



RÉGION ACADÉMIQUE
GRAND EST

MINISTÈRE
DE L'ÉDUCATION NATIONALE
ET DE LA JEUNESSE

MINISTÈRE
DE L'ENSEIGNEMENT SUPÉRIEUR,
DE LA RECHERCHE
ET DE L'INNOVATION



Les microcontrôleurs dans les nouveaux programmes de Physique-Chimie

*D'après un atelier du Plan
National de Formation
« Nouveaux programmes de
physique-chimie de la voie
générale » - Mars 2019*



Plan

- Pourquoi introduire des microcontrôleurs en activités expérimentales ?
- Les programmes
- ECE
- Qu'est-ce qu'un microcontrôleur ?
- Suggestions pédagogiques pour réaliser des activités expérimentales avec microcontrôleur
- Evaluation

Pourquoi introduire
des activités
expérimentales
avec microcontrôleurs ?

L'apport en physique-chimie

L'apport en physique-chimie (physical computing)

La pensée informatique permet d'améliorer la compréhension des concepts de la physique et de la chimie :

Approche expérimentale (chaîne d'acquisition de l'information : capteur -> données)

Approche conceptuelle (modélisation / simulation)

- *prise d'initiatives en identifiant des informations ou des grandeurs importantes, afin de tester une hypothèse dans l'élaboration d'un modèle avec une démarche d'essai erreur.*
- *élaboration d'une solution, étape par étape (algorithmic thinking), en vérifiant l'adéquation des prévisions avec la réalité.*

Les programmes

1. En classe de Seconde
2. En classe de Première spécialité

En classe de seconde

Ondes et signaux

<p>Emission et perception d'un son <i>Signal sonore périodique, fréquence et période.</i></p>	<p>Utiliser un dispositif comportant un microcontrôleur pour produire un signal sonore.</p>	
<p>Signaux et capteurs <i>Capteurs électriques</i></p>	<p>Mesurer une grandeur physique à l'aide d'un capteur électrique résistif. Produire et réaliser une courbe d'étalonnage reliant la résistance d'un système avec une grandeur d'intérêt (température, pression, intensité lumineuse, etc.). Utiliser un dispositif avec microcontrôleur et capteur</p>	

En classe de première spécialité

Capacité expérimentales :

- Mesurer et traiter un signal au moyen d'une interface de mesure ou d'un microcontrôleur.
- Commander la production d'un signal grâce à un microcontrôleur.

<u>Mouvements et interactions</u> Description d'un fluide au repos <i>Loi de Mariotte</i>	Tester la loi de Mariotte, par exemple en utilisant un dispositif comportant un microcontrôleur.	
<u>Ondes et signaux</u> Ondes mécaniques <i>Célérité d'une onde, retard</i>	Déterminer, par exemple à l'aide d'un microcontrôleur ou d'un smartphone, une distance ou la célérité d'une onde. Illustrer l'influence du milieu sur la célérité d'une onde.	
Absorbance <i>Mesure d'une concentration</i>	Déterminer la concentration d'une espèce colorée en solution par des mesures d'absorbance.	

Pesanteur et arduino *Le but de cette épreuve est de mesurer l'intensité de la pesanteur terrestre à l'aide d'un montage utilisant la carte Arduino UNO.*

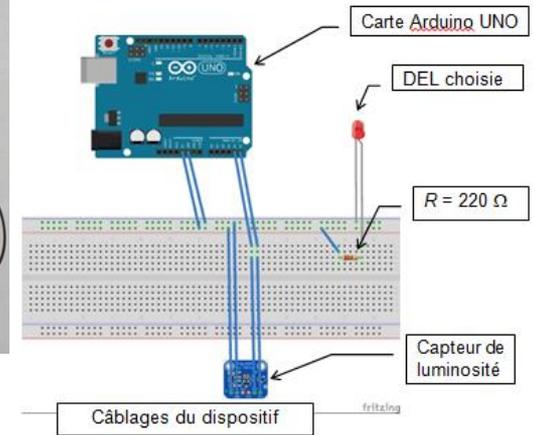
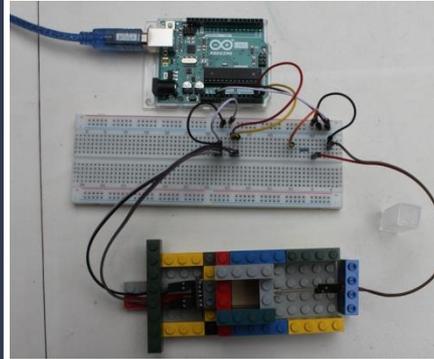
L'uart de l'arduino *Le but de cette épreuve est de décoder un signal (ou trame) UART émise par une carte Arduino UNO et d'évaluer le débit binaire de la transmission.*

Spectrophotomètre et arduino *Le but de cette épreuve est de discuter de la pertinence de l'utilisation d'une carte Arduino UNO pour fabriquer un spectrophotomètre.*

S'accorder avec un microcontrôleur *Le but de cette épreuve est de déterminer s'il est possible d'accorder une guitare avec un microcontrôleur.*

Nager sans frissonner *Le but de cette épreuve est de déterminer si un dispositif reliant une thermistance à une carte Arduino pourrait être utilisé pour suivre la température d'une eau de baignade chez un particulier.*

Spectrophotomètre et arduino *Le but de cette épreuve est de discuter de la pertinence de l'utilisation d'une carte Arduino UNO pour fabriquer un spectrophotomètre.*



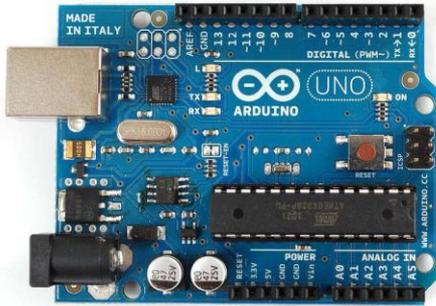
Qu'est-ce qu'un microcontrôleur ?

- Présentation
- Le microcontrôleur
- Les cartes
- La platine de câblage
- L'environnement de programmation
- Sécurité

Arduino en résumé



Une carte électronique



Arduino d'Ivrea
(Civico raccolta stampe - Milano)

Une communauté qui échange
<http://arduino.cc/>



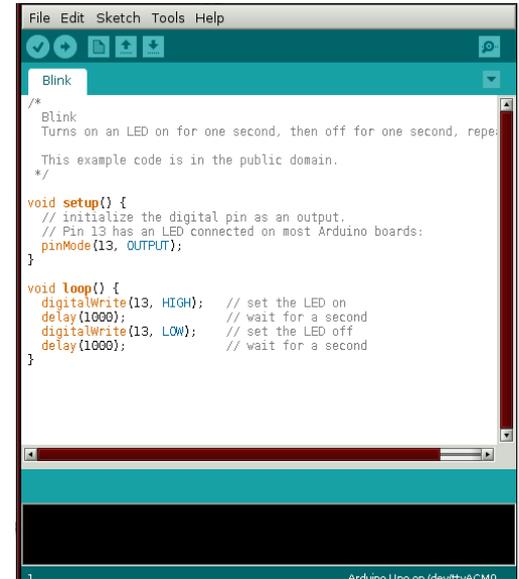
Photo by the Arduino Team

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be small alone or they can communicate with software running on a computer (e.g. Fluid, Processing, Max/MSP).

The boards can be **bought by hand** or purchased **pre-programmed**; the software can be **downloaded** for free. The hardware reference design (PCB files) are **available** under an open-source license, you are free to **adapt them to your needs**.

Un environnement de programmation



Microcontrôleur



Le microcontrôleur est composé de **quatre parties** :

Un **microprocesseur** qui va prendre en charge la partie traitement des informations et envoyer des ordres. C'est donc lui qui va exécuter le programme embarqué dans le microcontrôleur.

Une **mémoire de données** dans laquelle seront entreposées les données temporaires nécessaires aux calculs. C'est en fait la mémoire de travail qui est donc volatile.

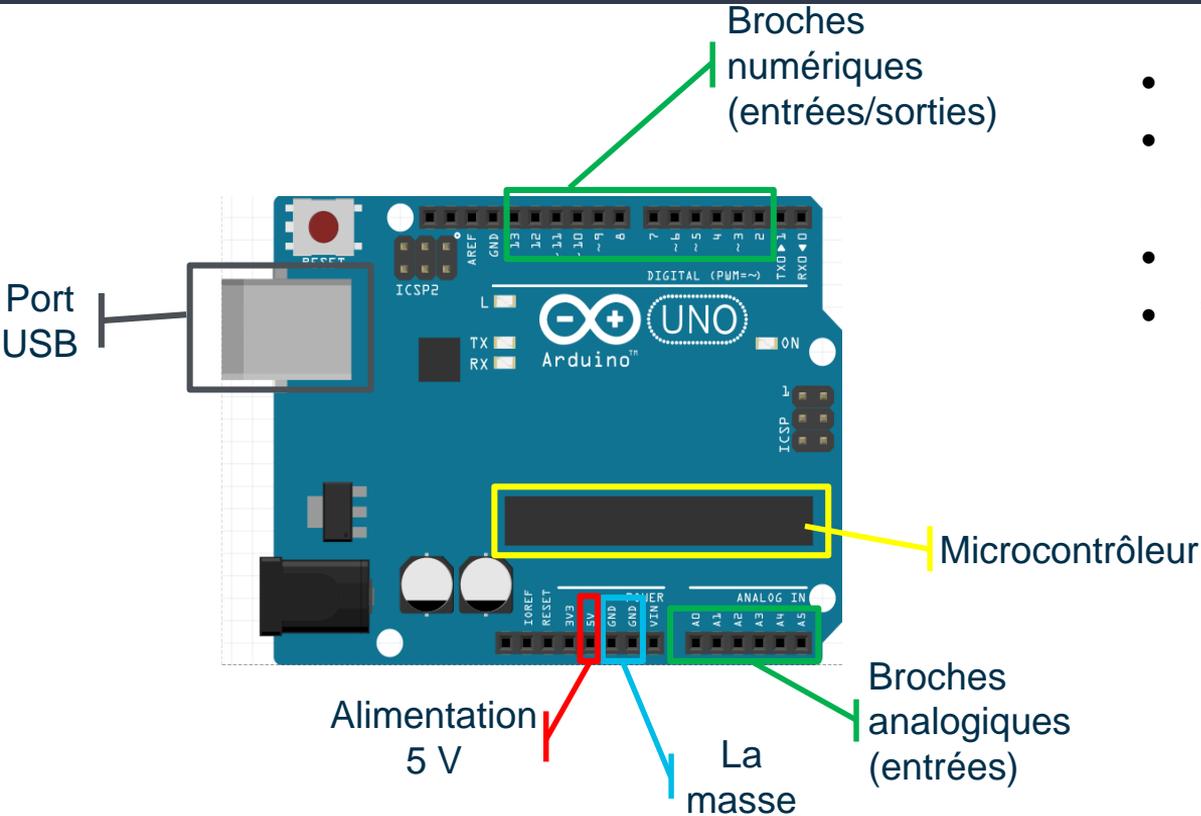
Une **mémoire programmable**, qui va contenir les instructions du programme pilotant l'application à laquelle le microcontrôleur est dédié. Il s'agit ici d'une mémoire non volatile puisque le programme à exécuter est à priori toujours le même.

La **dernière partie** correspond aux ressources auxiliaires. Celles-ci sont généralement :

- Ports d'entrées / sorties parallèle et série.
- Des timers pour générer ou mesurer des signaux avec une grande précision temporelle.
- Des convertisseurs A/N pour traiter les signaux analogiques.

<https://www.theengineeringprojects.com/2017/08/introduction-to-atmega328.html>

La carte Arduino



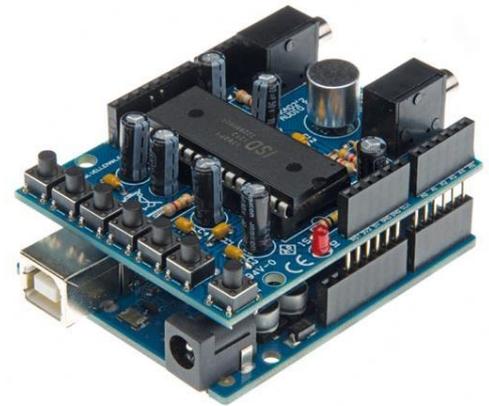
- Un langage basé sur C
- 14 entrées/sorties numériques, 6 entrées analogiques
- Mémoire flash 32 ko
- Vitesse d'horloge 16 MHz

La carte Arduino

L'alimentation

La carte ARDUINO UNO peut être alimentée ou par le **câble USB**, ou par un **bloc secteur externe** connecté grâce à une **prise « jack »** ou encore par un bloc de piles dont le raccordement est réalisé par l'intermédiaire des « **GND** » et « **Vin** » du connecteur d'alimentation. L'alimentation extérieure doit présenter une tension comprise entre **7 à 12V**.

La carte génère, par l'intermédiaire de régulateurs intégrés, deux tensions stabilisées : **5 V et 3,3 V**. Ces deux tensions permettent l'alimentation des composants électroniques de la carte ARDUINO. Étant disponibles sur connecteurs placés sur le pourtour des cartes, elles permettent également l'alimentation des modules Shields.



La carte Arduino

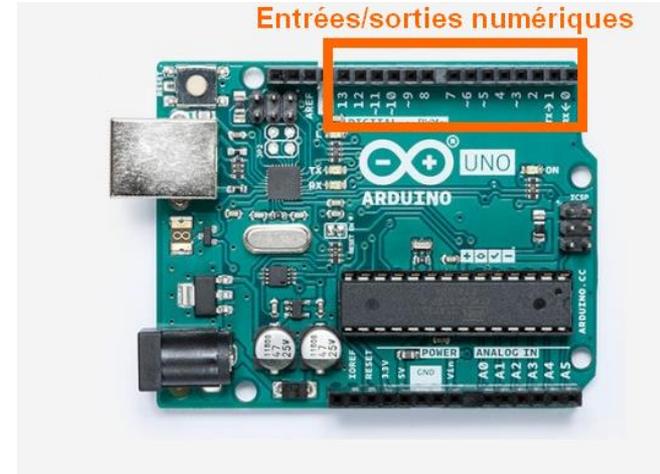
Les entrées-sorties

Quasiment toutes les broches d'un Arduino peuvent être programmées en entrée ou sortie numérique,

Une entrée : le programme peut lire une tension présente sur cette broche en utilisant `digitalRead(...)`. Comme cette tension est interprétée comme un chiffre binaire (0 ou 1), la datasheet l'Arduino Uno garantit que toute tension inférieure à $0,3xV_{cc}$, sera comprise comme un 0 et que toute tension supérieure à $0,6xV_{cc}$, sera comprise comme un 1.

Entre les deux, c'est flou. `digitalRead(...)` renverra de toutes façons un 0 ou un 1 mais de manière plus ou moins aléatoire selon l'Arduino utilisé.

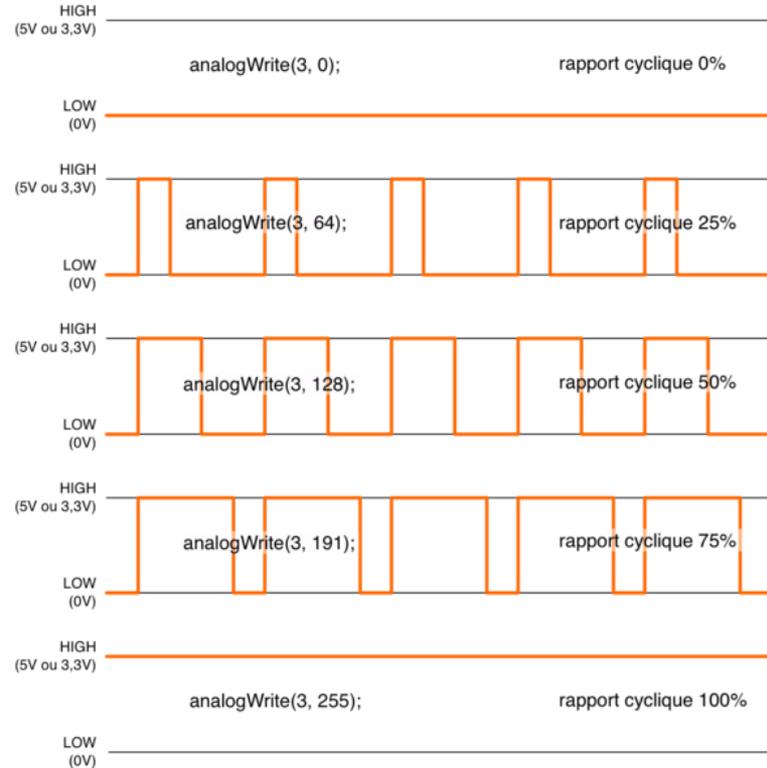
Une sortie : le programme peut écrire un chiffre binaire, au moyen de `digitalWrite(...)`, chiffre qui dans le programme sont nommées **HIGH** pour le 1 et **LOW** pour le 0, qui sera traduit en une tension de 5V pour le 1 et de 0V pour le 0. Attention toutefois, cette tension peut respectivement être plus basse ou plus haute si le courant qui est tiré de la broche commence à être important.



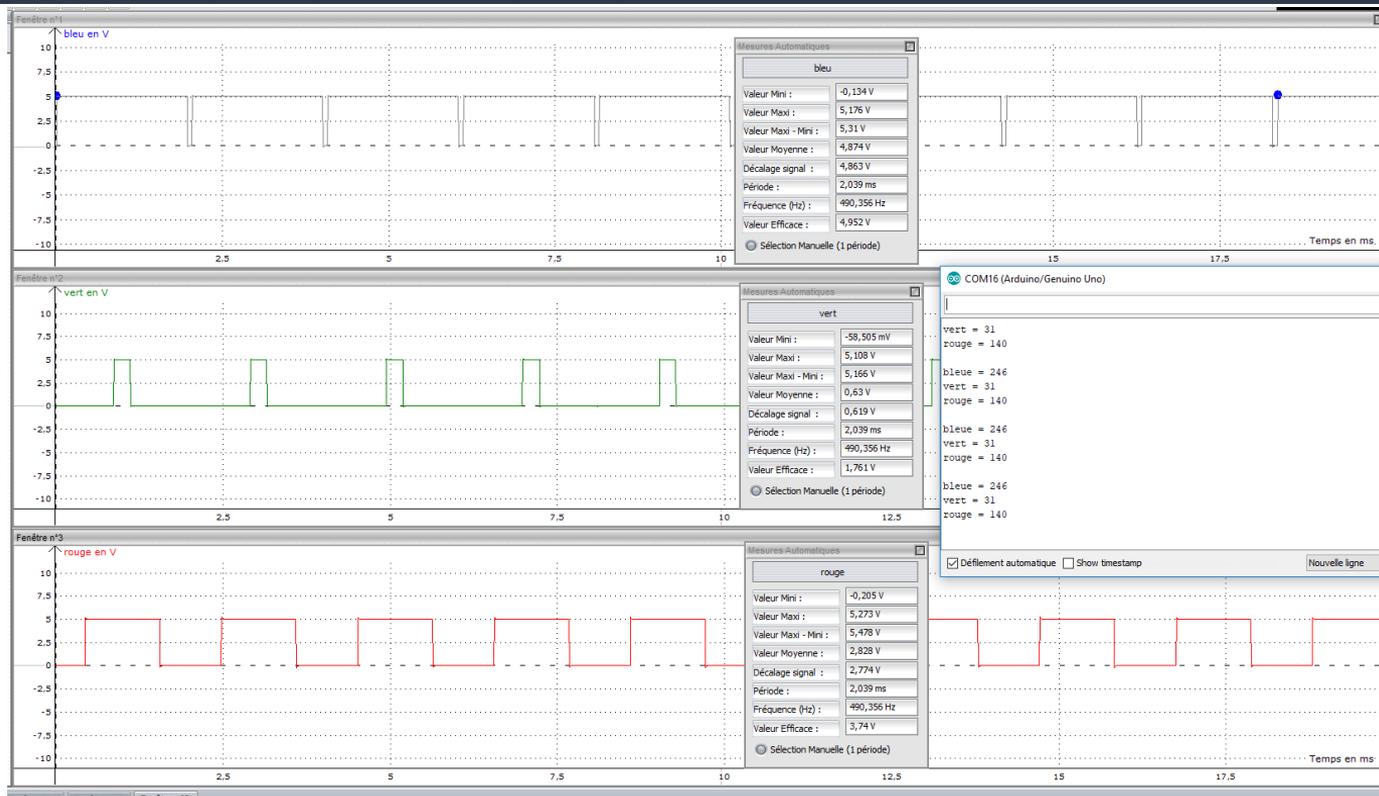
La carte Arduino

Les entrées-sorties

Pour l'Arduino Uno, 6 broches peuvent être configurées en sortie PWM (**Pulse Width Modulation** ou **modulation de largeur d'impulsion**). La configuration est effective dès que le programme utilise la fonction `analogWrite(...)` sur une des broches configurable en PWM. Le rapport cyclique de la PWM est réglé via cette même fonction entre 0, le signal est constamment à l'état bas, 0V, à 255, le signal est constamment à l'état haut, soit un rapport cyclique de 100%. La fréquence par défaut est de 500Hz



Analyse du signal de 3 sorties PMW par l'interface Sysam, pour éclairer 3 LEDS R,V,B



La carte Arduino

Les entrées analogiques

L'Arduino Uno possède 6 entrées analogiques. Les tensions, toujours entre 0 et 5V, appliquées sur ces broches, peuvent être numérisées via un convertisseur analogique-numérique. La fonction `analogRead(...)` permet de réaliser l'opération.

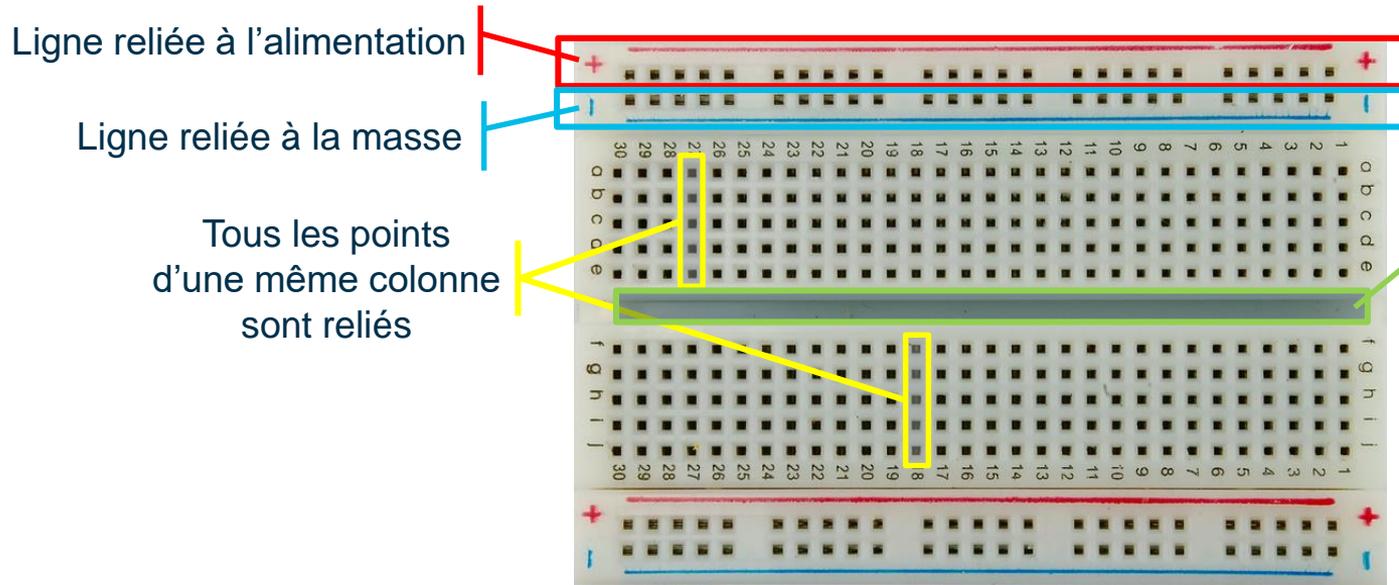
Le convertisseur des Arduino effectue une conversion sur 10 bits, c'est à dire qu'il convertit la tension en un nombre entier ayant une valeur de 0 à 1023. La résolution est donc d'environ 5mV.

Il est possible de changer la tension de référence de la conversion analogique-numérique, c'est à dire la tension comprise comme 1023.

Le MCU des Arduino possède une référence de tension interne de 1,1V qui peut être sélectionné à la place de la tension d'alimentation. Cela permet une meilleure résolution, environ 1mV, sur les faibles valeurs de tension à convertir.



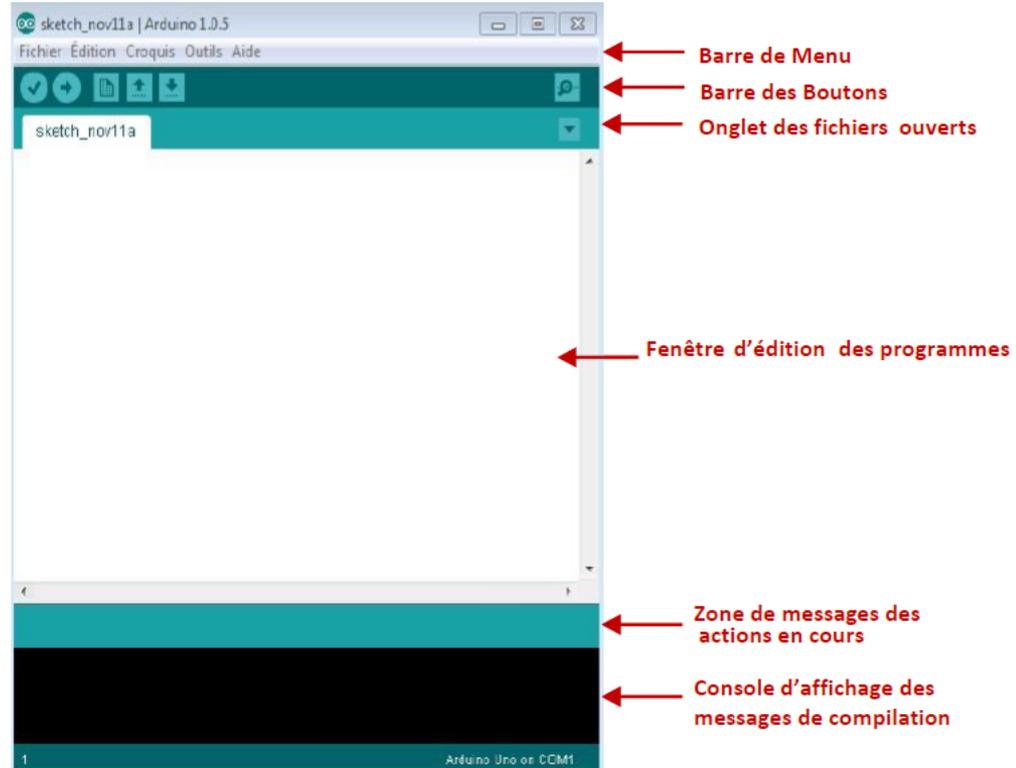
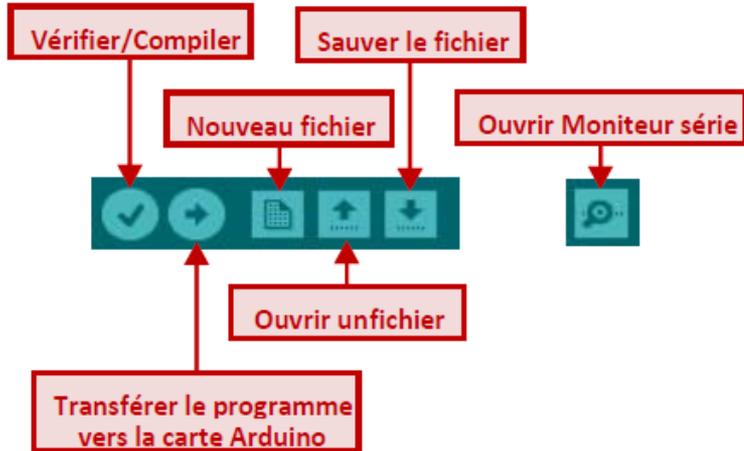
La platine de câblage (breadboard)



La plaque est séparée en deux parties symétriques et indépendantes

L'environnement de programmation

L'interface de programmation



L'environnement de programmation

La structure d'un programme

Zone de définition des constantes ou des variables ou d'inclusion des bibliothèques

```
/* Ce programme fait clignoter une LED branchée sur la broche n°13  
et fait également clignoter la diode de test de la carte */
```

```
int ledPin = 13;           // LED connectée à la broche n°13
```

```
void setup()  
{  
    pinMode(ledPin,OUTPUT); // Définit la broche n°13 comme une sortie  
}
```

```
void loop()  
{  
    digitalWrite(ledPin,HIGH); // Met la sortie ledPin au NL1 (diode allumée)  
    delay(3000);               // Attendre 3 s  
    digitalWrite(ledPin,LOW); // Met la sortie ledPin au NLO (diode éteinte)  
    delay(1000);               // Attendre 1 s  
}
```

Fonction setup()
qui contient les instructions d'initialisation

Fonction loop()
qui contient les instructions du programme

L'environnement de programmation

1^{ère} prise en main : Faire clignoter la led 13

- Connecter Arduino en USB
- Ouvrir le logiciel Arduino IDE (*Integrated Development Environment*)
- Charger le programme « Blink » dans
Fichier/Exemples/ 01.Basics/Blink
- Téléverser le programme (En cas de message d'erreur,
vérifier que la carte Arduino est connectée sur le bon port USB)
- Faire clignoter la led 4 s allumée et 1 s éteinte

L'environnement de programmation

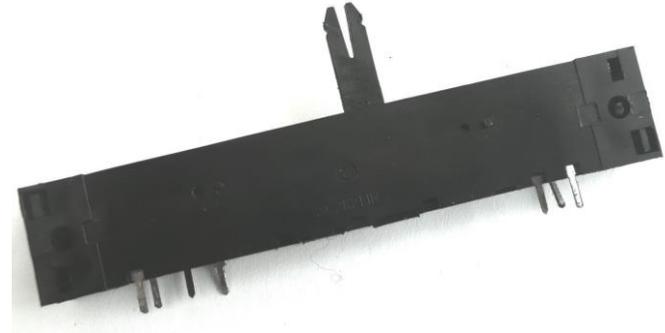
Utiliser un capteur résistif :

Au choix :

Capteur de position résistif à 1D

Capteur photo-résistif

Capteur thermo-résistif



L'environnement de programmation

Les bibliothèques de fonctions (quelques exemples)

`delay(1000)` ; // Pause de 1 s

`pinMode (12,OUTPUT)` ; // La broche 12 est configurée comme une sortie

`digitalWrite(12,HIGH)` ; // La sortie 12 est placée au niveau logique haut

`tension = analogRead(1)` ; // tension est égale au résultat de la CAN de l'entrée analogique 1

*Cette fonction permet de **lire le résultat de la conversion analogique numérique de la tension présente sur l'entrée analogique sélectionnée.***

La donnée retournée étant comprise entre 0 et 1023.

La durée de la conversion est d'environ 100 μ s.

`analogWrite(2,64)` ; // Signal PWM de rapport cyclique de 25 % sur la sortie analogique 2

<http://www.lyceebellevue.toulouse.net/4--chaine-information/4-2--langage>
(puis TP Arduino)

Sécurité



- Le microcontrôleur placé sur la carte est prévu pour fonctionner entre 3,3 et 5V.
- Le courant de sortie de chaque broche ne doit pas dépasser 40 mA (idéalement 20mA).
- Le courant issu du port USB ne doit pas dépasser 500 mA.

Conseils de sécurité :

- Pour éviter qu'un fil ou qu'un composant branché au + vienne endommager un port USB dans l'ordinateur, isoler le métal du port USB avec un adhésif d'électricien (souvent l'ordinateur détecte le court-circuit et désactive le port mais pas toujours....)
- Pour éviter les courts-circuits :
 - La carte ne doit pas être posée sur un support conducteur car elle possède sur son verso des zones nues qui ne doivent pas être mises en contact afin de ne pas court-circuiter les composants entre eux.
 - Ne jamais connecter directement le port noté « Gnd » avec la broche 5 V.

Sécurité



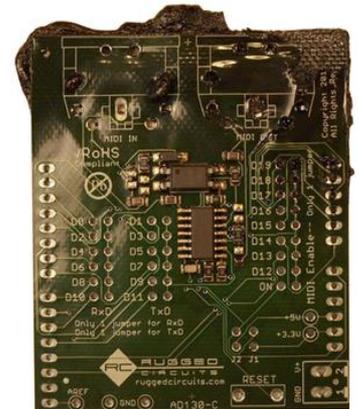
La plupart du temps les broches numériques sont des sorties (le courant va du microcontrôleur à la masse).

Dans le cas où les broches numériques du microcontrôleur sont des entrées (au lieu d'être raccordée à la masse, la broche sera raccordée au +5V), il faudra être très vigilant à ce que la broche soit configurée comme INPUT. Si elle devait être configurée en OUTPUT et réglée à 0V (LOW) par erreur, il est presque certain que le microcontrôleur finira grillé !

Pour protéger les entrées numériques :

- Connecter sur la patte du microcontrôleur utilisé comme INPUT une résistance d'une centaine d'Ohms qui limitera le courant en cas de fausse manœuvre.

Lien : <https://www.rugged-circuits.com/10-ways-to-destroy-an-arduino>

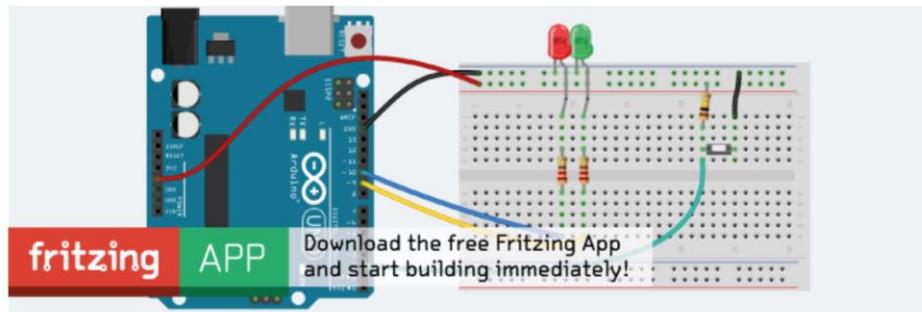


Suggestions pédagogiques pour réaliser des activités expérimentales avec microcontrôleur

1. Fritzing
2. Exemple de contrôle en classe
3. Organisation matérielle
4. Déroulement d'une activité expérimentale
5. Difficultés à gérer
6. Organisation de l'activité expérimentale en fonction de la classe
7. Evaluations (activité expérimentale, maison, classe)
8. Liens

Une aide pour la redaction des documents : l'éditeur de schémas, Fritzing

<http://fritzing.org/home/>



Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone. We offer a software tool, a community website and services in the spirit of Processing and Arduino, fostering a creative

Download and Start

Download our **latest version 0.9.3b** released on June 2, 2016 and start right away.

Produce your own board

With **Fritzing Fab** you can easily and inexpensively

[FAQ](#) [ABOUT](#) [CONTACT](#)

Blog

Fritzing Fab now powered by AISLER
Feb. 21, 2017

New fritzing release 0.9.3b!
Jun. 3, 2016

New Book: "Fritzing for Inventors"
Dec. 6, 2015

[More posts...](#)

Projects

asfasf4wr 34w433232
[gercerber](#)

Hand By Hand, arobot
[janaroman](#)

Led RGB humbjunior134
[humbjunior](#)

[More projects...](#)

En contrôle

Afin de réaliser une décoration pour Noël à partir de deux petites maisons en bois, on ajoute une LED pour éclairer chaque maison de l'intérieur. On souhaite que les deux LED clignotent à une fréquence de 0,5 Hz et s'allument alternativement.

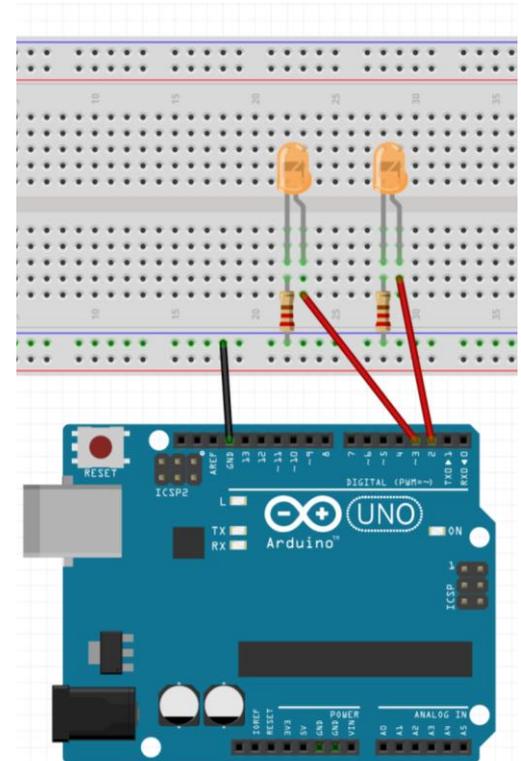
On décide pour cela d'utiliser un microcontrôleur Arduino.



Le programme pour contrôler les LED :

```
// Programme
int ledOrange1= 2; //indique que la LED 1 est branchée sur la broche 2
int ledOrange2 = 4; //indique que la LED 2 est branchée sur la broche 4
/* On stocke la valeur 2 dans la variable "ledOrange1",
pour indiquer que la led orange notée 1 se trouve sur la broche 2,
même chose pour la led orange notée 2 sur la broche 3
*/
void setup() {
pinMode(ledOrange1, OUTPUT); // indique que la broche sur laquelle est branchée la LED 1 est une sortie
pinMode(ledOrange2, OUTPUT); // indique que la broche sur laquelle est branchée la LED2 est une sortie
}
void loop() {
digitalWrite(ledOrange1,HIGH);
delay(1000);
digitalWrite(ledOrange1,LOW);
digitalWrite(ledOrange2,HIGH);
delay(100);
digitalWrite(ledOrange2,LOW);
}
```

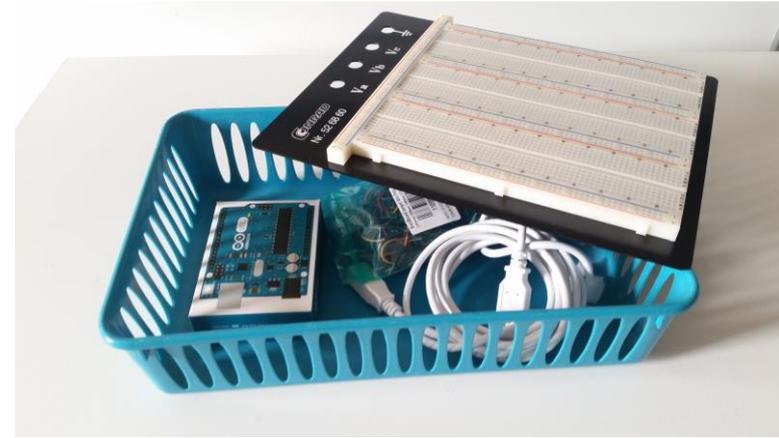
- 1) D'après le montage réalisé, chaque LED a-t-elle été déclarée sur la broche à laquelle elle est connectée ? Si non, surligner la(ou les) ligne(s) de code fausse(s) et la (ou les) réécrire en la (ou les) corrigeant.
- 2) Quelle est la partie du code qui se répète tant que l'Arduino est branché ?
- 3) Ajouter un commentaire pour expliquer chaque ligne de code de la partie `void loop()`.
- 4) Il y a un problème, les LED ne clignotent pas avec la bonne fréquence. Surligner les lignes à corriger, puis les réécrire de manière à ce que les LED clignotent avec la fréquence attendue. Justifier par un calcul.



Organisation matérielle



→ Placer les différents composants dans des boites



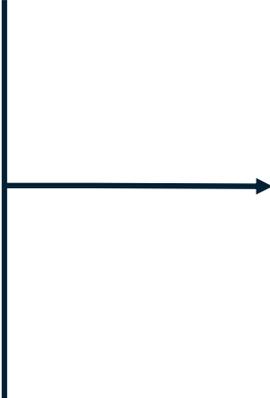
→ Le microcontrôleur et les fils dans une autre boîte

Déroulement d'une activité expérimentale avec microcontrôleur

- Réaliser le circuit
- Recopier, écrire ou compléter le code et le valider
- Téléverser le code
- Vérifier que le code commande correctement le circuit
- Commenter, modifier les paramètres, exploiter...

Difficultés à gérer / solutions

- Composants petits donc difficiles à manipuler
- Vérification des connexions sur la platine



Le montage est donné

- Vérification du code
- Niveau hétérogène
- Gestion du groupe



Le code est donné



Différentiation du sujet



Les élèves qui ont terminé rapidement aident les autres

Organisation de l'activité expérimentale en fonction de la classe

- Préparation de l'activité expérimentale à la maison
- Faire réaliser le circuit / Donner le circuit
- Faire écrire le code / Donner le code (en entier ou en partie)
- Proposer différents niveaux

En activité expérimentale

- Réalisation du montage
- Ecriture du code et capacité de l'élève à comprendre les messages d'erreurs
- Ajouter des lignes de code, commenter le code
- Interprétation des résultats
- Eventuellement le rangement des composants électriques dans la boîte

A la maison

- Exercices :
- Réaliser le schéma électrique correspondant au montage fourni
 - Commenter le code
 - Ajouter des lignes de codes

- Utiliser un simulateur :
- Réaliser un circuit et écrire un code
 - Vérifier que cela fonctionne correctement
 - Envoyer le lien au professeur

En contrôle

- Réaliser le schéma électrique correspondant au circuit
- Vérifier de la cohérence entre le circuit et le code
- Ajouter des commentaires au code
- Chercher les erreurs dans le code et le modifier
- Ajouter des lignes de codes

Liens

- Histoire de l'Arduino : <https://framablog.org/2011/12/10/arduino-histoire/>
- Tutoriel pas à pas pour la prise en main des microcontrôleurs : <https://phychim.ac-versailles.fr/spip.php?article1076>
- FUNMOOC : "Programmer un objet avec Arduino" : <https://www.fun-mooc.fr/courses/course-v1:MinesTelecom+04017+session06/about>
- Cours en ligne : <https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino?status=published>
- Le blog d'Eksimon : <https://eskimon.fr/>
- Supports pdf pour apprendre pas à pas : http://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php?n=MAIN.ATELIERS
- Les librairies : http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.Librairies
- Références du langage Arduino : http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.Reference
- Fiches de Julien Bobroff (Paris XI) : <https://opentp.fr/card/>
- Exemples d'activités : https://ent2d.ac-bordeaux.fr/disciplines/sciences-physiques/wp-content/uploads/sites/7/2018/10/physique_computationnelle.pdf
- Projets Arduino : <https://create.arduino.cc/projecthub> et <https://www.carnetdumaker.net/articles/categories/tutoriels/arduino/>
- Activités expérimentales de Julien Bobroff (Paris XI) : <https://opentp.fr/>
- Site québécois : <http://edu.mrpigg.ca/>
- Electronique en amateur : <http://electroniqueamateur.blogspot.com/p/arduino.html>