

# Introduction à Arduino

<https://www.arduino.cc>

Octobre 2022



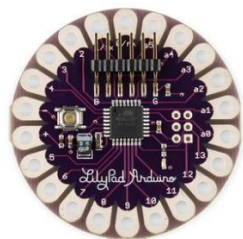
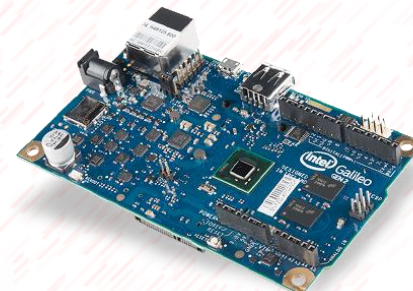
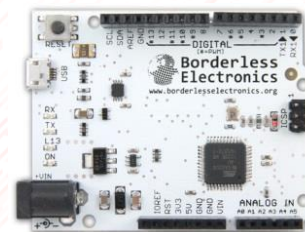
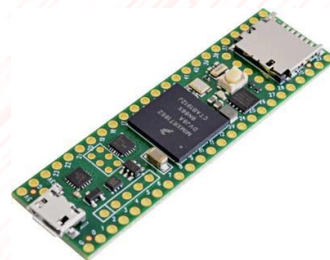
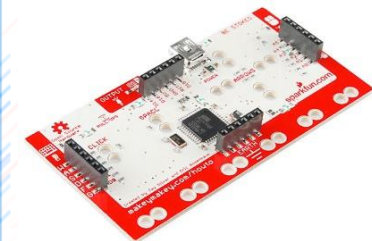
# Qu'est ce qu'Arduino ?

sketch\_feb08a

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

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

D'abord du matériel !



avec plein de versions différentes !



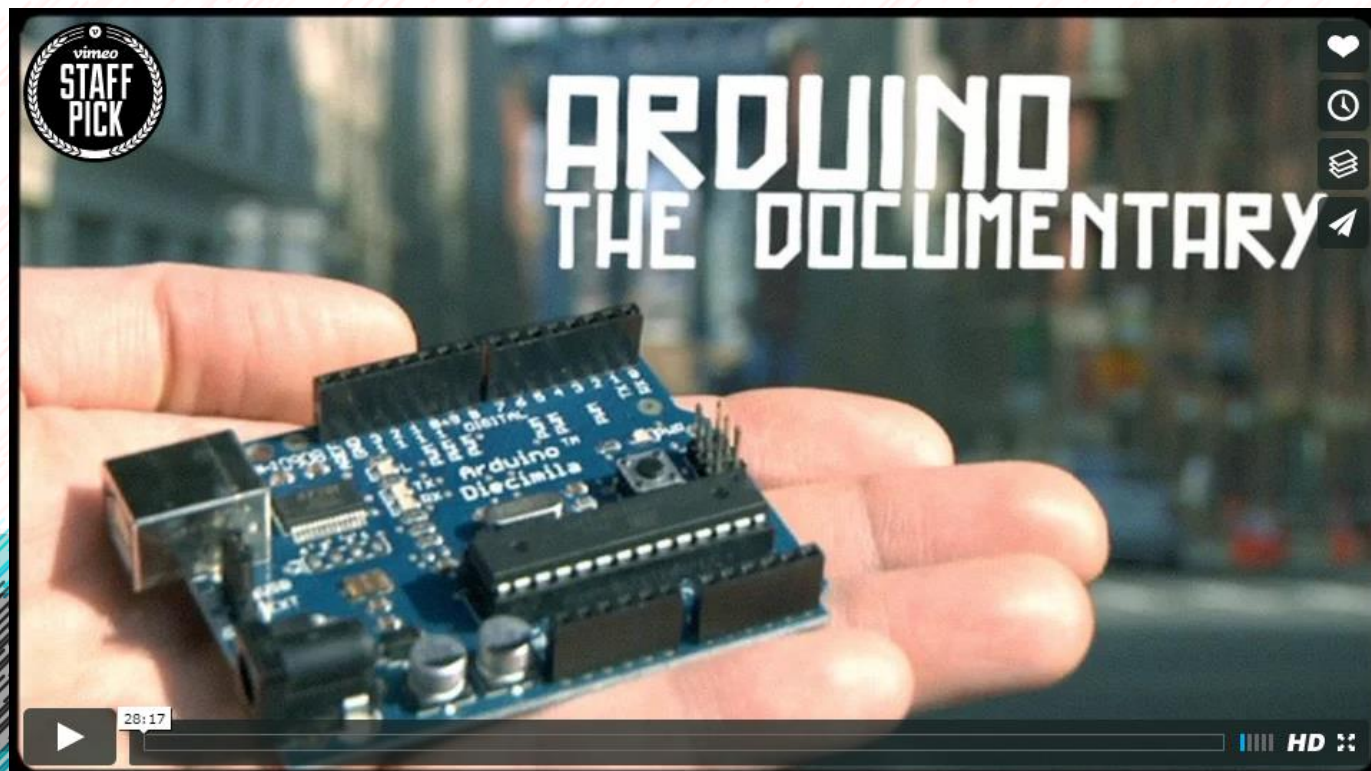
# Une histoire d'Arduino ...

sketch\_feb08a

```
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://youtu.be/D4D1WhA_mi8)

<https://arduinohistory.github.io>





# Historique

sketch\_feb08a

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

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

## Design by Numbers

<http://dbn.media.mit.edu>

Date : 1999-2001

Lieu : MIT Media Lab

John Maeda



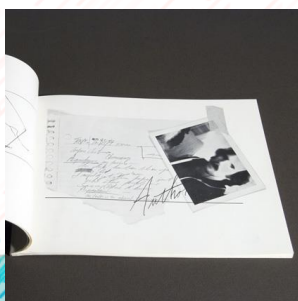
## Visible Language Workshop

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

Date : 1975

Lieu : MIT

Muriel Cooper



## Processing

<https://www.processing.org>

Date : Printemps 2001

Lieu : MIT Media Lab

Ben Fry / Casey Reas



Processing 4



p5.js

## Wiring

<http://wiring.org.co>

Date : 2003

Lieu : IDII

Hernando Barragán



## Arduino

<https://www.arduino.cc>

Date : 2005

Lieu : IDII

Massimo Banzi



# IDE – Environnement de dév.

sketch\_feb08a

```
void setup() {  
  // put your set
```

```
void loop() {  
  // put your main
```

}

Software | Arduino

arduino.cc/en/software

PROFESSIONAL EDUCATION STORE

Search on Arduino.cc

SIGN IN

HARDWARE **SOFTWARE** CLOUD DOCUMENTATION COMMUNITY BLOG ABOUT

up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

**CODE ONLINE** **GETTING STARTED**

## Downloads

### Arduino IDE 2.0.0

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

**SOURCE CODE**

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

#### DOWNLOAD OPTIONS

**Windows** Win 10 and newer, 64 bits  
**Windows** MSI installer  
**Windows** ZIP file

**Linux** ApplImage 64 bits (X86-64)  
**Linux** ZIP file 64 bits (X86-64)

**macOS** 10.14: "Mojave" or newer, 64 bits

**Help**



# 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:  
}  
  
void loop() {  
  // 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, ...)



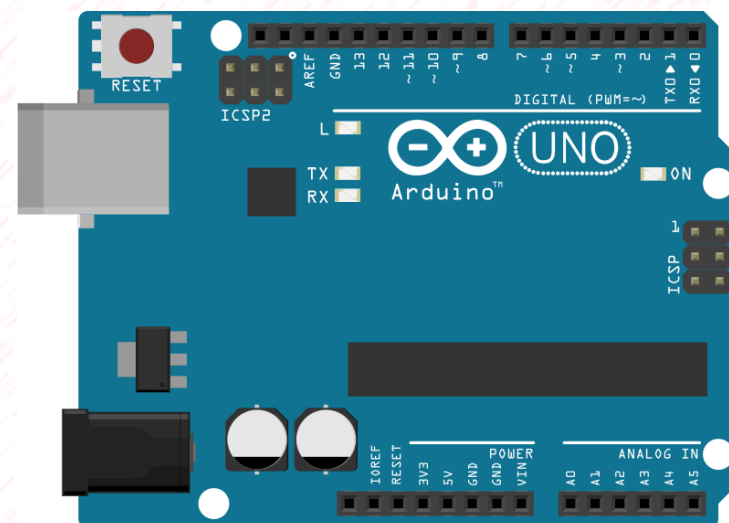
# La carte arduino UNO ...

sketch\_feb08a

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

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

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





# arduino UNO pinout

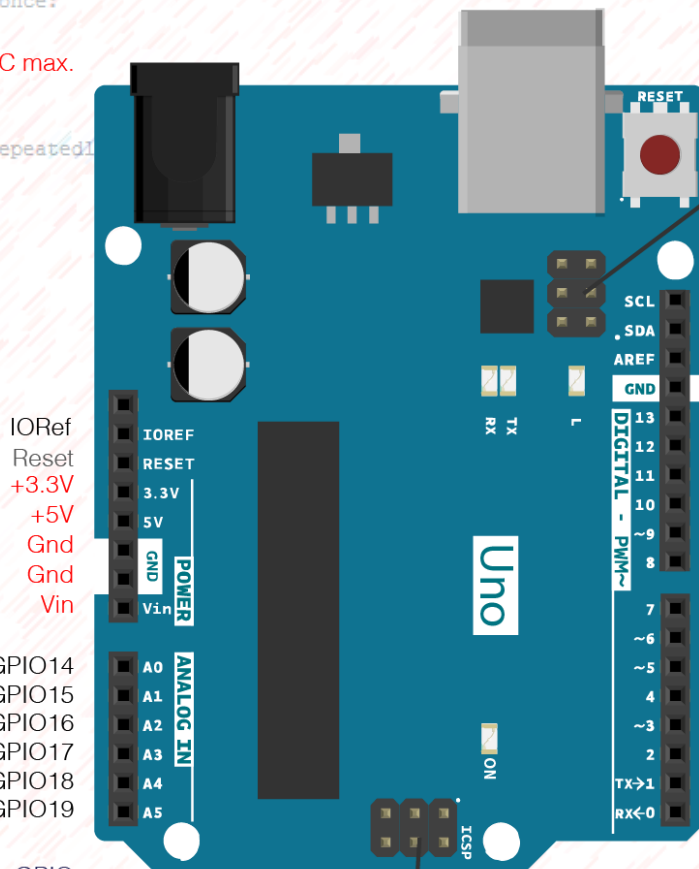
sketch\_feb08a

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

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

I<sub>OREf</sub>: 5V

V<sub>in</sub>: 7-12V DC max.



Serial: Serial is attached to pins 0 and 1, and to the USB-Serial microcontroller on board.

The Uno has a second microcontroller on board to handle USB-to-serial communications. This is the ICSP header for that microcontroller.

ADC0 GPIO14  
ADC1 GPIO15  
ADC2 GPIO16  
ADC3 GPIO17  
SDA ADC4 GPIO18  
SCL ADC5 GPIO19

Comm. ADC GPIO

GPIO18 ADC4 SDA  
GPIO19 ADC5 SCL  
AREF

Gnd  
GPIO13 SCK LED  
GPIO12 MISO  
GPIO11 MOSI PWM11  
GPIO10 CS PWM10  
GPIO9 PWM9  
GPIO8  
GPIO7 PWM6  
GPIO6 PWM5  
GPIO4 PWM3 INT1  
GPIO3 INT0  
GPIO2 TX  
GPIO1 RX  
GPIO0

GPIO ADC Comm. PWM Interrupts

ICSP:  
Reset RST  
Gnd  
MISO  
MOSI  
+5V



# Programmation arduino

sketch\_feb08a

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

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

Arduino est « ***un langage commun*** » (syntaxe C++)  
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** »

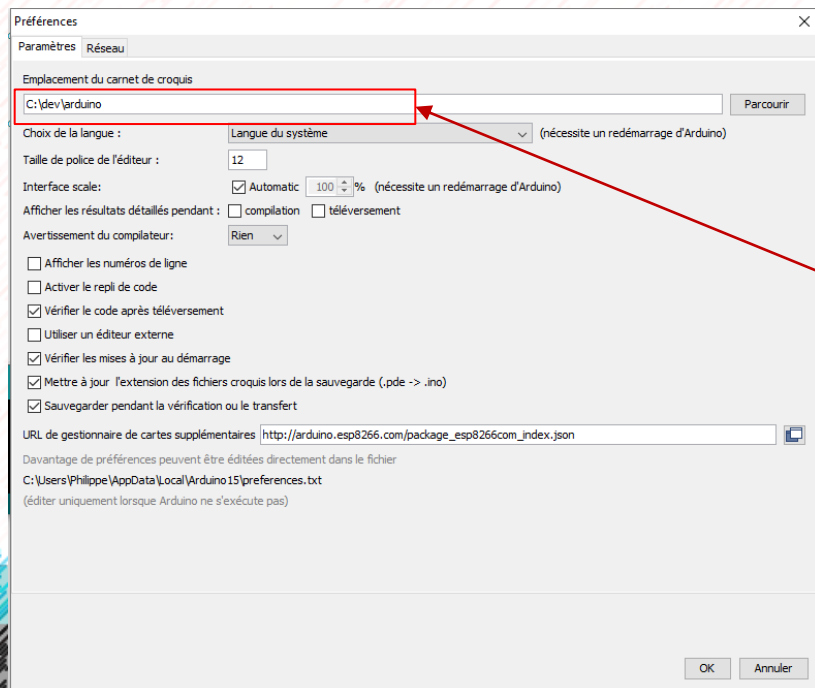


# Structure

sketch\_feb08a

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

- Les « sketches » (programmes) sont localisés dans le répertoire « préférences »



sketch\_may09a | Arduino 1.6.8

Fichier Édition Croquis Outils Aide

Nouveau	Ctrl+N
Ouvrir...	Ctrl+O
Ouvert récemment	>
Carnet de croquis	>
Exemples	>
Fermer	Ctrl+W
Enregistrer	Ctrl+S
Enregistrer sous...	Ctrl+Maj+S
Mise en page	Ctrl+Maj+P
Imprimer	Ctrl+P
Préférences	Ctrl+Virgule
Quitter	Ctrl+Q



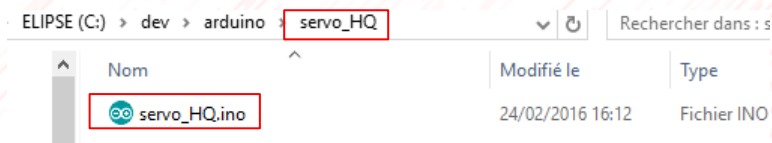
# Structure

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

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

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

- **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.setPWMFreq(60); // Analog servos run at ~60 Hz updates  
  yield();  
}
```

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



sketch\_feb08a

Arduino - Reference

← → ↺ arduino.cc/en/Reference/HomePage

Philippe



Buy

Download

Products ▾

Learning ▾

Forum

Support ▾

Blog

LOG IN

SIGN UP

## Structure

- setup()
- loop()

### Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

### Further Syntax

- ; (semicolon)
- {} (curly braces)
- // (single line comment)
- /\* \*/ (multi-line comment)
- #define
- #include

### Arithmetic Operators

- = (assignment operator)
- + (addition)
- - (subtraction)
- \* (multiplication)
- / (division)
- % (modulo)

## Variables

### Constants

- HIGH | LOW
- INPUT | OUTPUT | INPUT\_PULLUP
- LED\_BUILTIN
- true | false
- integer constants
- floating point constants

### Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int
- word
- long
- unsigned long
- short
- float
- double
- string - char array
- String - object
- array

### Conversion

- char()
- byte()

## Functions

### Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

### Analog I/O

- analogReference()
- analogRead()
- analogWrite() - PWM

### Due only

- analogReadResolution()
- analogWriteResolution()

### Advanced I/O

- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

### Time

- millis()
- micros()
- delay()
- delayMicroseconds()

### Math

- min()
- max()



# Un premier exemple

```
void setup() {  
  // put your setup code here,
```

```
void loop() {  
  // put your main code here, t
```

Blink | Arduino 1.6.7

Fichier Édition Croquis Outils Aide

Blink

```
/*  
  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.  
// Pin 11 has the LED on Teensy 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  
}
```



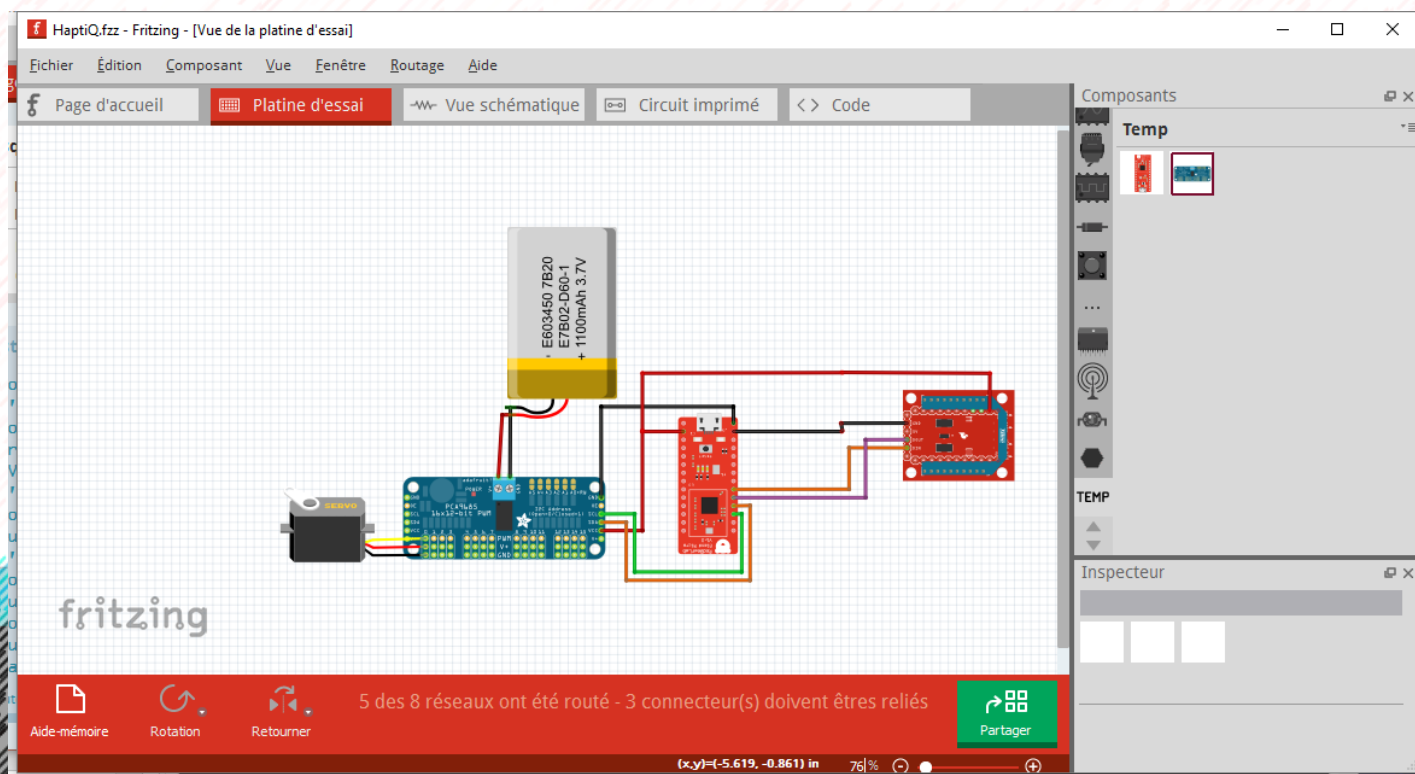
# Un outil d'aide au montage

sketch\_feb08a

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

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

- **Fritzing** - <https://fritzing.org> (payant depuis 2019)  
<https://www.softpedia.com/get/Science-CAD/Fritzing.shtml#download>





# Un simulateur en ligne

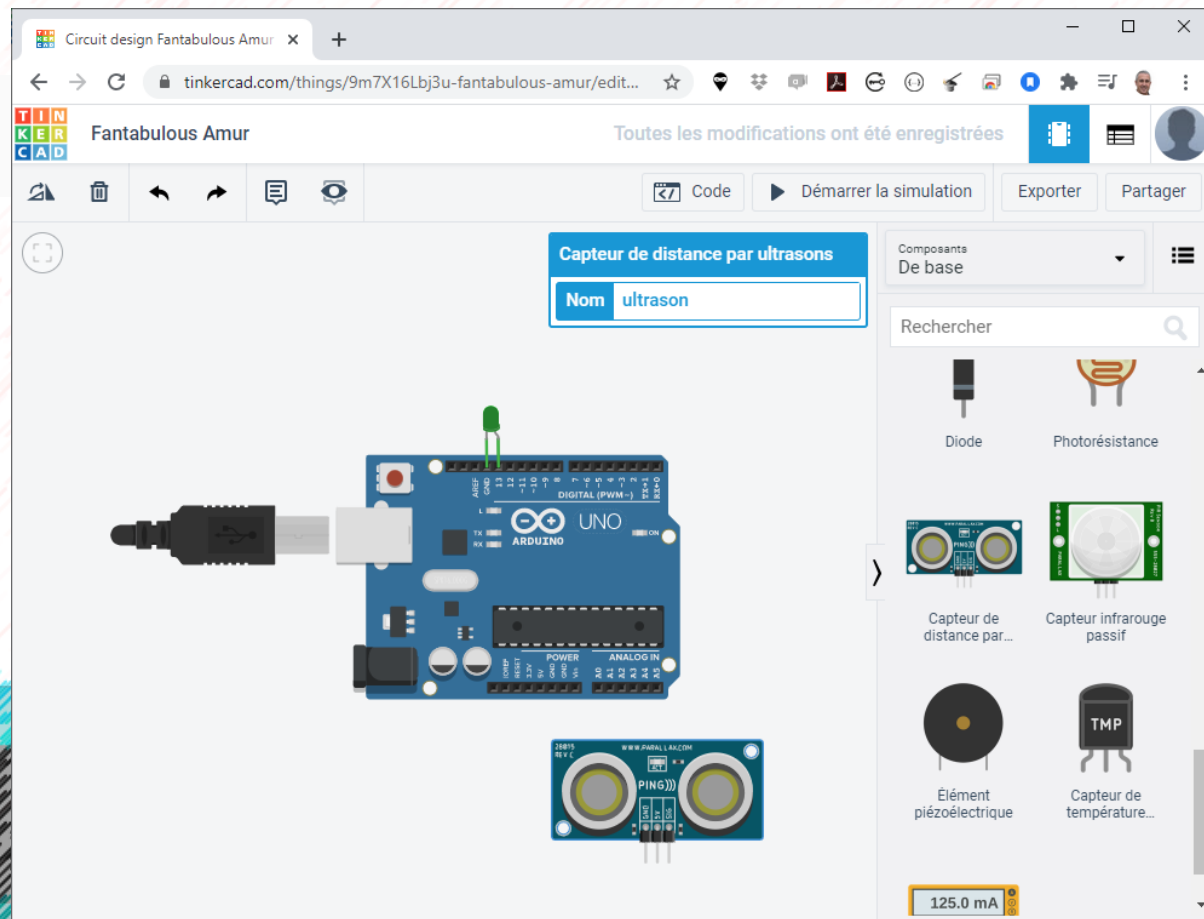
sketch\_feb08a

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

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

- <https://www.tinkercad.com/dashboard>

- Choisir **Circuits**



# Exercices de chauffe

sketch\_feb08a

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

- Des leds
- Un capteur de distance

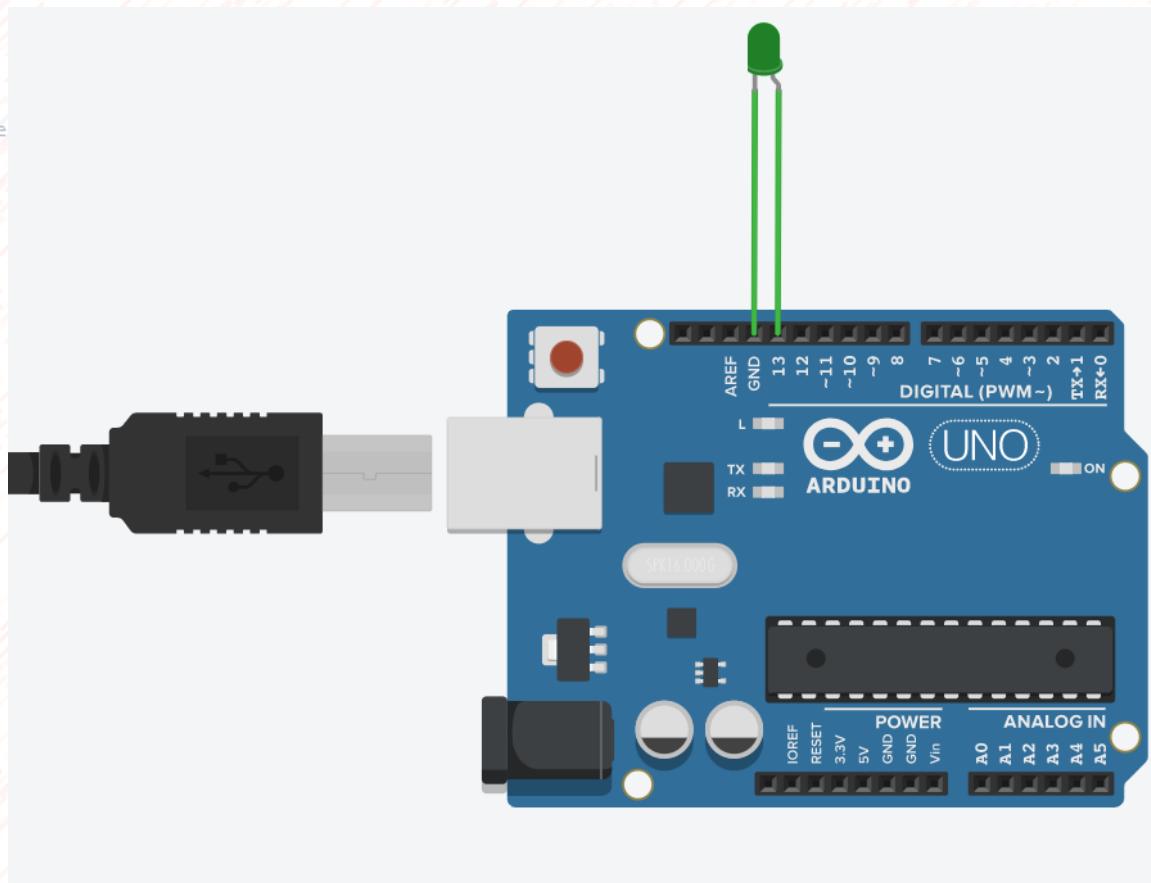


# Led

sketch\_feb08a

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

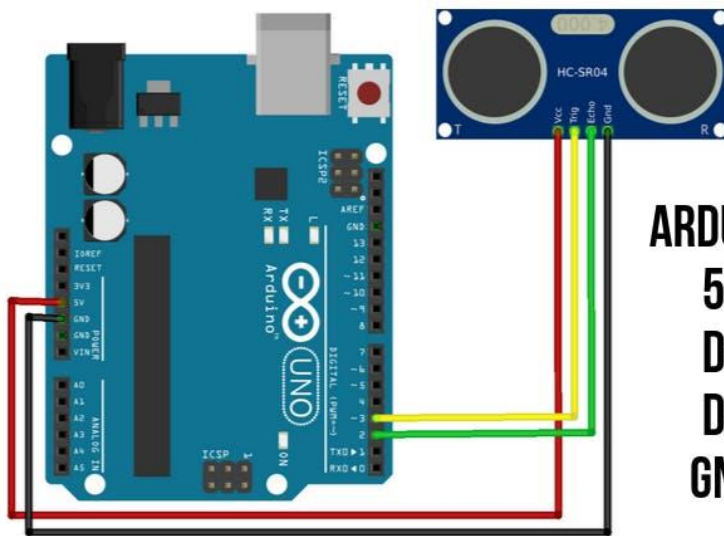
```
void loop() {  
  // put your main code here  
}
```



# Capteur de distance

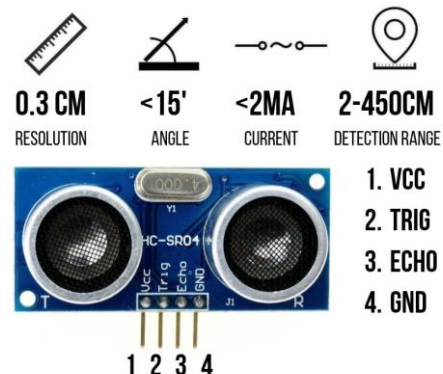
sketch\_feb08a

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



ARDUINO >> HC-SR04

5V	—	VCC
D2	—	ECHO
D3	—	TRIG
GND	—	GND





# Liaison série

sketch\_feb08a

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

- interruption

```
void serialEvent() { // instructions }
```

**/!** Ne fonctionne pas pour tous les arduino



# Machine à états

sketch\_feb08a

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  typedef enum {ETEINT=LOW, ALLUME=HIGH} MAE; //  
  l'énumération est définie sous le type MAE  
  
  MAE mae;  
  
  ...  
  
  void loop() {  
    switch(mae) {  
      case ALLUME: ...
```



# ESP32

sketch\_feb08a

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

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

- ESP32 est une série de micro-contrôleurs intégrant la gestion du wifi et du bluetooth (jusqu'à BLE)
- Il est peu cher et très apprécié dans le domaine de l'IoT !

# Installer ESP8266 et ESP32

sketch\_feb08a

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

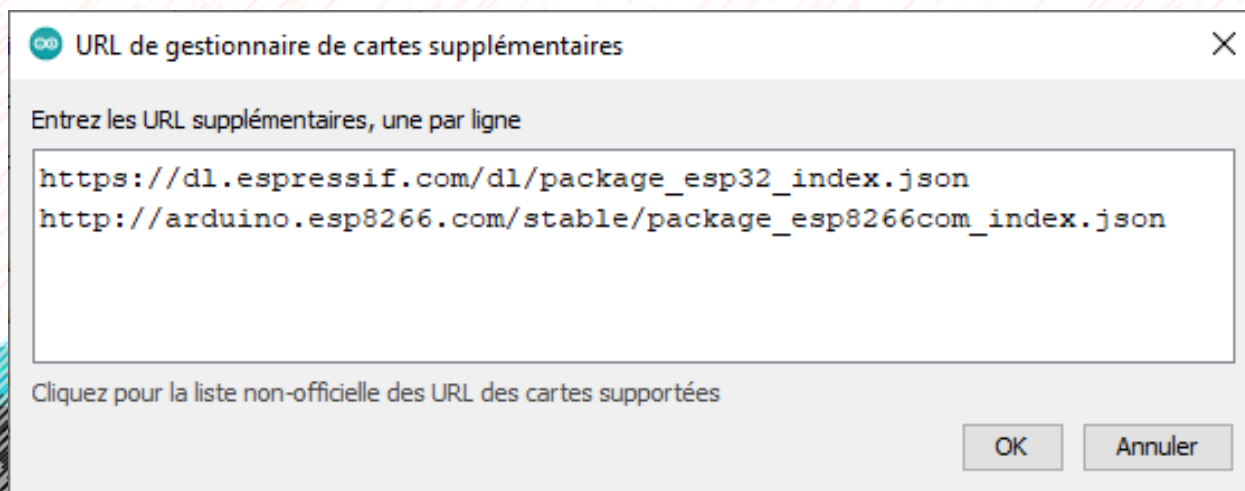
```
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

- Dans Fichier | Préférences

URL de gestionnaire de cartes supplémentaires

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



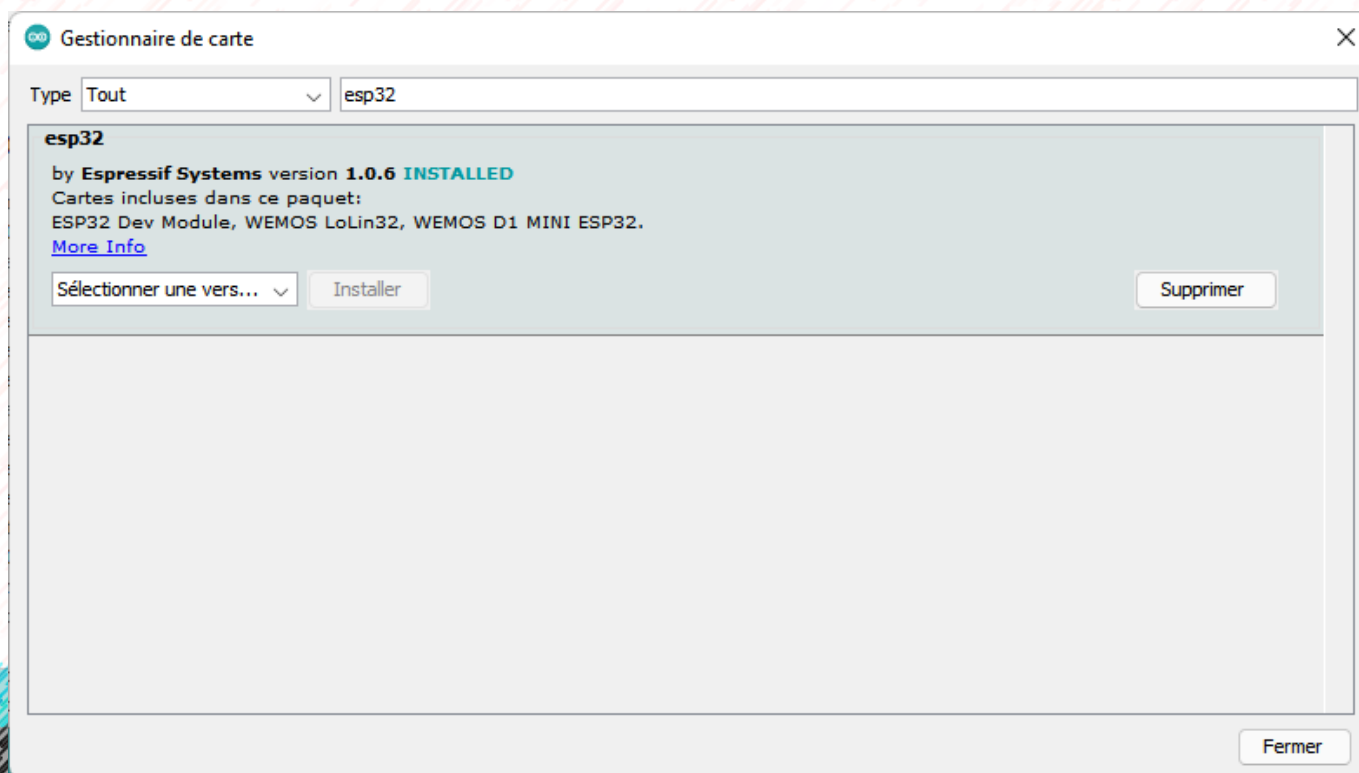


# Installer ESP8266 et ESP32

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

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

- Dans Outils | Type de carte | Gestionnaire de carte

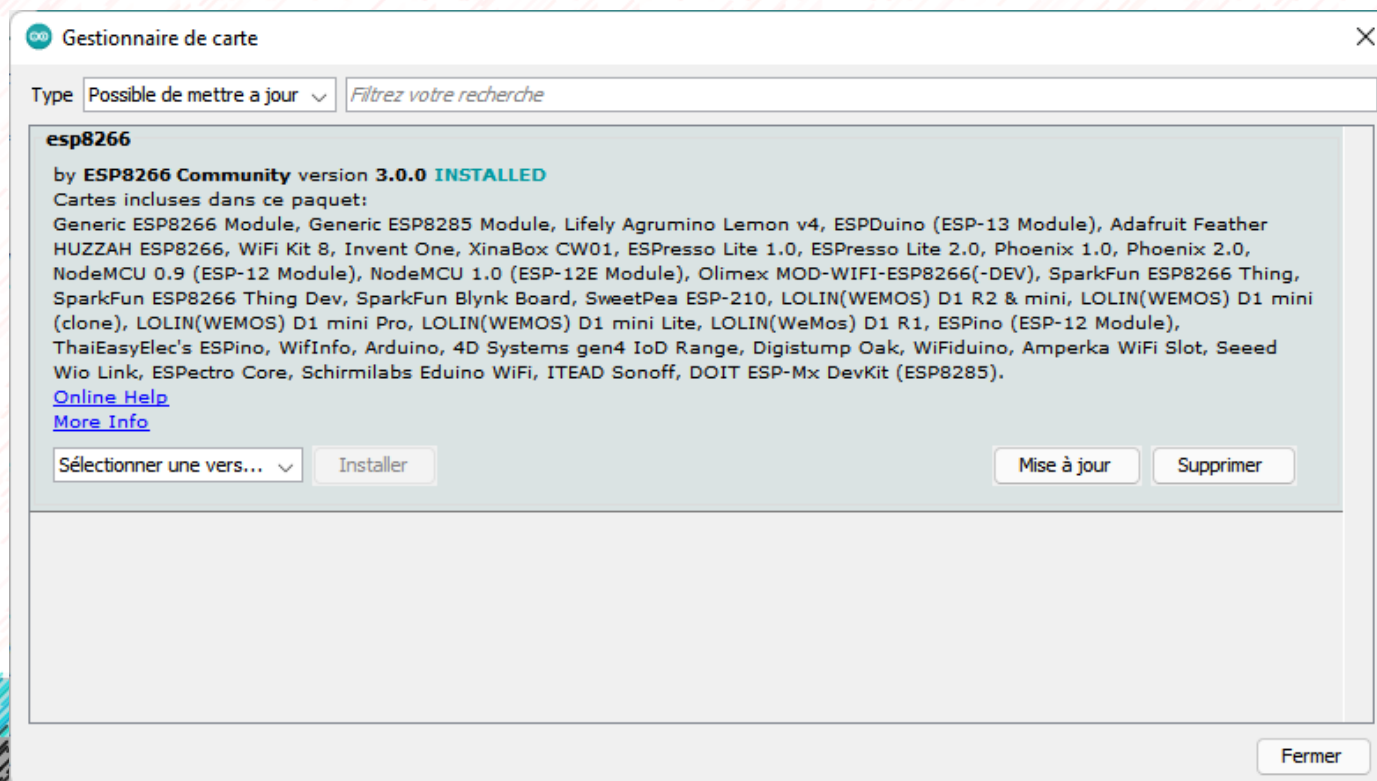


# Installer ESP8266 et ESP32

sketch\_feb08a

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

- Dans Outils | Type de carte | Gestionnaire de carte





# Installer ESP8266 et ESP32

sketch\_feb08a

```
void setup() {
  // put your setup code here
```

```
void loop() {
  // put your main code here
```

Blink

/\*  
Blink

Turns an LED on

Most Arduinos have

it is attached to

the correct LED

If you want to know

model, check the

<https://www.arduino.cc>

modified 8 May 2015

by Scott Fitzgerald

modified 2 Sep 2015

by Arturo Guadalquivir

modified 8 Sep 2015

by Colby Newman

This example code is in the public domain.

Téléversement terminé

Leaving...

Hard resetting via RTS pin...

1

DOIT ESP32 DEVKIT V1, 80MHz, 921600, No

Formatage automatique

Ctrl+T

Archiver le croquis

Réparer encodage &amp; recharger

Gérer les bibliothèques

Ctrl+Maj+I

Moniteur série

Ctrl+Maj+M

Traceur série

Ctrl+Maj+L

WiFi101 / WiFININA Firmware Updater

Type de carte: "DOIT ESP32 DEVKIT V1"

Upload Speed: "921600"

Flash Frequency: "80MHz"

Core Debug Level: "Rien"

Port: "COM6"

Récupérer les informations de la carte

Programmeur

Graver la séquence d'initialisation

Gestionnaire de carte

Arduino AVR Boards &gt;

Arduino i586 Boards &gt;

ESP32 Arduino

ESP8266 Boards (3.0.2) &gt;

NodeMCU-32S

MH ET LIVE ESP32DevKIT

MH ET LIVE ESP32MiniKit

ESP32vn IoT Uno

• DOIT ESP32 DEVKIT V1

DOIT ESPduino32

OLIMEX ESP32-EVB

OLIMEX ESP32-GATEWAY

OLIMEX ESP32-PoE

OLIMEX ESP32-PoE-ISO

OLIMEX ESP32-DevKit-LiPo

ThaiEasyElec's ESPino32

M5Stack-Core-ESP32

M5Stack-FIRE

M5Stick-C

M5Stack-ATOM

M5Stack-Core2

M5Stack-Timer-CAM

M5Stack-CoreInk

ODROID ESP32

Heltec WiFi Kit 32

Heltec WiFi LoRa 32

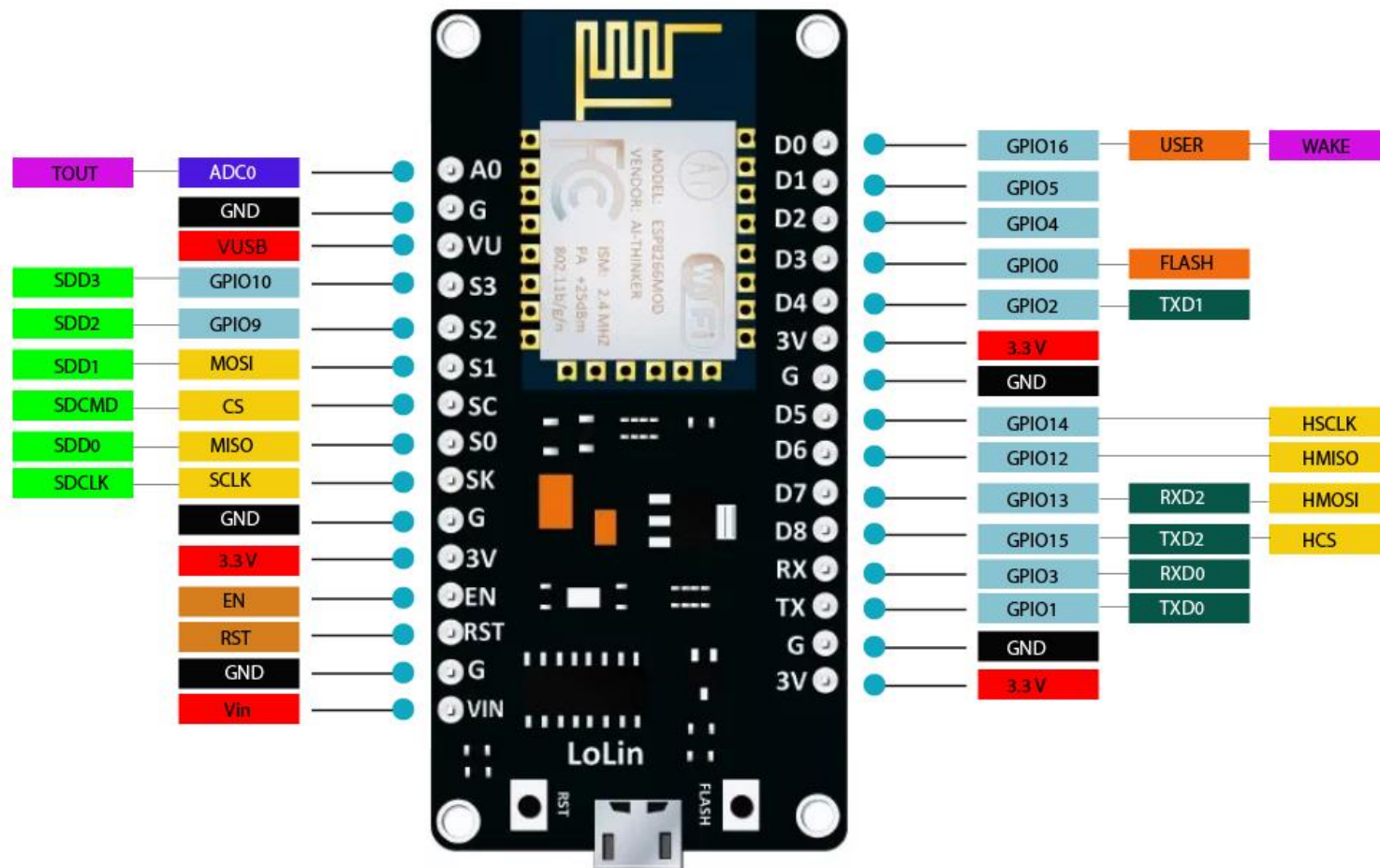
Heltec WiFi LoRa 32(V2)

Heltec Wireless Stick

Heltec Wireless Stick Lite

# ESP8266 pinout

sketch\_feb08a



NodeMCU V3 Pinout

[www.TheEngineeringProjects.com](http://www.TheEngineeringProjects.com)

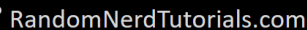
D0 GPIO16  
D1 GPIO05  
D2 GPIO04  
D3 GPIO00  
D4 GPIO02  
D5 GPIO14  
D6 GPIO12  
D7 GPIO13  
D8 GPIO15  
D9 GPIO03  
D10 GPIO01



sketch\_feb08a

# ESP32 DEVKIT V1 – DOIT

version with 30 GPIOs



# ESP32 pinout

sketch\_feb08a

```
void set
// put

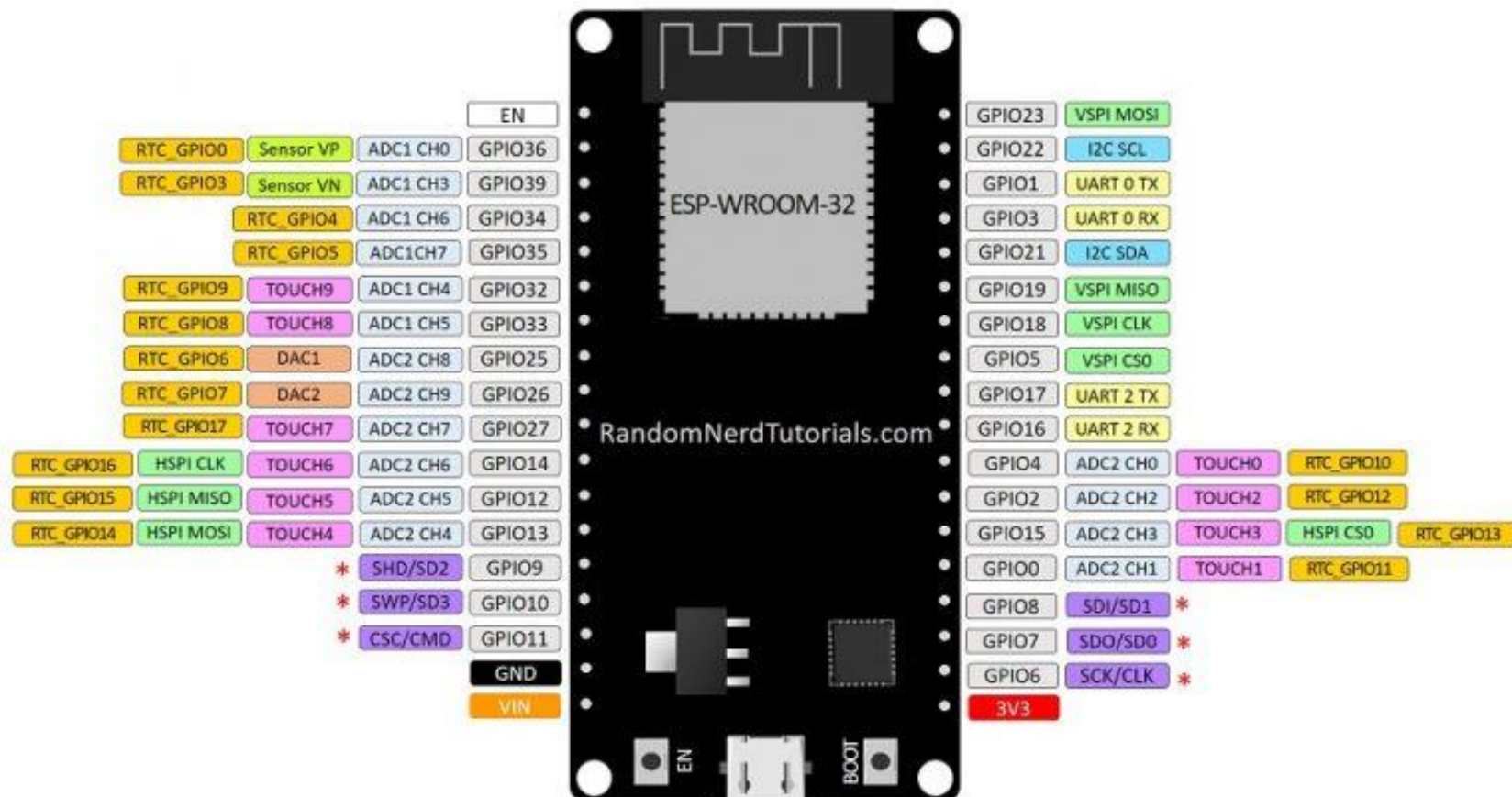
}

void loop
// put

}
```

## ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs



\* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and CSC/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.



# ESP32

sketch\_feb08a

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

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

- 3 interfaces UART (liaisons série)
- 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

sketch\_feb08a

```
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(rapidity_modulation)  
Serial.println()
```

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



# 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:  
}
```



- Utiliser une librairie externe → capteur ultrason HC-SR04
  - [https://bitbucket.org/teckel12/arduino-new-ping/downloads/NewPing\\_v1.9.4.zip](https://bitbucket.org/teckel12/arduino-new-ping/downloads/NewPing_v1.9.4.zip) (ou via la bibliothèque)

Please Notice This



Pour être utilisable sur ESP32, il faut le modèle HC-SR04P ou modifier le capteur (<https://www.instructables.com/Modify-Ultrasonic-Sensors-for-3-Volts-Logic-prepar/>)





# 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:  
}
```



→ Écrire un programme qui envoie sur le port série la distance perçue par l'arduino avec le plus proche objet et allume la led **LED\_BUILTIN** si la distance est inférieure à 20 cm



# Exercices de démarrage

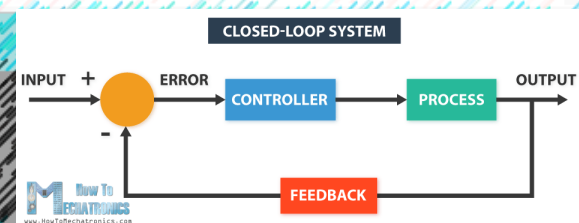
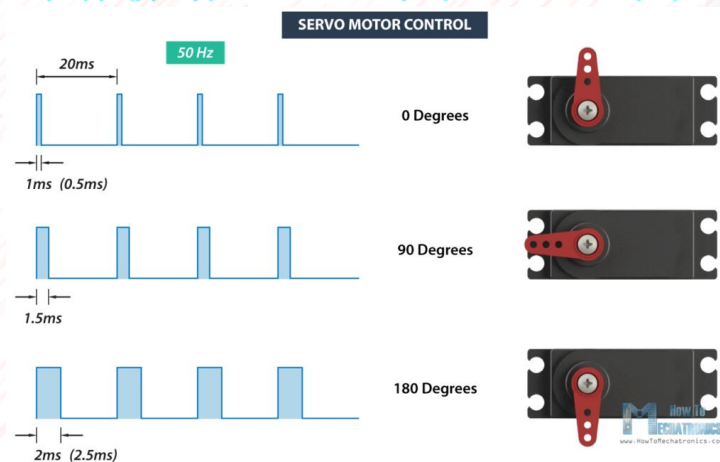
sketch\_feb08a

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

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

- Servomoteur et PWM  
(**P**ulse **W**idth **M**odulation)

→ Écrire un programme qui permet de tourner un servomoteur entre 0 et 180° suivant la distance perçue entre 10 et 60 cm (15 cm = 0° et 60 cm = 180°)

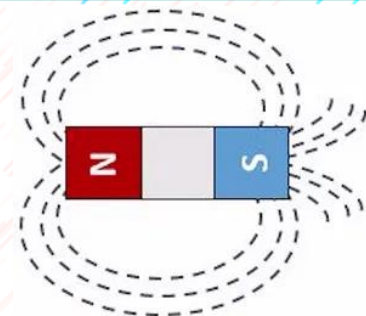




# Effet Hall

sketch\_feb08a

```
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

sketch\_feb08a

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



- **touchRead()**
  - Des capteurs de toucher (T0 à T9)

→ Écrire un programme qui allume la LED interne quand on touche le capteur T0



# Capteur T° et Humidité

sketch\_feb08a

```
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

## DHT sensor library

by **Adafruit** Version **1.4.1** **INSTALLED**

Arduino library for DHT11, DHT22, etc Temp &amp; Humidity Sensors

[More info](#)

Sélectionner une version ▾

Installer

DHT sensor library for ESP8266

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



# Neopixel

sketch\_feb08a

```
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

## Adafruit NeoPixel

by Adafruit Version 1.7.0 **INSTALLED**

Arduino library for controlling single-wire-based LED pixels and strip. Arduino library for controlling single-wire-based LED pixels and strip.

[More info](#)

Sélectionner une version ▾

Installer

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



# Lecteur NFC

sketch\_feb08a

```
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

SDA	GPIO21
SCK	GPIO18
MOSI	GPIO23
MISO	GPIO19
IRQ	NOT USED
GND	GND
RST	GPIO22
3v3	3v3

→ Modifier le code fourni qui permet d'allumer/éteindre une LED quand on présente une carte NFC spécifique



# Borne wifi

sketch\_feb08a

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

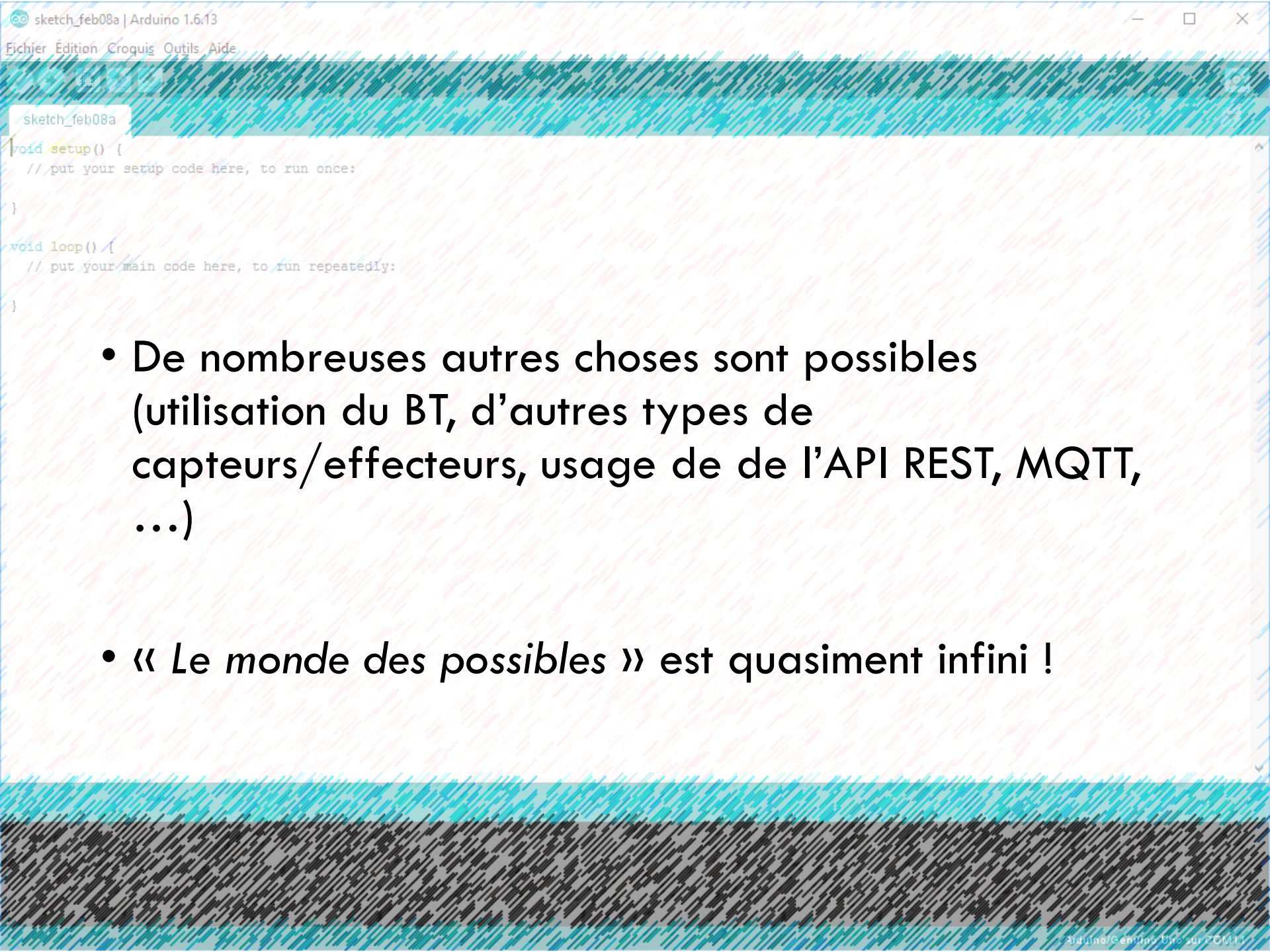
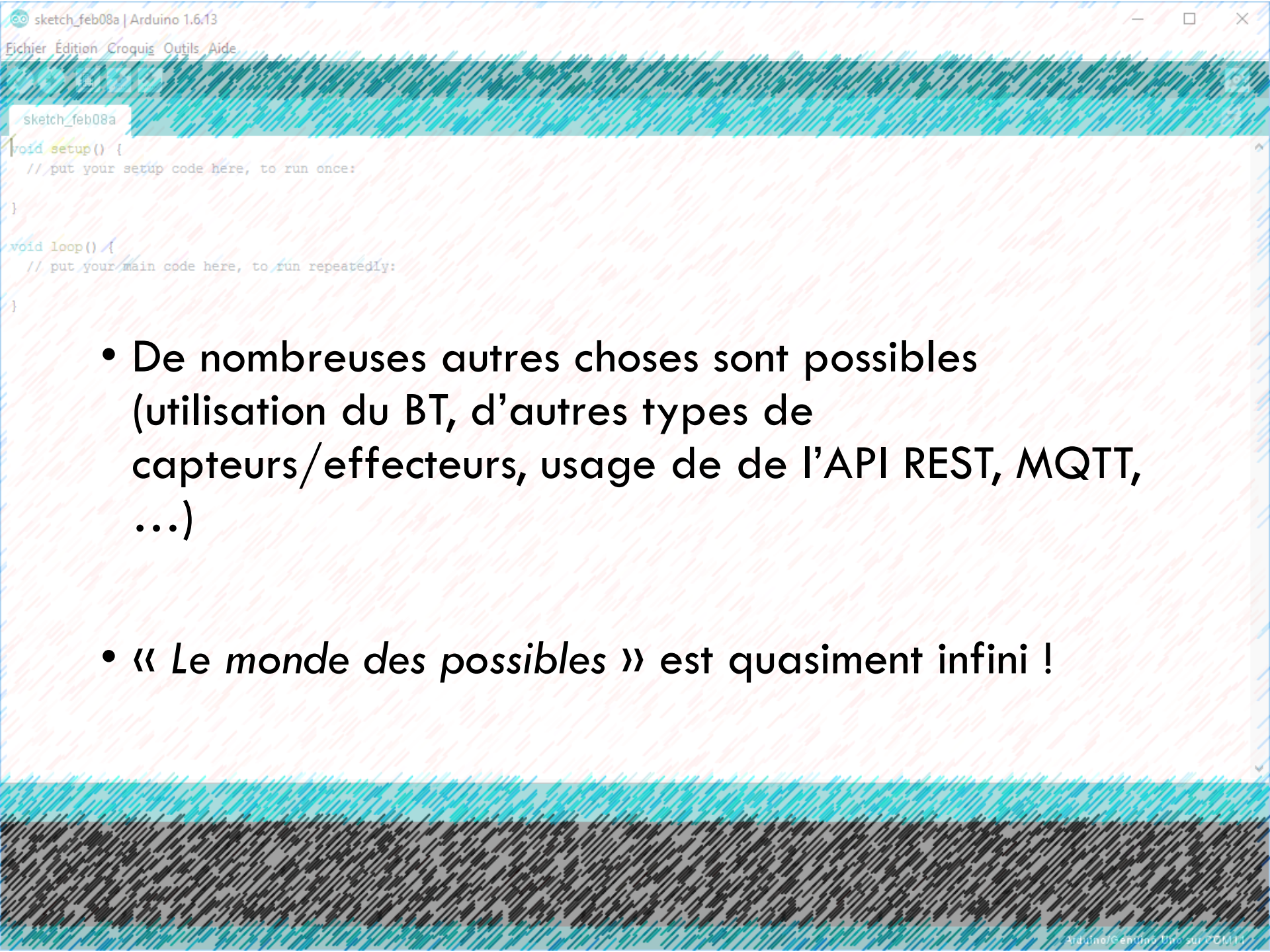


- **Une borne et/ou un client wifi**

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





- De nombreuses autres choses sont possibles (utilisation du BT, d'autres types de capteurs/effecteurs, usage de de l'API REST, MQTT, ...)
- « *Le monde des possibles* » est quasiment infini !



# Mini-Projet

sketch\_feb08a

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



# « Astuces »

sketch\_feb08a

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

- Outils | Moniteur série

