

Introduction à Arduino

<http://www.arduino.cc>

Février 2021



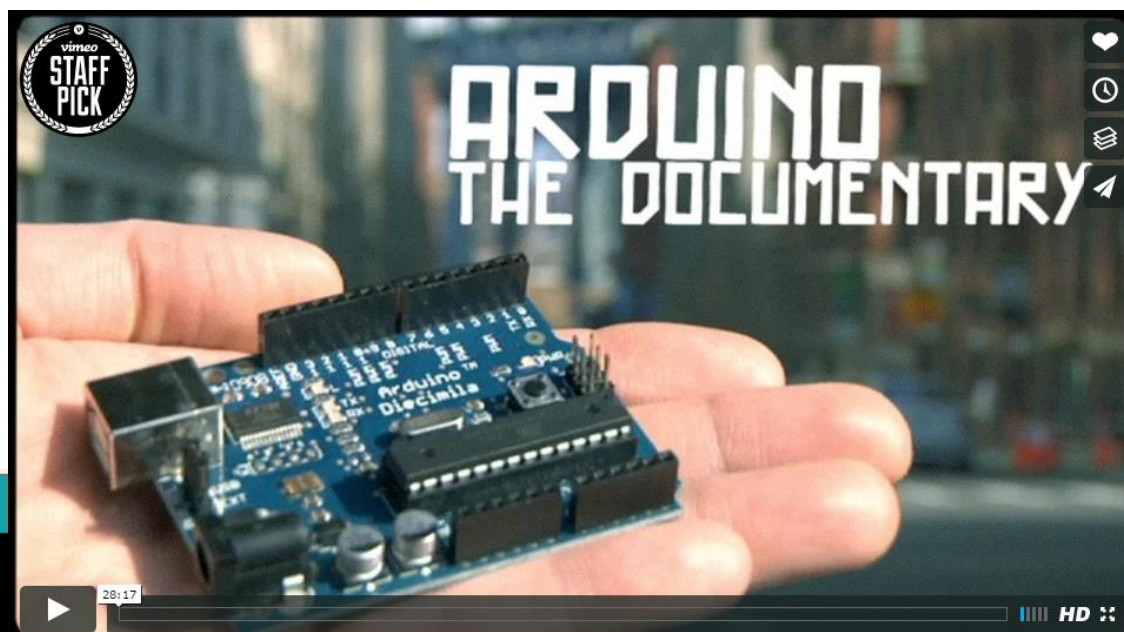
Une histoire d'Arduino ...

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://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



<http://www.processing.org>

Date : Printemps 2001

Lieu : MIT Media Lab

Ben Fry / Casey Reas



Processing 3



p5.js



Wiring

<http://wiring.org.co>

Date : 2003

Lieu : IDII

Hernando Barragán



Arduino



<http://www.arduino.cc>

Date : 2005

Lieu : IDII

Massimo Banzi



IDE – Environnement de dév.

sketch_feb08a

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

```
void loop() {  
  // put your ma  
}
```

The screenshot shows the Arduino Software IDE interface. The top bar displays the file name 'sketch_feb08a' and the version 'Arduino 1.6.13'. The menu bar includes 'Fichier', 'Édition', 'Croquis', 'Outils', and 'Aide'. The main workspace shows a code editor with the following code:

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

Below the code editor, a browser window is open to the Arduino Software page at arduino.cc/en/software. The page features the Arduino logo, a navigation menu, and a list of sketches. A prominent banner reads 'up-to-date version of the IDE includes all libraries and also supports new Arduino boards.' Below this, there are buttons for 'CODE ONLINE' and 'GETTING STARTED'. Further down, there are promotional banners for 'The Arduino® Student Kit: bring the buzz home'. The 'Downloads' section is visible, featuring the Arduino IDE 1.8.13 logo and a description: 'The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.'

The 'DOWNLOAD OPTIONS' section lists the following options:

- Windows** Win 7 and newer
- Windows** ZIP file
- Windows app** Win 8.1 or 10 [Get](#)
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM 32 bits
- Linux** ARM 64 bits
- Mac OS X** 10.10 or newer

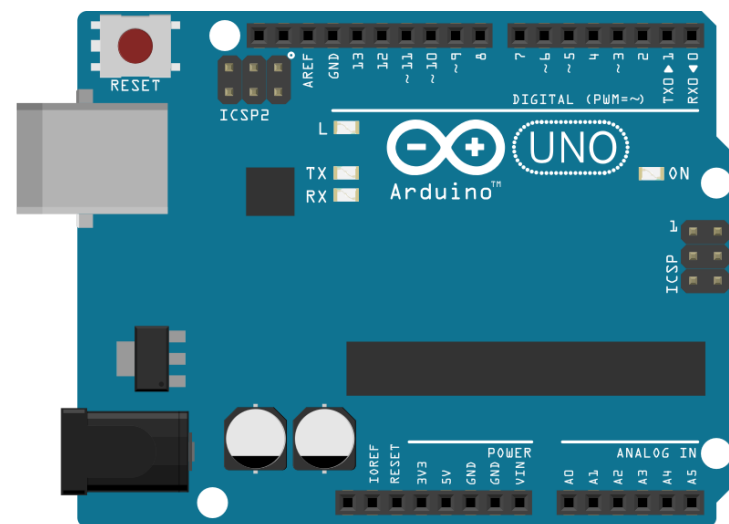
A 'Help' button is located at the bottom right of the download section.

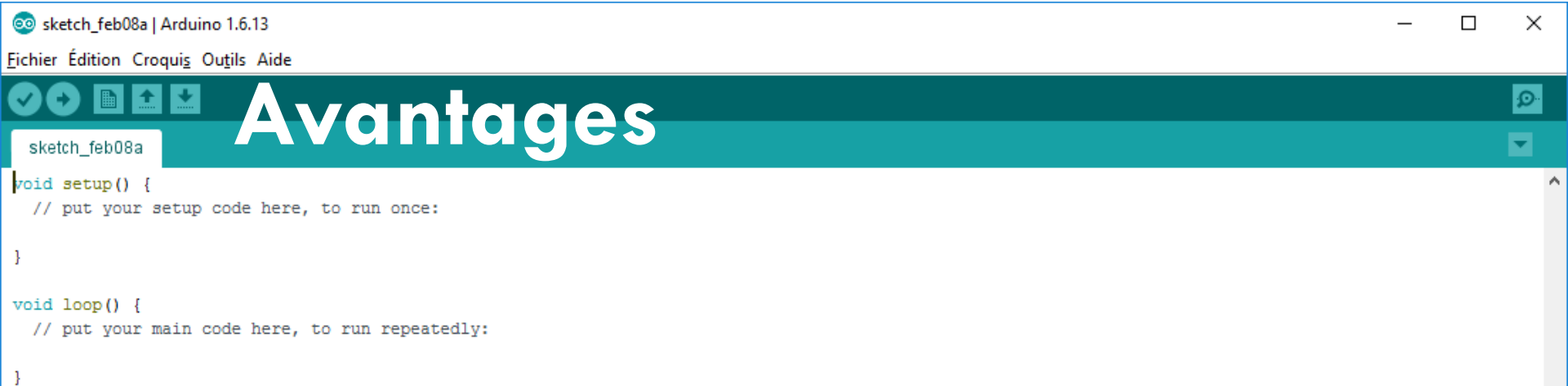
La carte générale ...

sketch_feb08a

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

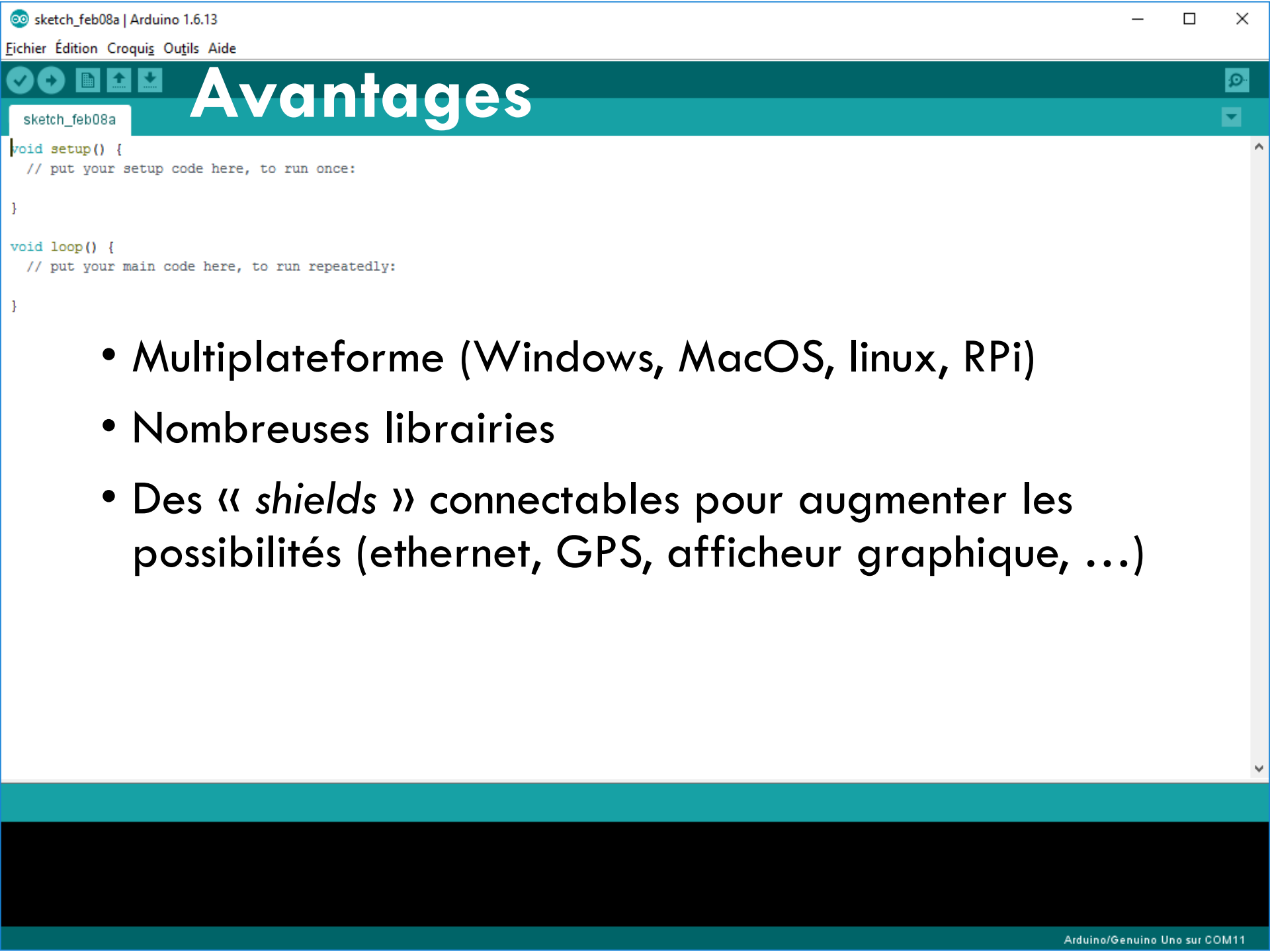
- Des entrées/sorties numériques
- Des entrées analogiques (A)
- ...





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

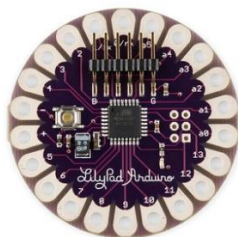
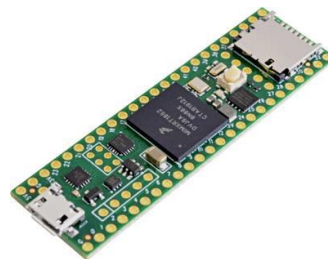
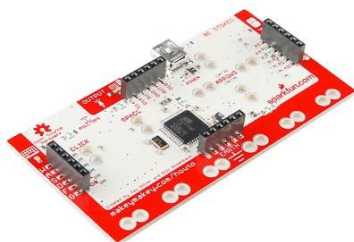
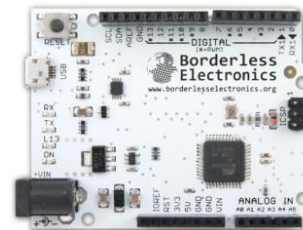
- Multiplateforme (Windows, MacOS, linux, RPi)
- Nombreuses librairies
- Des « *shields* » connectables pour augmenter les possibilités (ethernet, GPS, afficheur graphique, ...)

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:  
}
```

De multiples versions disponibles



... plein d'autres !

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:  
}
```

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

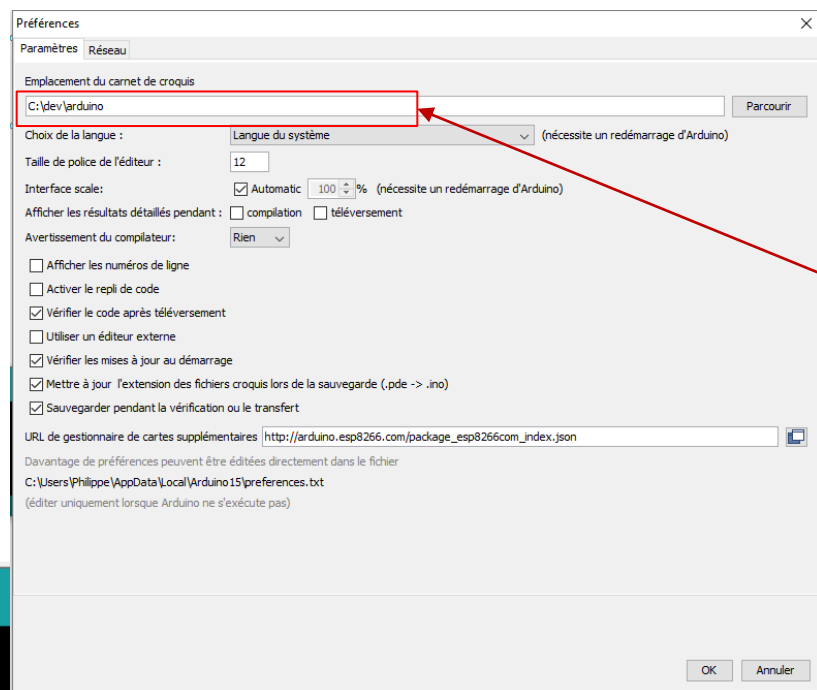
La base du programme Arduino est le « *sketch* »
(programme, prototype)
L'extension est le « **.ino** »

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

ELIPSE (C:) > dev > arduino > servo_HQ			Rechercher dans : s
Nom	Modifié le	Type	
servo_HQ.ino	24/02/2016 16:12	Fichier INO	

Deux fonctions basiques

sketch_feb08a

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

}

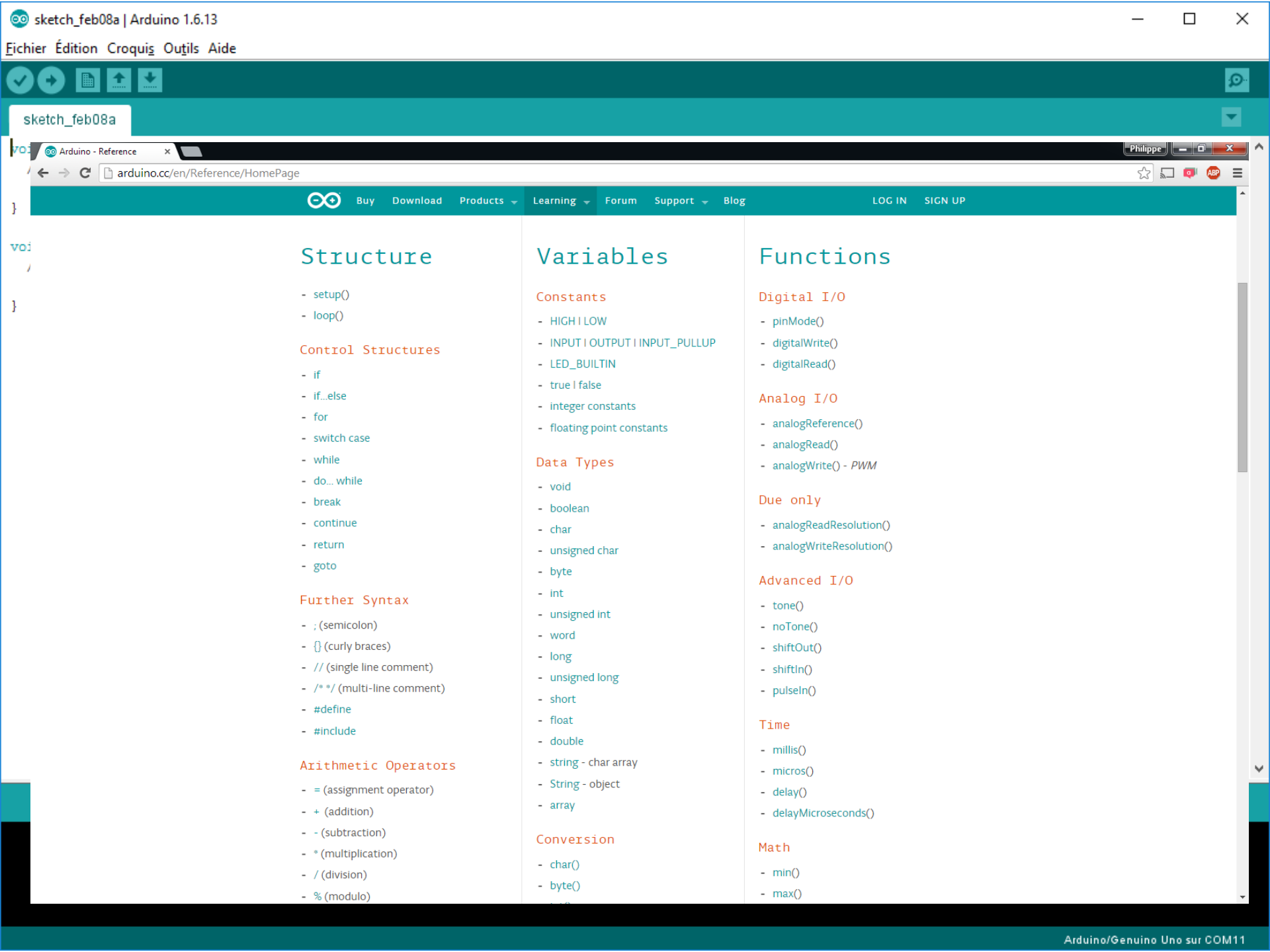
```
void loop() {  
  // put your main code here, to run repeatedly:
```

}

- **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 1.6.13

FichierÉditionCroquisOutilsAide

sketch_feb08a

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

IORef: 5V
Vin: 7-12V DC max.

IORef
Reset
+3.3V
+5V
Gnd
Gnd
Vin

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

Comm. ADC GPIO

ICSP:
Reset
Gnd

SCK
MISO
MOSI
+5V

GPIO18
GPIO19
Gnd
GPIO13
GPIO12
GPIO11
GPIO10
GPIO9
GPIO8
GPIO7
GPIO6
GPIO5
GPIO4
GPIO3
GPIO2
GPIO1
GPIO0

ADC4
ADC5
AREF
SCK
MISO
MOSI
CS
PWM11
PWM10
PWM9
PWM6
PWM5
PWM3
TX
RX

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.

GPIO
ADC
Comm.
PWM
Interrupts

LED

INT1
INT0

Arduino/Genuino Uno sur COM11

Un premier exemple

sketch_feb08a

```
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  
}
```

« A ne pas oublier »

sketch_feb08a

```
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



« Astuces »

sketch_feb08a

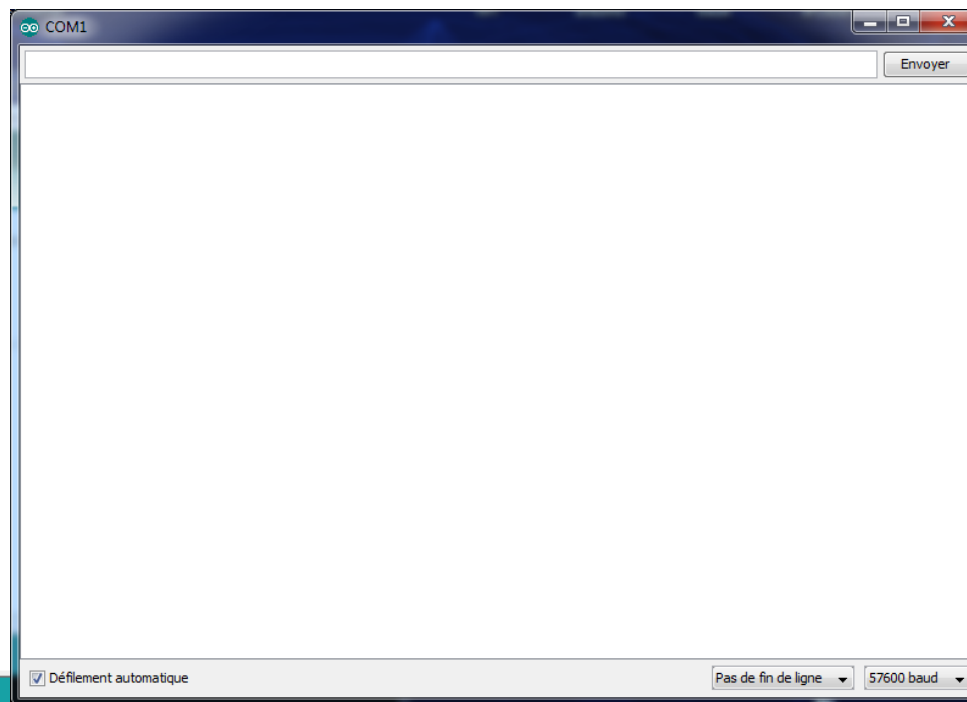
```
void setup() {  
  // put your setup code here, to run once:
```

```
}
```

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

```
}
```

- Outils | Moniteur série



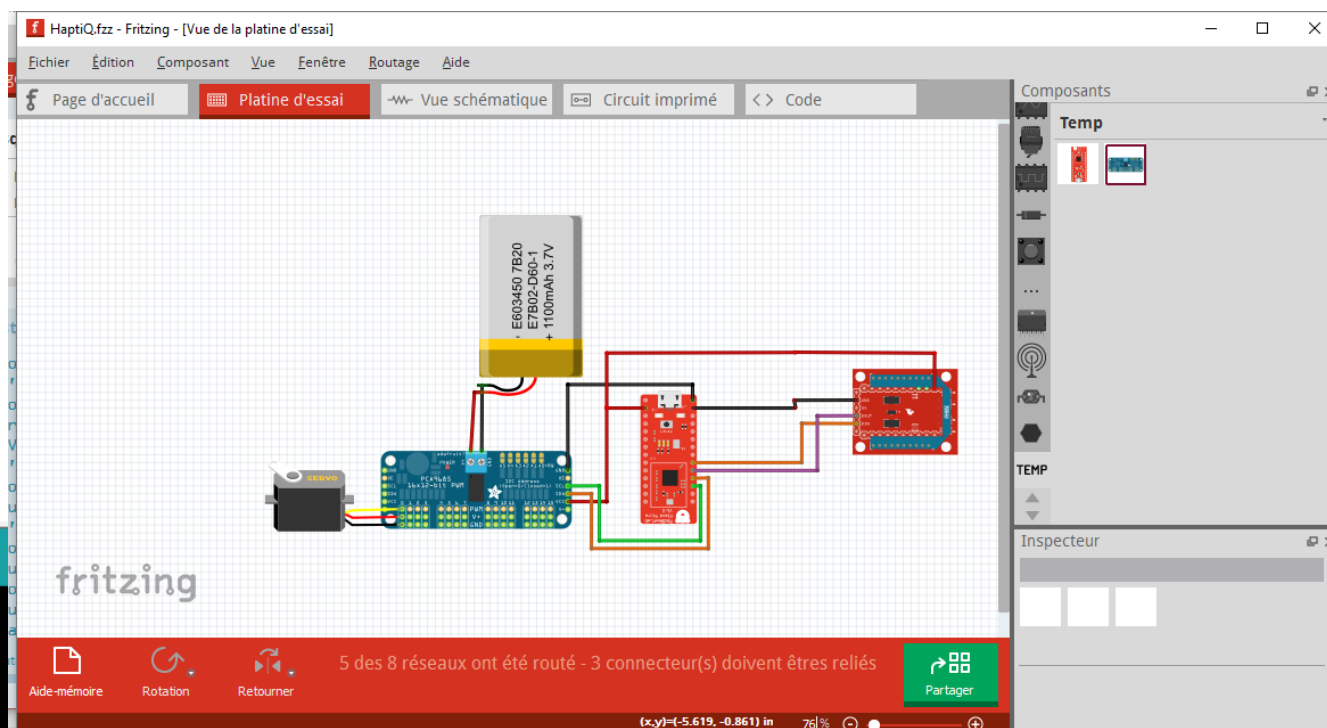
Un outil d'aide au montage

sketch_feb08a

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

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

- **Fritzing** - <http://fritzing.org> (payant depuis 2019)
<http://fritzing.org/download/0.9.3b/windows-64bit/fritzing.0.9.3b.64.pc.zip>



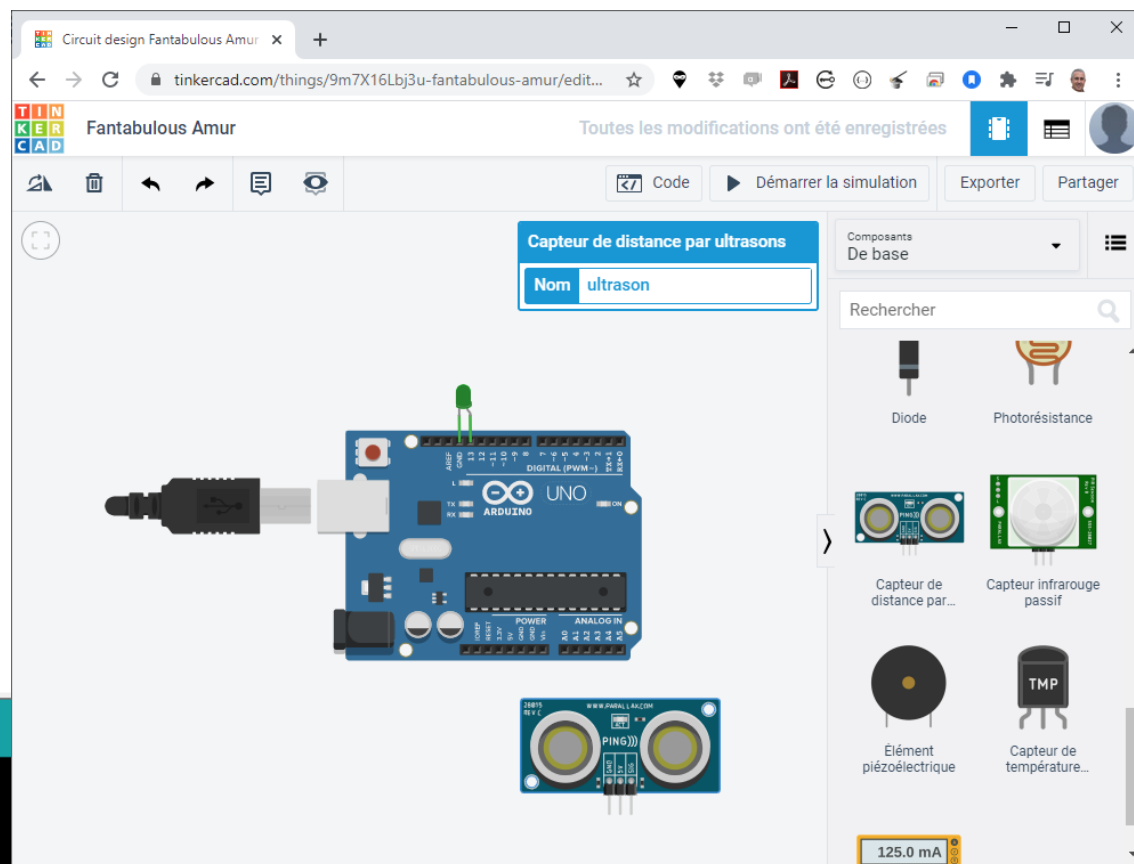
Un simulateur en ligne :

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

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

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

- Choisir **Circuits**





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***

→ modification de la durée du clignotement

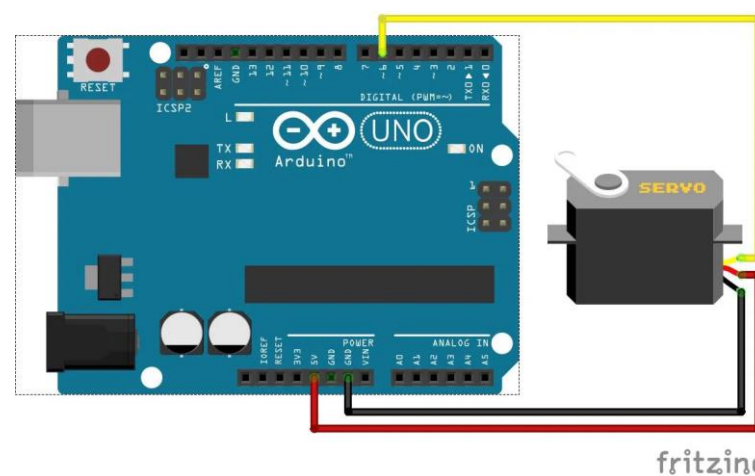
Exercices de démarrage

sketch_feb08a

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



- **Librairie et servomoteur (positionnement angulaire)**



fritzing

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_v1.9.1.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/>)



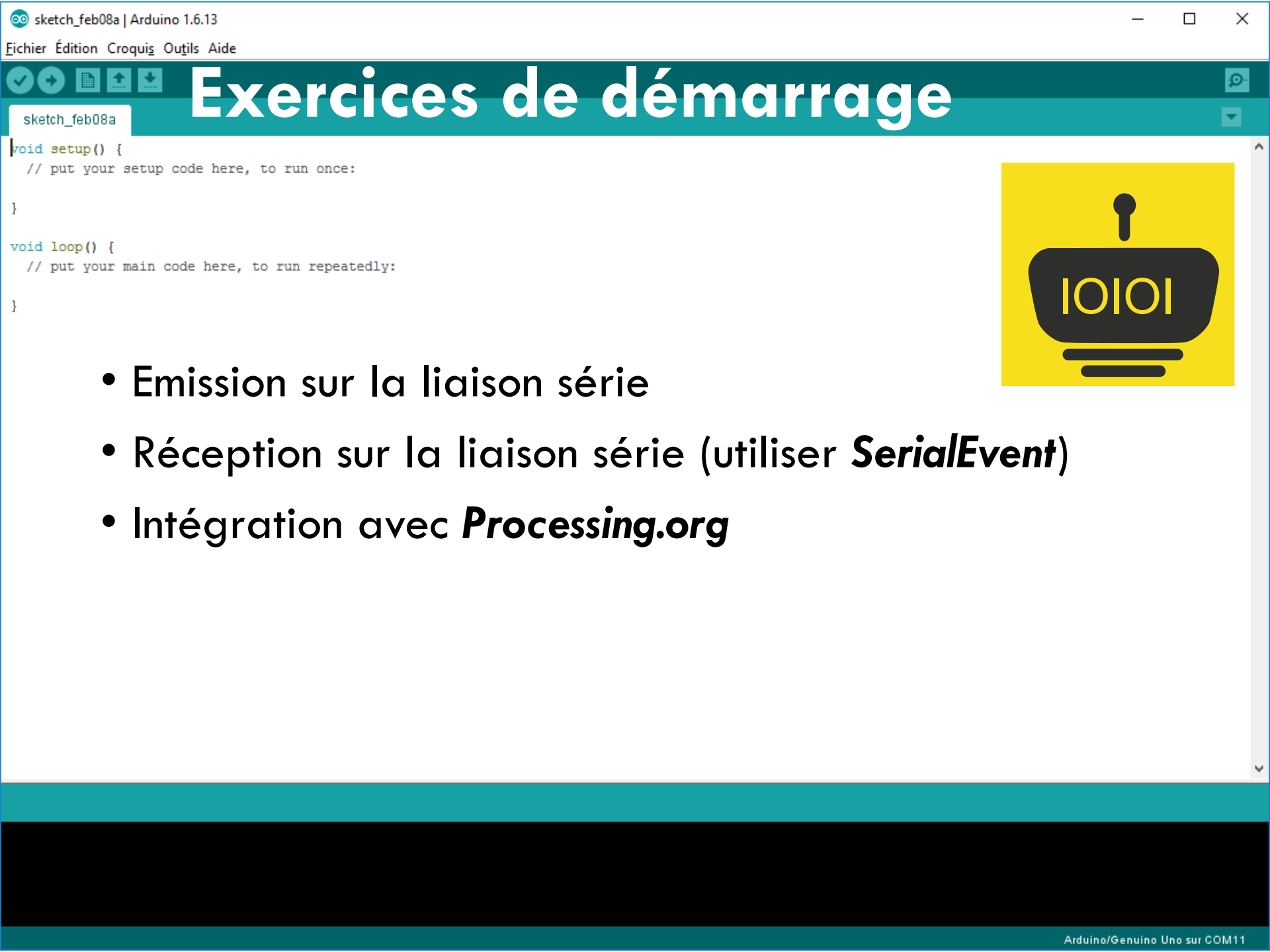
Exercices de démarrage

sketch_feb08a

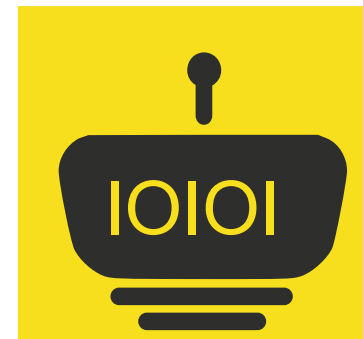
```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```



→ 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



- Emission sur la liaison série
- Réception sur la liaison série (utiliser **SerialEvent**)
- Intégration avec **Processing.org**

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

