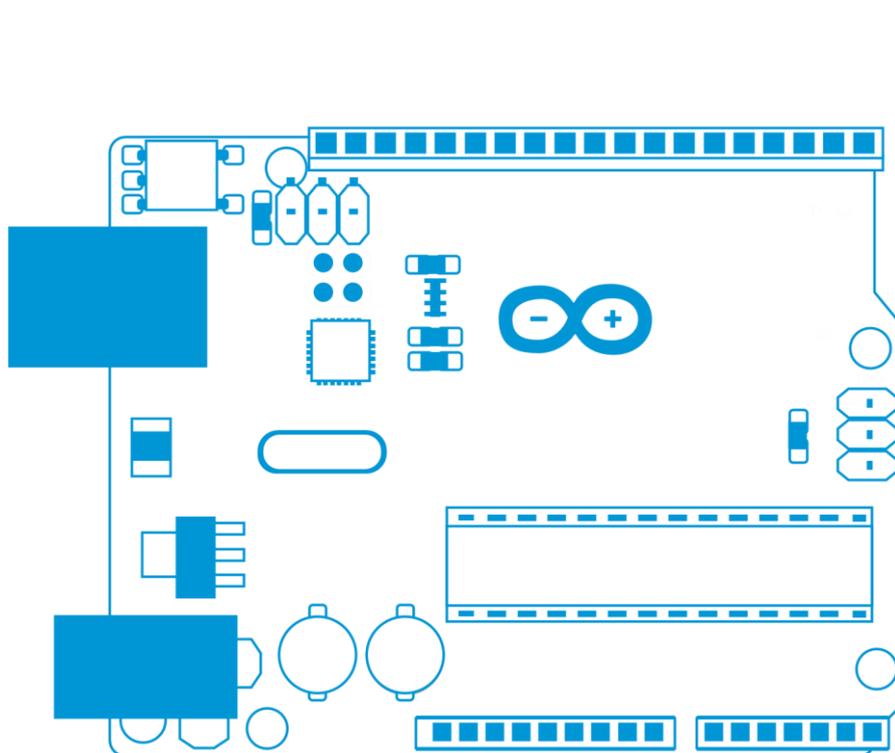
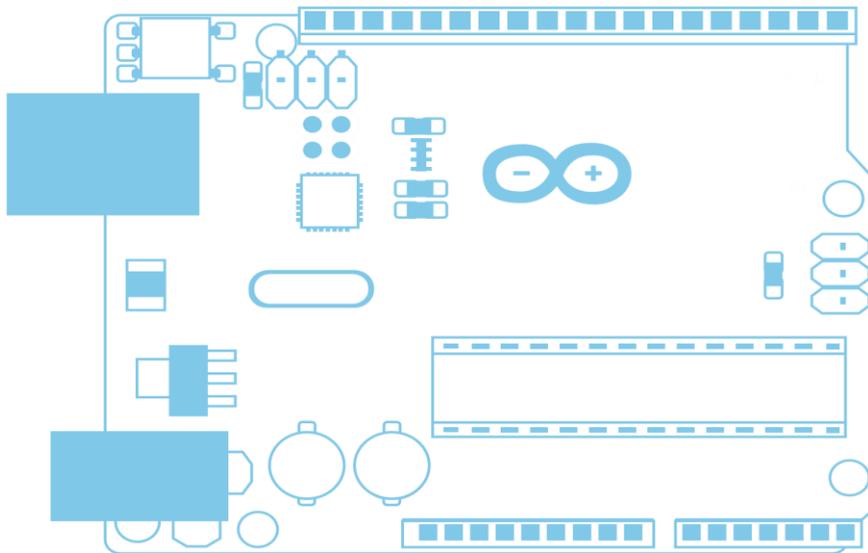
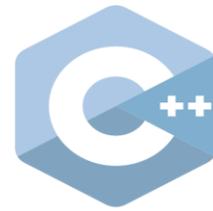


teach with space

→ Initiation à Arduino

Introduction à la programmation Arduino en C++





Eléments clés	page 3
Résumé des activités	page 4
GUIDE DU PROFESSEUR	
Activité 0: Démarrage	page 5
Activité 1: Faites clignoter votre carte !	page 8
Activité 2: S.O.S en morse !	page 11
Activité 3: Mesurer la température	page 12
Activité 4: Mesurer la pression	page 16
Activité 5: Mesurer l'altitude	page 19
Fiche élève	page 20
Liens Utiles	page 38

teach with space – Initiation à arduino | T04.1

www.esa.int/education (EN)

www.esero.fr ou <https://www.planete-sciences.org/espace/CanSat/CanSat-Lycee> (FR)

Vos commentaires et retours d'expérience sont les bienvenus, contacter :

ESA Education à education@esa.int

ESERO France à esero.france@cnes.fr

Planète Sciences sur espace@planete-sciences.org

Une production ESA Education

traduite et adaptée en français par ESERO France

Copyright 2018 © European Space Agency

Copyright 2020 © ESERO France, CNES, Planète Sciences

→ Initiation à Arduino

Introduction à la programmation Arduino en C++

Informations clés

Tranche d'âge : 14-20 ans

Liens avec le programme scolaire :
programmation, électronique

Complexité : Moyen

Temps de cours requis : 90-120 minutes

Localisation : Intérieur

Comprend l'utilisation de :

Logiciels et matériels Arduino

Mots-clés : Arduino, Capteur, Code

Résumé

Les élèves découvriront des technologies utilisées dans l'espace grâce à l'outil Arduino. Ils construiront des circuits pour faire clignoter une LED et pour mesurer la température, la pression et l'altitude. Ils découvriront les bases de la programmation en C++ à l'aide du logiciel Arduino IDE (Integrated Development Environment).

Objectifs d'apprentissage

- Améliorer les compétences analytiques
- Développer la compréhension d'un langage de programmation
- Développer la compréhension des circuits électroniques
- Comprendre comment les capteurs peuvent être utilisés pour collecter des données
- Améliorer les compétences en matière de travail d'équipe

Activités - Contenus et objectifs

	Titre	Description	Objectifs	Prérequis	Durée
1	Pour commencer	Une introduction aux Composantes utilisées dans toute l'activité	Les élèves se familiarisent avec les composants et leurs fonctions	Compréhension des bases de l'électronique	10 minutes
1	Faites clignoter une LED !	Les élèves construiront leur premier circuit électronique Arduino	Les étudiants seront capable de contrôler une LED à l'aide de la carte Arduino	Activités précédentes	15 minutes
2	S.O.S en morse	Les élèves apprendront à envoyer des signaux plus complexes avec une LED	Les élèves seront capable d'écrire un programme envoyant un S.O.S	Activités précédentes	15 minutes
3	Mesurer la température	Les élèves utiliseront une thermistance pour mesurer la température de leur environnement	Les élèves seront capables d'écrire des programmes plus complexes et d'utiliser des capteurs pour récupérer des données	Activités précédentes	20 minutes
4	Mesurer la pression	Les élèves découvriront comment utiliser un capteur de pression pour mesurer la pression de l'air	Les élèves seront capables d'écrire des programmes plus complexes et d'utiliser des capteurs pour récupérer des données	Activités précédentes	20 minutes
5	Mesurer l'altitude	Les élèves découvriront comment calculer leur altitude à partir de la pression de l'air environnant	Les élèves seront capables d'écrire des programmes plus complexes et d'utiliser des capteurs pour récupérer des données	Activités précédentes	20 minutes

Activité 0 : Démarrer

Introduction

Arduino est une plateforme open-source utilisée pour construire une grande variété de projets électroniques, y compris pour simuler des missions spatiales réelles comme ExoMars (<http://exploration.esa.int/mars/>).

Arduino intègre à la fois un circuit imprimé programmable (matériel) et un logiciel ou un IDE (Integrated Development Environment) qui fonctionne sur un ordinateur et qui permet la programmation.

Le matériel et le logiciel Arduino ont été conçus pour les artistes, les designers, les amateurs, les pirates informatiques et les personnes intéressées par la création d'objets et d'environnements interactifs. Arduino peut être utilisé pour interagir avec des boutons, des LED, des capteurs, des moteurs, des haut-parleurs, des GPS, des appareils photo, l'internet et même votre smartphone ou votre télévision ! Cette flexibilité, combinée au fait que le logiciel Arduino est gratuit, que les cartes et les capteurs sont relativement bon marché et que le logiciel et le matériel sont intuitifs, a donné naissance à une large communauté d'utilisateurs (www.arduino.cc).

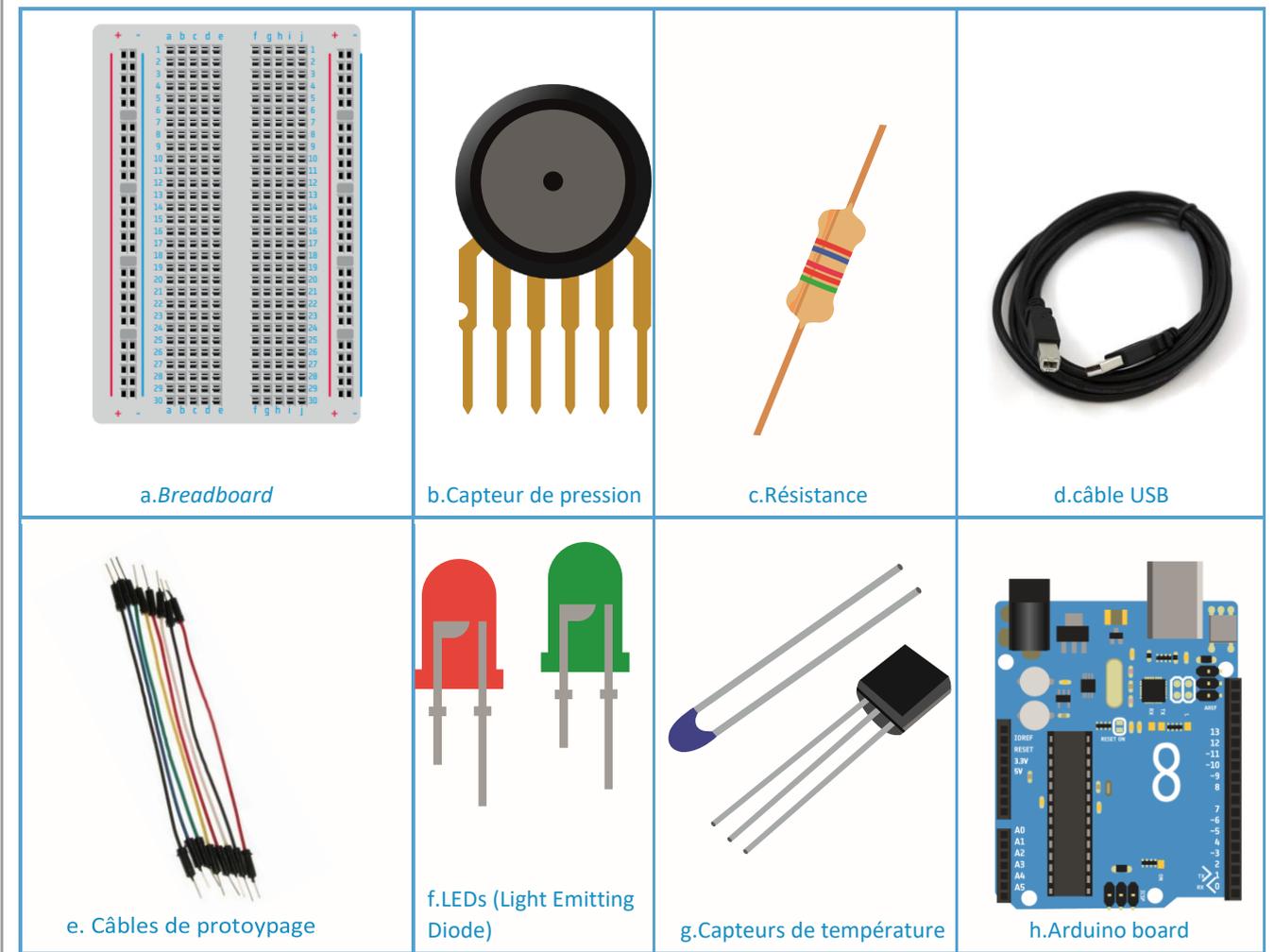
Matériel

Pour cette activité, nous allons utiliser l'Arduino Uno (Figure A1, boîte numéro h). L'Arduino peut fournir une tension à ses broches (pattes métalliques) et lire la tension à partir de celles-ci. Notez le numéro imprimé à côté de chaque broche, avec **GND** (masse) agissant comme pôle négatif et **Vin** ou **5V** agissant comme pôle positif (comme les côtés positif et négatif d'une batterie).

Nous allons utiliser les éléments suivants, illustrés dans la figure A1 :

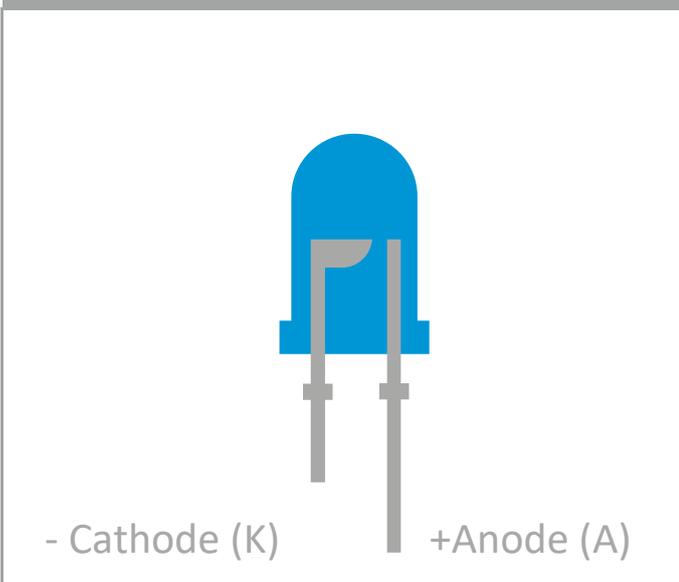
- a. Breadboard:** Une structure de support permettant de connecter facilement des composants électroniques.
- b. Capteur de pression MPX4115A:** Composant qui mesure la pression ambiante en valeur absolue
- c. Résistances de 220 Ω et 10k Ω :** Pour contrôler l'intensité. La résistance est mesurée en ohms (Ω).
- d. Câble USB :** Utilisé pour connecter l'Arduino à l'ordinateur et/ou à la source d'alimentation
- e. Câbles de prototypage :** Utilisés pour connecter électriquement des composants.
- f. LEDs (Light Emitting Diode):** Composants électroniques qui génèrent de la lumière lorsqu'ils sont traversés par de l'électricité.
- g. Capteur de température:** Un capteur analogique (comme un LM35) et une thermistance (une résistance particulièrement sensible à la température).
- h. Carte Arduino:** Fonctionne comme un simple ordinateur qui contient un processeur, de la mémoire et quelques broches d'entrée/sortie

Figure A1



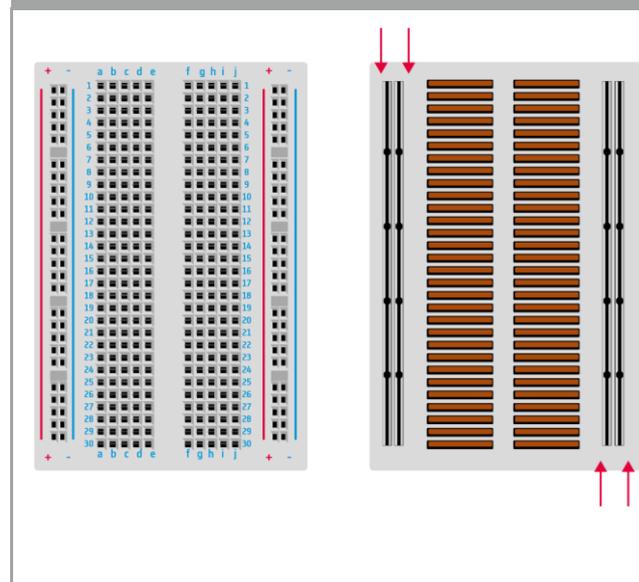
↑ Composants de base du kit Arduino

Figure A2



↑ LED : Notez que le LED a une jambe longue et une jambe courte. La longue branche est appelée anode et la courte cathode.

Figure A3



↑ LED : Notez que le LED a une jambe longue et une jambe courte. La longue branche est appelée anode et la courte cathode.

Lorsque vous connectez l'Arduino, votre ordinateur doit le reconnaître et lancer le processus d'installation du pilote. Si vous avez des problèmes pour installer les pilotes, vous pouvez trouver des instructions supplémentaires ici : www.arduino.cc/en/Guide/ArduinoUno

*Veuillez-vous assurer de sélectionner la bonne carte (essayez Genuino Uno) et le bon port série (COM) dans le menu Outils !

Software

Le logiciel Arduino peut être facilement téléchargé et installé ici : www.arduino.cc/en/Main/Software

Les programmes écrits à l'aide du logiciel Arduino (IDE) sont appelés "sketches". Ces « esquisses » sont écrites dans l'éditeur de texte et sont enregistrées avec l'extension de fichier **.ino**.

L'éditeur comporte des fonctions de couper/coller et de rechercher/remplacer du texte. La zone de message en bas (figure A4) donne des informations sur l'enregistrement et l'exportation et affiche également les erreurs. La carte et le port série sélectionnés sont affichés dans le coin inférieur droit de la fenêtre. Les boutons de la barre d'outils vous permettent de vérifier et de télécharger des programmes, de créer, d'ouvrir et d'enregistrer des croquis et d'ouvrir le moniteur série (pour visualiser les données collectées par les capteurs). La figure A4 est une capture d'écran de l'interface graphique de l'IDE Arduino.

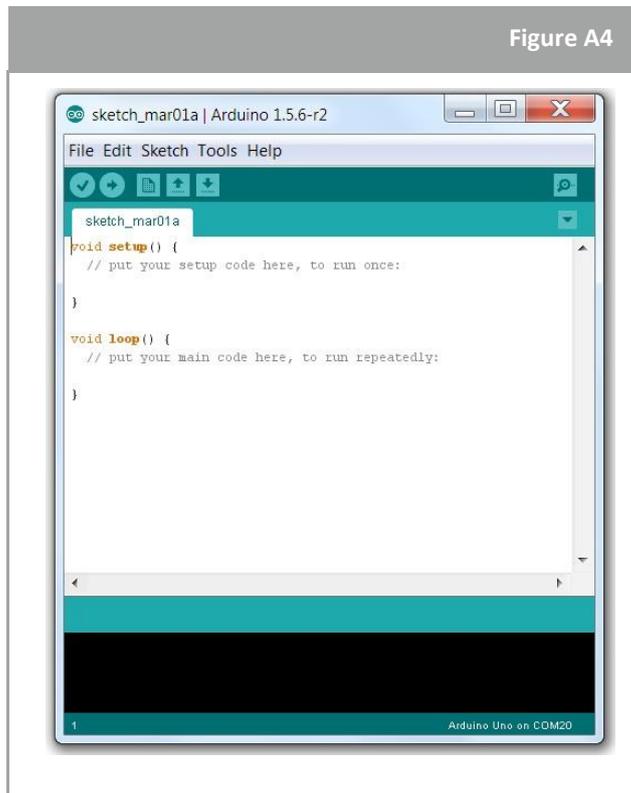


Figure A4

↑ Interface Arduino IDE

Commandes Arduino - Table A1

	Vérifiez: Vérifie que votre code ne contient pas d'erreurs lors de sa compilation.
	Téléversez: Compile le code et l'envoie sur la carte Arduino
	Nouveau: Créé un nouveau fichier de code
	Ouvrir: Ouvre un fichier
	Enregistrer : Enregistre le fichier en cours d'utilisation
	Serial Monitor : Ouvre le moniteur, vous permettant de voir les données collectées par les capteurs.

Des commandes supplémentaires se trouvent dans les cinq menus : Fichier, Édition, Croquis, Outils et Aide.

Activité 1: Faîtes clignoter votre carte !

Dans cette activité, les élèves contrôlent une LED avec l'Arduino. L'objectif est de contrôler le clignotement de cette LED et de comprendre comment le code est écrit/utilisé pour contrôler chaque LED. Vous trouverez plus d'informations en effectuant une recherche rapide sur Google : *Clignotement de la LED de l'Arduino*.

Equipment

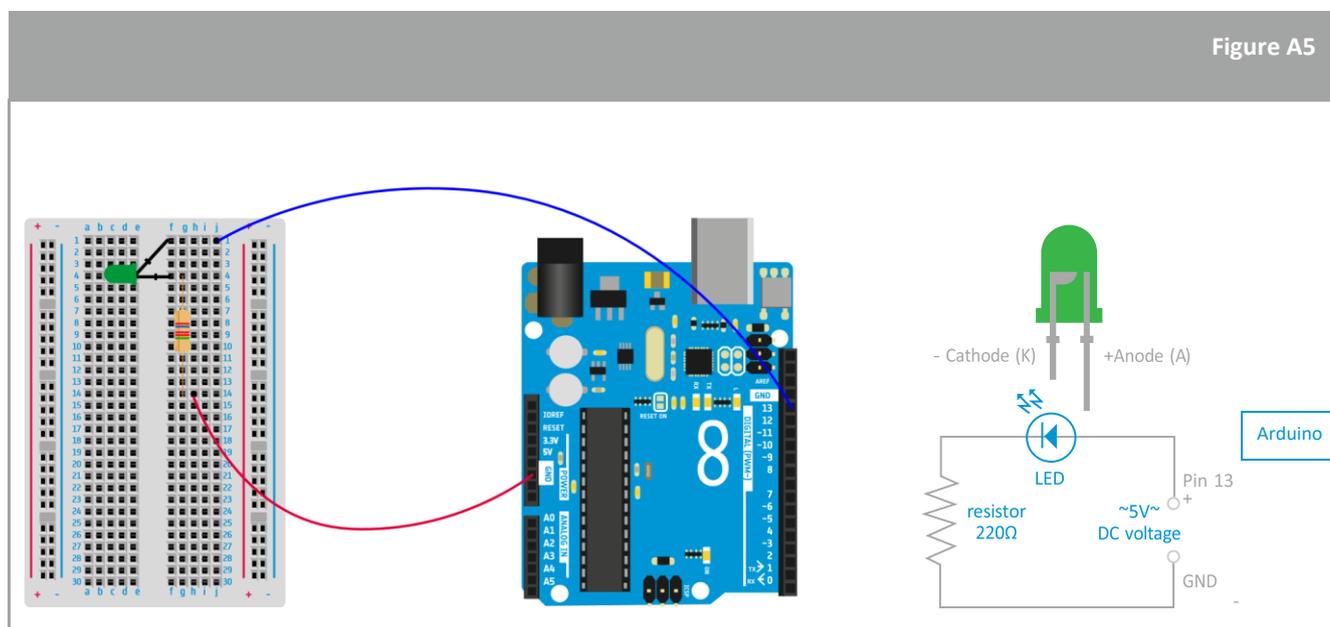
- 1 Arduino Uno
- 1 planche de prototypage
- 1 LED (vert)
- 1 résistance 220Ω (rouge/rouge/noir/noir/marron)
- 2 fils

Exercice

Configuration

Le circuit est configuré comme suit :

- Connectez la LED verte sur la *breadboard* comme indiqué sur la figure A5.
- Connectez la résistance 220Ω (rouge/rouge/noir/noir/marron) à la branche courte de la LED comme indiqué sur la figure 5.
- A l'aide d'un fil, connectez l'anode de la LED à la broche numéro 13 de l'Arduino Uno
- A l'aide d'un fil, connectez la résistance à la broche GND de l'Arduino Uno.
- Enfin, connectez l'Arduino UNO à l'ordinateur à l'aide du câble USB.



↑ Le circuit Arduino est prêt à faire clignoter la LED

Code

Les élèves vont maintenant télécharger un programme pour contrôler la LED. Ils utiliseront un exemple de code que l'on peut trouver en ouvrant le logiciel Arduino (IDE) et en cliquant sur :

Dossier → Exemples → Notions de base → Clignotant

Le code clignotant apparaîtra à l'écran et est illustré dans l'image ci-dessous. Les commentaires sont affichés en gris après les barres obliques et expliquent chaque ligne du code.

Téléversez le code sur la carte

```

1  /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
6  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN takes care
7  of use the correct LED pin whatever is the board used.
8  If you want to know what pin the on-board LED is connected to on your Arduino model, check
9  the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
10
11 This example code is in the public domain.
12
13 modified 8 May 2014
14 by Scott Fitzgerald
15
16 modified 2 Sep 2016
17 by Arturo Guadalupi
18 */
19
20
21 // the setup function runs once when you press reset or power the board
22 void setup() {
23   // initialize digital pin LED_BUILTIN as an output.
24   pinMode(LED_BUILTIN, OUTPUT);
25 }
26
27 // the loop function runs over and over again forever
28 void loop() {
29   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
30   delay(1000); // wait for a second
31   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
32   delay(1000); // wait for a second
33 }
34

```

Le texte en gris est un commentaire,
il explique ce que le code fait

LED_BUILTIN est le n° / nom du pin auquel la LED est connectée

HIGH = la LED est allumée

LOW = la LED est éteinte

1000ms est un délai d'attente, ici le
délai entre éteindre et allumer la LED

Les éléments principaux du code sont expliqués ci-dessous :

```
void setup() { ... }
```

Cette commande est toujours écrite au début d'un fichier ; elle initialise les variables qui vont être utilisées dans le reste de ce fichier. Elle ne sera exécutée qu'une seule fois, après chaque mise sous tension ou réinitialisation de la carte Arduino

```
pinMode(LED_BUILTIN, OUTPUT) ; -> pinMode(13, OUTPUT) ;
```

Configure la broche spécifiée pour qu'elle se comporte soit comme une entrée (**INPUT**), soit comme une sortie (**OUTPUT**).

Dans l'exemple du clignotement, remplacez le **LED_BUILTIN** par le numéro de la broche où la LED est connectée, c'est-à-dire **13**.

```
void loop() { ... }
```

Cette commande fait précisément ce que son nom suggère et répète successivement toutes les commandes qu'elle contient.

```
digitalWrite(13, HIGH)
```

Allume la LED. HIGH règle le voltage à environ 5V sur la broche 13.

```
delay(1000)
```

Met le programme en pause pendant la durée indiquée (en millisecondes).

```
digitalWrite(13, LOW)
```

Eteignez la LED en mettant le voltage à un niveau bas. LOW signifie donner un voltage de 0V à la broche 13.

Une liste complète de toutes les commandes possibles peut être trouvée grâce à une recherche rapide sur Google : *Commandes Arduino*.

Activité 2 : S.O.S en morse!

Dans cette activité, les élèves utilisent le circuit Arduino construit dans l'activité 1. Ils contrôleront la LED verte pour envoyer un message en morse afin de démontrer comment ils pourraient communiquer avec un rover sur Mars. Pour envoyer un O, les élèves doivent modifier le délai du code afin d'augmenter le temps de la LED verte allumée. Dans ce cas, nous avons changé 1000 ms à 5000 ms, mais il pourrait s'agir de n'importe quel nombre supérieur à 1000 ms. Le code est illustré à la figure A6 :

Figure A6

```
void loop() {  
  digitalWrite(13, HIGH);  
  delay(5000);  
  digitalWrite(13, LOW);  
  delay(1000);  
  
  digitalWrite(13, HIGH);  
  delay(5000);  
  digitalWrite(13, LOW);  
  delay(1000);  
  
  digitalWrite(13, HIGH);  
  delay(5000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

[↑ Code pour coder un O en morse](#)

Activité 3 : Mesurer une température

Dans cette activité, les élèves mesurent la température à l'aide de l'Arduino. Le but est de lire la température à partir du capteur de température (thermistance) et de comprendre comment le code est écrit/utilisé pour contrôler le capteur. Vous trouverez plus d'informations en effectuant une recherche rapide sur Google : *Capteur thermistance Arduino*.

Matériel

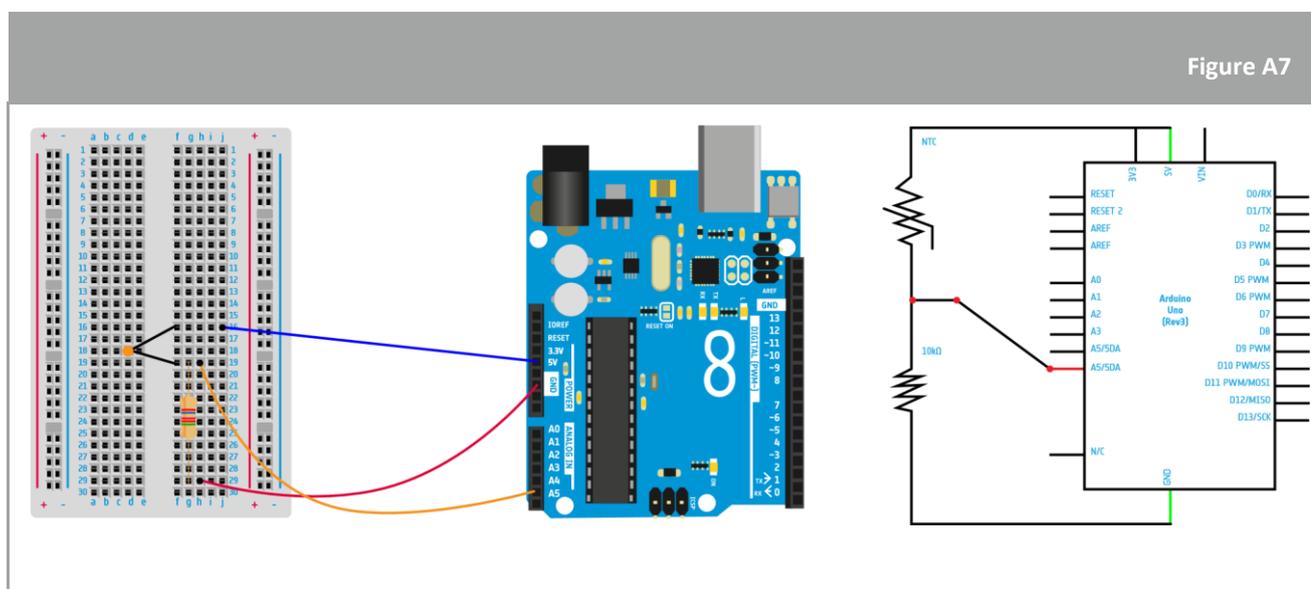
- 1 Arduino Uno
- 1 planche de prototypage
- 1 capteur de température (thermistance*)
- 1 résistance 10k Ω (marron/noir/noir/rouge/brun)
- 3 fils

Exercice

Configuration

- Connectez la thermistance sur la plaque d'affichage comme indiqué sur la figure A7 .
- Connectez la résistance 10k Ω (marron/noir/noir/rouge/marron) à la thermistance comme indiqué sur la figure A7.
- A l'aide d'un fil, connectez la thermistance à la broche **5V de l'Arduino Uno** d. A l'aide d'un fil, connectez la résistance à la broche **GND de l'Arduino Uno**.
- A l'aide d'un fil, connectez la thermistance et la résistance à la broche **A5 de l'Arduino**.
- Enfin, connectez l'Arduino UNO à l'ordinateur à l'aide du câble USB.

* thermistance : une résistance électrique dont la résistance est fortement réduite par le chauffage



↑ Le circuit Arduino est prêt à mesurer la température

Code

Les élèves vont maintenant écrire leur propre code. Une liste complète de toutes les commandes possibles peut être trouvée grâce à une recherche rapide sur Google : *Commandes Arduino*. La figure A8 illustre un exemple de code qui mesure la température. Dans cet exemple, la température lue par le capteur du circuit est appelée **AnalogT**, et est exprimée en unités de Volts car il s'agit d'un signal électrique ; elle n'a pas de signification physique. La température qui est convertie en degrés Celsius (l'unité de température utilisée en Europe) est appelée **AnalogTf**. Le nom de la broche de l'Arduino Uno est **A5**. Une fois que le code est téléchargé et commence à fonctionner, ouvrez la fenêtre de *Serial Monitor** afin de voir les données collectées.

Figure A8

```
1 void setup() {
2   // put your setup code here, to run once:
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   float AnalogT;
9   float AnalogTf;
10  AnalogT= float(analogRead(A5));
11  AnalogTf = (-40000/AnalogT) + 100;
12  Serial.println();
13  Serial.print("Temperature: ");
14  Serial.print(AnalogTf);
15  Serial.print("C");
16  delay(1000);
17
18 }
```

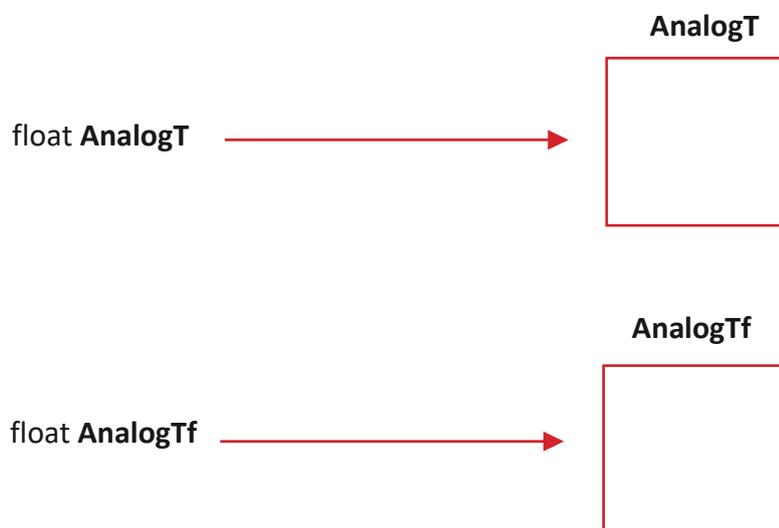
↑ [Arduino code for measuring the temperature](#)

*voir Tableau 1 : Commandes Arduino

Pour des explications sur la configuration du vide et la boucle du vide, veuillez-vous référer à l'Activité 1, exercice 3.

Ligne 3 : Comme la température sera affichée sur l'écran de l'ordinateur, il est nécessaire de prédéfinir le débit de données (9600 bits/seconde) régissant la transmission des données entre l'Arduino et l'ordinateur.

Ligne 8- 9 : ces deux lignes de code sont utilisées pour définir les variables appelées **AnalogT** et **AnalogTf**. Ces noms sont choisis par le programmeur. Pour modéliser ce processus, nous utiliserons une explication graphique. Lorsque vous écrivez **float AnalogT**, vous créez dans la mémoire de l'ordinateur une boîte ou un espace vide appelé **AnalogT** qui a une taille spécifique indiquée par "float". Flottant signifie que le nombre à l'intérieur de la boîte aura une virgule (décimale).



Ligne 10 : `AnalogT = float(analogRead(A5)) ;`

L'ordinateur inscrira le numéro lu sur la broche A5 de l'Arduino Uno (par exemple 500.00) à l'intérieur de la boîte **AnalogT**. Nous utilisons la fonction **analogRead()** pour lire n'importe quelle broche de l'Arduino.



Ligne 11 : `AnalogTf = (-40000/AnalogT) + 100 ;`

Cette ligne est l'opération mathématique qui permet de convertir la mesure en volts (**AnalogT**) en une température en degrés Celsius (**AnalogTf**).



Lignes 12 à 15 : Pour afficher la mesure de la température à l'écran, vous devez utiliser la fonction d'impression. Cette fonction permet de récupérer le numéro à l'intérieur de n'importe quelle boîte et de l'imprimer dans le Serial monitor. Dans cet exemple, la fonction imprimera le numéro à l'intérieur de la boîte appelée **AnalogTf**. Tous les mots contenus dans le "..." sont directement imprimés sur le moniteur de série.

Ligne 16 : la fonction de délai permet d'attendre 1000 millisecondes pour afficher la température sur le moniteur en série.

Discussion

Les élèves devront probablement calibrer leur propre capteur de température. Ils peuvent utiliser un autre thermomètre pour mesurer la température réelle comme référence pour calibrer leur capteur. Pour calibrer grossièrement, ils exécutent le code tel qu'il est actuellement écrit et si leur sortie de données n'est pas la même que la température réelle de référence, ils essaient de changer le dernier chiffre (+ 100) de la ligne 11 (**AnalogTf**) du code de la figure A8.

Pratique

On demande aux élèves de calibrer le capteur de température comme s'ils étaient à la surface de Mars. Comme la température moyenne de la surface martienne est considérée comme étant d'environ -60°C, et de +14°C pour la Terre, les élèves doivent calculer un décalage de -74. Le code doit être adapté de cette façon :

$$\text{AnalogTf} = (-40000/\text{AnalogT}) + 100 - 74 ;$$

Activité 4 : Mesurer la pression

Dans cette activité, les élèves mesurent la pression ambiante avec l'Arduino. Le but est de lire la pression du capteur (MPX4115A) et de comprendre comment le code est écrit/utilisé pour contrôler le capteur. Vous trouverez plus d'informations en effectuant une recherche rapide sur Google : *Arduino MPX4115A*.

Matériel

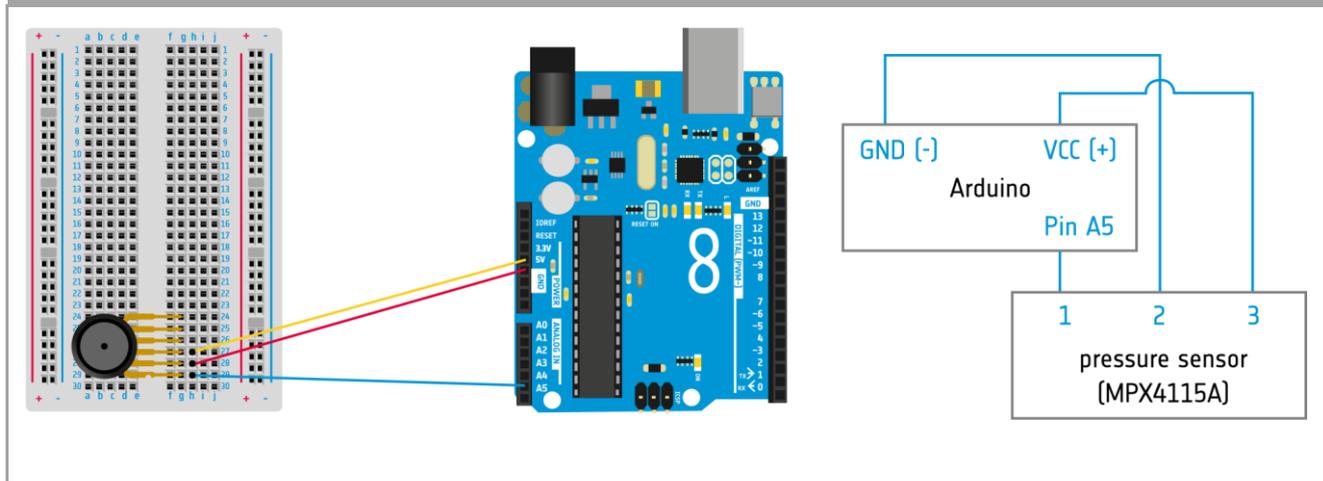
- 1 Arduino Uno
- 1 planche de prototypage
- 1 capteur de pression
- 3 fils

Exercice

Configuration

- Connectez le capteur de pression à la planche à pain.
- À l'aide d'un fil, connectez la branche 1 du capteur de pression (marquée d'une encoche) à la broche **A5**.
- A l'aide d'un fil, connectez la branche 2 du capteur de pression dans la rangée **GND de l'Arduino**
- A l'aide d'un fil, connectez la branche 3 du capteur de pression à la broche **5V de l'Arduino**.
- Enfin, connectez l'Arduino UNO à l'ordinateur à l'aide du câble USB.

Figure A9



↑ Arduino circuit ready to measure the pressure

Code

Dans ce cas, les élèves utilisent les mêmes commandes que lors de l'activité précédente sur la température ; là encore, ils se concentreront sur l'étalonnage du capteur. Une étape importante et très utile dans l'utilisation des capteurs est de consulter leurs fiches techniques, qui contiennent des informations détaillées sur le fonctionnement du capteur. Dans ce cas, nous pouvons facilement trouver des informations utiles grâce à une recherche rapide sur Google : Fiche technique MPX4115A.

Le capteur de pression convertit la pression mesurée en une tension. Il s'agit d'un signal analogique qui sera traduit en un signal numérique. Pour traduire les tensions mesurées en valeurs de pression ambiante (en unités Pa ou en pascals dans le système international d'unités), la fonction de transfert du capteur est nécessaire. Cette fonction décrit la relation mathématique entre la tension de sortie du capteur et la pression équivalente. Cette fonction se trouve dans les fiches techniques du capteur. Cette fonction peut varier d'un capteur à l'autre.

Si nous regardons la fiche technique, nous trouverons la fonction de transfert suivante :

$$V_{out} = V_s \cdot (P \cdot 0,009 - 0,095)$$

Il suffit d'isoler la valeur de P (pression) et de prendre en compte que $V_s = 1024$ (la tension lue est divisée en 1024 étapes).

V_{out} = Valeur analogique que nous lirons sur la broche **A5**.

Vous trouverez ci-dessous un exemple de code qui imprime les valeurs de la pression. Utilisez-le comme référence.

Figure A10

```

1 void setup() {
2   // put your setup code here, to run once:
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   float AnalogP;
9   float AnalogPf;
10  AnalogP= float(analogRead(A5));
11  AnalogPf = ((AnalogP/1024)+0.095)/0.009;
12  Serial.println();
13  Serial.print("Pressure: ");
14  Serial.print(AnalogPf);
15  Serial.print("kPa");
16  delay(1000);
17
18 }
```

[↑ Code Arduino pour mesurer la pression](#)

Le nom donné pour la « pression » en volts est **AnalogP** et pour la pression en pascal est **AnalogPf**. Le nom de la broche où le capteur de pression est connecté est A9.

Pour calibrer leur capteur de pression, les étudiants peuvent vérifier la pression ambiante de leur lieu sur Internet. Pour effectuer un calibrage approximatif, ils exécutent le code tel qu'il est actuellement écrit et, si la

sortie de leurs données n'est pas la même que la pression de référence réelle, ils peuvent adapter la ligne 11 (**AnalogPf**) du code.

Activité 5 : Mesurer l'altitude

La pression varie en fonction de l'altitude, et la pression peut donc être utilisée pour mesurer l'altitude. Pour mesurer l'altitude, les élèves doivent ajouter cette partie du code à l'intérieur de la boucle du code mesurant la pression.

Vous trouverez ci-dessous un exemple de code qui permet d'afficher l'altitude en mètres. Utilisez-le comme référence.

Figure A11

```
float Altitude;
float Altitudef;
  Altitude = pow(AnalogPf/101.325,0.1903);
  Altitudef = (1-Altitude)*44300 + 100;
Serial.print("Altitude: ");
Serial.print(Altitudef); Serial.print ("meters");
```

[↑ Code Arduino pour mesurer la pression](#)

Le nom donné à l'altitude en mètres est **Altitudef**. La pression en unités de kPa est ensuite donnée par **AnalogPf**, dans cet exemple.

D'après les indications données dans la case "Saviez-vous que", les élèves doivent se rendre compte qu'à mesure que l'altitude augmente, la température de l'atmosphère diminue, et cette relation doit être visible dans leur graphique de l'altitude et de la température.

Comment utiliser Arduino en classe

Arduino est un excellent outil pour initier les étudiants à la programmation de base et au large éventail d'applications qu'elle offre. Il peut également être utilisé pour enseigner et apprendre des sujets liés aux STEM. En voici quelques exemples :

- **Introduction aux principes de base des circuits** - L'activité 1 peut être utilisée pour présenter aux élèves un circuit simple.
- **Introduction à l'analyse des données** - Les élèves recueillent des données du monde réel (température et/ou pression) et analysent les données à l'aide de graphiques. Les élèves lisent les informations et tirent des conclusions.

Introduction aux éléments météorologiques et à la collecte des données - Les élèves peuvent développer leur propre station météorologique et prévoir les jours ensoleillés/nuageux.

Le saviez-vous ?



La mission 2020 du programme ExoMars livrera un rover européen et une charge utile scientifique russe à la surface de Mars pour établir si la vie y a jamais existé !

Activité 0 : Débuter

Tout le monde, tous les jours, utilise la technologie ! Dans les missions spatiales, les ingénieurs utilisent la technologie pour commander des robots et communiquer avec eux afin d'étendre nos connaissances scientifiques. Dans cette activité, vous deviendrez l'ingénieur chargé d'étudier Mars, la planète rouge, en utilisant un outil technologique appelé Arduino. Découvrons comment identifier les composants utilisés dans Arduino!

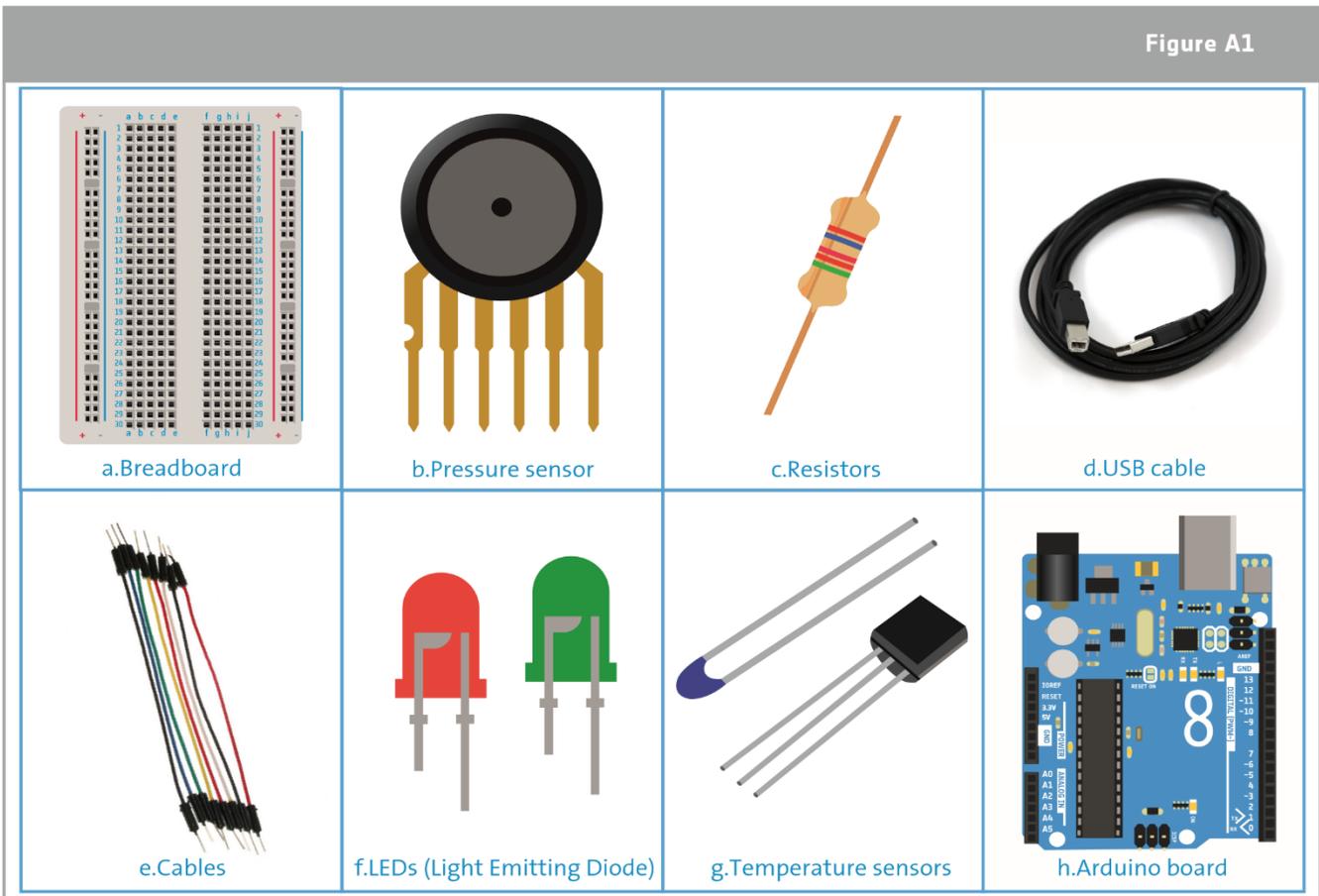
Matériel

- The Arduino basic kit

Composants

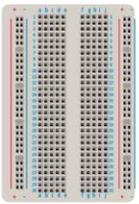
- Travailler en binôme. Ouvrez votre kit Arduino et faites correspondre les composants avec les descriptions de la page suivante.

Figure A1

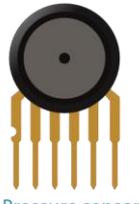


↑ Arduino basic kit components

COMPOSANTS



Breadboard



Pressure sensor



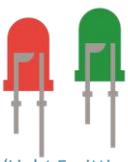
Resistors



USB cable



Cables



LEDs (Light Emitting Diode)

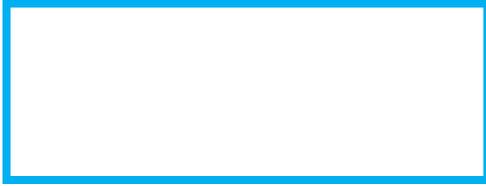


Temperature sensors



Arduino board

DESCRIPTIONS



A une forme circulaire et mesure la pression ambiante



A des entrées et des sorties et fonctionne comme un petit ordinateur



a deux longues jambes connectées à un système diriger et mesurer la température



Sont rouges, vertes... et émettent de la lumière lorsque l'électricité passent par elles



Forme cylindrique avec bandes de couleur, ils réduisent l'intensité dans le circuit



Est un support blanc utilisé pour connecter facilement les composants électroniques

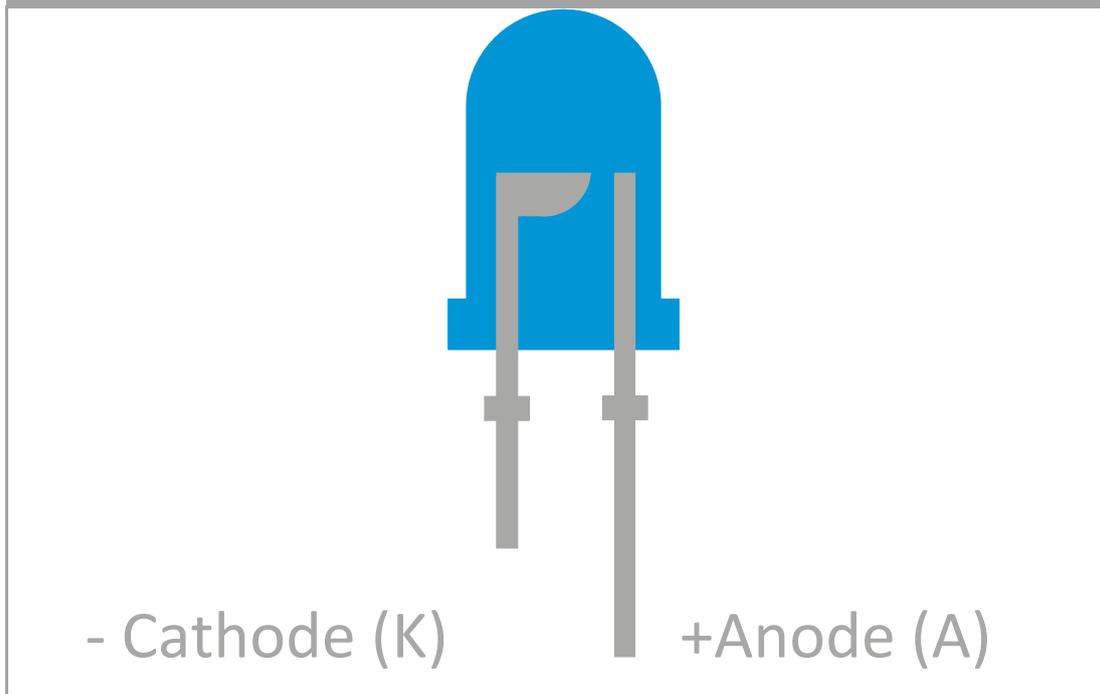


Ont des couleurs différentes et sont utilisés pour connecter des composants en conduisant l'électricité



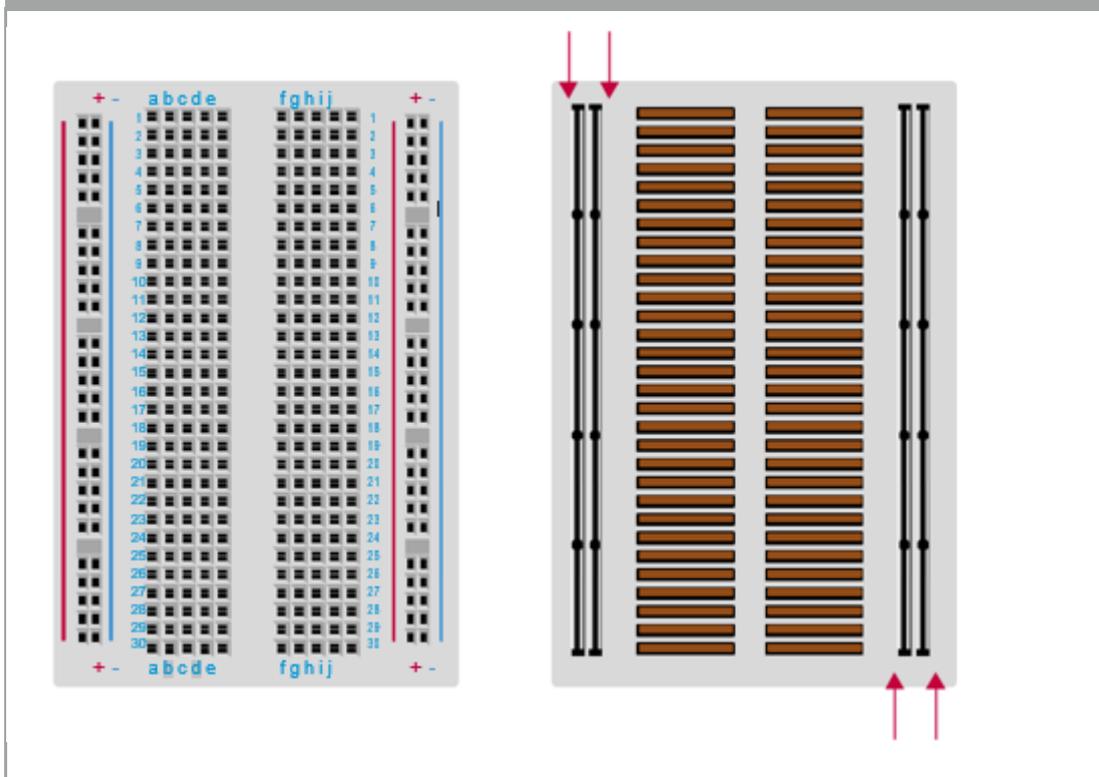
Un long câble noir utilisé pour connecter l'Arduino Uno à un ordinateur

Figure A2



↑ LED : Notez que le LED a une jambe longue et une jambe courte. La longue branche est appelée anode et la courte cathode.

Figure A3



↑ Breadboard: Notez que les broches sont connectées par rangées

Activité 1 : Faites clignoter votre carte !

Pour commander des objets interactifs, vous devez utiliser un outil tel qu'un Arduino, composé de plusieurs composants électroniques. Comme vous êtes l'ingénieur chargé d'étudier Mars, vous devez communiquer avec le rover pour commander ses expériences et collecter ses données. Dans cette activité, vous apprendrez le langage de programmation C++ afin de communiquer avec votre Arduino Uno, et vous apprendrez à contrôler la lumière !

Matériel

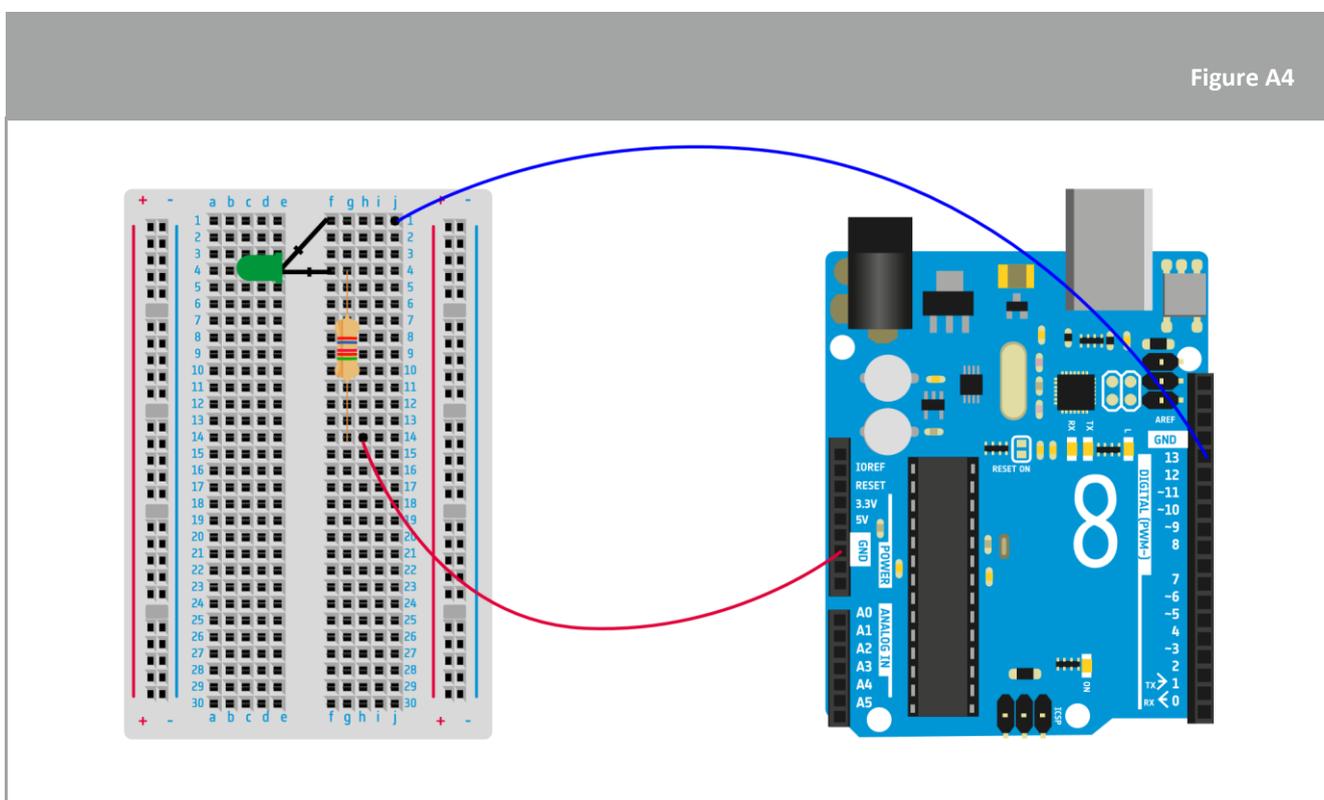
- 1 Arduino Uno
- 1 planche de prototypage
- 1 LED verte
- 1 résistance 220 Ω (rouge/rouge/noir/noir/marron)
- 2 fils

Exercice

Suivez attentivement ces instructions pour installer la planche à pain et contrôler le voyant vert :

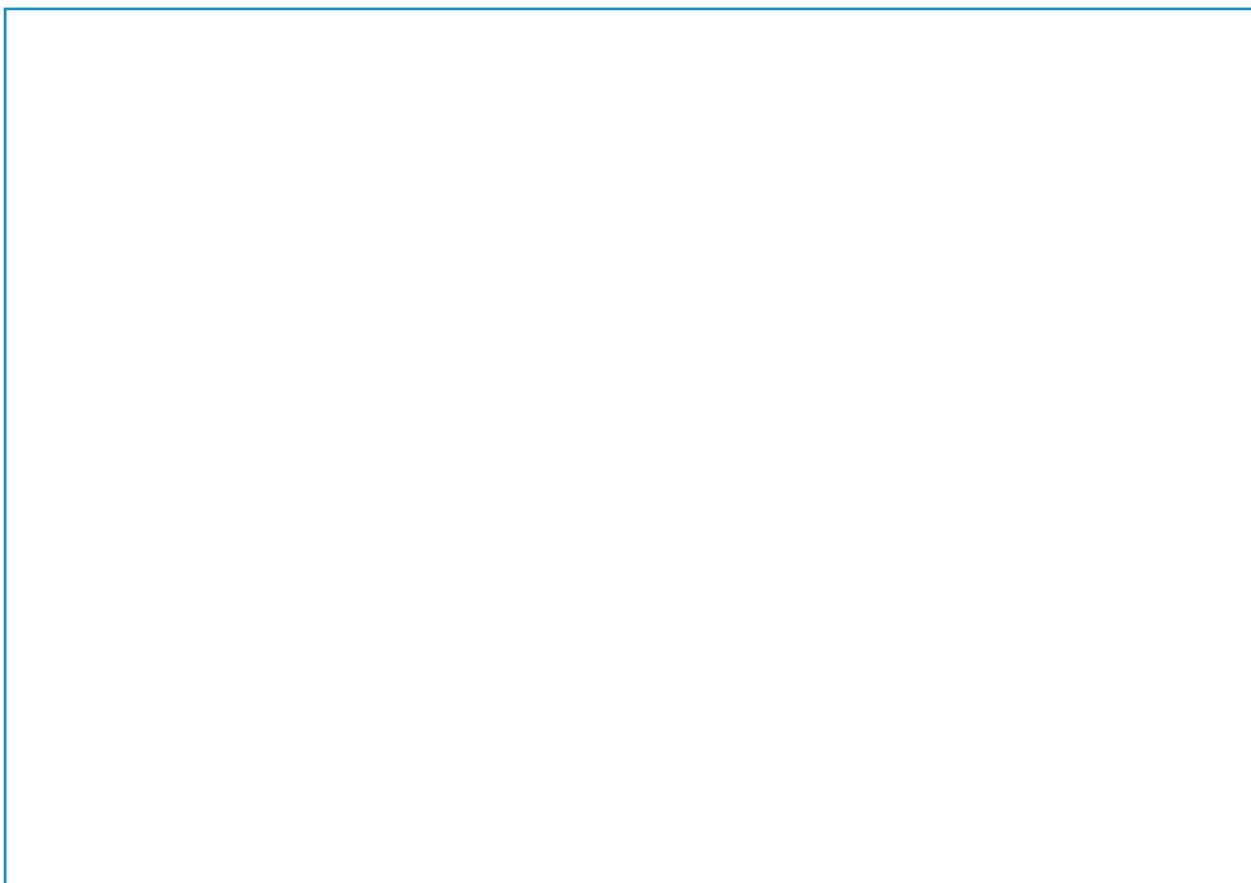
- Connectez la LED verte (plastique transparent) sur la planche à pain comme indiqué sur la figure A4.
- Connectez la résistance 220 Ω (rouge/rouge/noir/noir/marron) à la branche courte de la LED comme indiqué sur la figure A4.
- A l'aide d'un fil, connectez la longue jambe de la LED à la broche numéro **13** de l'Arduino Uno.
- A l'aide d'un fil, connectez la résistance à la broche **GND** (masse) de l'Arduino Uno
- Connectez l'Arduino Uno à l'ordinateur à l'aide du câble USB

Figure A4



↑ Le circuit Arduino est prêt à faire clignoter la LED verte

2. Dans l'encadré ci-dessous, dessinez le circuit électrique correspondant aux connexions électriques de la figure A4. Utilisez ces symboles :



3. Maintenant que la breadboard est prête, vous devez envoyer des instructions à l'Arduino Uno pour faire clignoter la LED.

Ouvrez le logiciel Arduino (IDE) sur l'ordinateur et cliquez sur :

Fichier → Exemples → Notions de base → Clignez des yeux.

Le code clignotant apparaîtra à l'écran et est illustré dans la figure ci-dessous. Lisez attentivement l'explication pour bien comprendre :

Téléversez le code sur la carte

```
1 /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
6  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN takes care
7  of use the correct LED pin whatever is the board used.
8  If you want to know what pin the on-board LED is connected to on your Arduino model, check
9  the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
10
11  This example code is in the public domain.
12
13  modified 8 May 2014
14  by Scott Fitzgerald
15
16  modified 2 Sep 2016
17  by Arturo Guadalupi
18 */
19
20
21 // the setup function runs once when you press reset or power the board
22 void setup() {
23   // initialize digital pin LED_BUILTIN as an output.
24   pinMode(LED_BUILTIN, OUTPUT);
25 }
26
27 // the loop function runs over and over again forever
28 void loop() {
29   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
30   delay(1000); // wait for a second
31   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
32   delay(1000); // wait for a second
33 }
34
```

Le texte en gris est un commentaire,
il explique ce que le code fait

LED_BUILTIN est le n° / nom du pin auquel la LED est connectée

HIGH = la LED est allumée

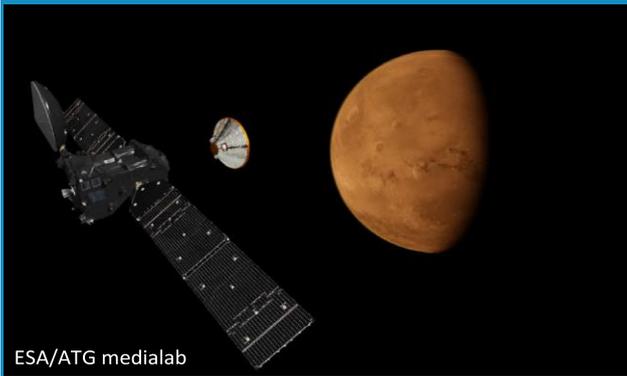
1000ms est un délai d'attente,
ici le délai entre éteindre et
allumer la LED

LOW = la LED est éteinte

- Avant de télécharger le code sur l'Arduino Uno, vous devez faire correspondre le nom de la broche dans le code avec le numéro de la broche où la LED verte est connectée à l'Arduino Uno. Expliquez ce que vous avez changé dans le code pour contrôler la LED verte :

- Téléchargez le code de l'Arduino Uno en cliquant sur le bouton  en haut à gauche de l'écran. Félicitations, le voyant vert clignote !

Le saviez-vous ?



Mars est tellement éloignée que les signaux prennent du temps à voyager depuis le vaisseau spatial et le rover jusqu'à la Terre. Le délai de communication rend difficile d'avoir une conversation avec le rover, ou de réagir rapidement si quoi que ce soit arrive.

Activité 2: S.O.S en morse !

Maintenant que vous savez comment faire clignoter une LED en utilisant grâce à une Arduino, cette activité vous apprendra à communiquer à l'aide d'une LED et à envoyer un message à un rover sur Mars. En fait, une très forte tempête de sable prévoit de passer sur le rover ! Pour éviter d'endommager le rover, faites-lui envoyer un S.O.S. !

Le Saviez-vous ?



NASA/JPL-Caltech/MSSS

Cette image d'une tempête de poussière sur Mars montre les nuages qui résultent des changements de la pression atmosphérique, de la température et de la hauteur en raison du déplacement vertical, comme lorsque le vent souffle sur une montagne ou un mur de cratère

Matériel

- Circuit construit dans l'activité 1

Exercice

Utilisez le code Morse pour faire clignoter la LED verte. Sur la figure A5, un point signifie un signal court, qui allumera la LED pendant un court instant et un tiret signifie un signal long, qui allumera la LED pendant plus longtemps. Le code pour envoyer un S est donné sous forme d'exemple (A6)

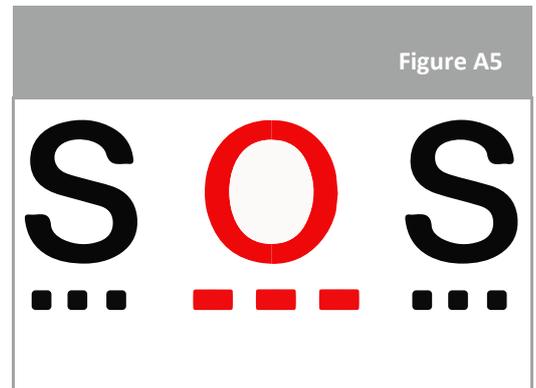


Figure A5

Figure A6

```

28 void loop() {
29   digitalWrite(13, HIGH);
30   delay(1000);
31   digitalWrite(13, LOW);
32   delay(1000);
33
34   digitalWrite(13, HIGH);
35   delay(1000);
36   digitalWrite(13, LOW);
37   delay(1000);
38
39   digitalWrite(13, HIGH);
40   delay(1000);
41   digitalWrite(13, LOW);
42   delay(1000);
43 }
44

```



↑ Code pour envoyer un S en morse

Expliquer ce que vous devez changer dans le code pour envoyer un O :

Adaptez le code pour envoyer un S.O.S. , puis téléversez-le sur l'Arduino et testez-le !

Activité 3 : Mesurer la température

La technologie Arduino vous permet de contrôler une LED connectée à une Arduino Uno à l'aide d'un code écrit sur un ordinateur. Lorsque vous téléversez le code, il envoie les instructions à l'Arduino Uno, qui le traite pour activer les composants électroniques. Dans la mission ExoMars, la technologie sera utilisée pour étudier l'environnement de Mars grâce à certains capteurs. Dans cette activité, vous découvrirez comment mesurer la température de la salle de classe et simuler la température sur Mars !

Matériel

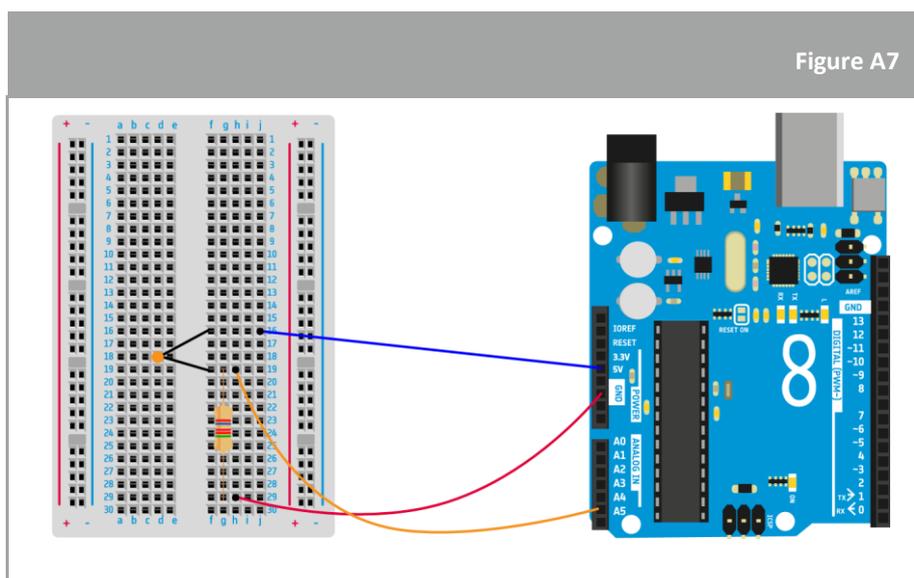
- 1 Arduino Uno
- 1 planche de prototypage
- 1 capteur de température (thermistance)
- 1 résistance 10k Ω (marron/noir/noir/rouge/brun)
- 3 fils de prototypage

Exercice

1. Pour mesurer la température dans la salle de classe, vous devez construire le circuit sur la planche à pain comme indiqué ci-dessous.

Suivez attentivement ces instructions :

- a. Connectez la thermistance sur la planche à pain comme indiqué sur la figure A7.
- b. Connectez la résistance 10k Ω (marron/noir/noir/rouge/marron) à la thermistance comme indiqué sur la figure A7.
- c. A l'aide d'un fil, connectez la thermistance à la broche **5V** de l'Arduino Uno.
- d. A l'aide d'un fil, connectez la résistance à la broche **GND** de l'Arduino Uno.
- e. À l'aide d'un fil, connectez la thermistance et la résistance à la broche **A5** de l'Arduino.
- f. Connectez l'Arduino Uno à l'ordinateur à l'aide du câble USB.



↑ Le circuit Arduino est prêt à mesurer la température

2. La figure A8 illustre un code partiellement manquant pour mesurer la température. Trois valeurs, représentées par des cases (rouge, vert et bleu) doivent être remplies pour donner des instructions correctes à l'Arduino Uno afin de mesurer la température. Voyons comment remplir ce code.

Figure A8

```

1
2 void setup() {
3   Serial.begin(9600);
4 }
5
6 void loop() {
7
8   float ;
9   float ;
10
11    = float(analogRead());
12    = (-40000/)+100;
13
14   Serial.println();
15   Serial.print("Temperature in classroom: ");
16   Serial.print();
17   Serial.print("C");
18   delay(1000);
19
20 }
```

↑ [Arduino code for measuring the temperature](#)

- a. Quelle est l'unité utilisée pour déterminer la température en Europe ?

Le capteur de température connecté à l'Arduino Uno va d'abord relever une mesure dans une autre unité, le volt. Pour transformer ce nombre de volts en degrés Celsius, vous utiliserez la formule écrite sur la ligne 12 de la figure A8. Enfin, vous afficherez la température à l'écran en utilisant la fonction *Serial.print* dans le code.

- b. Définissez la variable contenant la température en volts

Définissez la variable contenant la température en degrés Celsius

Cherchez la broche de l'Arduino où le capteur de température est connecté :

- c. Vous êtes prêt à écrire le code sur votre ordinateur portable. Ouvrez le logiciel Arduino et écrivez le code dans la figure A8, en remplaçant les cases par les variables que vous avez définies pour les cases rouges, verte et bleue. Veillez à remplacer chaque case de couleur par sa variable respective !

Téléversez le code vers l'Arduino en cliquant sur le bouton



d. Pour voir les mesures de température, vous devez ouvrir une fenêtre dans l'Arduino

appelée *Serial monitor*. Cherchez ce symbole  pour y accéder.

Discussion

1. Discutez de la pertinence des mesures avec vos collègues. Comment pouvez-vous vérifier que le capteur de température mesure avec précision la température de la classe.

2. Quelle ligne de code, sur la figure A8, pouvez-vous adapter pour calibrer le capteur ?

3. Pour vous entraîner un peu plus avec l'Arduino, calibrez le capteur de température comme s'il était à la surface de Mars. Considérez que la température moyenne de la surface de Mars est d'environ -60°C .

- Température moyenne sur Terre = _____

b. Ecrivez ici la nouvelle formule d'étalonnage du capteur de température :

Activité 4 : Mesurer la pression

Dans cette activité, vous apprendrez à mesurer la pression atmosphérique à l'aide de l'Arduino et d'un capteur de pression.

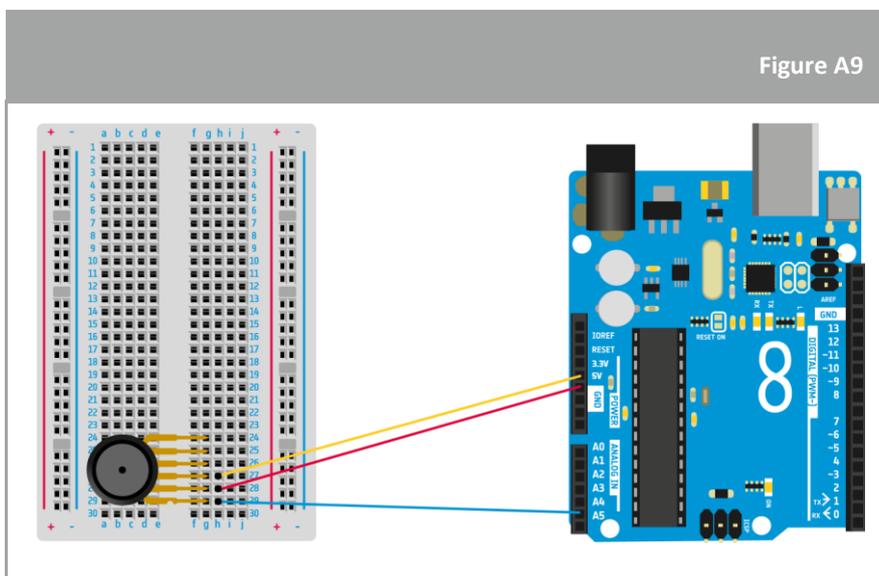
Matériel

- 1 Arduino Uno
- 1 planche de prototypage
- 1 capteur de pression
- 3 fils de prototypage

Exercice

1. Pour mesurer la pression dans la classe, vous devez d'abord construire le circuit sur *breadboard* (figure A9). Suivez attentivement ces instructions

Remarque : chaque broche du capteur de pression se comporte différemment, assurez-vous donc qu'elle est dans la même orientation que sur la figure :



↑ Circuit Arduino prêt à mesurer la pression

*

2. La Figure A10 est un code à compléter pour mesurer la pression. Trois variables, représentées par des cases (rouge, vert, bleu) doivent être remplies pour donner les instructions correctes à l'Arduino Uno afin qu'il puisse mesurer la pression. Découvrons comment remplir ce code.

Figure A10

```

1 void setup() {
2
3   Serial.begin(9600);
4 }
5 void loop() {
6
7   float [red] ;
8   float [green] ;
9
10  [red] = float(analogRead([blue]));
11  [green] = (([red]/1024)+0.095)/0.009;
12  Serial.println();
13  Serial.print("Pressure in classroom: ");
14  Serial.print([green]);
15  Serial.print("kPa");
16  delay(1000);
17
18 }

```

[↑ Code Arduino pour mesurer la pression](#)

a. Quelle est l'unité physique utilisée pour déterminer la pression dans le système international d'unités ?

b. Définissez la variable contenant la pression en volts

Définir la variable contenant la pression en kilopascal :

Cherchez la broche de l'Arduino où le capteur de pression est connecté :

Le capteur de pression connecté à l'Arduino Uno va d'abord récupérer une mesure (liée à la pression) dans une autre unité, le volt, qui est l'unité de mesure de l'électricité. Pour transformer ce nombre de volts en pascals, nous utilisons la formule de la fiche technique du capteur :

$$V_{out} = V_s \cdot (P \cdot 0,009 - 0,095)$$

Enfin, affichez la mesure de la pression à l'écran en utilisant la fonction *Serial.print* dans le code.

- c. En utilisant la formule $V_{out} = V_s \cdot (P \cdot 0,009 - 0,095)$; calculer la formule d'étalonnage pour compléter la ligne 12 de la figure A10. Comme la tension lue est divisée en 1024 étapes, considérez $V_s=1024$.

- d. Vous êtes prêt à écrire le code sur votre ordinateur. Ouvrez le logiciel Arduino et recopiez le code de la figure A10, avec les variables que vous avez définies pour les cases rouge, verte et bleue. Veillez à remplacer chaque case colorée par ses valeurs respectives !

Téléversez le code arduino sur l'ordinateur grâce au bouton :



- e. Pour voir les mesures de pression, vous devez ouvrir une fenêtre spécifique dans le logiciel Arduino appelée Serial monitor. Cherchez ce symbole sur la barre d'outils pour y accéder :



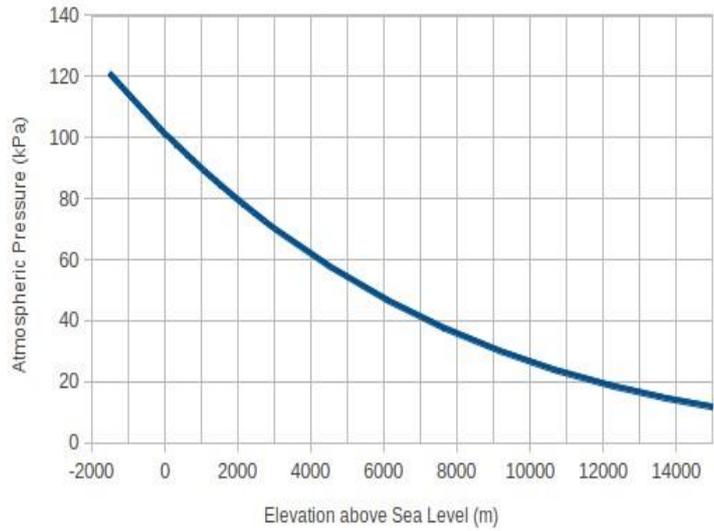
Discussion

Discutez de la pertinence des mesures de pression avec vos pairs. Comment pouvez-vous vérifier que le capteur de pression mesure avec précision la pression dans la classe ?

Quelle ligne du code de la figure A10 devez-vous adapter pour calibrer le capteur ?

Activité 5 : Mesurer l'altitude

La pression de l'air varie en fonction de l'altitude. Par exemple, la pression de l'air en mer est plus élevée que la pression de l'air en montagne. Vous pouvez voir comment elle varie dans le graphique. Avec un peu de mathématiques et une certaine compréhension de la physique qui sous-tend les lois des gaz, nous pouvons calculer exactement quelle est la relation entre la pression et l'altitude. Pour l'instant, nous n'avons pas besoin de nous inquiéter de cela. Dans cette activité, vous utiliserez vos mesures de pression pour déterminer votre altitude en utilisant une version simplifiée de la formule de la relation pression-altitude connue sous le nom de formule barométrique.



Matériel

- Circuit électronique réalisé à l'activité 4
- Code pour mesurer la pression dans l'activité 4

Exercice

1. Ouvrez le code mesurant la pression que vous avez écrit dans l'activité 4 du logiciel Arduino. Ajoutez ce code partiellement manquant à l'intérieur de la boucle vide :

Figure A11

```

1 float [ ] ;
2 float [ ] ;
3
4 [ ] = pow( [ ] /101.325,0.1903);
5 [ ] =(1- [ ] ) *44300+100;
6
7 Serial.println();
8 Serial.print("Altitude in classroom: ");
9 Serial.print( [ ] );
10 Serial.print("meters");
11 delay(1000);

```

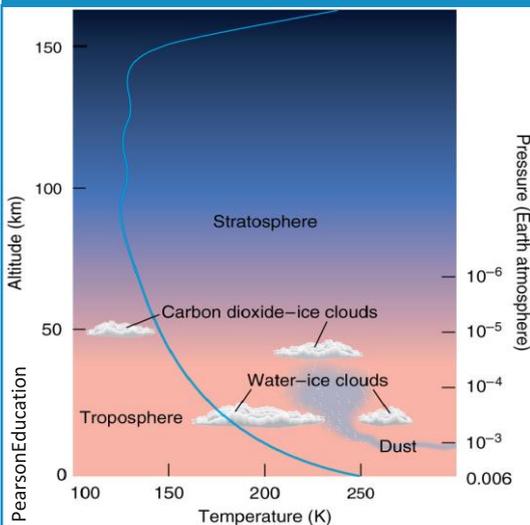
↑ [Code Arduino pour mesurer l'altitude](#)

2. Pour déterminer l'altitude, vous utiliserez les mesures du capteur de pression dans une formule écrite sur la ligne 4 de la figure A11. Comme vous l'avez découvert dans les exercices

précédents, un capteur connecté à l'Arduino Uno va d'abord récupérer une mesure en volts. Pour transformer ce nombre en mètres, vous utiliserez une formule écrite sur la ligne 5 de la figure A11. Enfin, les données d'altitude peuvent être affichées à l'aide de la fonction *Serial.print* dans le code. Si la lecture n'est pas calibrée, elle peut être ajustée en changeant la ligne 5 du code.

3.
 - a. Définir la variable qui contiendra :
 - L'altitude en volts :
 - L'altitude en mètres :
 - La pression en Pascals
 - b. Complétez le code avec les variables que vous avez définies pour les cases jaune, violette et verte. Veillez à remplacer chaque case colorée par sa valeur respective ! Ensuite, téléchargez le code
 - c. Pour afficher les données d'altitude, vous devez ouvrir une fenêtre spécifique dans le logiciel Arduino appelée *Serial monitor*.
4. Combinez les codes de température et d'altitude en un seul croquis et téléchargez-le sur l'Arduino Uno.
 Recueillez les données de température et d'altitude du sol au plafond de votre classe, puis dessinez un graphique affichant les résultats.
 Pouvez-vous expliquer la forme du graphique ?

Le saviez-vous ?



L'atmosphère martienne est une feuille extrêmement mince de gaz, principalement du dioxyde de carbone, qui s'étend de la surface de Mars jusqu'aux confins de l'espace. Le soleil chauffe la surface de Mars et une partie de cette chaleur sert à réchauffer le gaz près de la surface. Le gaz chauffé se diffuse ou se convecte ensuite dans l'atmosphère. Ainsi, la température du gaz est la plus élevée près de la surface et diminue à mesure que nous augmentons l'altitude.

LIENS UTILES

Arduino blog, useful for finding the latest Arduino information:
<https://blog.arduino.cc/>

Guides to using various Arduino components: <https://quarkstream.wordpress.com/>

Many Arduino based projects you can try at home can be found on Instructables:
<http://www.instructables.com/howto/arduino/>

HackADay hosts many other Arduino projects: <https://hackaday.io/list/3611-arduino-projects>