



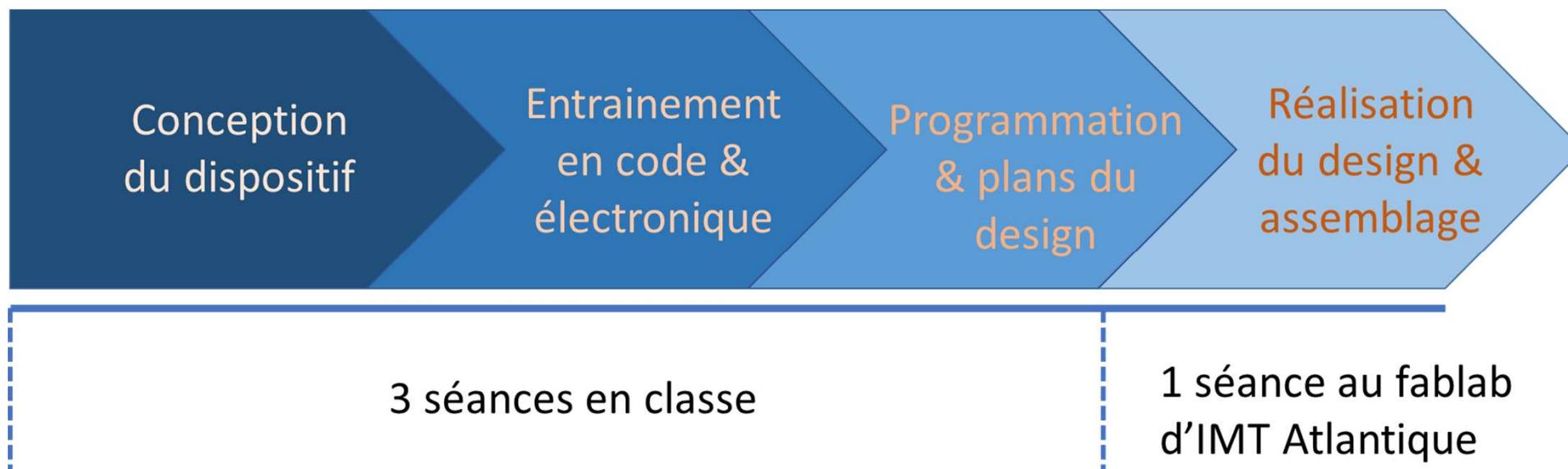
**IMT Atlantique**

Bretagne-Pays de la Loire  
École Mines-Télécom

**DÉFI AERE**

**LYCÉE DE L'IROISE  
2022-203**

**Maud TOURNERY**





**Concevoir, assembler et fabriquer un dispositif permettant de signaler lorsque le taux de CO<sub>2</sub> dans une salle est trop élevé**

**Norme NF EN 13779** : Concentration Co<sub>2</sub> mesurée en ppm (partie par millions)

*Concentration en Co<sub>2</sub> < 800ppm - **Qualité d'air excellente***

*800ppm < Concentration en Co<sub>2</sub> < 1500ppm - **Qualité d'air modérée***

*Concentration en Co<sub>2</sub> > 1500ppm - **Qualité d'air basse -> aérer***



15 min

Imaginer et faire un schéma d'un dispositif permettant de répondre au défi avec le matériel disponible au lycée

Par groupe, 5 groupes

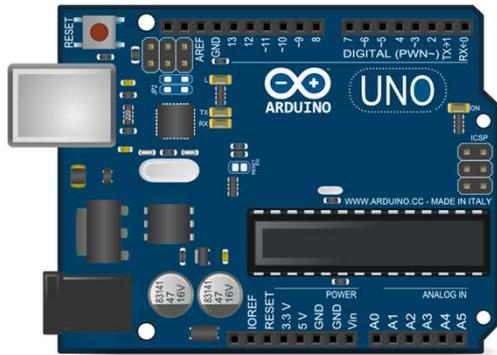


20 min

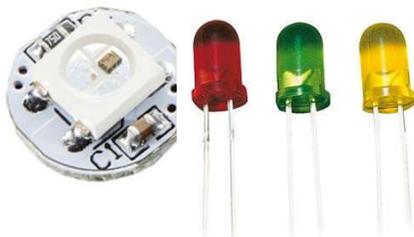
Présentation en 1 minute de l'idée de chaque groupe

Design, options, matériel etc.

# LE MATÉRIEL

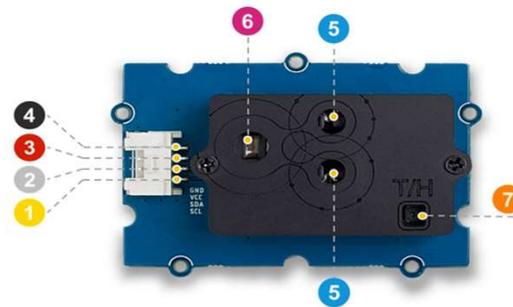


Micro-contrôleur Arduino = récupère et exécute le programme



LEDs ou neopixel

➔ afficher une lumière colorée

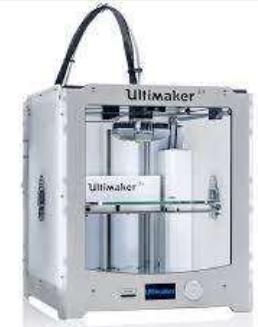


- 4 GND: connect this module to the system GND
- 3 VCC: you can use 5V or 3.3V for this module
- 2 SDA: I<sup>2</sup>C serial data
- 1 SCL: I<sup>2</sup>C serial clock
- 5 CO<sub>2</sub> Sensor Opening
- 6 Infrared Light Source
- 7 Temperature & Humidity Sensor Opening

Capteur du taux de CO<sub>2</sub>



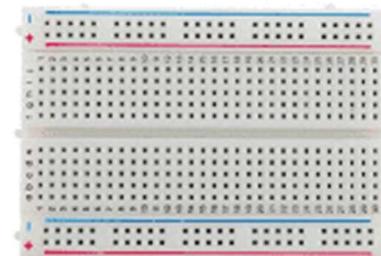
Découpeuse laser



Imprimante 3D



Fils électriques



05/03/2024

Breadboard

Logotype partenaire



10 min

Refaire un schéma du dispositif avec le nouveau matériel

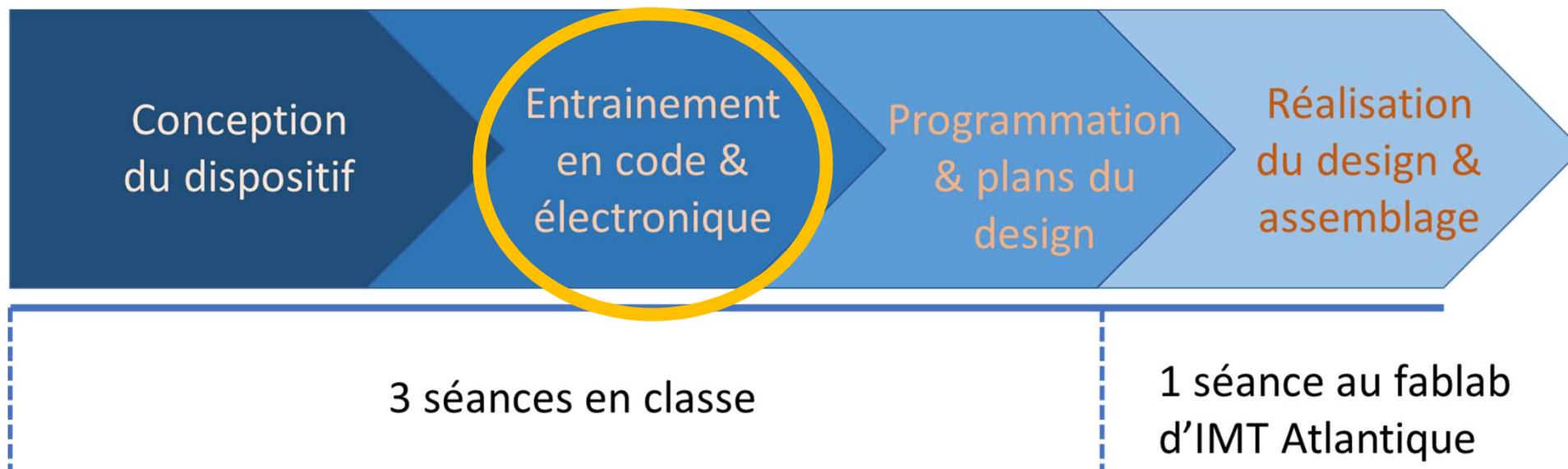
- Où positionner le dispositif dans la classe ?
  - Comment l'alimenter en énergie ?
  - Qui va s'en servir ? Quelles contraintes ?
  - Où seront les lumières ? Quelle forme & aspect ?
- Par groupe, 5 groupes



15 min

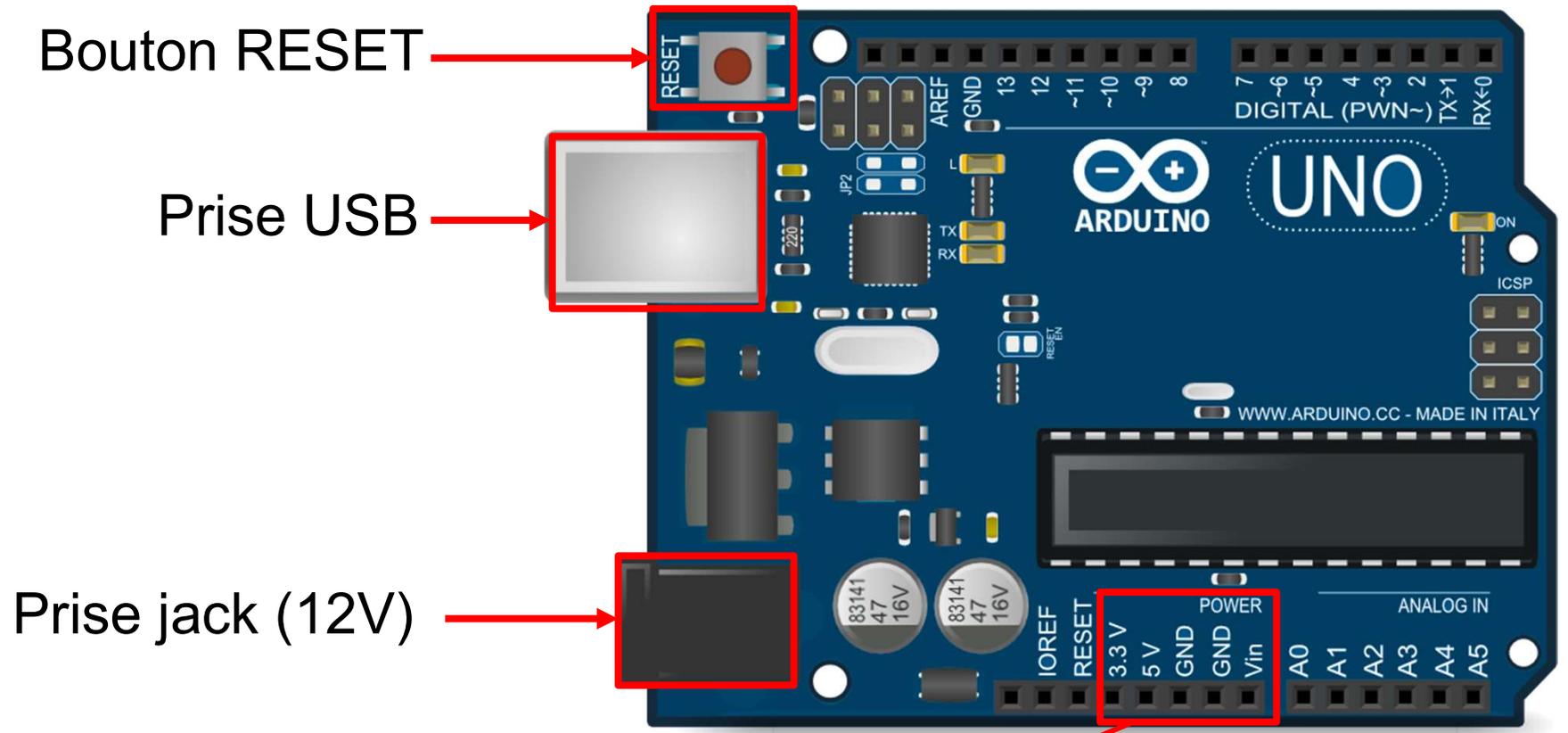
Présentation en 1 minute de l'idée de chaque groupe

Design, options, matériel etc.



Réaliser les défis (branchements et code) de la mallette MERITE :

- Apprendre à faire clignoter une puis deux LEDs
- Apprendre à programmer une LED avec un bouton pression (boucle si... sinon)
- Apprendre à utiliser une photorésistance (différence entre digital et analogique)

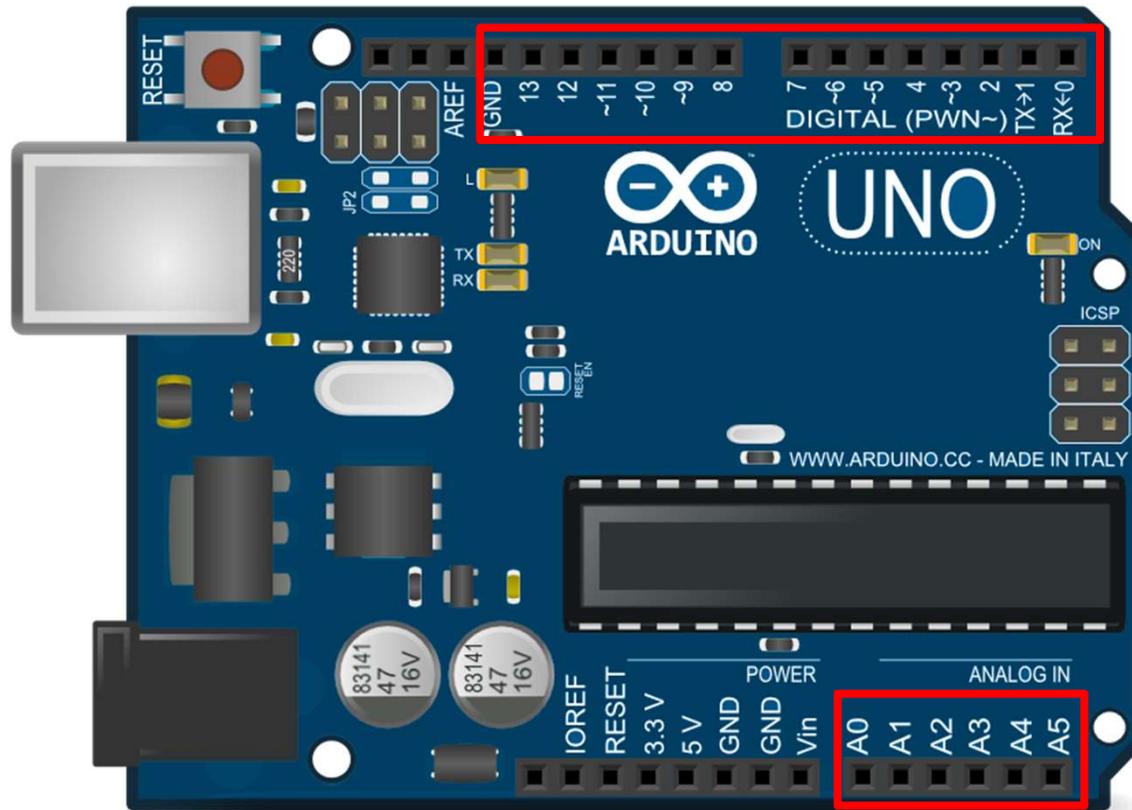


**Bornier d'alimentation**

**GND = terre**

**5V : alimente l'Arduino et les circuits**

Entrées & sorties numériques de 0 à 13

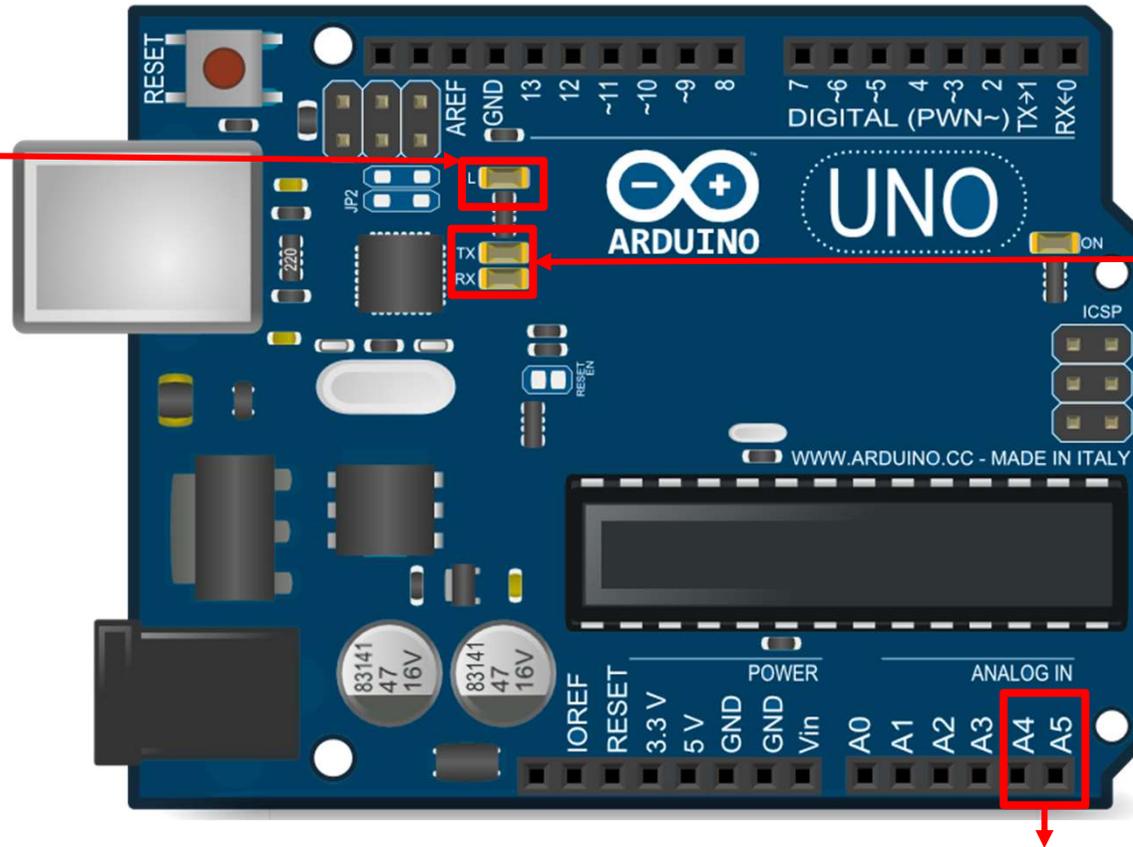


Entrées analogiques ou entrées & sorties numériques

# CODER EN BLOCS: MICROCONTRÔLEUR ARDUINO UNO <sup>14</sup>

(communication ordinateur, un

LED témoin  
de la borne 13  
(et sa  
résistance)



LED  
born

Borne A4 : SDA = signal de données  
Borne A5 : SCL = signal d'horloge

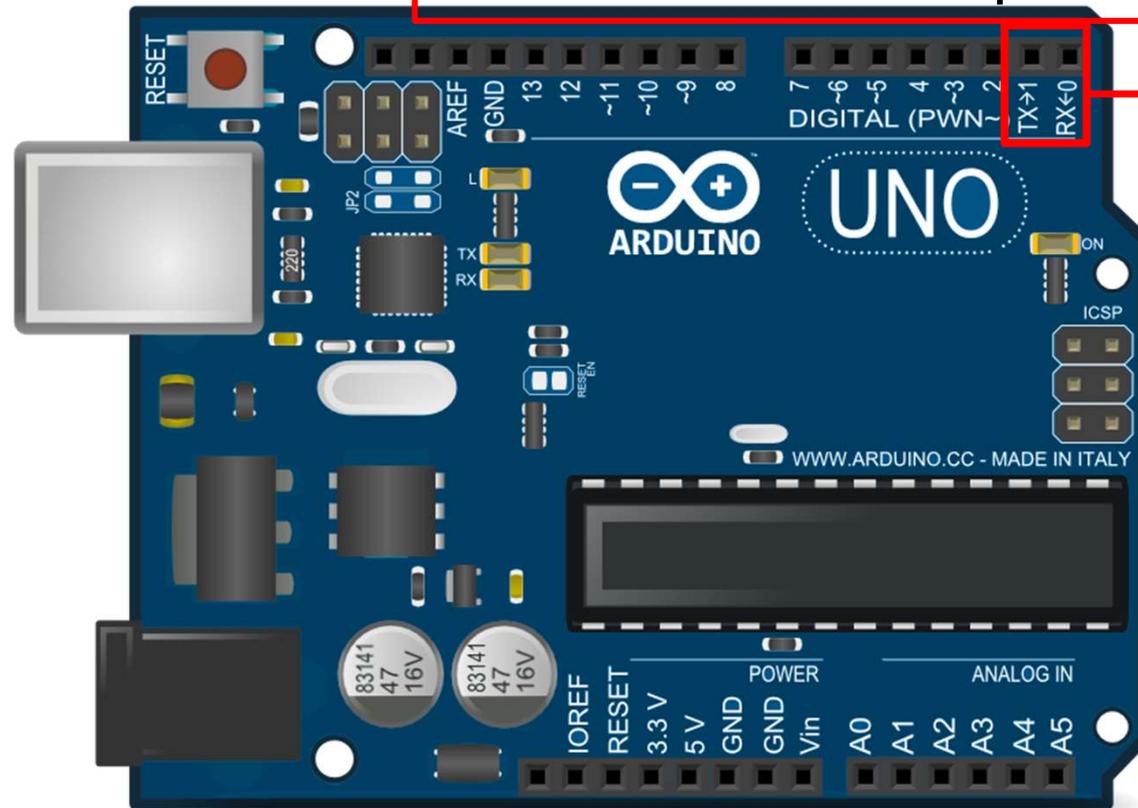
**Bus I2C**

(communication entre l'Arduino (=maître) et un ou plusieurs circuits comme des capteurs (=esclaves))

## Port série

(communication entre l'Arduino et l'ordinateur, un autre Arduino etc.)

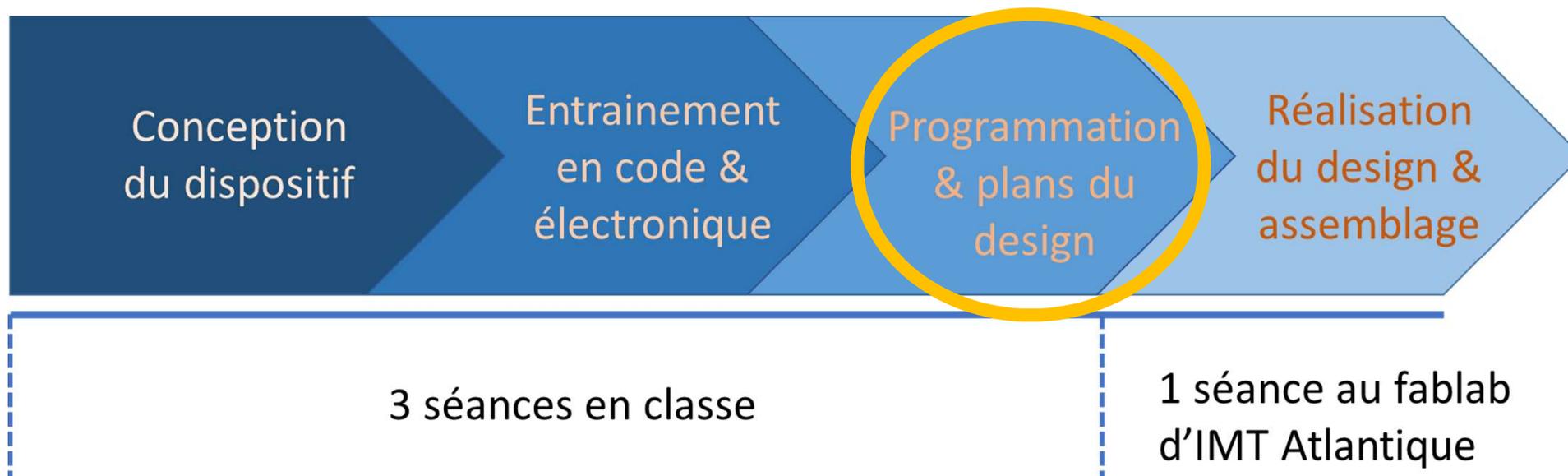
Borne 1 : Tx = émission de données  
Borne 0 : Rx = réception de données

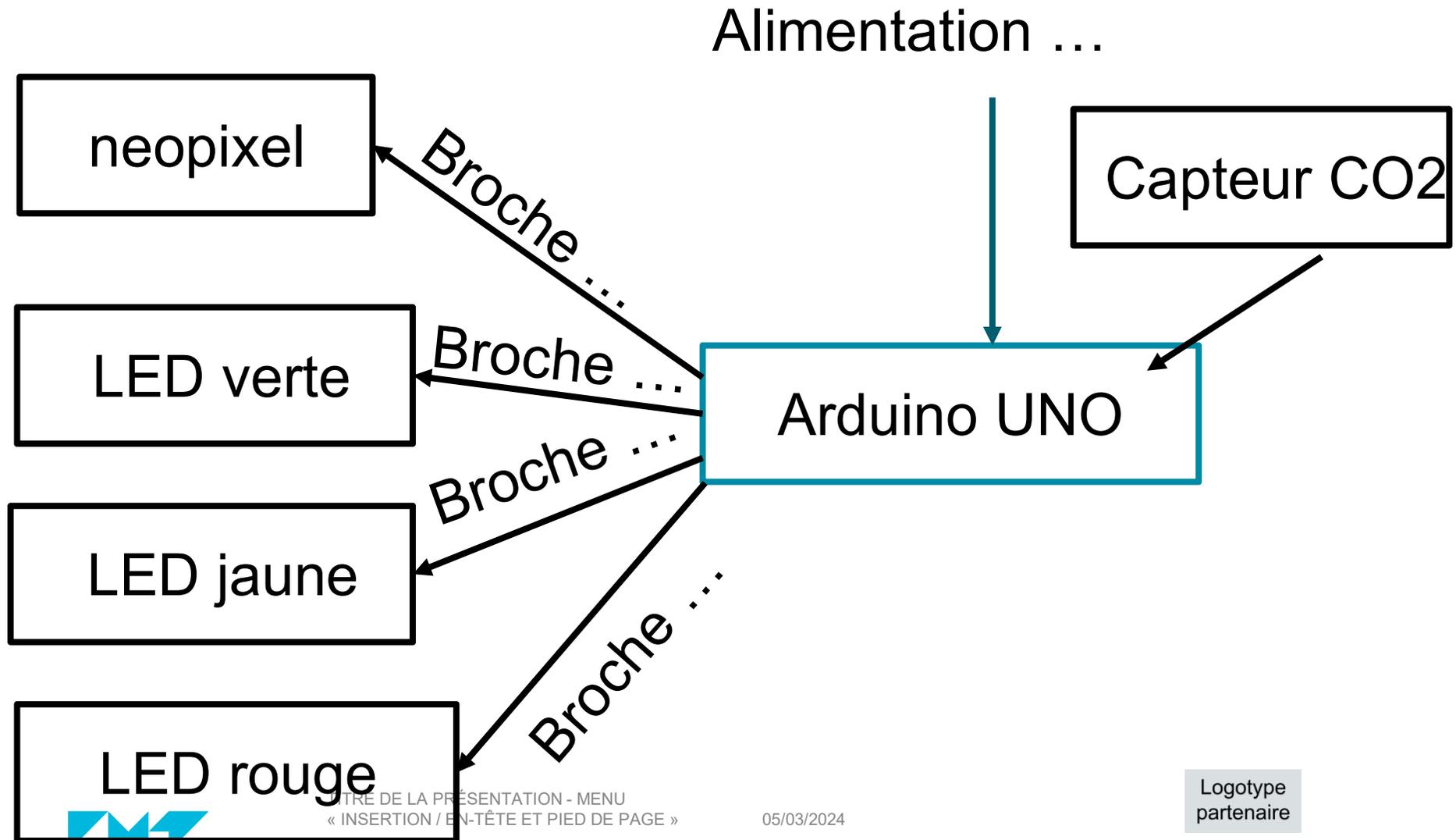


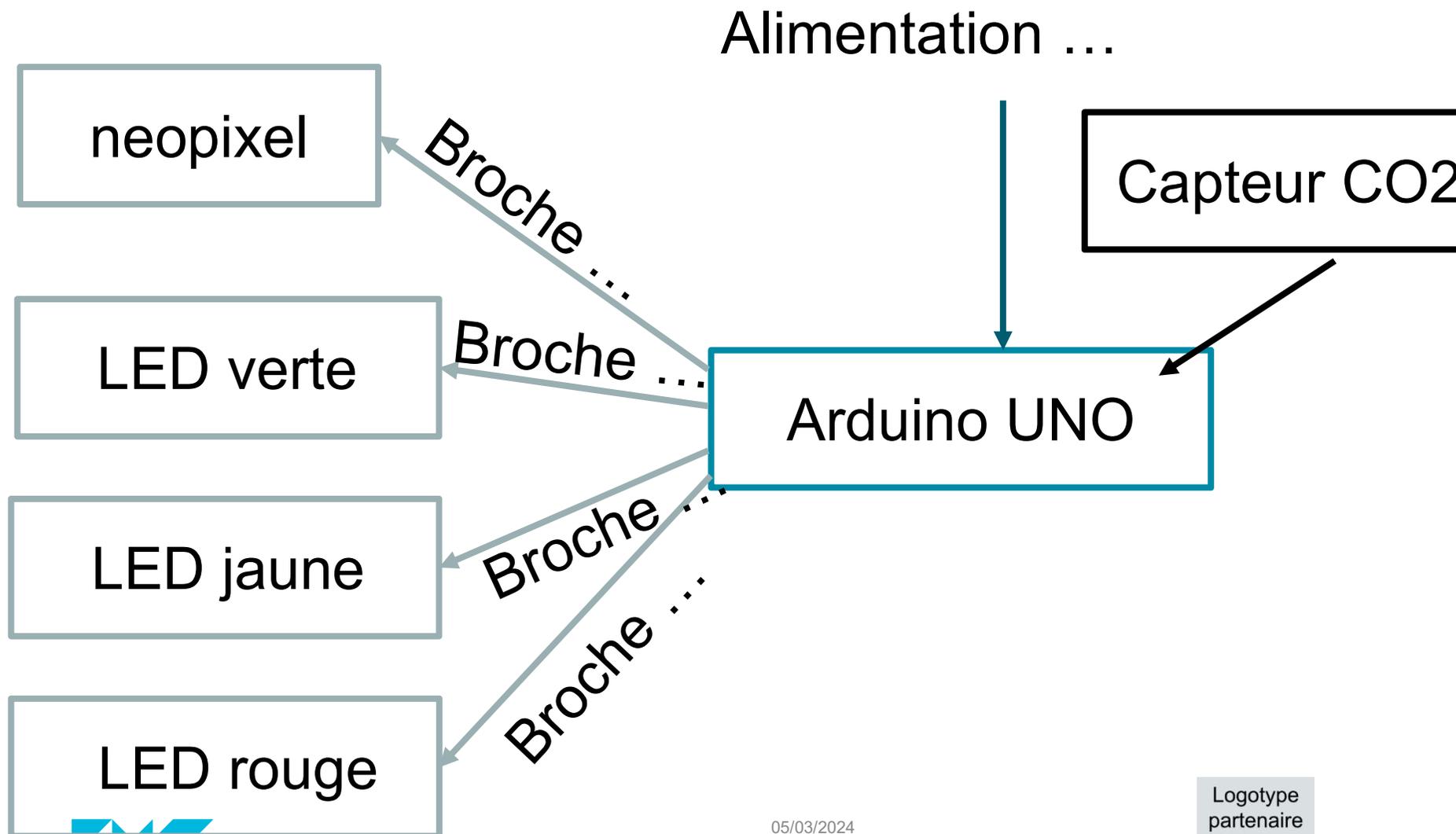
Chaque composant doit être reconnu par le micro-contrôleur, on lui indique pour cela la **broche** sur laquelle est branché le composant

Les LEDs peuvent être programmées pour clignoter, s'allumer, s'éteindre indépendamment ou de manière conditionnelle (*si le bouton est enfoncé alors la LED s'allume, sinon elle s'éteint*)

On peut afficher les valeurs physiques, réelles mesurées par un **capteur analogique** (comme la photorésistance) sur le **moniteur série**







Consommation en veille de l'Arduino = 30 mA / h

Consommation de l'Arduino allumé = 12mA / h

Capacité d'une pile 9V : 600 (alcaline) à 1200 (lithium) mA

= Quel durée d'autonomie ?

***Capacité de la pile / consommation de l'appareil = durée en heure***

Quelle autonomie ?

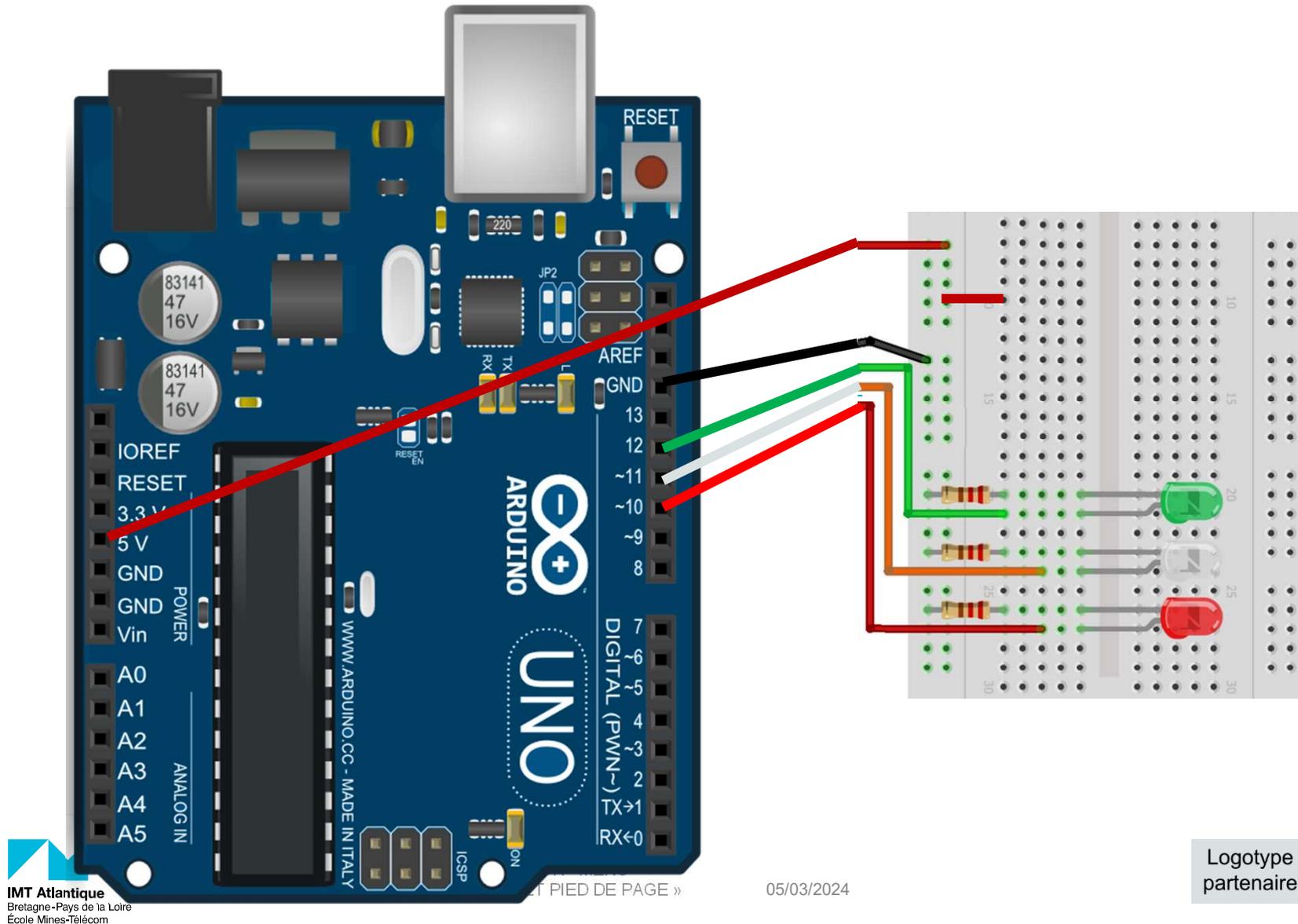
Conclusion ?

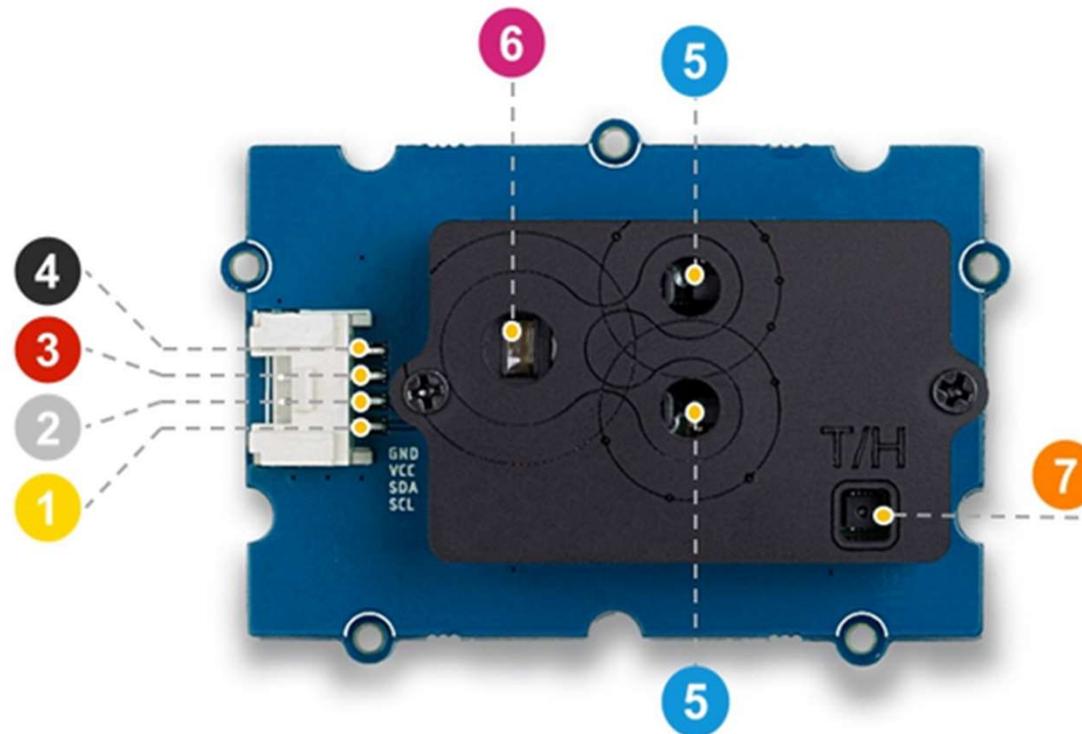
Pour alimenter l'Arduino, un câble microUSB-USB va très bien

Attention à bien le connecter

- soit directement à l'ordinateur
- soit via un chargeur de téléphone qui transforme le 220V de la prise en 12V.

**Conclusion : quelle alimentation ?**



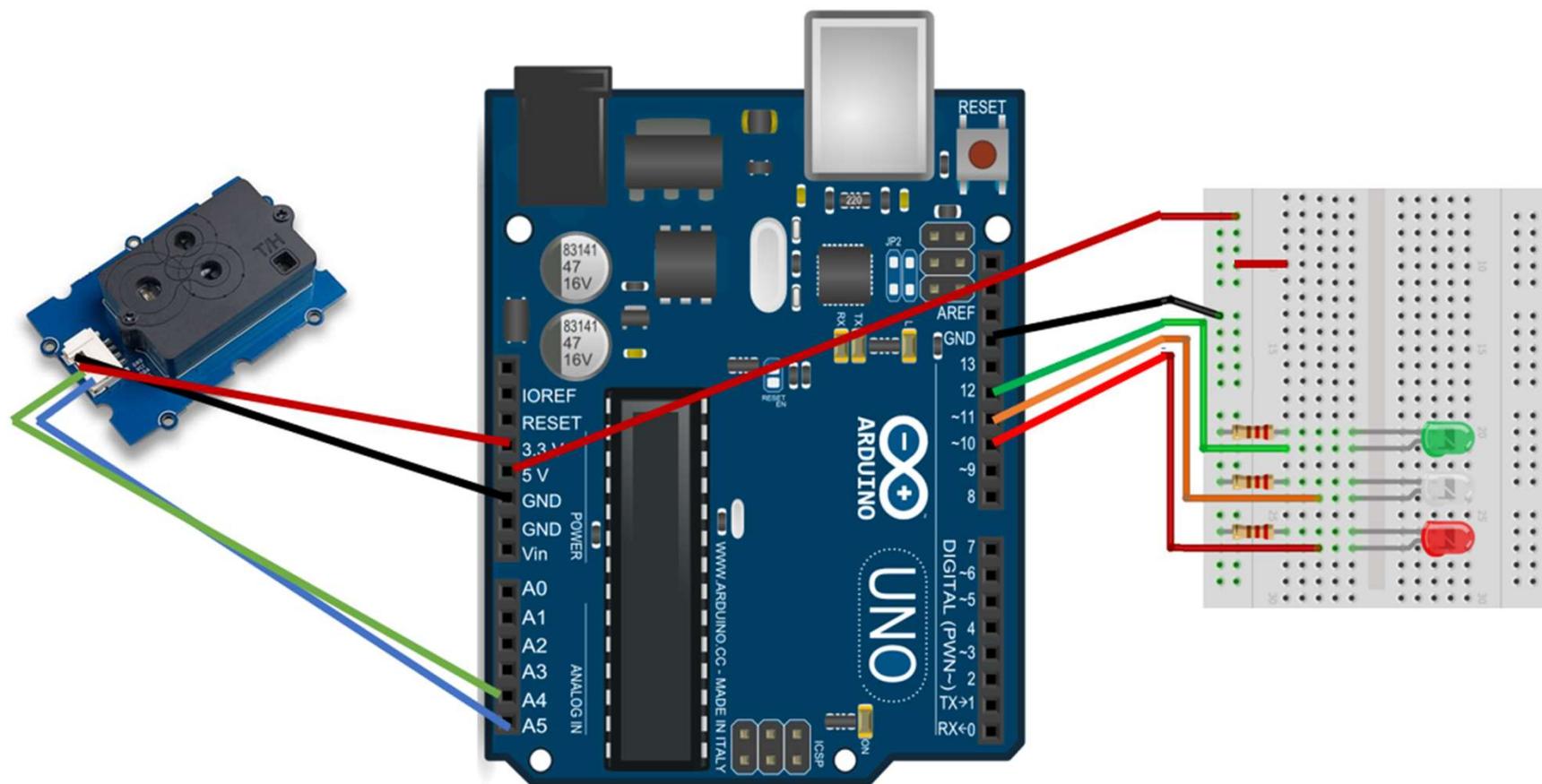


- 4 GND: connect this module to the system GND
- 3 VCC: you can use 5V or 3.3V for this module
- 2 SDA: I<sup>2</sup>C serial data
- 1 SCL: I<sup>2</sup>C serial clock

- 5 CO<sub>2</sub> Sensor Opening
- 6 Infrared Light Source
- 7 Temperature & Humidity Sensor Opening

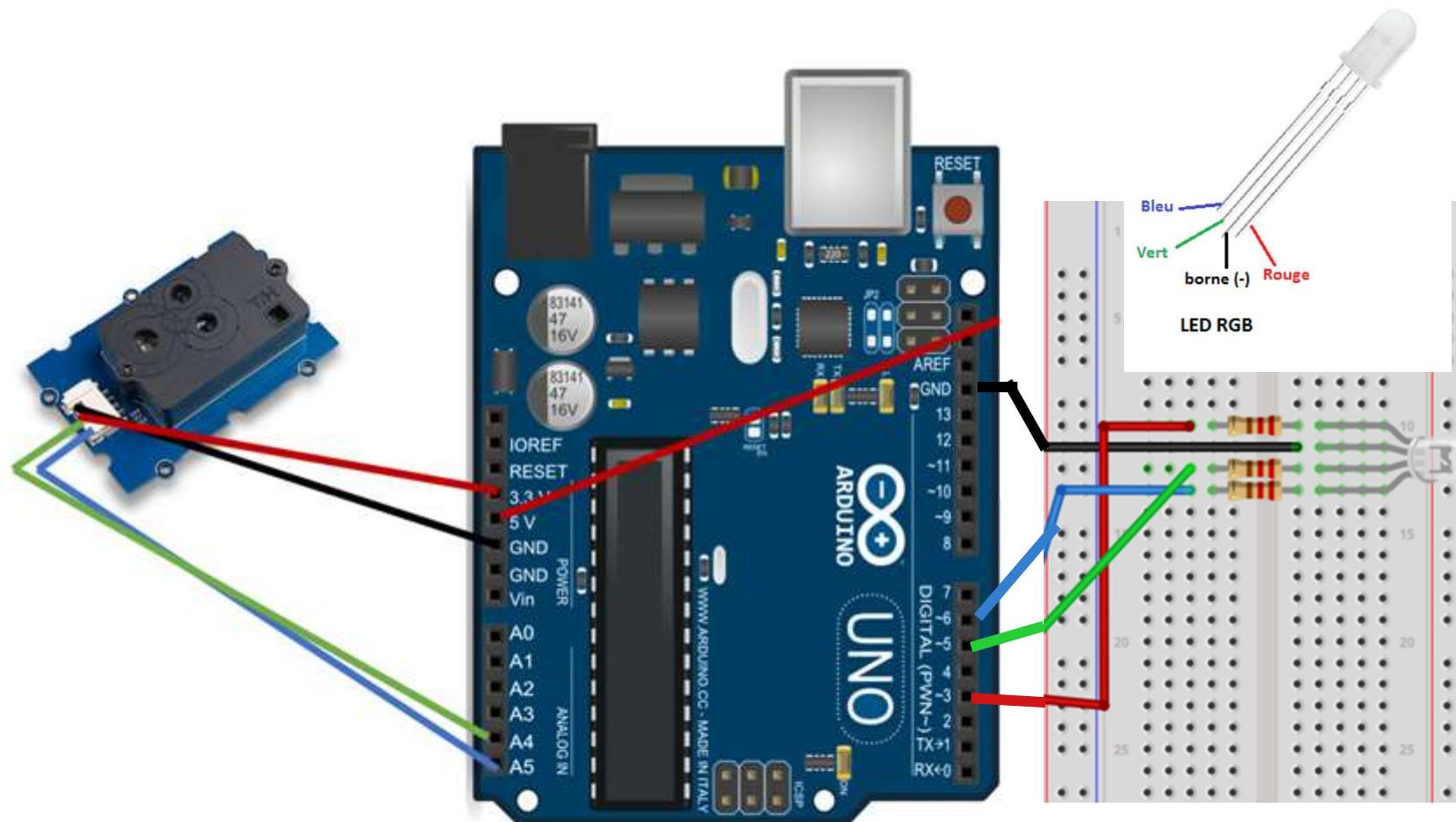
# LE DÉFI : SCHÉMA DE BRANCHEMENT- LEDS

24



# LE DÉFI : SCHÉMA DE BRANCHEMENT-LED RGB

25



## Vittascience

<https://fr.vittascience.com/arduino/?mode=mixed&console=bottom&toolbox=vittascience>

=> Programmer => Arduino

vitta science

Programmer Ressources Classe Matériel

**DÉCOUVREZ LA PROGRAMMATION INFORMATIQUE EN TOUTE SIMPLICITÉ**

Vittascience est une plateforme éducative pensée pour l'apprentissage du codage, qui propose des outils innovants pour l'enseignement. Programmer des cartes, robots, jeux, etc., la seule limite est l'imagination !

Programmer >

Bretagne-Pays de la Loire  
École Mines-Télécom

The screenshot shows a navigation bar with 'Programmer', 'Ressources', 'Classe', and 'Matériel'. Below it is a circular diagram with 'Python', 'Arduino', and 'Microbit' icons connected by dashed lines. A central image shows a laptop displaying the programming interface.

vitta science

Programmer Ressources Classe Matériel

**Programmer**

Retrouvez l'interface de programmation que vous souhaitez en cliquant sur les vignettes.

- Arduino**  
Interface de programmation pour la carte Arduino
- BBC microbit**  
Interface de programmation pour la carte BBC microbit
- Python**  
Interface de programmation pour Python 3 pensée pour l'éducation.

The screenshot shows a green header with the title 'Programmer' and a sub-header 'Retrouvez l'interface de programmation que vous souhaitez en cliquant sur les vignettes.' Below this are three white boxes, each containing an image of a hardware board and its corresponding programming interface name and description.



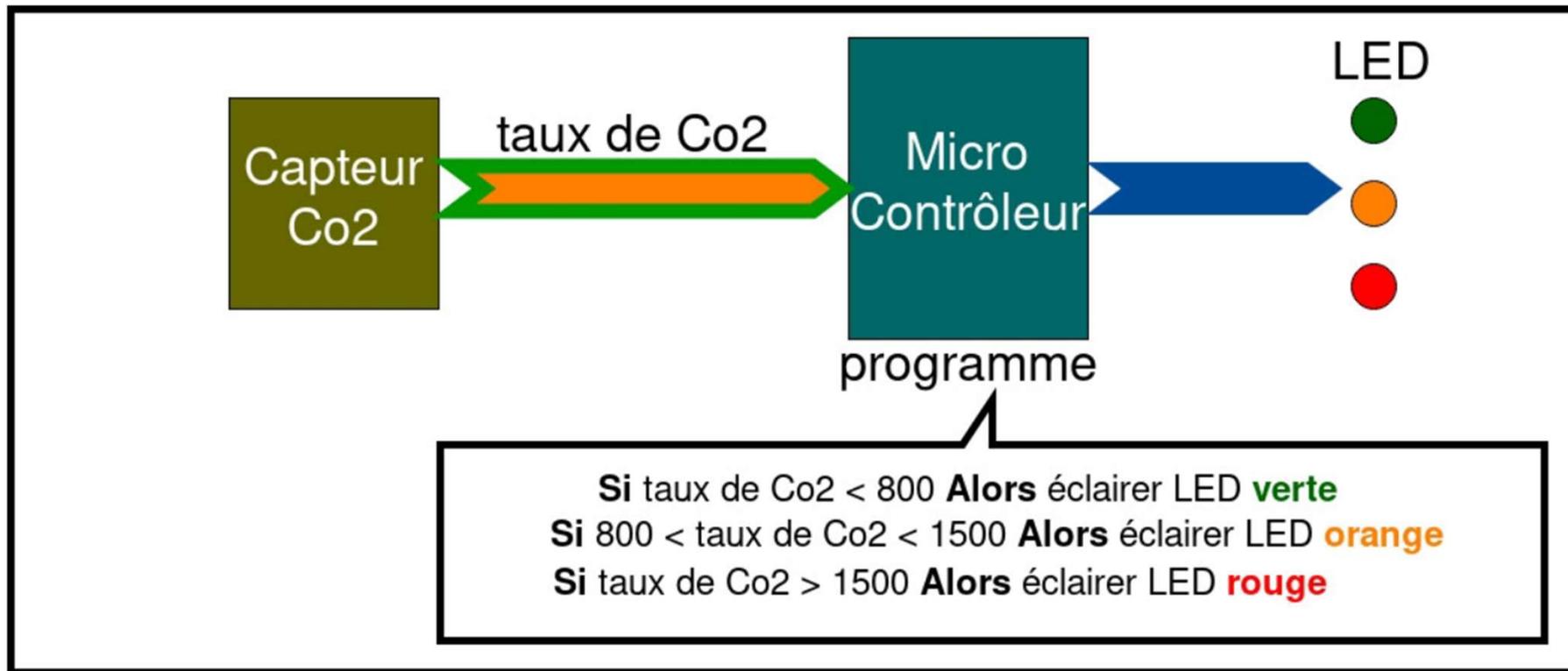
10 min

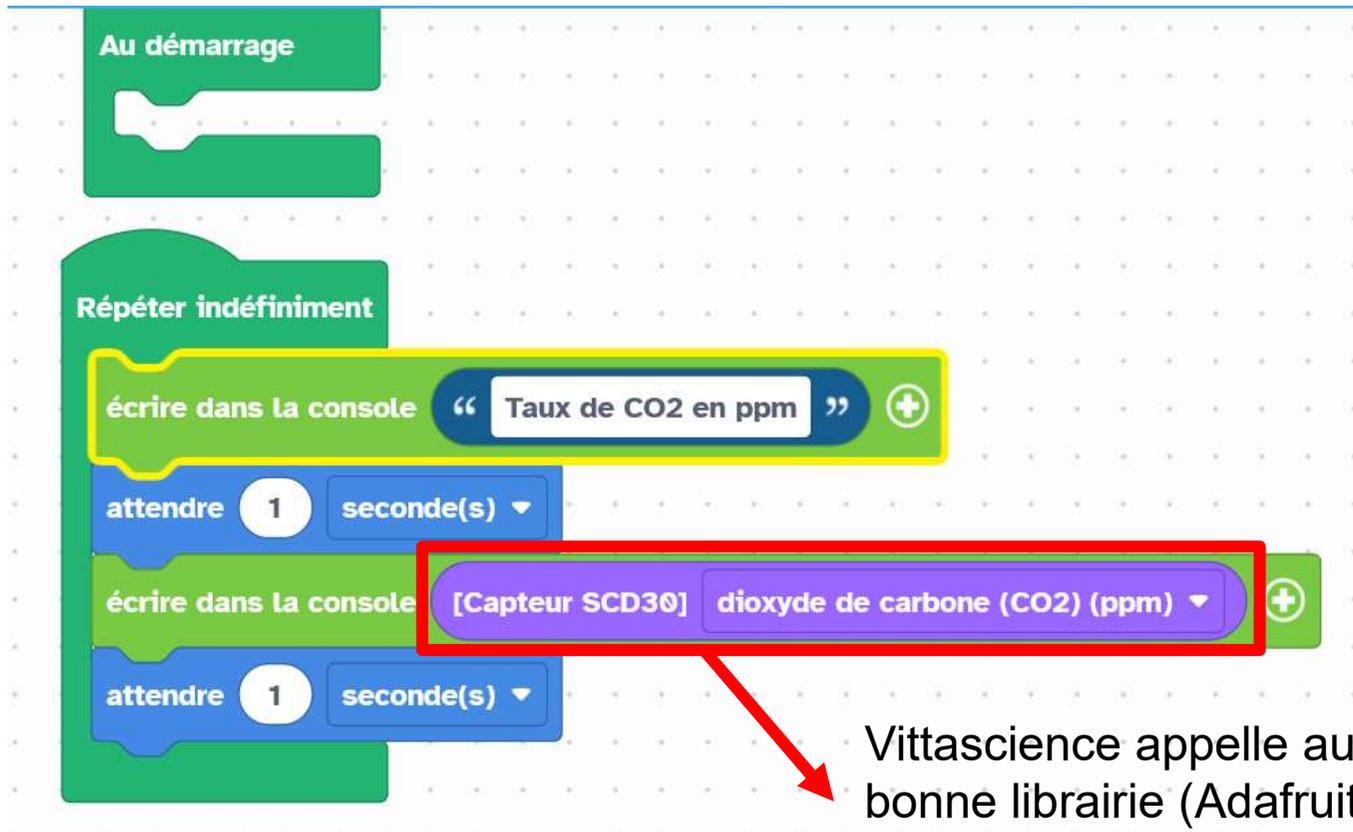
Coder pour que les LEDs soient identifiées par l'Arduino, s'allument, s'éteignent  
Par groupe, 5 groupes



15 min

Coder pour que l'Arduino reconnaisse le capteur et pour conditionner l'état des LEDs aux mesures du capteur



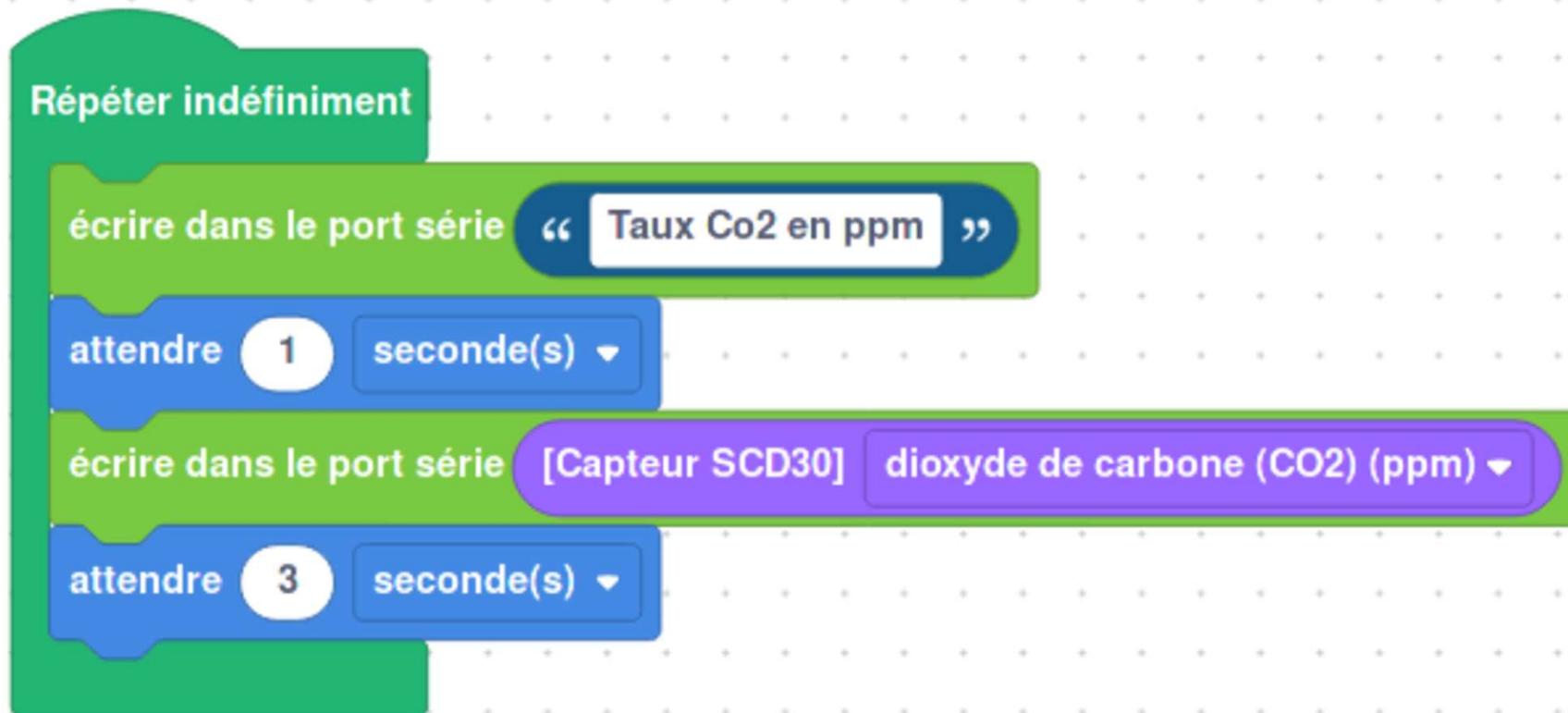


The image shows a Scratch code editor with the following blocks:

- Au démarrage** (When green flag clicked):
  - écrire dans la console** " Taux de CO2 en ppm "
  - attendre** 1 seconde(s)
  - écrire dans la console** [Capteur SCD30] dioxyde de carbone (CO2) (ppm)
  - attendre** 1 seconde(s)
- Répéter indéfiniment** (Repeat forever loop):
  - écrire dans la console** " Taux de CO2 en ppm "
  - attendre** 1 seconde(s)
  - écrire dans la console** [Capteur SCD30] dioxyde de carbone (CO2) (ppm)
  - attendre** 1 seconde(s)

A red box highlights the "Capteur SCD30" and "dioxyde de carbone (CO2) (ppm)" options in the second "écrire dans la console" block of the "Répéter indéfiniment" loop. A red arrow points from this box to the explanatory text below.

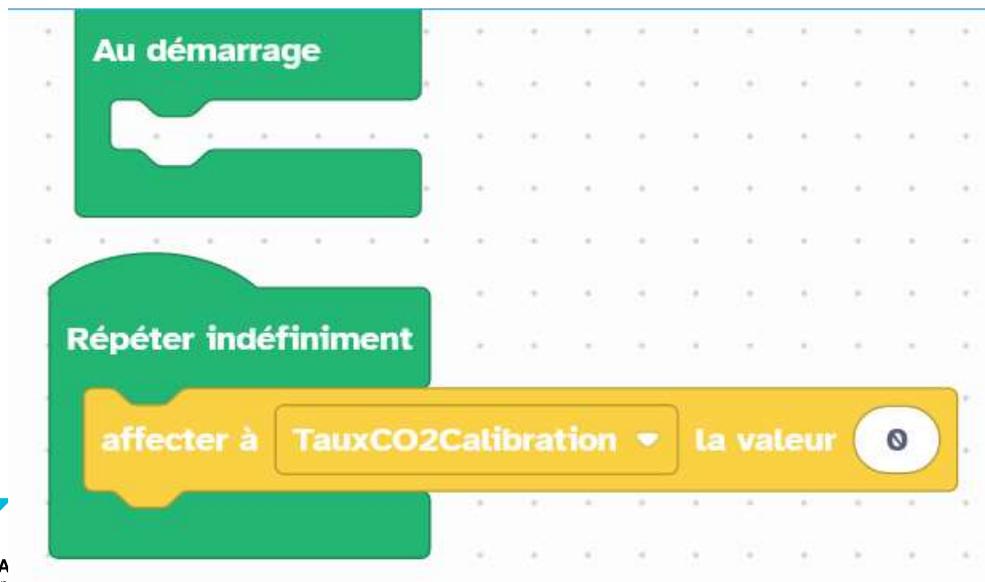
Vittascience appelle automatiquement la bonne librairie (Adafruit, qui permet la calibration) contenant tout le nécessaire pour prendre en charge le capteur.



The image shows a Scratch code block titled "Répéter indéfiniment" (Repeat indefinitely). It contains four sub-blocks: 1. "écrire dans le port série" (write to serial port) with the text "Taux Co2 en ppm". 2. "attendre" (wait) for 1 "seconde(s)". 3. "écrire dans le port série" with the sensor "[Capteur SCD30]" and the variable "dioxyde de carbone (CO2) (ppm)". 4. "attendre" for 3 "seconde(s)".

```
Scratch code block: Répéter indéfiniment  
- écrire dans le port série "Taux Co2 en ppm"  
- attendre 1 seconde(s)  
- écrire dans le port série [Capteur SCD30] dioxyde de carbone (CO2) (ppm)  
- attendre 3 seconde(s)
```

1. Laisser le capteur 10 minutes à l'air libre
2. Création de la variable calibrée  
( = différence entre la valeur mesurée et la valeur prévue de 400 ppm)



On veut avoir un témoin lumineux indiquant si le taux de CO2 mesuré est :

- Bon ( < 800ppm)
- Moyen (entre 800 et 1500 ppm)
- Mauvais (> 1500 ppm)

=> boucle du type « **si sinon si sinon** ». Pour faire apparaître cette boucle en entier, on sélectionne une **boucle « si sinon »** et on **clique sur le « + »** situé en bas de la boucle.

Ajouter le bloc pour la lumière verte

The image shows a Scratch code editor with a 'Répéter indéfiniment' (Repeat indefinitely) loop. The code includes:

- An 'Au démarrage' (When green flag clicked) block.
- An 'affecter à' (set) block: 'TauxCO2Calibration' la valeur 0.
- An 'écrire dans la console' (log) block: '[Capteur SCD30] dioxyde de carbone (CO2) (ppm) + TauxCO2Calibration +
- A 'si' (if) block with condition: '[Capteur SCD30] dioxyde de carbone (CO2) (ppm) + TauxCO2Calibration < 800'.
- A 'si' (if) block with condition: '[Capteur SCD30] dioxyde de carbone (CO2) (ppm) + TauxCO2Calibration ≥ 800 et [Capteur SCD30] dioxyde de carbone (CO2) (ppm) + TauxCO2Calibration ≤ 1500'.
- A 'sinon' (else) block.

A black box highlights the second 'si' block. An arrow points from the text 'Ajouter le bloc pour la lumière verte' to the right side of the code editor.

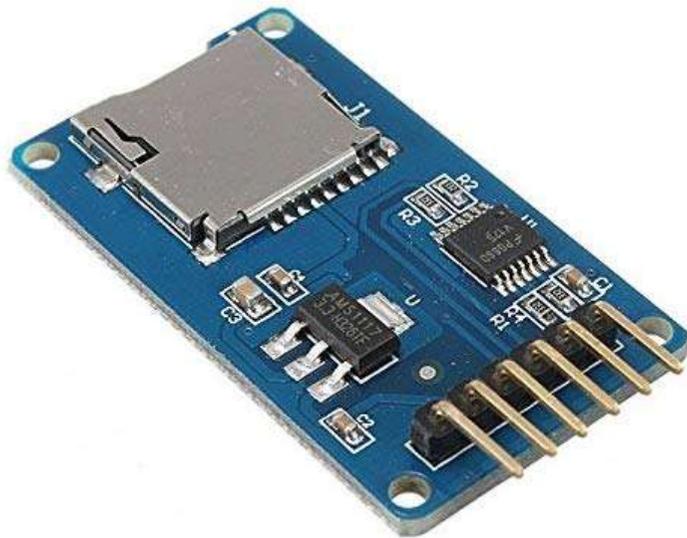
The image shows a Scratch script on a grid background. It starts with a green 'Au démarrage' block. Below it is a green 'Répéter indéfiniment' loop. The first block in the loop is yellow: 'affecter à TauxCO2Calibration la valeur 0'. The second is green: 'écrire dans la console [Capteur SCD30] dioxyde de carbone (CO2) (ppm) + TauxCO2Calibration'. The third is an orange 'si' block: '[Capteur SCD30] dioxyde de carbone (CO2) (ppm) + TauxCO2Calibration <= 800 alors'. The fourth is another orange 'sinon si' block: '[Capteur SCD30] dioxyde de carbone (CO2) (ppm) + TauxCO2Calibration > 1500 alors'. The fifth is an orange 'sinon' block. A black arrow points from the text 'Ajouter le bloc pour la lumière verte' to the 'sinon si' block.

Ajouter le bloc pour la lumière verte

The code is a Scratch script for a microcontroller. It starts with a 'Répéter indéfiniment' (Repeat indefinitely) loop. Inside the loop, the first block is 'fixer TauxCo2Calibration à 0'. The second block is 'écrire dans le port série [Capteur SGP30] gaz Dioxyde de carbone (CO2) (ppm) + TauxCo2Calibration'. This is followed by a conditional 'si' block: 'si [Capteur SGP30] gaz Dioxyde de carbone (CO2) (ppm) + TauxCo2Calibration ≤ 800 alors'. The 'alors' branch contains three 'écrire sur la broche numérique' blocks: D2 to HAUT, D3 to BAS, and D4 to BAS. The 'sinon si' branch is: 'sinon si [Capteur SGP30] gaz Dioxyde de carbone (CO2) (ppm) + TauxCo2Calibration > 800 et [Capteur SGP30] gaz Dioxyde de carbone (CO2) (ppm) + TauxCo2Calibration ≤ 1500'. This branch contains three 'écrire sur la broche numérique' blocks: D2 to BAS, D3 to HAUT, and D4 to BAS. The final 'sinon' branch contains three 'écrire sur la broche numérique' blocks: D2 to BAS, D3 to BAS, and D4 to HAUT. The script ends with an 'attendre 2 seconde(s)' block.







Enregistrement des données sur une carte micro SD

=> Lecteur de carte microSD (module SD SPI à 4,90 euros)

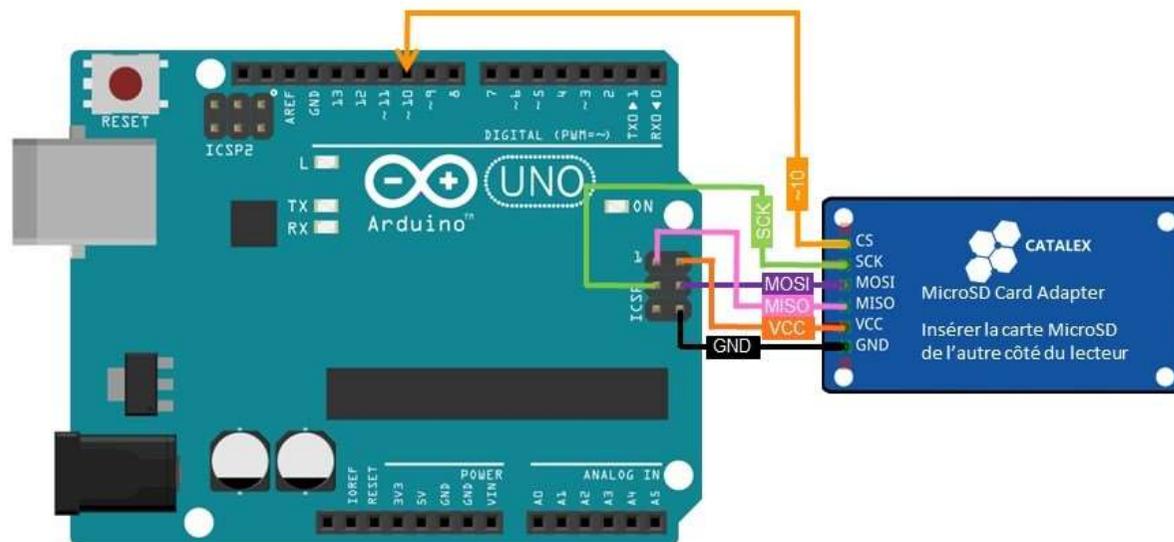
# POUR ALLER PLUS LOIN : BRANCHEMENT DU LECTEUR DE CARTE MICROSD...

39

 La couleur des câbles n'a pas d'incidence sur le montage.

 Câble femelle-mâle

 Câble femelle-femelle





## **Pour récupérer les données :**

1. Enlever la carte microSD de l'Arduino,
2. La connecter à un ordinateur (adaptateur microSD/USB ou microSD/SD)
3. Ouvrir le fichier données.txt