

CARTE A MICROCONTROLEUR DE TYPE ARDUINO



<https://www.arduino-france.com/>

A Introduction

➤ Vidéos de présentation :

vidéo 1 (3 min) : Capteurs et Microcontrôleurs - Florence Raffin

https://www.youtube.com/watch?v=RcHwDMV9fEA&list=UUHYqoNtWtk6sR1BUXb_sERQ&index=98

vidéo 2 (10 min) : Qu'est-ce qu'Arduino et puis-je l'utiliser pour mon projet ?

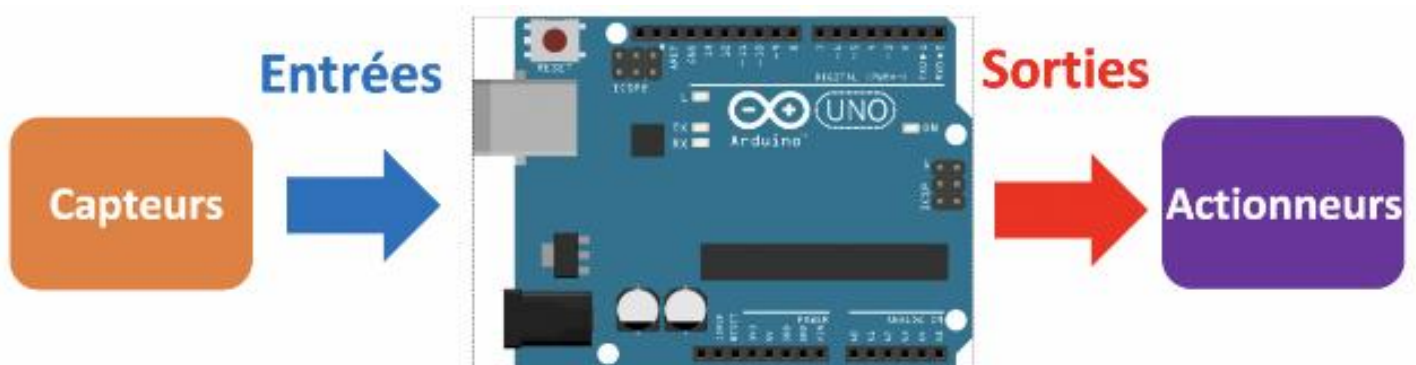
<https://www.youtube.com/watch?v=CSx6k-zXILE>

➤ Propriétés générales :

Un microcontrôleur est un circuit intégré de faible dimension qui rassemble les éléments essentiels nécessaires au fonctionnement d'un ordinateur : processeur, mémoires (ROM et RAM), unités périphériques et interfaces d'entrées-sorties.

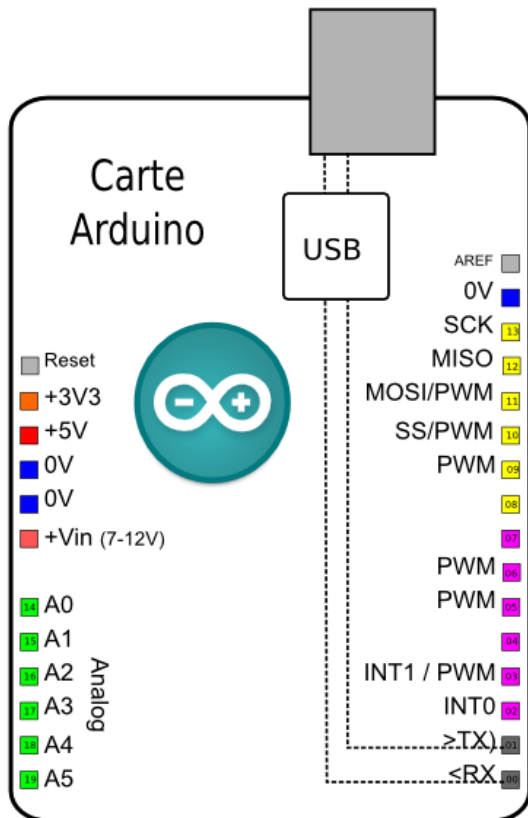
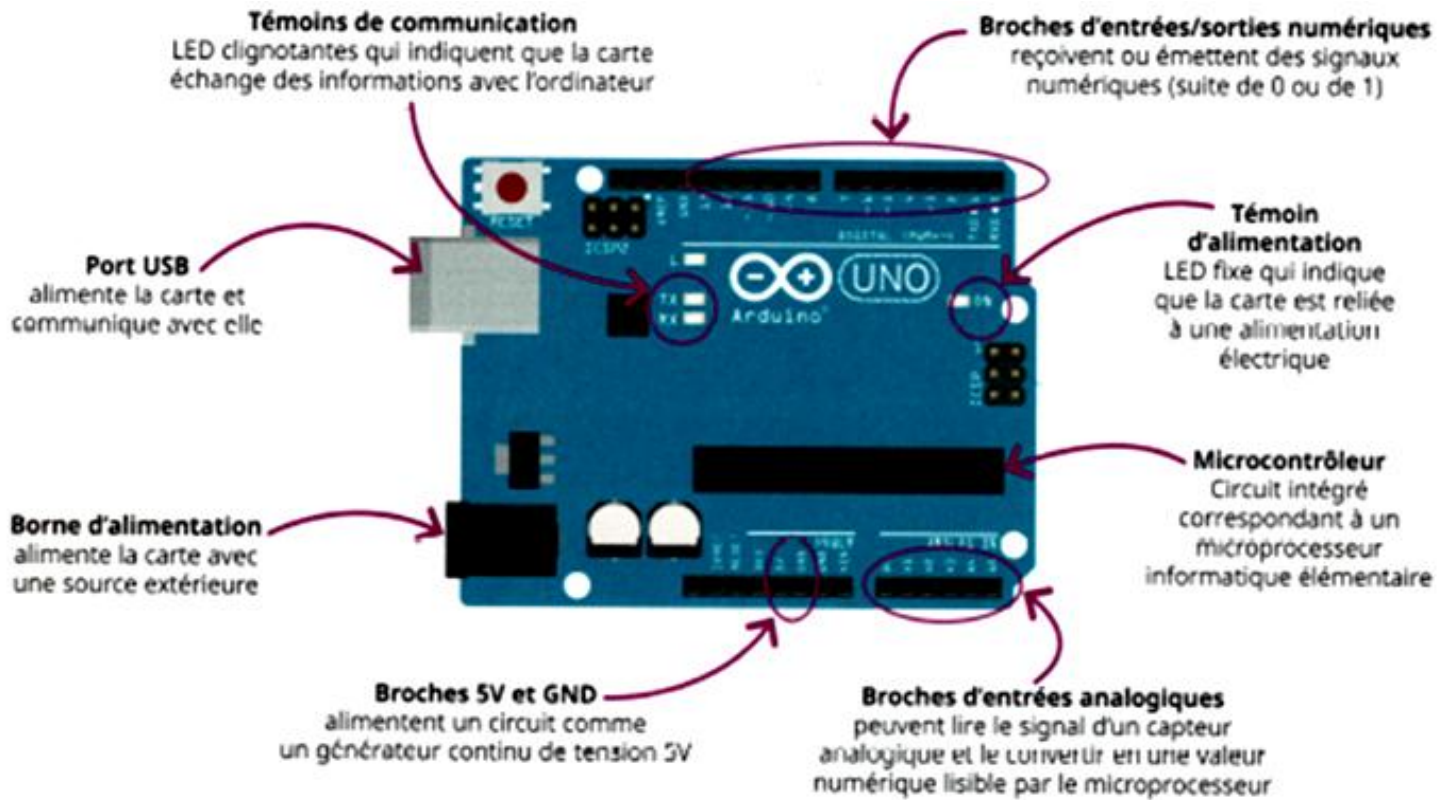
Les microcontrôleurs sont fréquemment utilisés dans des systèmes embarqués, en robotique et en domotique. Relié à différents types de capteurs, un microcontrôleur peut être programmé afin de produire des signaux électriques ou acquérir des mesures de grandeurs physiques afin de mieux comprendre les lois régissant un phénomène. Les cartes à microcontrôleur de type ARDUINO™ sont les plus populaires en raison de leur robustesse et de leur faible coût.

Un microcontrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs. Une carte Arduino™ est donc une interface programmable.



➤ Carte Arduino Uno :

Les principaux éléments suivants sont présents sur une carte Arduino Uno :



Broches numériques :

14 entrées/sorties (ou ports) numériques dont 6 sont utilisables en sortie pseudo-analogique grâce au PWM. Ce sont celles qui ont le symbole ~.

Les pins 0 et 1 sont utilisés par la liaison série synchrone (0 sur RX et 1 sur TX).

La sortie 13 est couplée à une LED sur la carte Arduino. Chaque broche numérique ne peut fournir ou absorber qu'une intensité maximale de 40mA.

Les 6 entrées analogiques (A0 à A5) :

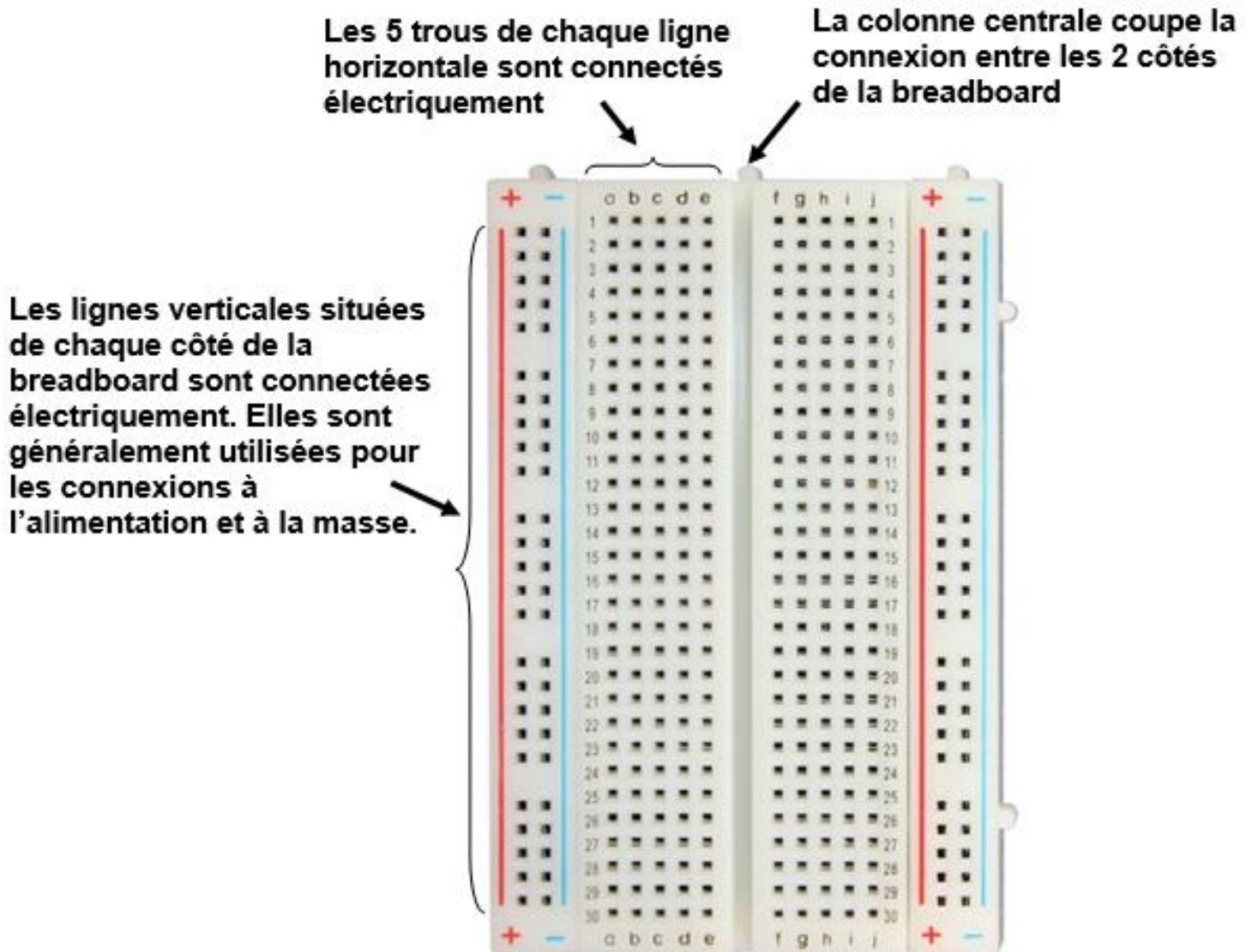
La tension d'entrée doit être inférieure à 5V. La conversion en signaux logiques destinés au microcontrôleur est assurée par un convertisseur analogique-numérique (CAN).

EN PRATIQUE, C'EST SIMPLE D'UTILISATION !!



➤ Connectique entre la carte Arduino et les autres éléments du montage :

Usuellement, on utilise une petite plaquette, appelée **breadboard**, sur laquelle on place le capteur pour ensuite le connecter par des fils à la carte Arduino, ou bien sur laquelle on réalise un circuit plus complet (dipôles et capteur) que l'on connecte ensuite à la carte Arduino.



➤ Autre présentation possible de la carte Arduino - connecteur de type « grove » :

<https://www.gotronic.fr/blog/dossiers/grove/>

Qu'est-ce que c'est ?

Grove est un système de connecteur plug-and-play open-source développé par Seeed Studio. Les modules Grove ont été conçus pour l'éducation et le prototypage rapide. Grove propose une large gamme de capteurs et d'actionneurs. Les modules sont enfichables sans soudure.

Quel intérêt ?

Vos montages seront bien plus simples, sûrs et rapides à faire.
Simple : car vous n'avez pas besoin de réfléchir à « quel fil va où ? »
Sûr : car il y a un détrompeur pour éviter de vous tromper avec la polarité
Rapide : car vous n'avez aucune soudure à faire. Branchez, codez, alimentez et c'est fait ! De plus, le système peut accueillir d'autres cartes de connexion



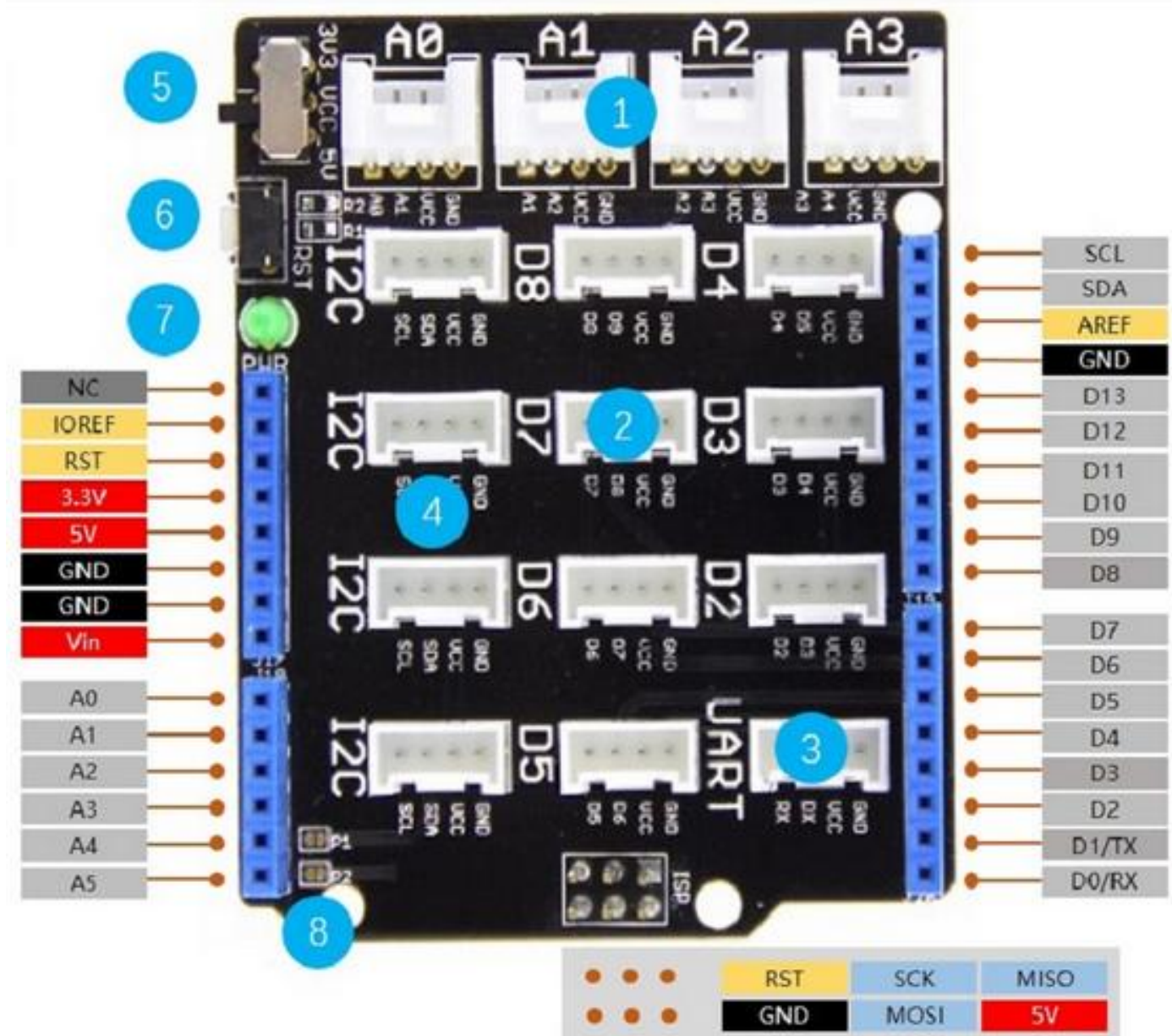
Pour qui et pour quoi ?

Débutant ou expert, le système Grove est accessible à toutes et à tous, et quel que soit le projet.

La documentation de Seeed Studio est très détaillée, complète, et comporte des exemples de code pour vous aider à démarrer vos projets. Il existe plus de 300 références de composants avec le système Grove.

plaquette disponible au lycée :

https://wiki.seeedstudio.com/Base_Shield_V2/



- **1-Analog Ports:** include 4 analog ports, A0, A1, A2 and A3.
- **2-Digital Ports:** include 7 digital ports, D2, D3, D4, D5, D6, D7 and D8.
- **3-UART Port:** 1 UART port.
- **4-I2C Ports:** 4 I2C ports.
- **5-Power Switch:** when using Arduino UNO with Base Shield v2, please turn the switch to 5v position; While using Seeeduno Arch with Base Shield v2, please turn the switch to 3.3v.
- **6-Reset Button:** reset the arduino board.
- **7-PWR LED:** The Green LED turns on when power on.
- **8-P1, P2:** please solder the pads P1 and P2, if use Base Shield v2 with Seeeduno V3.
- **Dimension:** 2.1 * 2.7 inch

C Exemple d'utilisation dans le cadre du T.I.P.E.

1°) déroulé opérationnel

➤ 1^{ère} étape : le capteur

_ analyse de l'expérience envisagée afin de déterminer le type de capteur (capteur de température, ou de distance, ..., accéléromètre, ...) et ses caractéristiques souhaitées (domaine d'utilisation, précision, temps de réponse ...)

_ choix du capteur parmi les capteurs disponibles au lycée ou choix dans une liste de capteurs disponibles facilement (et à bas coût le plus souvent) dans un site marchand

_ étalonnage du capteur si besoin (manipulation préliminaire)

remarque : la courbe d'étalonnage du capteur est parfois fournie sur le site marchand.



➤ 2^{ème} étape : le codage (« sketch »)

_ codage des instructions pour commander le microcontrôleur (langage C++ pour carte Arduino) et enregistrement (« versement ») de ce code sur la microcontrôleur.

remarque : les lignes de code souvent fournies sur le site marchand du capteur (et il n'y a plus, éventuellement, qu'à les adapter un petit peu à la situation expérimentale que l'on étudie).



➤ 3^{ème} étape : la manipulation

_ câblage du capteur au microcontrôleur

remarque : le câblage est souvent fourni dans la notice du capteur ou sur le site marchand.



2°) exemple d'illustration

On décide par exemple d'utiliser un accéléromètre pour étudier les vibrations d'un objet.













On recherche d'abord la liste des capteurs disponibles dans un site marchand, par exemple ici le site de **Gotronic** : (ou dans le site Seed Studio pour une connectique grove <https://wiki.seeedstudio.com/>) :

<https://www.gotronic.fr/cat-capteurs-1114.htm>

Cartes programmables, prototypage, capteurs






Capteurs

- Kits de capteurs
- Accéléromètres et capteurs spatiaux
 - Accéléromètres
 - Boussoles
 - Gyroscopes
 - Modules 6 degrés de liberté
 - Modules 9 degrés de liberté
 - Modules 10 degrés de liberté et plus
- Air
- Biométrie
- Caméras
- Caméras thermiques
- Chocs et vibrations
- Codes-barres et QR codes
- Conductivité
- Couleur
- Courant
- Débitmètres
- Déplacement
- Distance
- Effet Hall
- Encodeurs rotatifs
- Environnement
- Fin de course

CAPTEURS			
 KITS DE CAPTEURS	 ACCÉLÉROMÈTRES ET CAPTEURS SPATIAUX	 AIR	 BIOMÉTRIQUES
 CAMÉRAS	 CAMÉRAS THERMIQUES	 CHOC ET VIBRATIONS	 CODES-BARRES ET QR CODES
 CONDUCTIVITÉ	 COULEUR	 COURANT	 DÉBITMÈTRES

... puis, parmi les différents accéléromètres disponibles : <https://www.gotronic.fr/cat-accelerometres-1985.htm>

(et bien d'autres encore)

	Accéléromètre 3 axes MMA8452Q pour Raspberry Pi et Arduino Module à faible consommation basé sur un accéléromètre MMA8452Q permettant la mesure d'accélérations sur 3 axes jusqu'à 8 g. Ce module communique avec un microcontrôleur Raspberry Pi ou Arduino via le bus I2C. Code: 36680 3,83 € HT 4,60 € TTC Ajouter au panier En stock  Quantité <input type="text" value="1"/>
	Circuit accéléromètre MMA7361LC Ce circuit accéléromètre 3 axes MMA7361LC offre une plage de mesure entre 1,5 g et 6 g. La sortie est proportionnelle à la valeur mesurée. Code: 33486 3,67 € HT 4,40 € TTC Ajouter au panier En stock  Quantité <input type="text" value="1"/>
	Accéléromètre 3 axes BMA220 SEN0168 Module accéléromètre Gravity 3 axes basé sur le BMA220 de Bosch permettant de mesurer les accélérations jusqu'à ±16 g. Il communique avec un microcontrôleur compatible Arduino® via le bus I2C. Code: 33724 4,33 € HT 5,20 € TTC

... on choisit (par exemple) le 1^{er} accéléromètre de la liste :

<https://www.gotronic.fr/art-accelerometre-3-axes-mma8452q-31493.htm>

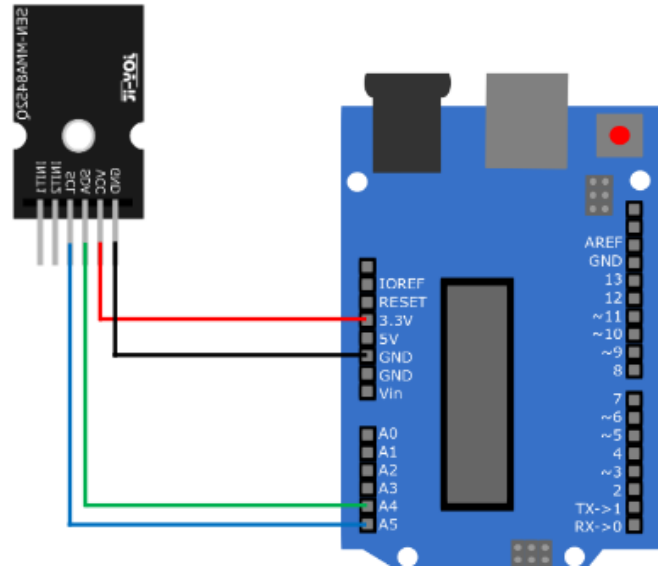
	Accéléromètre 3 axes MMA8452Q pour Raspberry Pi et Arduino Code article : 36680 Module à faible consommation basé sur un accéléromètre MMA8452Q permettant la mesure d'accélérations sur 3 axes jusqu'à 8 g. Ce module communique avec un microcontrôleur Raspberry Pi ou Arduino via le bus I2C. > Description complète Quantité : <input type="text" value="1"/>   ✓ Quantité en stock : 53 🚚 Livraison à partir de 2,90€  3,83 € HT 4,60 € TTC   dont 0,02 € d'éco-part
DESCRIPTION FICHE TECHNIQUE AVIS	
<p>Module à faible consommation basé sur un accéléromètre MMA8452Q permettant la mesure d'accélérations sur 3 axes jusqu'à 8 g. Ce module communique avec un microcontrôleur Raspberry Pi ou Arduino via le bus I2C.</p> <p>Ce module est livré sans cordons de raccordement, voir BB15 pour Raspberry Pi et BB121 pour Arduino. Ce capteur peut également être enfilé dans une plaque de montage rapide.</p> <p>Caractéristiques:</p> <ul style="list-style-type: none">• Alimentation: 1,95 à 3,6 Vcc• Consommation: 6 à 165 µA• Sortie digitale 8 et 12 bits• Plage de mesure: ±2 g, 4 g ou ±8 g• Fréquence de sortie: 1,56 à 800 Hz• Connecteurs: 6 broches mâles au pas de 2,54 mm• Dimensions: 35 x 20 x 7 mm <p>Référence Joy-It: SEN-MMA8452Q</p>	

... pour lequel on dispose en plus de **nombreuses informations** dans le guide d'utilisation : <https://joy-it.net/files/files/Produkte/SEN-MMA8452Q/SEN-MMA8452Q-Manual-20210111.pdf>



→ a) le **câblage** préconisé du capteur avec la carte Arduino :

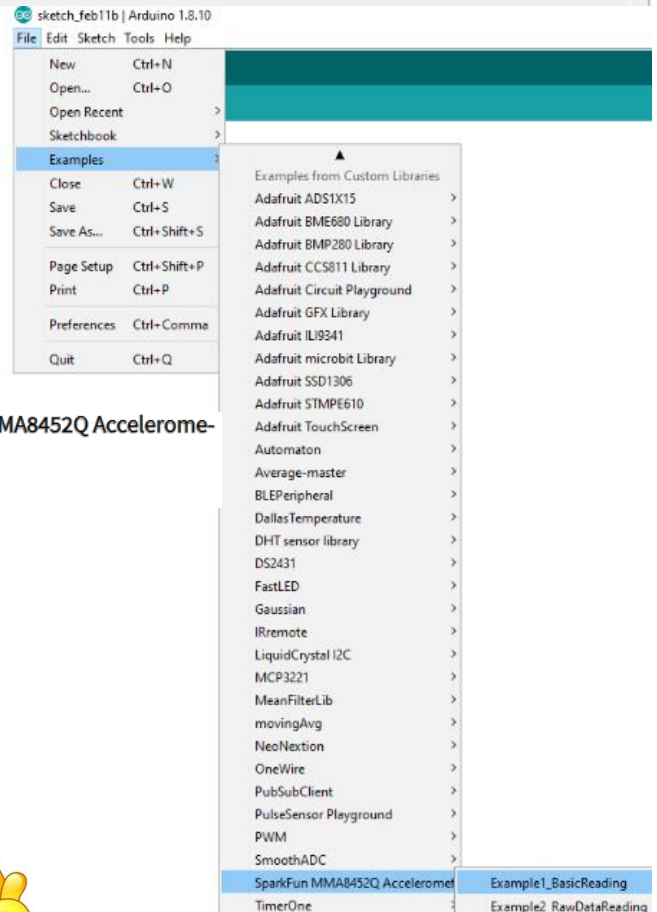
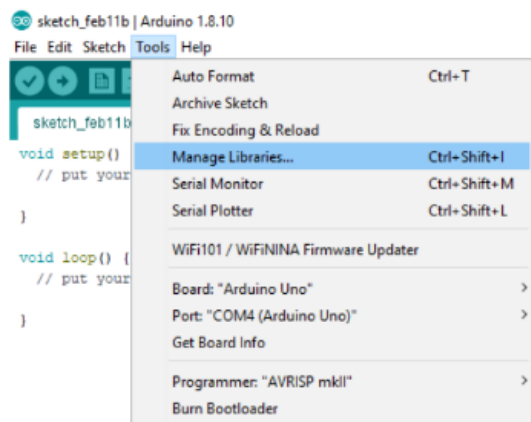
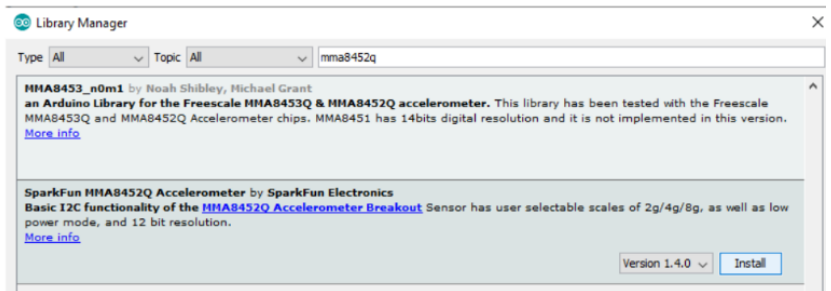
ARD-Uno	MMA8452Q
3.3V	VCC
GND	GND
A4 (SDA)	SDA
A5 (SCL)	SCL



→ b) le **code de commande** de la carte Arduino, à copier ou à télécharger directement sur la carte, ainsi qu'une éventuelle **librairie** de fonctions particulières permettant d'utiliser le capteur avec la carte Arduino (même principe que le téléchargement en en-tête de programme python de certaines bibliothèques optionnelles) :

→ téléchargement d'une librairie :

To use the sensor with the Arduino, you must first install a library.
Go to Tools -> Manage libraries...



→ Téléchargement d'un exemple de code :

You can now select one of the example files under File -> Examples -> SparkFun MMA8452Q Accelerometer and upload it to your Arduino.

Go to Tools and make sure you selected the correct board and port.

... et c'est parti :-)

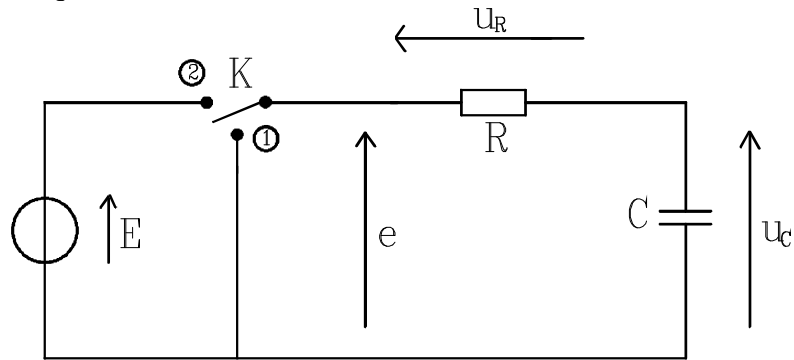


C Exemple d'utilisation dans le cadre d'un T.P.

On désire déterminer la constante de temps de charge d'un condensateur de capacité C à travers une résistance R . Pour cela, on va réaliser le montage sur la **breadboard**, l'alimentation du circuit ainsi que le suivi de la tension aux bornes du condensateur étant assuré par la carte Arduino.

1°) montage théorique

On reprend le montage classique étudié en cours :



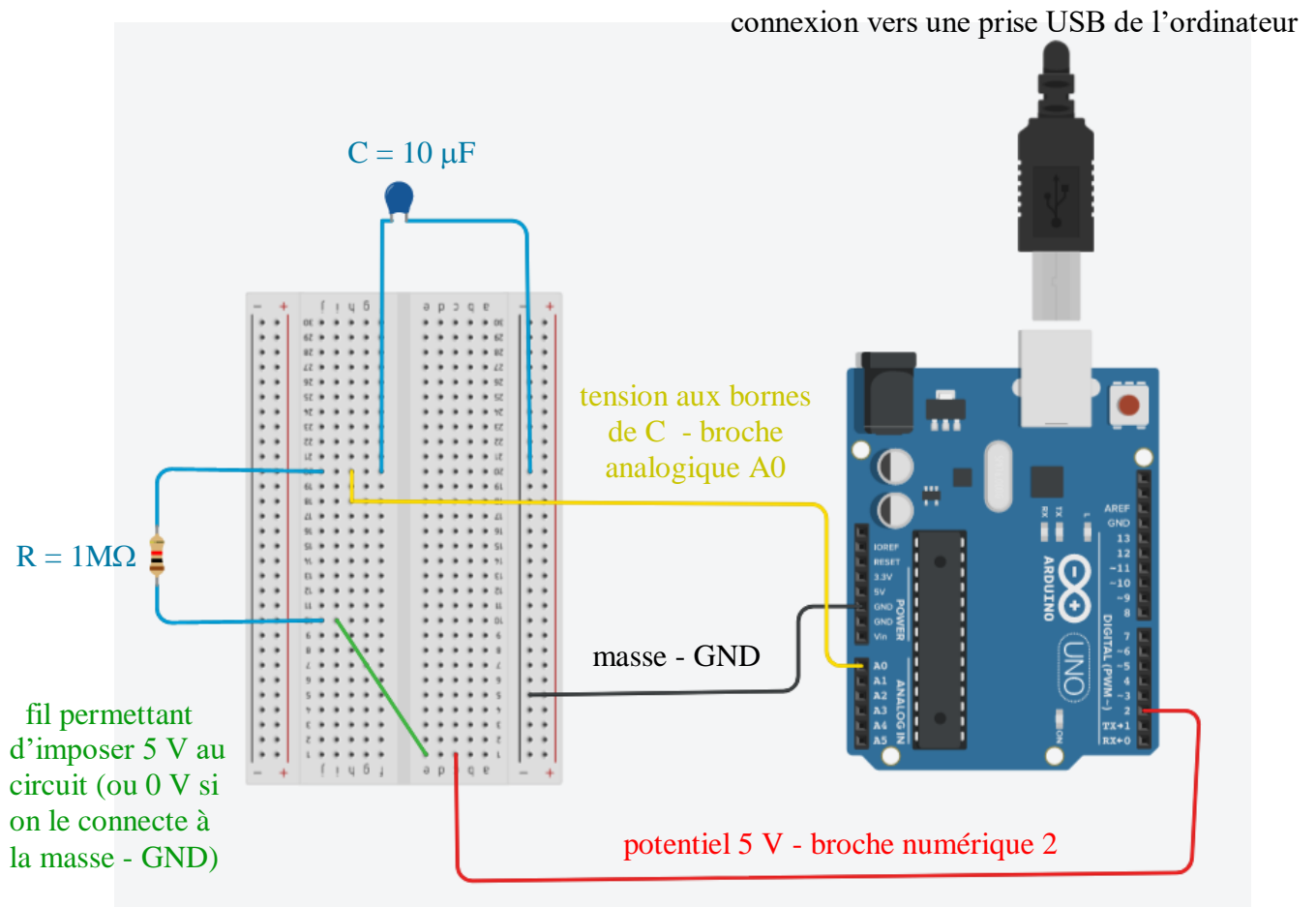
2°) montage avec la carte Arduino

a) simulation avec Tinkercad : <https://www.tinkercad.com/>

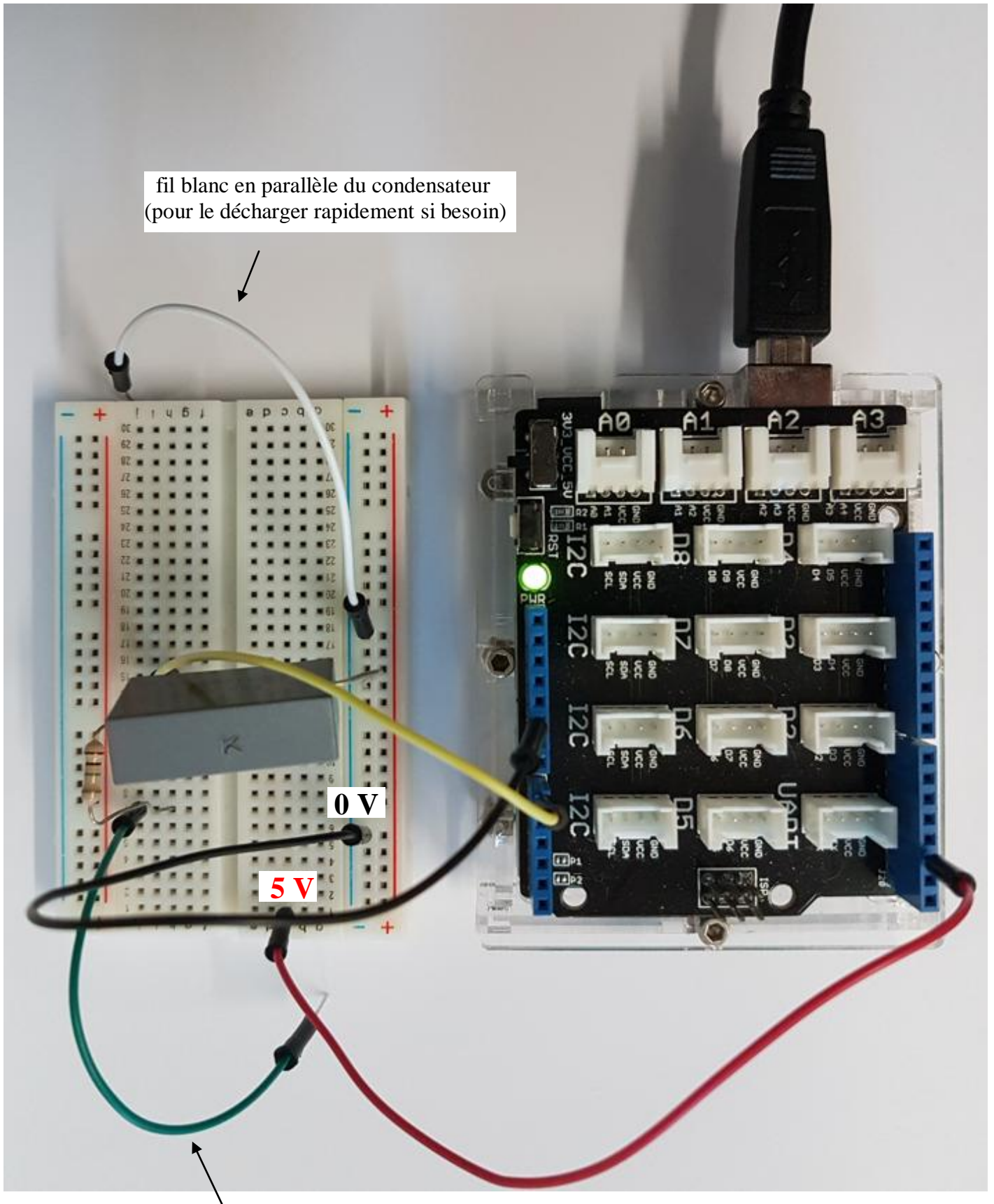
Le microcontrôleur est utilisé, d'une part pour alimenter le circuit RC série sous 5 V (broche numérique 2) et d'autre part pour relever mesurer la tension aux bornes du condensateur (broche analogique A0).

Un fil (non représenté ci-dessous) placé en parallèle du condensateur pourra permettre une décharge instantanée avant de mesurer la durée de la charge du condensateur.

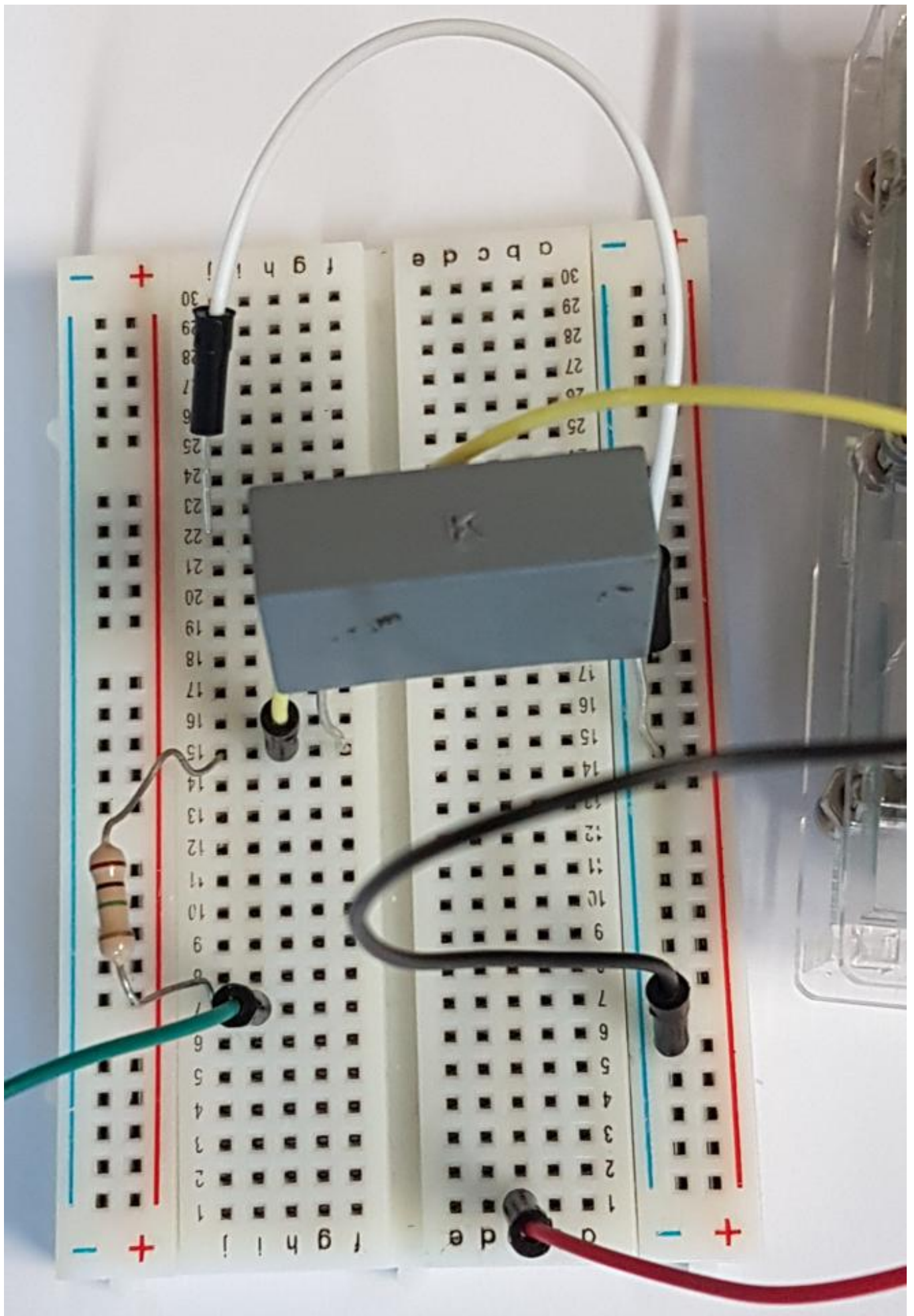
Une fois le condensateur déchargé, la sortie numérique 2 de la carte passe de l'état bas à l'état haut et alimente le circuit série sous une tension de 5,0 V.



b) quelques photos du montage réalisé :



fil vert permettant d'imposer **5 V** au circuit (si connecté à la broche numérique 2 - fil rouge) ou **0 V** (si connecté à la masse GND - fil noir)



2°) script Arduino

→ fichier : « **script_Arduino__charge_du_condensateur.ino** »

script_Arduino__charge_du_condensateur | Arduino 1.8.16

Fichier Édition Croquis Outils Aide



script_Arduino__charge_du_condensateur

```
// Définition de l'entrée numérique utilisée
const int alimentation = 2; // La broche numérique 2 alimente le circuit RC

// Définition de l'entrée analogique utilisée et des variables décimales
int Lecture = A0; // La broche analogique A0 permet la mesure de la tension aux bornes du condensateur
float temps; // "temps" est une variable décimale

// Définition et mise à zéro des durées mesurées et intervenant dans le calcul de "temps"
unsigned long tempsDebut = 0; // unsigned long : variable de type long non signé
unsigned long tempsFin = 0;

// BOUCLE D'INITIALISATION : exécutée une seule fois
void setup() {
  Serial.begin(9600); // Ouvre le port série et fixe le débit de communication à 9600 bauds
  pinMode(alimentation, OUTPUT); // "alimentation" est une broche de sortie
  digitalWrite(alimentation, LOW); // A l'état initial "alimentation" est à l'état BAS
  pinMode(Lecture, INPUT); // "Lecture" est une broche d'entrée
}


// BOUCLE PRINCIPALE : exécutée à l'infini
void loop() {
  digitalWrite(alimentation, LOW);
  Serial.println("Décharge du condensateur");
  Serial.println("Appuyez le bouton poussoir de court-circuit si la décharge trop longue");
  while (analogRead(Lecture) > 0) { // On s'assure que le condensateur se décharge
    float Vmesuree = analogRead(Lecture);
    float Vc = (Vmesuree/1023)*5; // On convertit "Vmesuree" en une tension comprise entre 0 et 5V
    Serial.print("La tension aux bornes du condensateur est égale à : ");
    Serial.print(Vc); // On lit la tension aux bornes du condensateur (valeur entre 0 et 1023)
    Serial.println(" V");
    delay(15000); // Temporisation de 15 000 ms soit 15 s
  }
  Serial.println("Etude de la charge du condensateur dans 10 secondes");
  Serial.println("Relâchez l'interrupteur de court-circuit");
  delay(15000); // Temporisation de 15 000 ms soit 15 s
  digitalWrite(alimentation, HIGH); // On alimente le circuit RC avec une tension de 5 V
  tempsDebut = micros(); // Mémorisation du temps au début de l'alimentation du circuit RC
  Serial.print("Vc");
  Serial.print("\t"); // Ajout d'un espace
  Serial.println("Temps");
  Serial.print("V");
  Serial.print("\t");
  Serial.println("s");
  while (analogRead(Lecture) < 1023) { // Tant que le niveau lu est inférieur à 1023
    tempsFin = micros(); // Durée écoulée depuis le début du lancement du programme
    temps = (tempsFin - tempsDebut) * 1e-6; // Calcul de "temps" exprimé en s
    float Vmesuree = analogRead(Lecture);
    float Vc = (Vmesuree/1023)*5;
    Serial.print(Vc,2); // "Vc" avec 2 décimales
    Serial.print("\t");
    Serial.println(temps,1); // "temps" avec 1 décimale
    delay(1000); // Temporisation de 1 000 ms, soit 1 s, entre chaque mesure
  }
  Serial.println("Fin de la charge du condensateur");
  delay(20000); // Temporisation de 20 000 ms soit 20 s
}
```


3°) manipulation


→ montage :

- _ réaliser le câblage du montage comportant la résistance R, le condensateur C, la carte Arduino.
- _ fil blanc connecté de manière à court-circuiter le condensateur : cela permet d'avoir un condensateur initialement non chargé.
- _ fil vert connecté à la masse de manière à ce que la tension imposée au circuit soit égale à 0 V au départ.
- _ alimenter la carte Arduino par une connexion USB de l'ordinateur

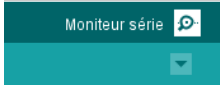
→ script Arduino :

- _ via l'icône  ou programme ...accéder au script Arduino « `script_Arduino_charge_du_condensateur.ino` » présent dans le fichier `;;;;;;;;;;;;;` (présent dans le dossier partagé PCSI)

- _ vérifier la syntaxe du script en cliquant sur l'icône 

- _ téléverser le script dans la carte en cliquant sur l'icône 

→ manipulation :

- _ suivre les instructions qui s'affichent dans le moniteur série. Cliquer pour cela sur l'icône située en haut à droite à l'affichage du script. 

- _ afin de permettre la charge du condensateur, il faudra successivement agir sur le fil blanc pour ne plus court-circuiter le condensateur (et donc permettre sa charge) puis agir sur le fil vert afin d'imposer la tension 5 V au circuit.

- _ suivre la charge du condensateur sur le moniteur série. Un tableau de valeurs (tension, temps) se complète au fur et à mesure de la charge (cf extrait ci-contre).

```
Etude de la charge du condensateur dans 10 secondes
Lachez l'interrupteur de court-circuit
Vc      Temps
V       s
0.07    0.0
0.53    1.0
0.96    2.0
1.35    3.0
1.72    4.0
2.03    5.0
```

→ récupération des valeurs :

- _ lorsque le message « *Fin de la charge du condensateur* » s'affiche, décocher le **Défilement automatique** et copier dans le presse-papier les valeurs de « Vc » et de « temps » associées à la charge (sélectionner les valeurs puis « contrôle C »).

- _ copier ces valeurs dans **Latispro** (Traitement → Tableur → Edition → Copier)

→ traitement à l'aide de Latispro :

- _ représenter la courbe de la tension en fonction du temps
- _ en déduire la constante de temps du circuit et la comparer à la valeur attendue