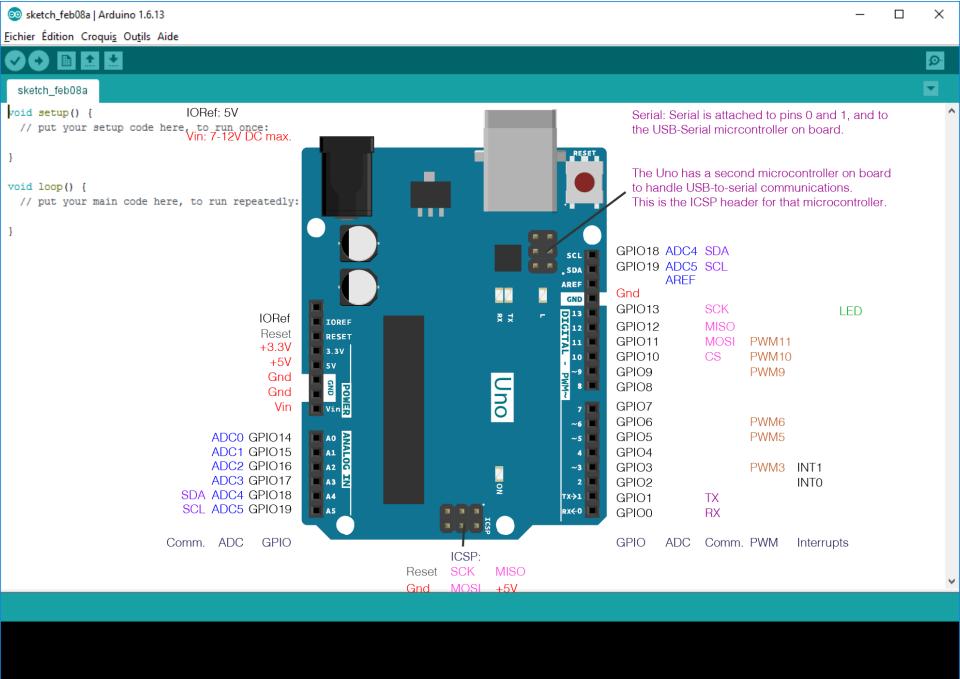
```
void setup() {
  // put your setup code here, to run once:
void loop() {
```

• setup : exécuté une seule fois au démarrage – permet d'initialiser les variables du programme

```
void setup() {
  Serial.begin(9600);
  Serial.println("16 channel Servo test!");
  pwm.begin();
  pwm.setPWMFreg(60); // Analog servos run at ~60 Hz updates
  vield();
```

• loop: c'est la boucle de traitement des capteurs exécutée « à l'infini » (mainloop)







void setup() {

void loop() {

Un premier exemple sketch feb08a

```
Blink | Arduino 1.6.7
                                                                                                                   X
// put your setup code here,
                                 Fichier Édition Croquis Outils Aide
                                   Blink
// put your main code here, t
                                   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.
                                 // Pin 11 has the LED on Teensv 2.0
                                 // Pin 6 has the LED on Teensy++ 2.0
                                 // Pin 13 has the LED on Teensy 3.0
                                 // 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
```

Arduino/Genuino Uno on COM15

Notes Commentaires

« A ne pas oublier »

```
:
:
```

```
void setup() {
   // put your setup code here, to run once:
}
void loop() {
   // put your main code here, to run repeatedly:
```

 Outils | Type de carte >> type de la carte utilisée

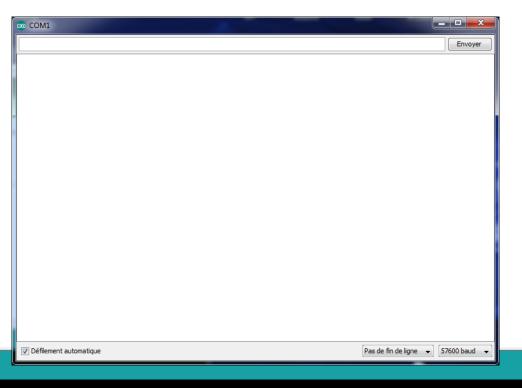
• Outils | Port >> port série utilisé par la carte

```
õ
```

```
void setup() {
   // put your setup code here, to run once:
}

void loop() {
   // put your main code here, to run repeatedly:
```

• Outils | Moniteur série



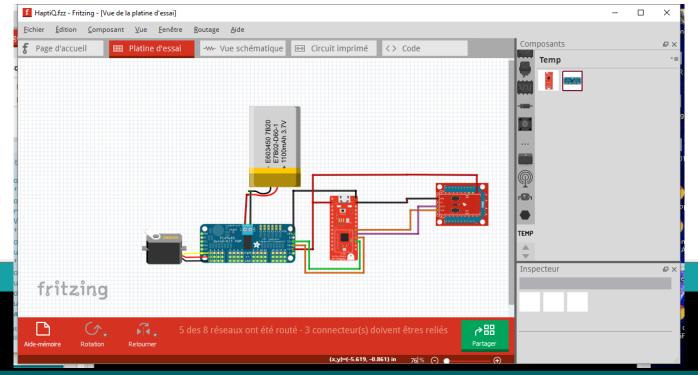
// put your setup code here, to run once:

void setup() {

Un outil d'aide au montage

```
• Fritzing - http://fritzing.org (payant depuis 2019)

http://fritzing.org/download/0.9.3b/windows-
64bit/fritzing.0.9.3b.64.pc.zip
```

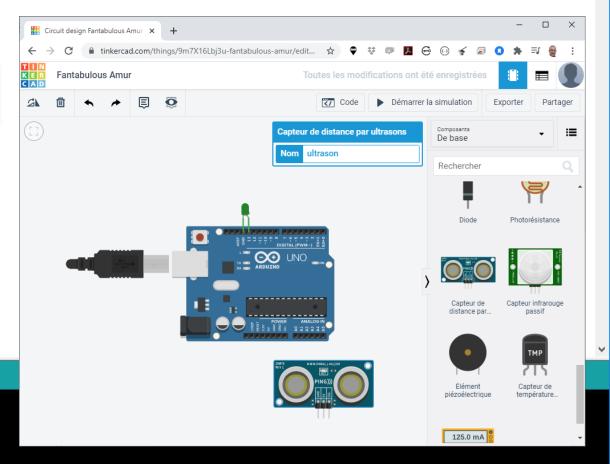


Un simulateur en ligne :

```
sketch_feb08a
void setup() {
  // put your setup cod/ .er , % r . on e:
```

```
void loop() {
* https://www.tinkercad.com
```

Choisir Circuits



Cliquez pour la liste non-officielle des URL des cartes supportées

Annuler

NodeMCU V3 Pinout

LoLin

www.TheEngineeringProjects.com

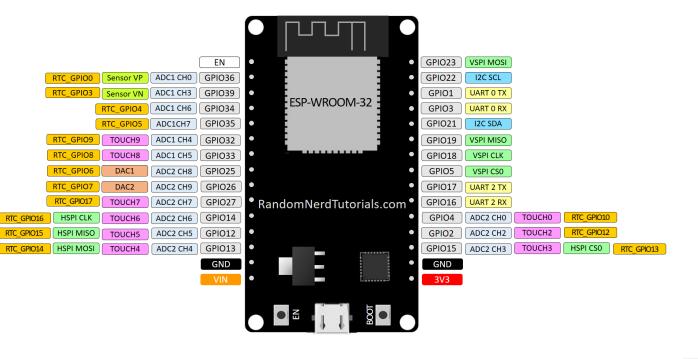
ESP32

```
Ð
```

```
void setup() {
   // put your setur
}
void loop() {
   // put your main
```

ESP32 DEVKIT V1 - DOIT

version with 30 GPIOs



integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

- 10 capteurs capacitifs
- 18 entrées analogiques
- 16 sorties PWM
- Interfaces I2C, SPI
- ...

Exercices de démarrage sketch feb08a

```
void setup() {
  // put your setup code here, to run once:
void loop() {
  // put your main code here, to run repeatedly:
```



- Allumer/Eteindre une led Ouvrir Fichier | Exemples | 01.Basics | Blink **LED BUILTIN** → GPIO 02 sur ESP32
- → Modifier la durée du clignotement

Exercices de démarrage

```
Ω
•
```

```
void setup() {
   // put your setup code here, to run once:
}

void loop() {
   // put your main code here, to run repeatedly:
}
```

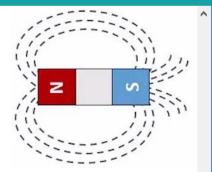
→ Modifier le programme et envoyer l'état de la LED sur la liaison série

Serial.begin(rapidite_modulation)
Serial.println()

→ Modifier le programme pour piloter l'état de la LED depuis le PC

Effet Hall

```
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
```



- hallRead()
 - Un capteur à effet Hall
- → Écrire un programme qui allume la LED interne quand on approche un aimant (déterminer le seuil de déclenchement)

Capteur de toucher

```
Ø
```

```
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```



- analogTouch()
 - Des capteurs de toucher (TO à T9)

→ Écrire un programme qui allume la LED interne quand on touche le capteur TO

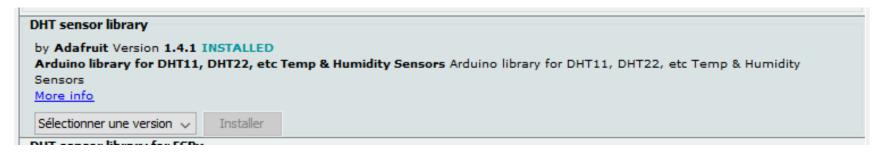
```
<u>ه</u>
```

```
void setup() {
   // put your setup code here, to run once:
}

void loop() {
   // put your main code here, to run repeatedly:
}
```



• **DHT** – capteur de T° et d'humidité



→ Ecrire un programme qui renvoie la température et l'humidité quand on touche le capteur T0

```
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
```





• Adafruit Neopixel – des pixels RGB



→ Modifier le code de telle manière que l'on puisse changer de couleur (aléatoire) quand on utilise un capteur de toucher

Lecteur NFC



```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
```



• RFID-522 — Un lecteur NFC

MFRC522

by GithubCommunity Version 1.4.7 INSTALLED

Arduino RFID Library for MFRC522 (SPI) Read/Write a RFID Card or Tag using the ISO/IEC 14443A/MIFARE interface.

More info

Sélectionner une version
Installer

→ Modifier le code fourni qui permet d'allumer/éteindre une LED quand on présente une carte NFC spécifique **SDA GPIO21 SCK GPIO18** MOSI **GPIO23 GPIO19** MISO **NOT USED** IRO **GND** GND **RST** GPIO22 3v3 3v3

Borne wifi

```
void setup() {
  // put your setup code here, to run once:
 // put your main code here, to run repeatedly:
```



Une borne et/ou un client wifi

L'ESP32 peut servir d'AP (Access Point) et/ou être client d'un routeur Wifi

On peut y implémenter un (mini)-serveur web par exemple

```
sketch_feb08a

void setup() {
   // put your setup code here, to run once:
}

void loop() {
   // put your main code here, to run repeatedly:
}
```

• De nombreuses autres choses sont possibles (utilisation du BT, d'autres types de capteurs/effecteurs, usage d'API REST, MQTT, ...)

• « Le monde des possibles » est quasiment infini!

<u>Fichier Édition Croquis</u> Ou<u>t</u>ils Aide

```
٥
```

```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Nous souhaitons gérer un feu tricolore (alternance vert 6 s, orange 1 s, rouge 3 s).

- Un accès par liaison série est possible ce qui permet de récupérer l'état courant du feu tricolore.
- Nous voulons gérer le feu piéton associé

 modifier le code
- Un appui sur un bouton d'appel (simulé par la liaison série) doit permettre de passer à l'orange puis au rouge

```
void setup() {
  // put your setup code here, to run once:
  // put your main code here, to run repeatedly:
```

Nous souhaitons créer une station météo : T° et taux d'humidité avec un affichage (avec de belles visualisations) sur une application graphique sur PC

- https://docs.espressif.com/projects/espidf/en/latest/esp32/apireference/peripherals/timer.html
- https://projetsdiy.fr/esp32-timers-alarmes-codearduino/
- https://maker.pro/arduino/projects/timer-interrupts-improve-your-arduino-programming-skills