La Martinière Monplaisir

Microcontrôleurs dans les nouveaux programmes de Physique-Chimie : quelques exemples d'activités

Jacques Vince
jvince@ac-lyon.fr

www.prof-vince.fr/arduino

Une nouvelle mode ou un réel apport ?

Cette « nouveauté » ne sera pas omniprésente…

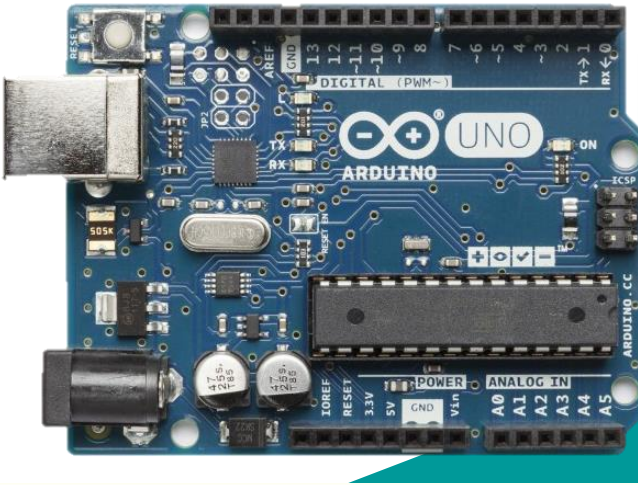
Mickaël M
@MickaIM1

Les cartes Arduino je n'arrive pas à comprendre l'intérêt pour un TP de physique.
Peut-être que je n'ai pas vu assez de sujets originaux, mais pour l'instant je trouve que c'est surtout grignoter du temps de physique pour faire de la techno/SI/info.

> **Julien Bobroff** @jubobroff · 14h
> "Enseigner la physique autrement", un article qu'on vient d'écrire dans @maglarecherche. On y raconte pourquoi il faut inventer de nouvelles formes de TP et comment : #Arduino #smartphone, #openTp #fiction et #smartphonePhysicsChallenge. @UnivParisSaclay @VillebonCharpak

Cette « nouveauté » ne sera pas omniprésente…

ARDUINO™

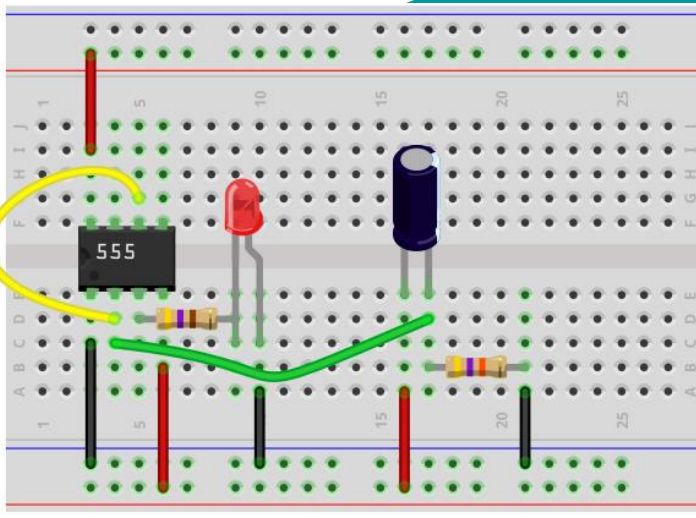


sketch_jan16a | Arduino 1.8.7

Fichier Édition Croquis Outils Aide

sketch_jan16a

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

}

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

}
```



PLAN

- Pourquoi introduire des microcontrôleurs en activités expérimentales ?
- Quelle place dans les programmes ?
- Qu’est-ce qu’un microcontrôleur ?
- Des programmes simples pour débuter
- Pour aller plus loin…

Pourquoi introduire des activités expérimentales avec microcontrôleurs ?

L’apport en physique-chimie

L’apport en physique-chimie

Améliorer la compréhension des concepts sur le plan de :

- l’approche expérimentale :
  - concevoir une chaîne d’acquisition de l’information (capteur → données → traitement → signal de sortie)

- l’approche conceptuelle
  - modéliser
  - simuler

Les programmes

1. En classe de Seconde

2. En classe de Première spécialité

En classe de seconde

Ondes et signaux

Emission et perception d'un son *Signal sonore périodique, fréquence et période.*	**Utiliser un dispositif comportant un microcontrôleur pour produire un signal sonore**.	Concevoir une chaîne d'acquisition
Signaux et capteurs *Capteurs électriques*	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 ...). **Utiliser un dispositif avec microcontrôleur et capteur.**	Concevoir une chaîne d'acquisition Modéliser

En classe de première spécialité

Mouvements et interactions **Description d'un fluide au repos** *Loi de Mariotte*	Tester la loi de Mariotte, **par exemple en utilisant un dispositif comportant un microcontrôleur**.	Concevoir une chaîne d'acquisition Traiter des données Modéliser
Ondes et signaux **Ondes mécaniques** *Célérité d'une onde, retard*	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.	Concevoir une chaîne d'acquisition Traiter des données Modéliser

Absorbance *Mesure d'une concentration*	Déterminer la concentration d'une espèce par une mesure d'absorbance. Mesurer et traiter un signal au moyen **d'une interface de mesure ou d'un microcontrôleur**.	Concevoir une chaine d'acquisition. Traiter des données Modéliser

En classe de terminale spécialité

Modèle du condensateur. Relation entre charge et tension ; capacité d'un condensateur.	Citer des ordres de grandeur de valeurs de capacités usuelles. *Identifier et tester le comportement capacitif d'un dipôle.* *Illustrer qualitativement, par exemple à l'aide d'un microcontrôleur, d'un multimètre ou d'une carte d'acquisition, l'effet de la géométrie d'un condensateur sur la valeur de sa capacité.*
Capteurs capacitifs.	Expliquer le principe de fonctionnement de quelques capteurs capacitifs. *Étudier la réponse d'un dispositif modélisé par un dipôle RC.* *Déterminer le temps caractéristique d'un dipôle RC à l'aide d'un microcontrôleur, d'une carte d'acquisition ou d'un oscilloscope.* **Capacité mathématique** : Résoudre une équation différentielle linéaire du premier ordre à coefficients constants avec un second membre constant.

Capacités expérimentales (bac)

Trois capacités expérimentales sont communes à l'ensemble des thèmes :

- respecter les règles de sécurité liées au travail en laboratoire ;
- mettre en œuvre un dispositif d'acquisition et de traitement de données : microcontrôleur, interface d'acquisition, tableur, langage de programmation ;
- utiliser un logiciel de simulation.

Ondes et signaux

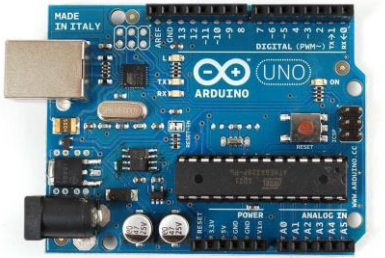
- Commander la production d'un signal grâce à un microcontrôleur.

Qu’est-ce qu’un microcontrôleur ?

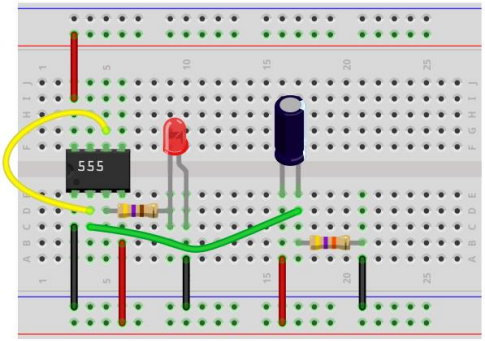
- Présentation
- Le microcontrôleur
- Les cartes
- L’environnement de programmation
- Sécurité

Arduino en résumé

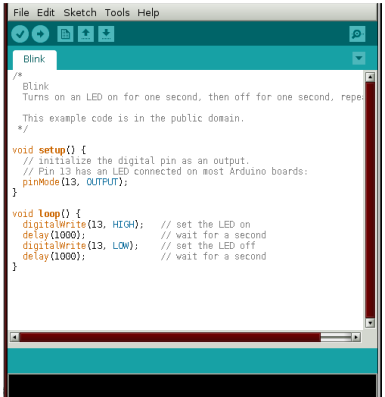
Une carte électronique



Un montage



Un environnement de programmation



Une communauté qui échange

http://arduino.cc/

Microcontrôleur

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires, unités périphériques et interfaces d'entrées-sorties, convertisseurs.

Les microcontrôleurs sont fréquemment utilisés dans les **systèmes embarqués**, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc.

Open source

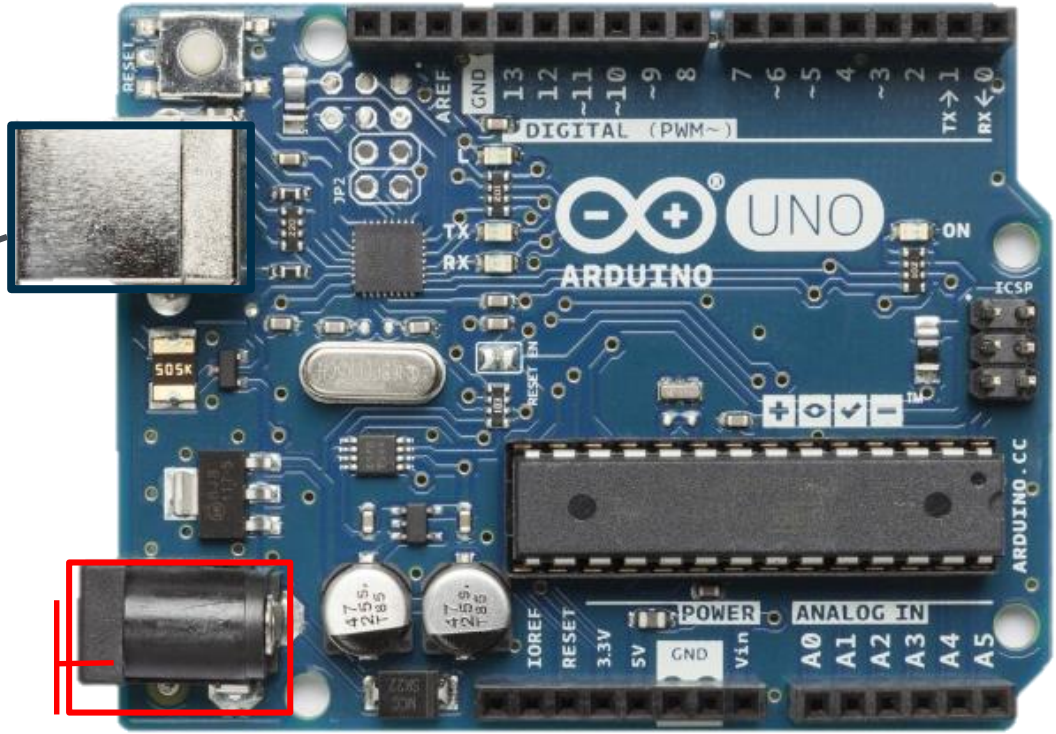
- Le matériel est « open source » :
  →On peut le copier, le fabriquer et le modifier librement.

- Le logiciel est libre :
  →On peut l'utiliser et le modifier librement.

- Sur Internet, on trouve :
  →Une communauté d'utilisateurs.
  →Des guides d'utilisation.
  →Des exemples.
  →Des forums d'entraide.

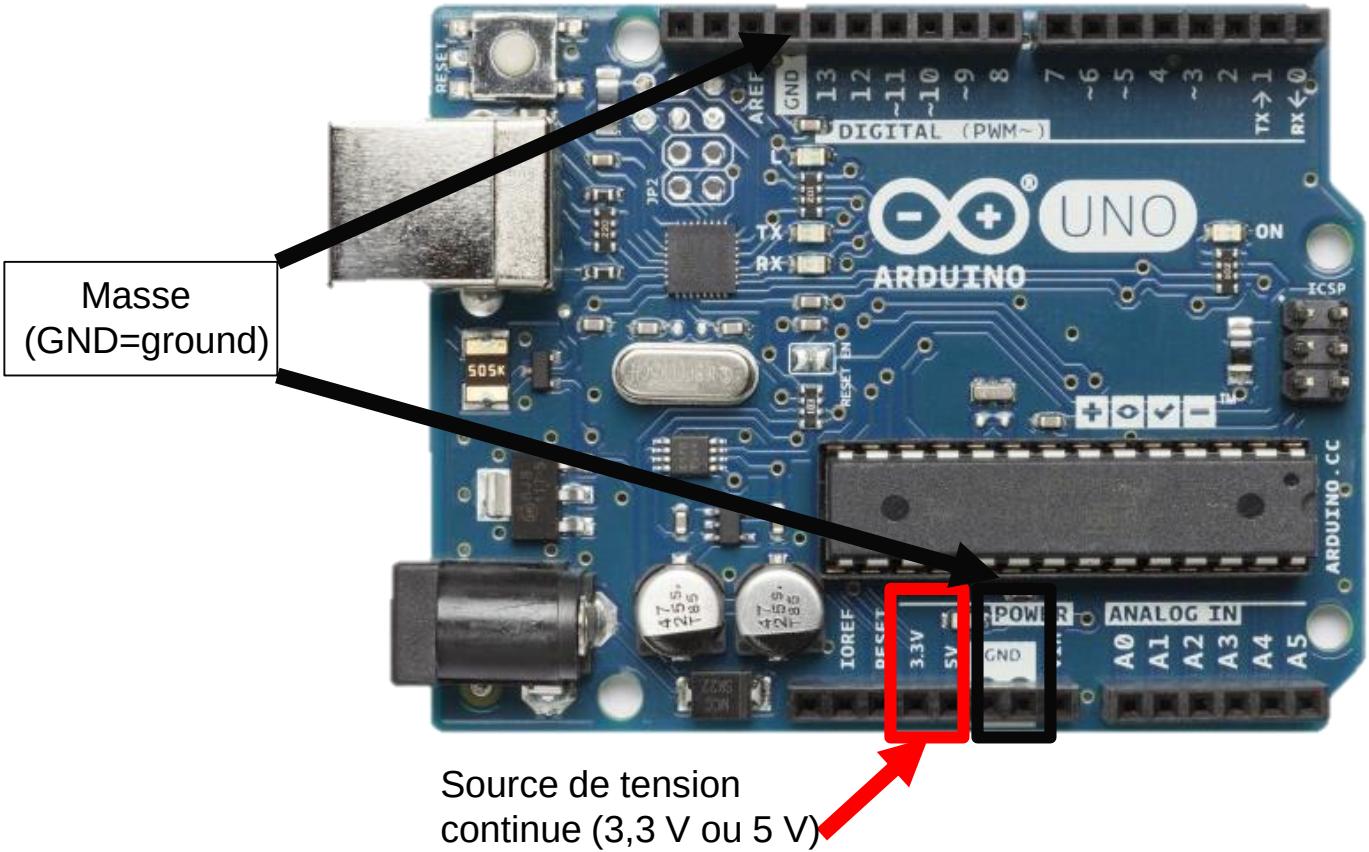
La carte Arduino

Port USB
Pour communiquer avec l'ordinateur et alimenter



Alimentation externe (entre 7 et 12 V) par piles : la carte devient nomade

La carte Arduino

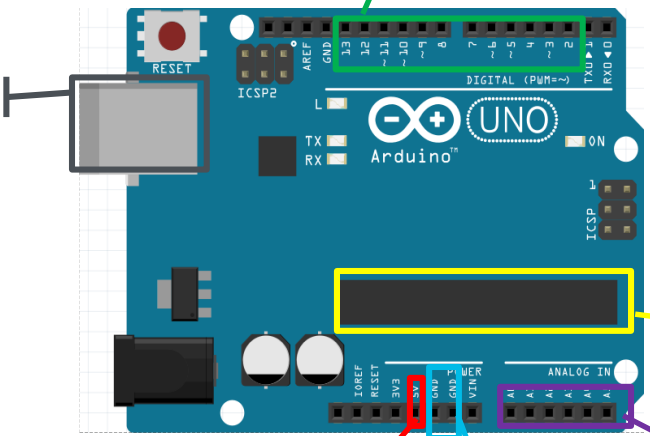


Les entrées et les sorties

- Un langage basé sur C/C++
- 14 entrées/sorties numériques
- 6 entrée analogiques
- Mémoire flash 32 ko
- Vitesse d'horloge 16 MHz
- Pas de communication wifi ou BT

Entrées (INPUT)	Sorties (OUTPUT)
Lues par le µC, pour connaître l'état du système auquel il est relié	**Déclenchement contrôlé par le µC => Source de tension**
Entrée numérique (DIGITAL) (ports 2 à 13) : lit l'état du port (LOW ou HIGH) *Exemple : interrupteur*	**Sortie numérique (DIGITAL)** (ports 2 à 13) : génère une tension de 0V (LOW) ou 5V (HIGH) *Exemples : moteurs, LED, buzzer*
Entrée analogique (ports A0 à A5) Voltmètre qui mesure une tension entre 0 et 5V (10 bits donc 1023 valeurs possibles puis CAN) *Exemples : photorésistance, thermistance*	

12 broches numériques (entrées/sorties)



Port USB

Microcontrôleur

Alimentation 5 V

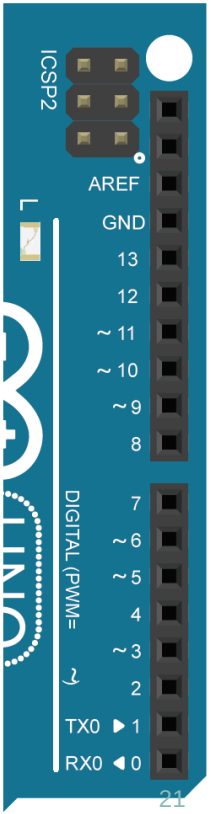
masse

6 broches analogiques (entrées)

Les sorties numériques

Il faut déclarer la broche utilisée en sortie numérique : broches numérotées de 2 à 13.

2 états possibles pour ces sorties : Haut (HIGH = 5 V) ou bas (LOW = 0 V).



Le courant est limité à 40 mA sur une sortie numérique (200 mA sur l'ensemble de ces sorties), 500 mA sur la sortie 5V et 50 mA sur la sortie 3,3 V.

Les sorties numériques

Ces broches peuvent aussi être des entrées !

Il faut les paramétrer dans le sketch ARDUINO.

Instruction à utiliser pour paramétrer les broches en sortie (=OUTPUT) :

pinMode(X,OUTPUT)

X : numéro de la broche compris entre 2 et 13

Pour fixer à l'état haut ou bas la broche X, on utilise l'instruction suivante :

digitalWrite(X,HIGH) digitalWrite(X,LOW)

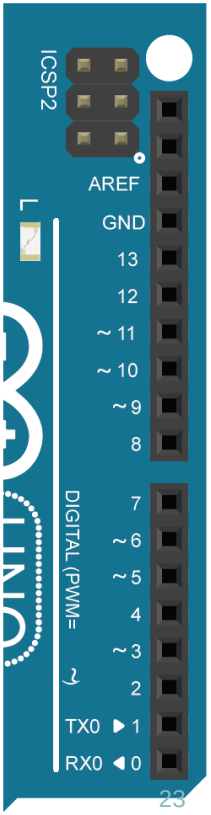
Respecter les minuscules et majuscules des instructions.

Le point virgule est le séparateur d'instruction.

Les sorties numériques

Il faut déclarer la broche utilisée en sortie numérique : broches numérotées de 2 à 13.

2 états possibles pour ces sorties : Haut (HIGH = 5 V) ou bas (LOW = 0 V).



Le courant est limité à 40 mA sur une sortie numérique (200 mA sur l'ensemble de ces sorties), 500 mA sur la sortie 5V et 50 mA sur la sortie 3,3 V.

Les entrées analogiques

Une entrée analogique est un voltmètre. Le microcontrôleur ne travaille cependant qu'avec des nombres binaires, il est alors utilisé un convertisseur analogique numérique (CAN).

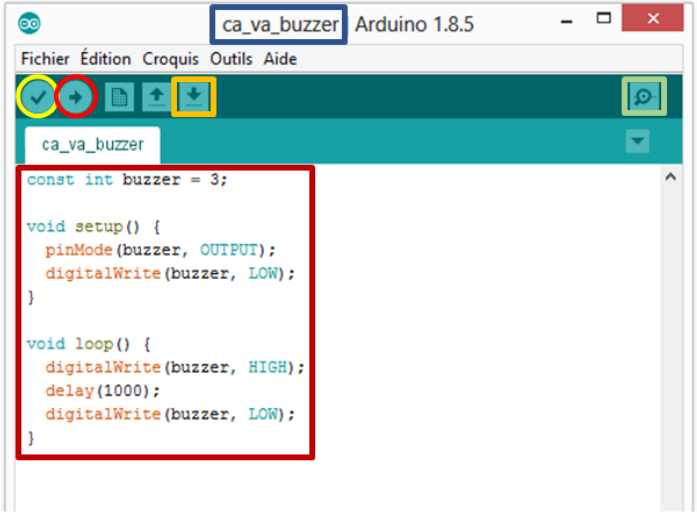
Tension mesurée (analogique) → CAN → Nombre binaire (numérique)

Le CAN des entrées analogiques travaille sur 10 bits.
La valeur binaire maximale est 1111111111 (= 1023 en base 10)

L'instruction analogRead(A0) permet de lire le nombre en base 10 associé au binaire de la tension mesurée sur A0

- Créer un nouveau fichier dans le programme Arduino (menu fichier puis nouveau)
- Saisir le code permettant de faire afficher en continu dans le moniteur série les valeurs lues sur l'entrée A0 du potentiomètre : `Serial.println(analogRead(A0));`
- Faire varier la valeur puis observer les valeurs lues dans le moniteur série

L’environnement de programmation



Nom du fichier

Enregistrer

Compilation

Téléverser

Moniteur

Programme

Télécharger l’environnement de programmation (IDE) à l’adresse officielle suivante : https://www.arduino.cc/en/Main/Software

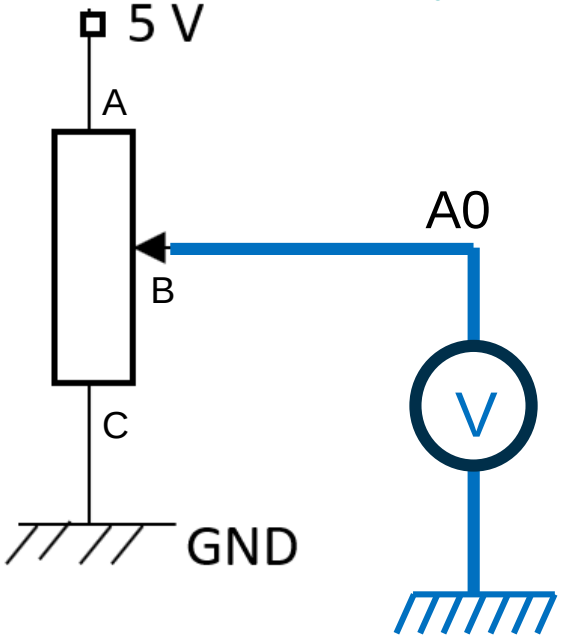
Structure d'un programme

- Ouvrir le logiciel Arduino ;
- Vérifier dans le menu « Outils » puis « Type de carte » que la carte Arduino/Genuino Uno est cochée ;
- Sélectionner le port USB utilisé : menu « Outils » puis « Port », choisir celui ou Arduino/Genuino Uno apparait ;

1. En-tête déclarative	Déclaration des constantes Déclarations des variables
2. Fonction Setup	Configuration initiale Déclaration de variables locales Configuration des broches Initialisation des variables Initialisation des fonctionnalités Initialisation des interruptions
3. Fonction loop	Instructions exécutées en boucle Boucle sans fin

Les entrées analogiques

- Réaliser le montage ci-dessous qui permet de faire faire varier la tension U_{BC}



- Créer un nouveau fichier dans le programme Arduino (menu fichier puis nouveau)
- Dans le bloc void setup(), saisir l'instruction : Serial.begin(9600);
- Saisir le code permettant de faire afficher en continu dans le moniteur série les valeurs lues sur l'entrée A0 du potentiomètre : Serial.println(analogRead(A0));
- Faire varier la valeur puis observer les valeurs lues dans le moniteur série

Les entrées analogiques

Valeur mesurée (analogique)	Valeur lue (numérique)
0 V	0
5,0 V	1023
U	valNum

$$U = valNum \times \frac{5.0}{1023}$$

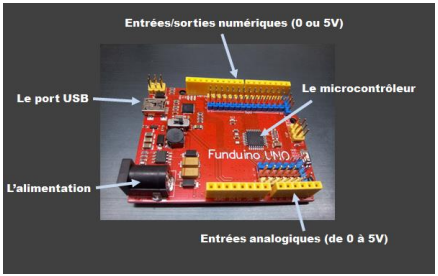
On peut faire afficher la tension U dans le moniteur série en modifiant votre code.

On utilisera la variable U qui est un nombre à virgule (un flottant) qui sera déclarée avec l'instruction suivante : float U

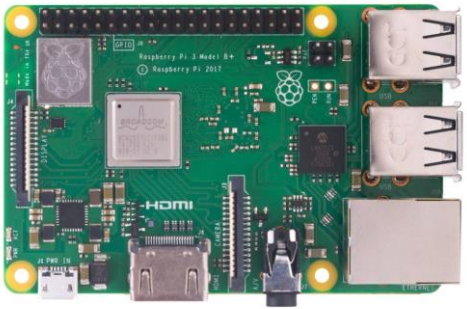
```
int valNum;
float U;

void setup() {
  Serial.begin(9600);
}

void loop() {
  valNum=analogRead(A0);
  U=5.0*valNum/1023;      // saisir 5.0 et non 5
  Serial.println(U);
}
```

Autres cartes

La carte Funduino



La carte Raspberry Pi

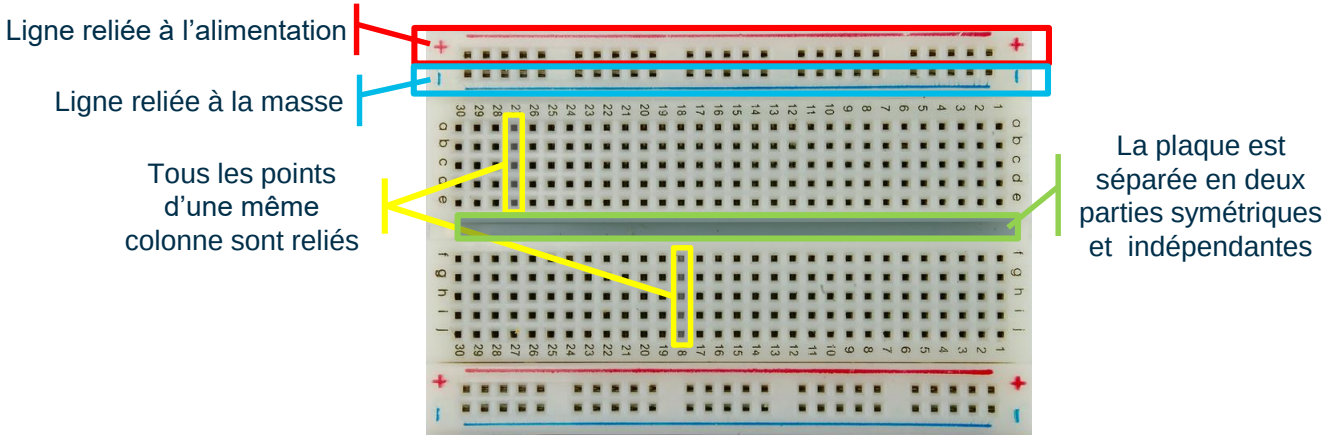


- Programmable en Python
- Mémoire flash 2 à 16 Go (vs 32 ko pour Arduino)
- Vitesse d'horloge 700 MHz (vs 16 MHz pour Arduino)

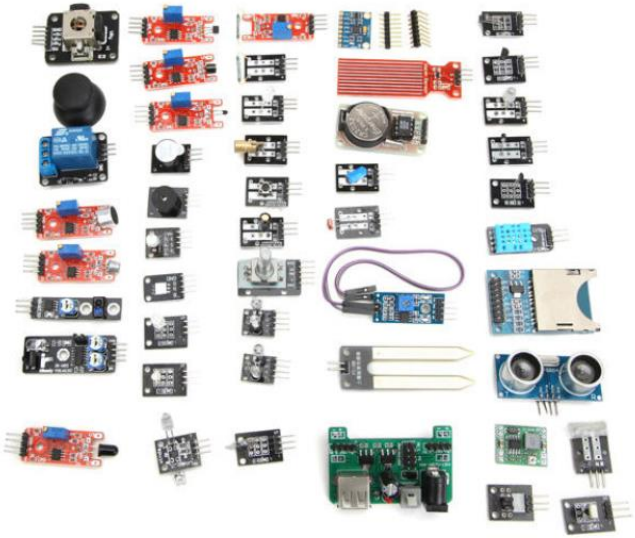
MAIS :

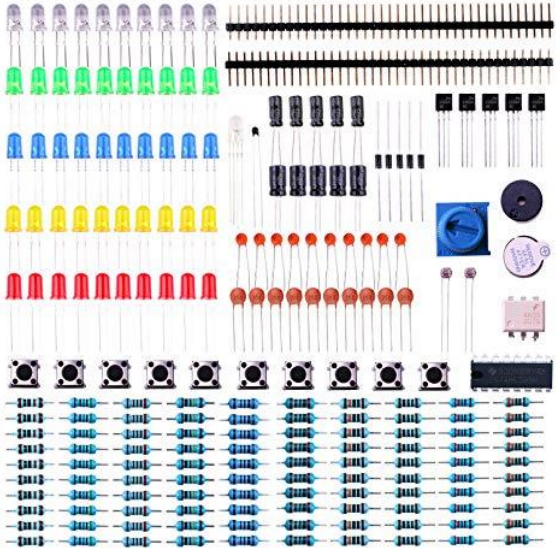
- Réponse en temps réel et en analogique moins performante que pour Arduino
- Moins de capteurs "simples"
- Communauté "débutante" moins développée que pour Arduino

La platine de câblage (breadboard)

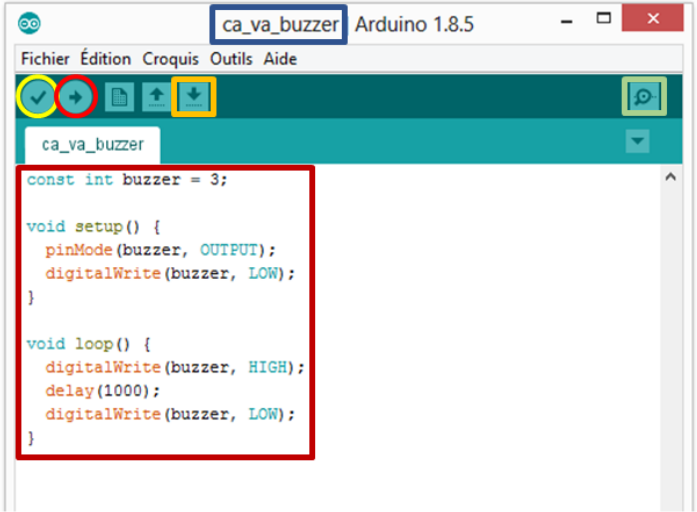


Des capteurs et des composants électroniques





L’environnement de programmation



Nom du fichier

Enregistrer

Compilation

Téléverser

Moniteur

Programme

Télécharger l’environnement de programmation (IDE) à l’adresse officielle suivante : https://www.arduino.cc/en/Main/Software

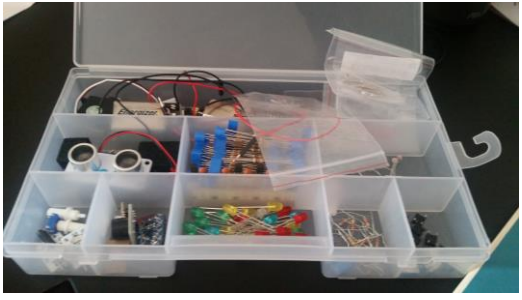
Sécurité

- 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.
- 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 êtres 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.

Organisation matérielle

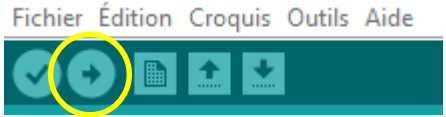


Déroulement d'une activité expérimentale avec microcontrôleur

→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 le breadboard

→ Le montage est donné ou module de type Groove

- Vérification du code → Le code est donné
- Niveau hétérogène → Différentiation du sujet
- Gestion du groupe → Les élèves qui ont terminé rapidement aident les autres

Des programmes simples pour débuter

1. Allumer une LED
2. Faire clignoter une LED
3. Capteur de lumière
4. Allumage automatique
5. Radar de recul (X1)
6. Feu piéton

Puis des exercices

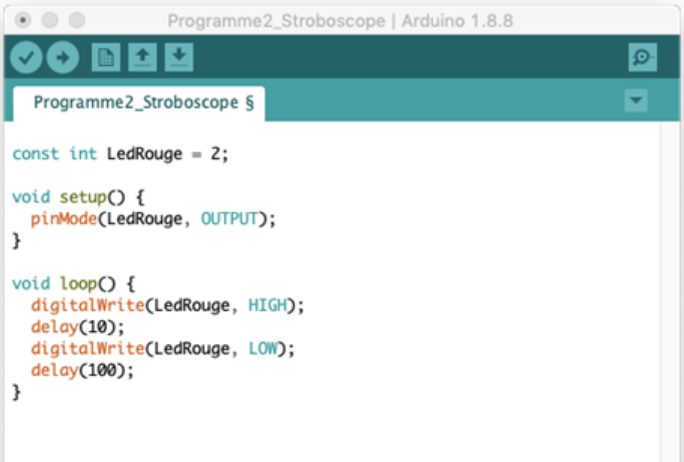
N’oubliez pas que votre meilleur ami, c’est…

l’ESSAI-ERREUR !

Puis pour s'amuser

- Jouer de la musique… au clair de la lune puis un PIANO
- Le principe du thérémine
- Un capteur d'humidité…
- Vitesse du son
- Modèle d'un spectrophotometre
- Loi de Mariotte

Programme 1 : Faire clignoter une DEL

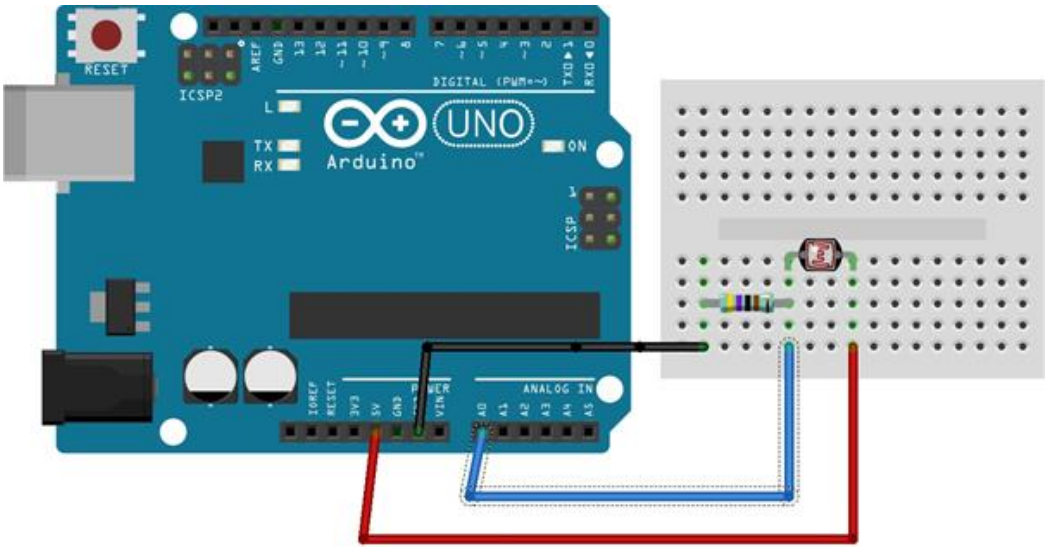


- La fonction delay()
- La fonction loop()

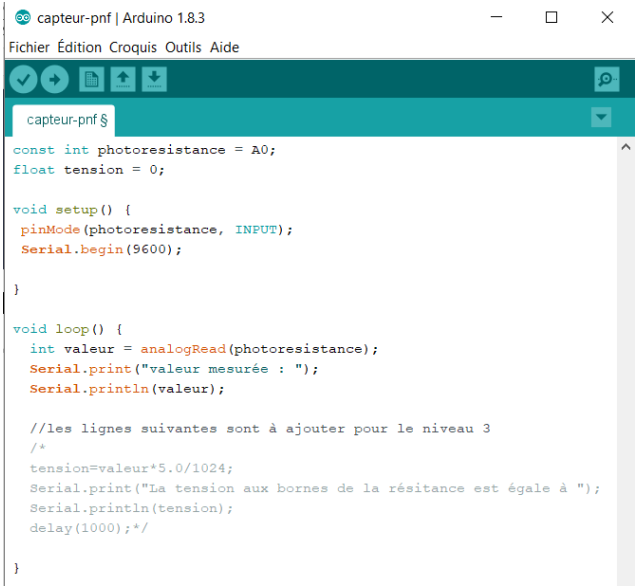
Programme 1 : Faire clignoter une DEL

- Niveau 1 : Brancher la LED sur la bonne broche, téléverser le programme, la DEL doit clignoter
- Niveau 2 : Modifier le programme de sorte à modifier la fréquence de clignotement.
- Niveau 3 : Modifier le programme de sorte à faire clignoter alternativement une LED rouge et une LED verte.

Programme 2 : Capteur de lumière



Programme 2 : Capteur de lumière

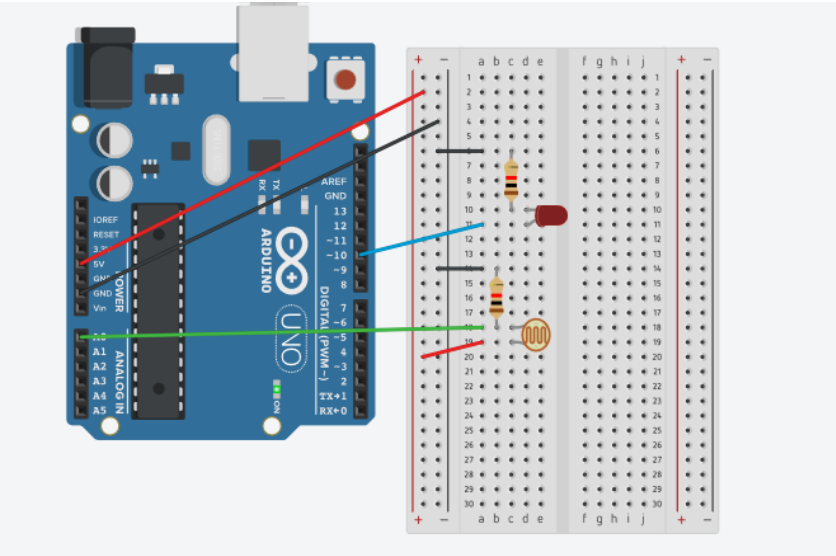


- Les entrées analogiques
- La fonction digitalRead() / analogRead()
- La fonction Serial.print() / Serial.println()

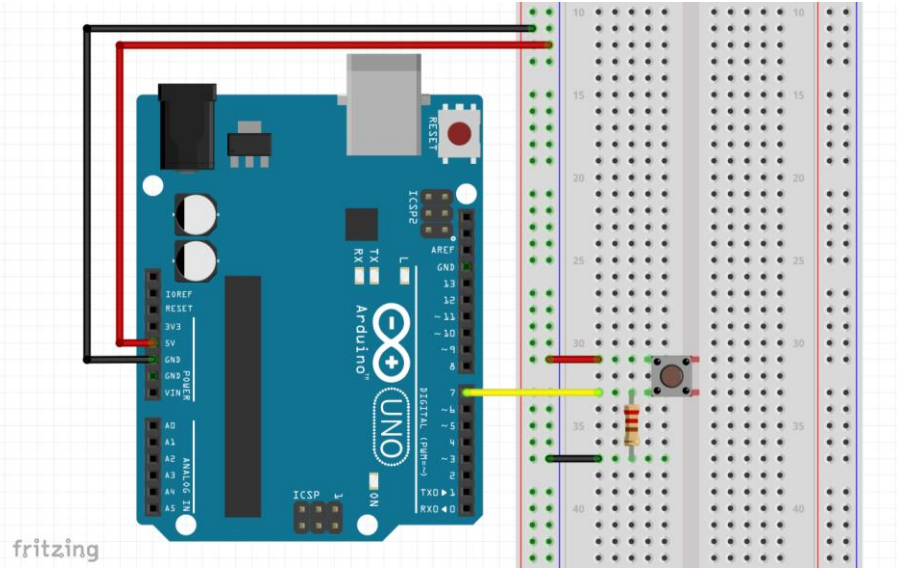
Programme 2 : Capteur de lumière

- Niveau 1 : Brancher la photorésistance sur l'entrée analogique A0, téléverser le programme puis ouvrir le moniteur série (bouton en haut à droite de la fenêtre)
- Niveau 2 : Modifier le programme de sorte à diminuer le temps entre deux mesures à 100ms au lieu de 1000ms... Téléverser puis dans l'onglet "Outils", cliquer sur "Traceur série" afin d'afficher la courbe de mesure.
- Niveau 3 : Compléter les lignes du programme de sorte à afficher dans le moniteur série la tension aux bornes de la photorésistance.
  La grandeur mesurée par la photorésistance est stockée sur 10 bits ce qui explique qu'elle prenne des valeurs entre 0 et 1023 (2^10 possibilités).
  En réalité, ces valeurs correspondent à 1024 valeurs de tension aux bornes de la résistance comprises entre 0 et 5V.

Programme 3 : Détecteur de présence



Programme 4 : Le feu piéton



1 bouton poussoir
1 résistance de 1 $k\Omega$
Fils
Platine de prototypage

Connecter correctement l'interrupteur

Programme 4 : Le feu piéton

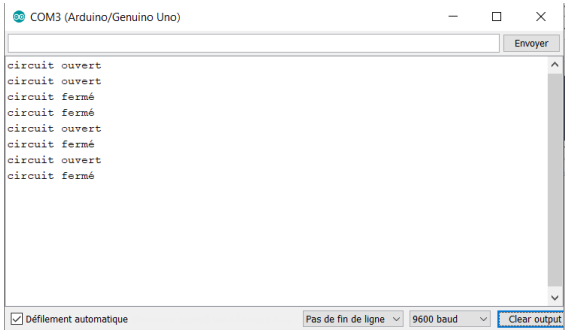
feupieton-pnf | Arduino 1.8.3

Fichier Édition Croquis Outils Aide

feupieton-pnf §

```
const int bouton = 7;
int etatBouton = 0;

void setup() {
pinMode(bouton,INPUT);
Serial.begin(9600);
}

void loop() {
  etatBouton=digitalRead(bouton);
  if(etatBouton==HIGH){
    Serial.println("circuit fermé");
  }
  else {
    Serial.println("circuit ouvert");
  }
  delay(1000);
}
```

- La commande “digitalRead”
- La condition if
- La condition else
- Le moniteur série



Liens pour se former

- Fiches de Julien Bobroff (Paris XI) : https://opentp.fr/card/
- Cours en ligne : https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino?status=published
- Exemples d'activités : https://ent2d.ac-bordeaux.fr/disciplines/sciences-physiques/wp-content/uploads/sites/7/2018/10/physique_computationnelle.pdf
- 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
- 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

Pour aller plus loin…

→ Envoyer les mesures réalisées avec un capteur et un microcontrôleur dans un programme Python

Exemple : Activité expérimentale sur la loi de Mariotte

→ Réaliser une série de mesures d'une grandeur physique et tracer un histogramme avec Python

(programme : Exploiter une série de mesures indépendantes d'une grandeur physique : histogramme, moyenne et écart-type.)

Exemple : Activité expérimentale sur la mesure de la vitesse du son

→ Ressources : http://www.f-legrand.fr/scidoc/