ARDUINO

DOSSIER RESSOURCE
POUR LA CLASSE

Sommaire

1. Présentation
2. Exemple d’apprentissage
3. Lexique de termes anglais
4. Reconnaître les composants
5. Rendre Arduino autonome
6. Les signaux d’entrée et de sortie
7. Câblage des composants

Arduino, qu’est-ce ?

Une carte électronique

Un environnement de programmation

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

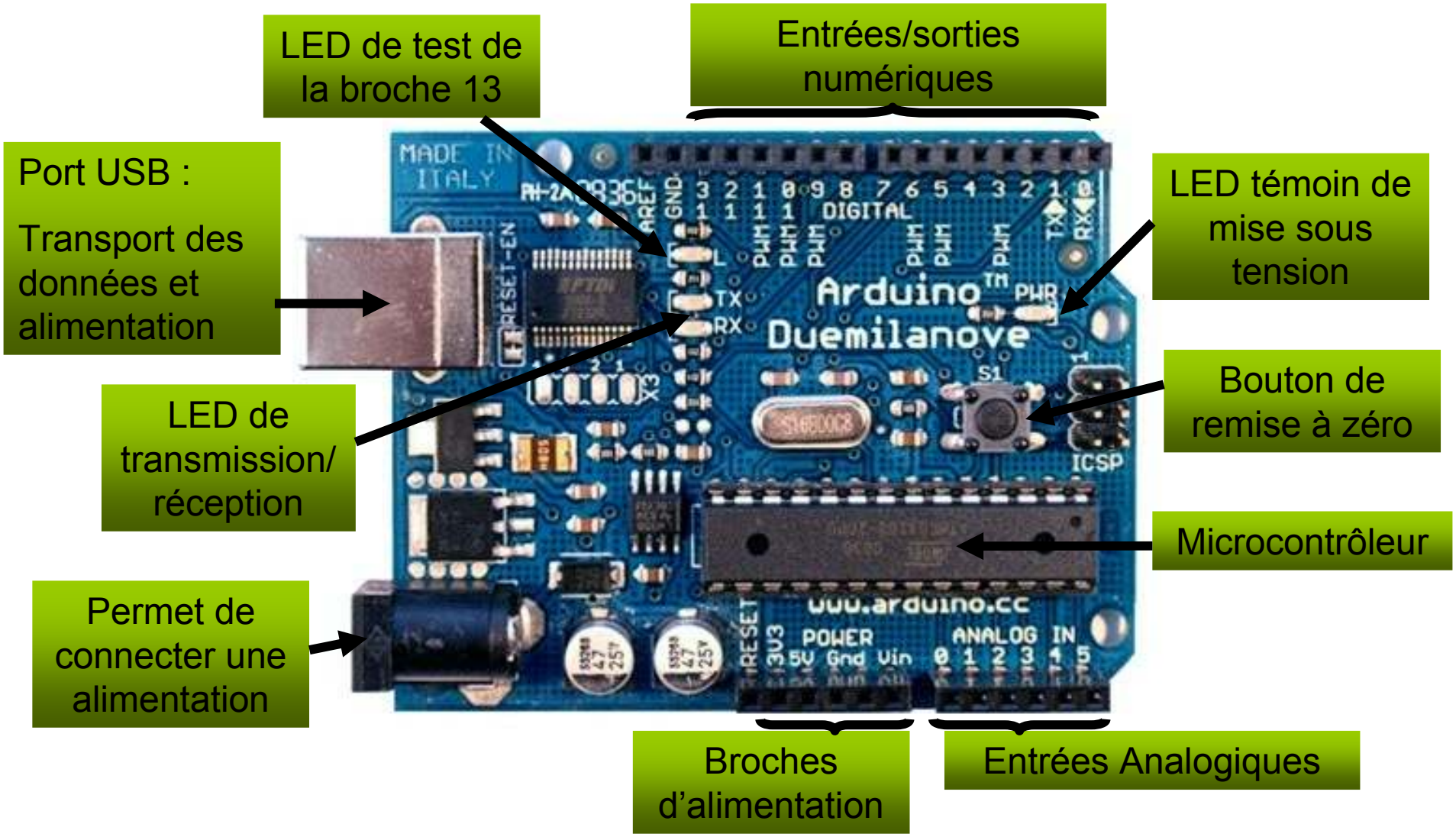
Arduino, une philosophie

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

Le logiciel est libre :
• On peu l'utiliser et le modifier librement.

Sur l'internet, on trouve :
• Une communauté d'utilisateurs.
• Des guides d'utilisation.
• Des exemples.
• Des forums d'entraide.

Arduino, la carte électronique



Arduino, le logiciel de programmation

Icône présente sur le bureau

Chemin d'accès des programmes : D: /Espace élève/Ressources/Quatrième/Programmes Arduino

Arduino, structure d'un programme

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne.
La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code.

Structure d'un programme

Il y a trois phases consécutives:

Commentaires multilignes pour se souvenir du patch ==>

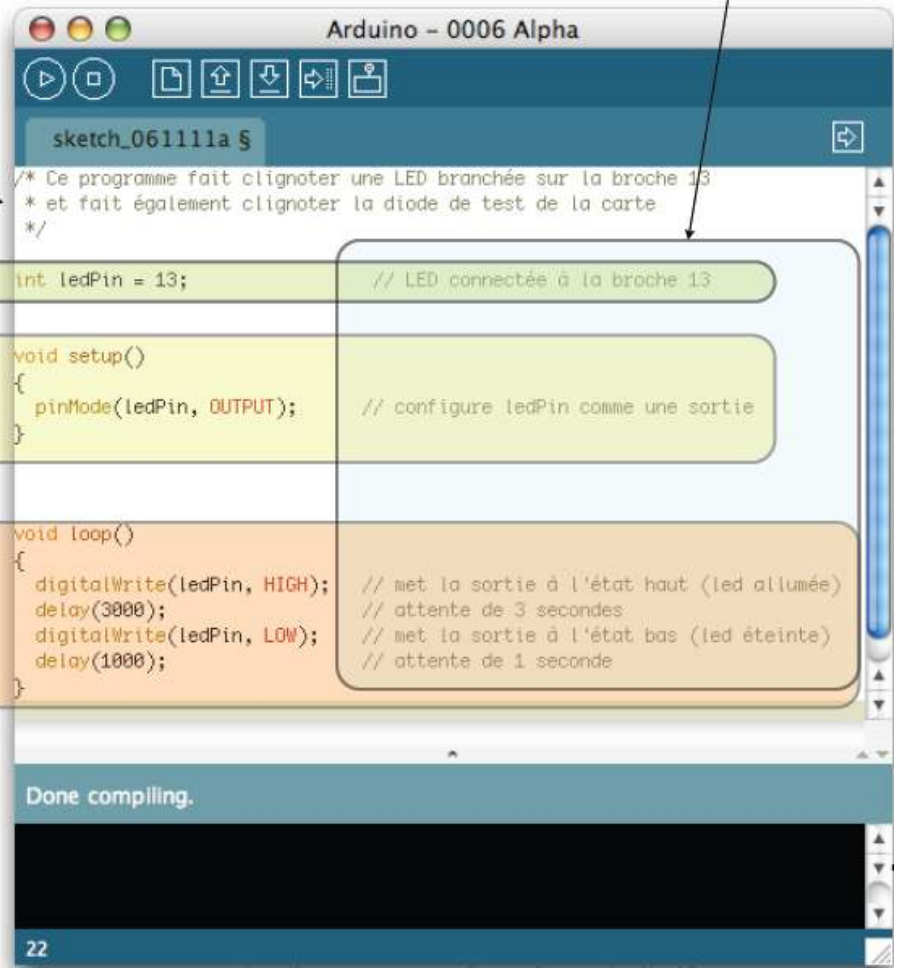
1/La définition des constantes et des variables

2/La configuration des entrées et sorties
void setup()

3/La programmation des interactions et comportements
void loop()

Une fois la dernière ligne exécutée, la carte revient au début de la troisième phase et recommence sa lecture et son exécution des instructions successives. Et ainsi de suite.

Cette **boucle** se déroule des milliers de fois par seconde et anime la carte.



Arduino, le principe de fonctionnement

1. On conçoit ou on ouvre un **programme** existant avec le logiciel Arduino.

2. On **vérifie** ce programme avec le logiciel Arduino (compilation).

3. Si des **erreurs** sont signalées, on **modifie** le programme.

4. On **charge** le programme sur la carte.

5. On **câble** le montage électronique.

6. L'exécution de programme est **automatique** après quelques secondes.

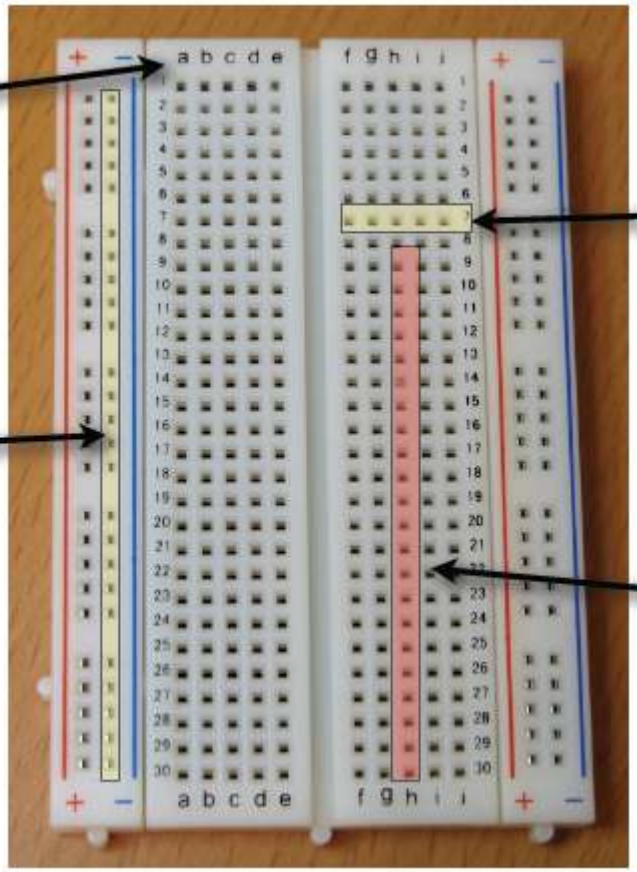
7. On **alimente** la carte soit par le port USB, soit par une source d'alimentation
8. autonome (pile 9 volts par exemple).

9. On **vérifie** que notre montage fonctionne.

Plaque d’essai

La plaque d’essai sans soudure nous permet de réaliser rapidement un montage électronique en insérant les pattes des composants et les fils dans les trous.

Les lettres repèrent les colonnes et les nombres repèrent les lignes



Chaque ligne numérotée relie 5 trous

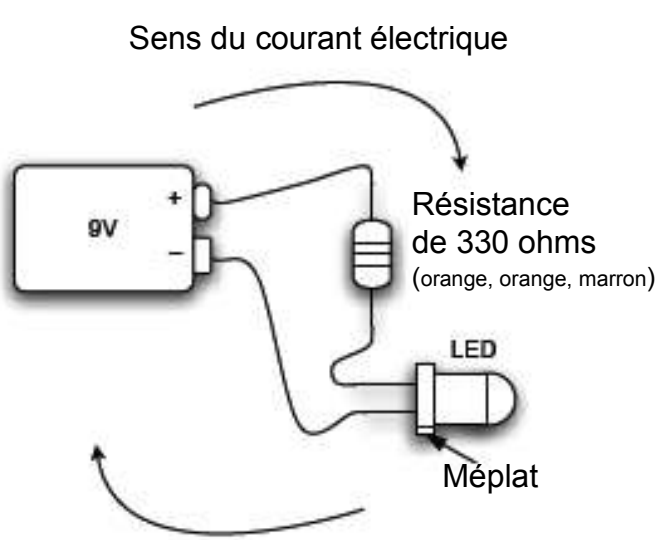
Les trous de la colonne « + » sont reliés et les trous de la colonne « - » sont reliés.

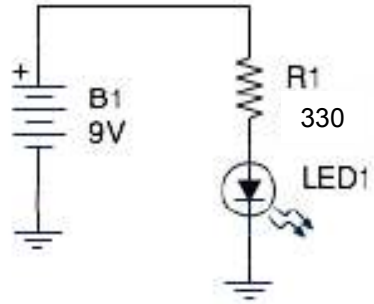
Ces colonnes sont dédiées à l’alimentation du circuit.

Les trous ne sont pas reliés en colonne.

Exemple d'apprentissage : allumer une LED

Une LED est polarisée
Une résistance n'a pas de sens imposé





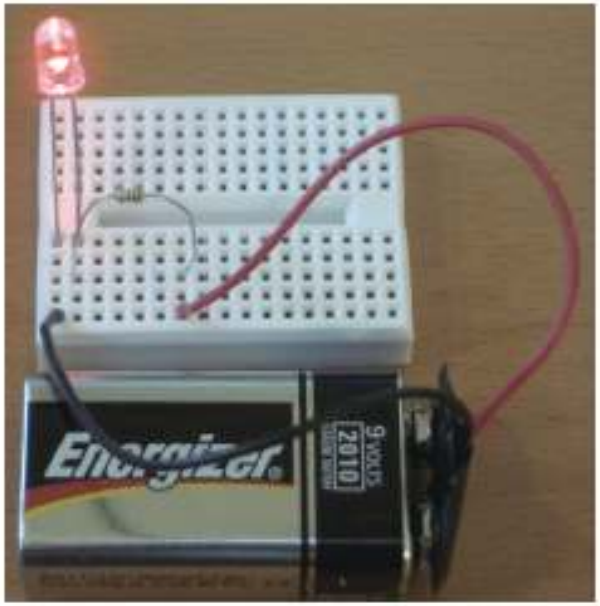
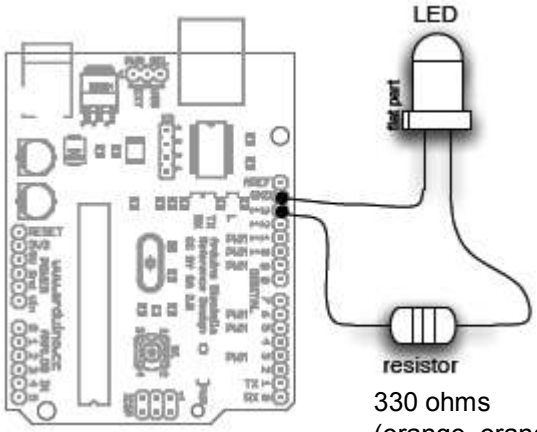


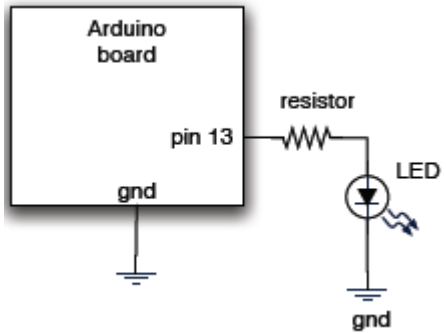
Schéma de câblage Schéma normalisé Montage d'essai

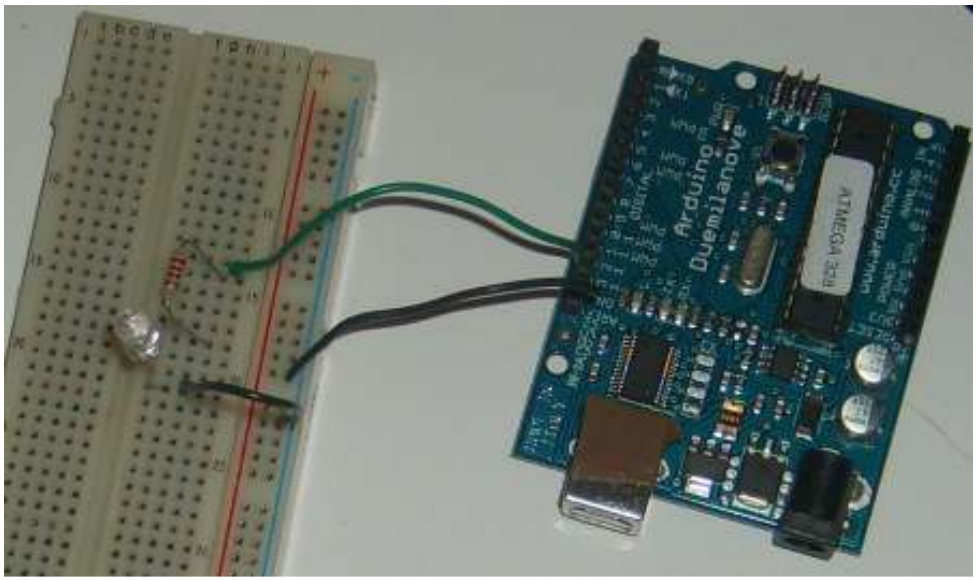
Pour éteindre la LED, il faut ouvrir le circuit à l'aide d'un interrupteur.

Exemple d'apprentissage : faire clignoter une LED



330 ohms
(orange, orange, marron)

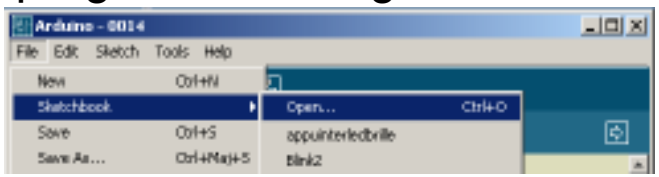




Exemple d'apprentissage : faire clignoter une LED

Etape 1 : lancer le logiciel Arduino

Etape 2 : ouvrir le programme « Clignoter LED »



Etape 3 : compiler le programme

Etape 4 : si il n'y a pas d'erreurs, brancher la carte Arduino avec le cordon USB

Etape 5 : charger le programme dans la carte Arduino

Exemple d'apprentissage : faire clignoter une LED

Commentaires
Toujours écrire des commentaires sur le programme: soit en multiligne, en écrivant entre des /****/, soit sur une ligne de code en se séparant du code avec //

Définition des variables:
Pour notre montage, on va utiliser une sortie numérique de la carte, qui est par exemple la 13 ème sortie numérique. Cette variable doit être définie et nommée ici: on lui donne un nom arbitraire BrocheLED . Le mot de la syntaxe est pour désigner un nombre *entier* est **int**

Configuration des entrées-sorties void setup():
Les broches numériques de l'Arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques. Ici on va configurer BrocheLED en sortie.
pinMode (nom, état) est une des quatre fonctions relatives aux entrées-sorties numériques.

Programmation des interactions void loop():
Dans cette boucle, on définit les opérations à effectuer, dans l'ordre:
- digitalWrite (nom, état) est une autre des quatre fonctions relatives aux entrées-sorties numériques.
- delay(temps en millisecondes) est la commande d'attente entre deux autres instruction
- Chaque ligne d'instruction est terminée par un point virgule
- Ne pas oublier les accolades, qui encadrent la boucle.

(Syntaxe en marron, paramètres utilisateur en vert)

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



int BrocheLED = 13;     // Définition de la valeur 13 et du nom de la broche à
utiliser




void setup()
{
  pinMode(BrocheLED, OUTPUT);      // configure BrocheLED comme une
sortie
}



void loop()
{
  digitalWrite(BrocheLED, HIGH);   // met la sortie num. à l'état haut (led
allumée)
  delay(3000);                     // attente de 3 secondes
  digitalWrite(BrocheLED, LOW);    // met la sortie num. à l'état bas (led
éteinte)
  delay(1000);                     // attente de 1 seconde
}
```

Lexique de termes anglais

ANALOG : Analogique.
AREF : Abréviation pour Analog REFérence, référence analogique.
AVAILABLE : Disponible.
BEGIN : Début.
BIT : bit, unité d'information informatique pouvant prendre soit la valeur 0 soit la valeur 1.
BUFFER : Tampon, dans le sens de "zone tampon". Mémoire temporaire
BYTE : Octet, soit un groupe de 8 bits.
bps : Abréviation pour Bits Per Second, Bits Par Seconde. Attention, abréviation toujours en minuscules.
BREADBOARD: plaque d'expérimentation
CAPACITOR: condensateur
CHAR : Pour CHARacter, caractère (typographique). Type de variable d'une taille d'un octet. C'est un synonyme de "byte" utilisé pour déclarer des variables stockant un caractère ou des chaines de caractères.
DEFINE : Définit.
DIGITAL : Numérique.
DO : Faire.
FALSE : Faux.
FOR : Pour. Jusqu'à ce que.
GND : Abréviation pour GrouND, la terre. C'est la masse, 0 Volt.
HIGH : Haut.
ICSP : Abréviation pour In Cicuit Serial Programming, programmation série sur circuit.
IF / THEN/ ELSE : Si / Alors / Sinon.
IN : Souvent l'abréviation pour INput, Entrée. Est toujours en rapport avec le sens extérieur vers carte Arduino.
INCLUDE : Inclut.
INPUT : Entrée.

Lexique de termes anglais

IS : Est (souvent dans le sens d'une question : Est ?).
INT : Abréviation pour INTeger, entier. Groupe de 16 bits, 2 octets groupés, considérés comme représentant un nombre entier négatif ou positif.
LONG : Abréviation pour "entier long". Groupe de 32 bits, 4 octets groupés, considérés comme représentant un nombre entier négatif ou positif.
LOOP : Boucle.
LOW : Bas.
OUT : Souvent l'abréviation pour OUTput, Sortie.
Est toujours en rapport avec le sens carte Arduino vers extérieur.
OUTPUT : Sortie.
PIN : Broche.
POWER : Puissance, alimentation.
PWM : Abréviation de (Pulse Width Modulation),
soit Modulation en Largeur d'Impulsion.
PWR : Abréviation pour PoWeR, puissance, alimentation.

Lexique de termes anglais

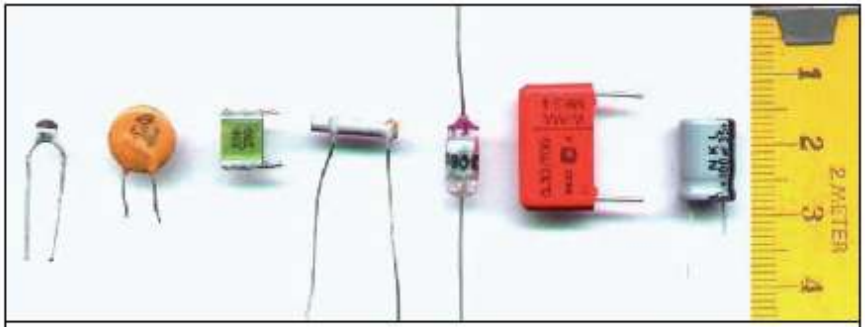
READ: Lire.
RESISTOR: résistance**.**
RELAY: relais.
RX : Abréviation pour Receive, réception.
SERIAL : Série.
SETUP : Initialisation.
SENSOR: capteur
SWITCH : basculer, interrupteur
TRUE : Vrai.
TX: Abréviation pour Transmit, transmission.
WIRE: câble, fil.
WHILE : Tant que.
WORD : mot, soit dans le sens de langage ;
soit dans le sens d'un groupe de 16 bits,
2 octets groupés considérés comme représentant un nombre entier positif (>= 0).
WRITE: Ecrire.

Reconnaître les composants

La résistance

La résistance s'oppose au passage du courant, proportionnellement à sa " résistance" exprimée en Ohm. Un code de couleurs, ci dessous permet de reconnaître cette valeur.

Symbole européen



Les condensateurs

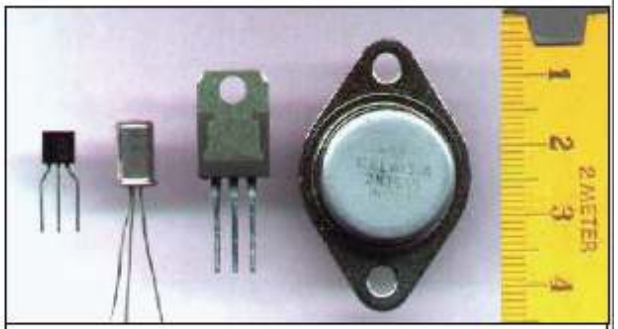
Les condensateurs peuvent stocker un peu de courant si on les charge, mais comme un tonneau percé, ils renvoient ce courant instantanément si ils sont branchés à un organe consommateur de courant.

Ils peuvent être polarisés ou non, dans des boitiers très divers. Leur valeur s'exprime en Farad (F)

non polarisé polarisé

Symbole

Reconnaissance de leur valeur:
A cause de la diversité des modèles, se reporter aux ressources sur le web



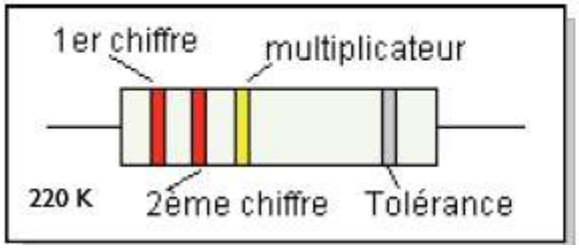
Le transistor

Le transistor est généralement utilisé comme une sorte de multiplicateur de puissance: lorsqu'on lui fait passer un courant faible, mais variable dans un de ses 3 pattes, il autorise proportionnellement le passage d'un gros courant dans une autre des 3 pattes.

transistor NPN transistor PNP

Code des couleurs des résistances

au delà de 1000 Ohms, on parle en KiloOhms, par exemple 10 K est 10 KiloOhms, puis en MegaOhms notés M



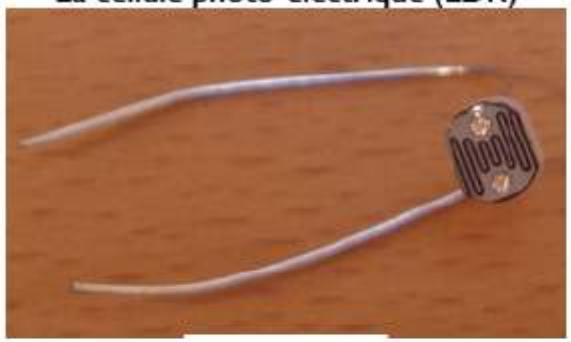
Chiffre	0	1	2	3	4	5	6	7	8	9	Or	Argent
Multiplicateur	1	10	10^2 100	10^3 1000	10^4 10000	10^5	10^6					
Précision	20%										5%	10%

Reconnaître les composants

L'interrupteur

L'interrupteur ouvre ou ferme un circuit. Il y a toutes sortes d'interrupteurs.
Sur l'Arduino, utiliser un interrupteur pour déclencher un événement nécessite d'utiliser un composant supplémentaire: une résistance de 10K ohms. Voir " Montages d'électronique interactive".

La cellule photo-électrique (LDR)



La cellule photo-électrique (LDR)

C'est une résistance variable, en fonction de la luminosité qu'elle reçoit. Sa résistance diminue quand elle reçoit de la lumière. On s'en sert donc de capteur de luminosité. Non polarisée. Pour lire sa valeur avec une Arduino, il faut également l'associer avec une résistance équivalente à sa résistance maxi (dans le noir) Voir " Montages d'électronique interactive".

Le piezo

Le transducteur piezo-électrique est un composant réversible: il peut aussi bien être utilisé en capteur de chocs ou de vibrations qu'en actionneur pouvant émettre des sons stridents parfois modulables.

Reconnaître les composants

Le servo moteur

Le servo-moteur est un moteur (rotatif) qui peut effectuer des rotations très précises (dans une portion de tour seulement) et en un certain nombre de pas (de micro-déplacements). Il y a toutes sortes de servo moteurs.. Un des avantages des servo moteurs est sa possibilité de maintenir avec force une position donnée. On peut piloter des rotations avec l'Arduino, quelques fois directement avec la carte si le moteur n'est pas trop puissant, sinon en passant par un montage associé.

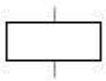
Le potentiomètre

Le potentiomètre

Le potentiomètre, rotatif comme ici, ou à glissière, est une résistance variable. Entre les extrémités, il y a la résistance maximale. La patte centrale est le curseur. C'est la résistance entre cette patte centrale et une extrémité que l'on peut faire varier en tournant le bouton. Le potentiomètre est donc un capteur. Il se branche sur les entrées analogiques de l'Arduino.
De très nombreux capteurs sont basés sur le principe de résistance variable et se cablent presque de la même façon: la cellule photo-électrique, le capteur de pression, le fil résistif, etc

Le relais

Le relais



Le relais est un composant à 4 broches minimum. C'est un interrupteur que l'on peut commander en envoyant un petit courant. Au repos, il est normalement fermé, ou normalement ouvert, selon le modèle. On peut s'en servir avec l'Arduino pour commander des machines en haute tension (230V par exemple), ou pour déclencher toute machine ou lumière.

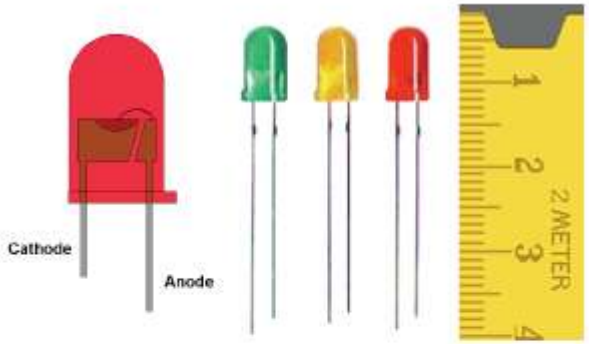
Reconnaître les composants

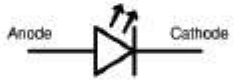
La diode

La diode

La diode ne laisse passer le courant que dans un seul sens. C'est un composant polarisé: on reconnait toujours son anneau coloré d'un coté du composant, correspondant à la cathode.

La LED

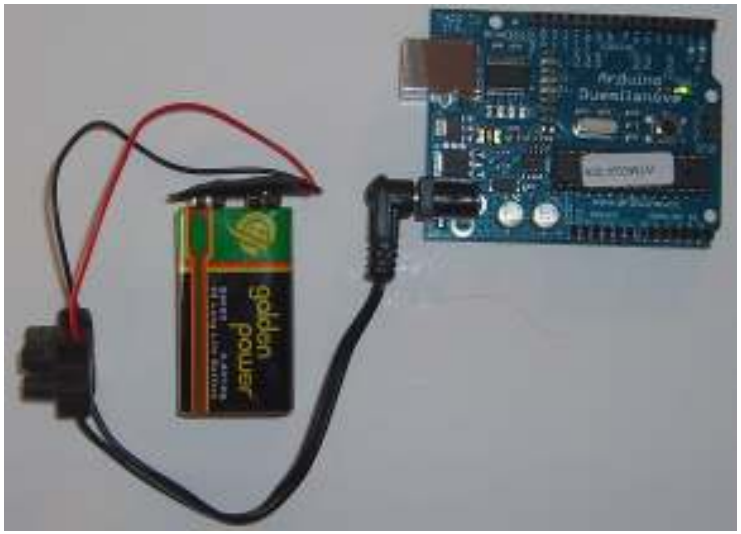




La diode électroluminescente (LED) émet de la lumière. Elle est polarisée: la patte "+" est la plus longue, l'autre patte est la patte "-" .
Les broches numériques de l'Arduino, lorsqu'elles sont configurées en sorties et qu'elles sont à l'état 1 ou haut (HIGH) , fournissent une tension de 5 volts, supérieure à ce que peut accepter une LED courante, sauf certaines LEDs. Les LED doivent donc être couplées en série avec une résistance.

Rendre Arduino autonome

Lorsque la carte Arduino est connectée au port USB de l'ordinateur, celui lui fournie l'énergie électrique nécessaire à son fonctionnement.
Une fois le programme chargé, on peu débrancher le cordon USB et connecter la carte soit à une pile, soit à un transformateur.



Une pile 9 volst et un connecteur de 2,1 mm
Avec le « + » au centre.

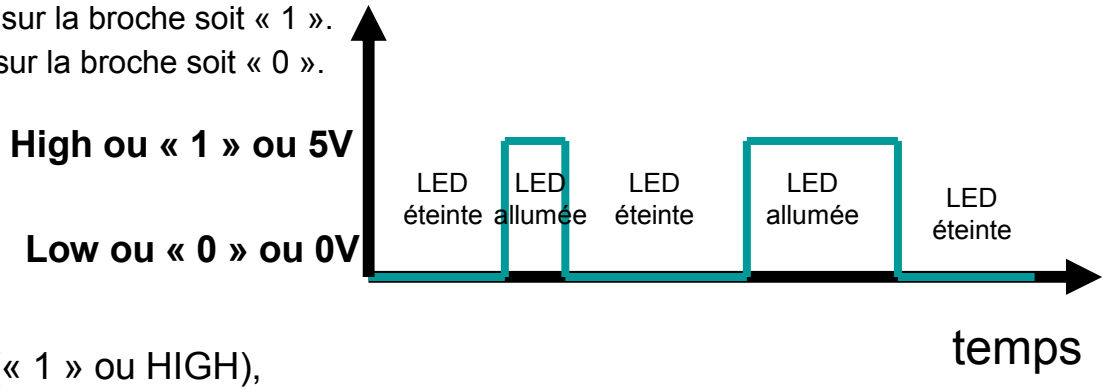
Un transformateur qui convertie la tension du secteur en une tension continue(DC) 9 volts et un connecteur de 2,1 mm
avec le « + » au centre.

Entrée/Sortie numérique

La carte Arduino possède **14 entrées / sorties numériques** (digital en anglais) D0 à D13.
Dans « void setup », il faut déclarer une broche comme une entrée ou comme une sortie par une des deux instructions suivantes :
pinMode (nom_de_broche, INPUT) ; // broche en entrée
pinMode (nom_de_broche, OUTPUT) ; // broche en sortie

En sortie, on envoie soit 5V sur la broche, soit 0V. Cela correspond à un « 1 » ou à un « 0 »,
à un niveau « haut » ou à un niveau « bas ».
Dans le programme cela correspond aux instructions suivantes :

digitalWrite(nom_de_broche, HIGH) ; // envoie 5V sur la broche soit « 1 ».
digitalWrite(nom_de_broche, LOW) ; // envoie 0V sur la broche soit « 0 ».



En entrée, la carte peut lire soit un niveau haut (« 1 » ou HIGH),
soit un niveau bas (« 0 » ou LOW).
Dans le programme cela correspond aux instructions suivantes :

digitalWrite(nom_de_broche, HIGH) ; // lit 5V sur la broche soit « 1 ».
digitalWrite(nom_de_broche, LOW) ; // lit 0V sur la broche soit « 0 ».

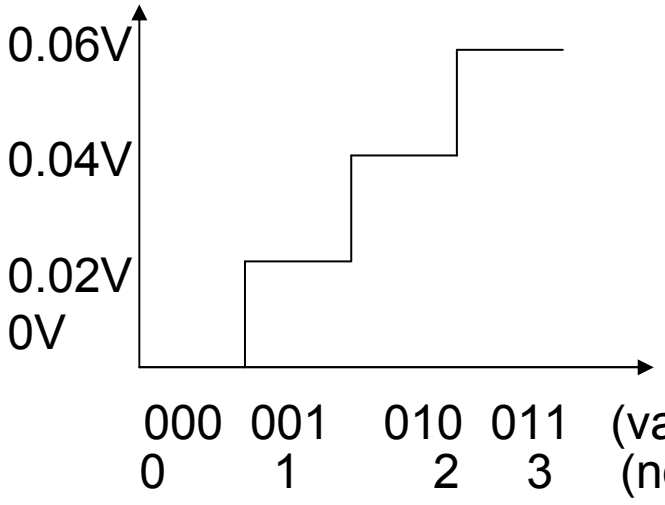
Signal numérique : signal qui ne prend que deux états distinct comme 0V et 5V soit « 0 » et « 1 ».

Les entrées analogiques

Contrairement au signal numérique qui ne peut prendre que deux états différents, Un signal analogique peut prendre une infinité de valeurs. Comme une tension que l'on fait varier progressivement de 0V à 5V.

La carte Arduino fonctionne en numérique, le microcontrôleur ne comprend que les « 0 » et les « 1 ». Les entrées de A0 à A5 sont dotées de convertisseurs analogique/numérique qui convertit une tension en une suite de « 0 » et de «1 » que la carte fait correspondre à un nombre variant de 0 à 1023.

On peut ainsi récupérer les informations d'un capteur.



Tension d'entrée :

0V ⟶ 2.5V ⟶ 5V

Nombre lu à utiliser dans le programme :

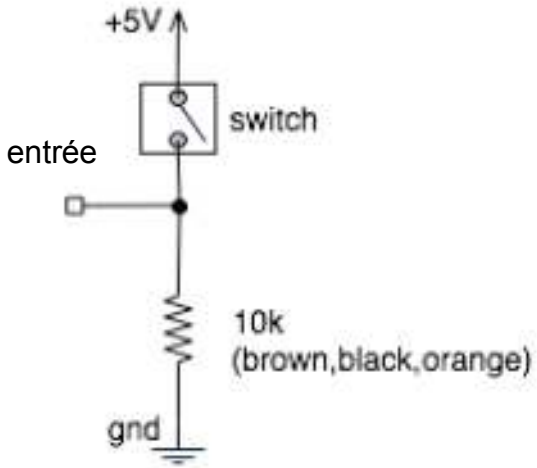
0 ⟶ 512 ⟶ 1023

Câbler un interrupteur

Lorsque l'interrupteur est **ouvert**, l'entrée de la carte lit un niveau bas ou **LOW**.

Lorsque l'interrupteur est **fermé**, l'entrée lit un niveau haut ou **HIGH**.

Donc, si on appuie sur un bouton poussoir ainsi câblé, la carte lira un « 1 » soit HIGH. Si on relâche le bouton poussoir, la carte lira un « 0 » soit LOW.



Exemple de programmation

```
void set up()
{
pinMode(inter, INPUT);
}
loop ()
{
int valinter = 0;    // on crée une variable valinter pour lire l'état de l'interrupteur
valinter = digitalRead(inter); // on lit la valeur de l'interrupteur (LOW ou HIGH)
if (valinter == LOW)  // Si valinter égale LOW, faire….
{instructions}
else   // sinon, faire….
{instructions}
}
```

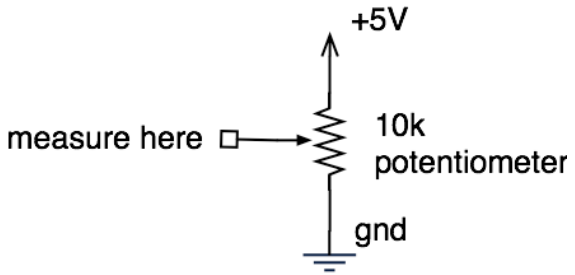
Câbler un potentiomètre

Lorsque l'on tourne le potentiomètre, on fait varier
la tension mesurée entre 0V et 5V.

Si on connecte le point de mesure sur un entrée analogique,
la valeur enregistrée variera entre 0 et 1023.

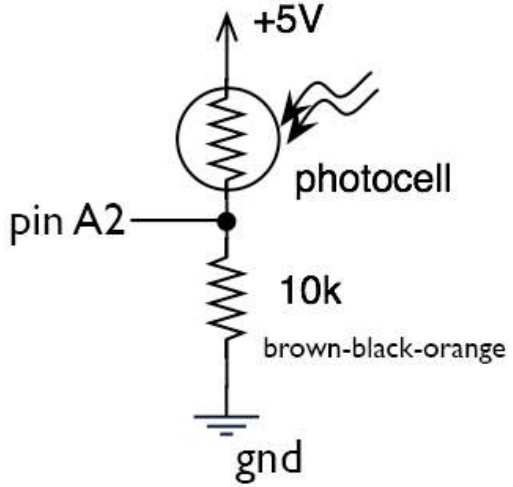
On peut enregistrer cette valeur et l'utiliser pour piloter une led ou un moteur.

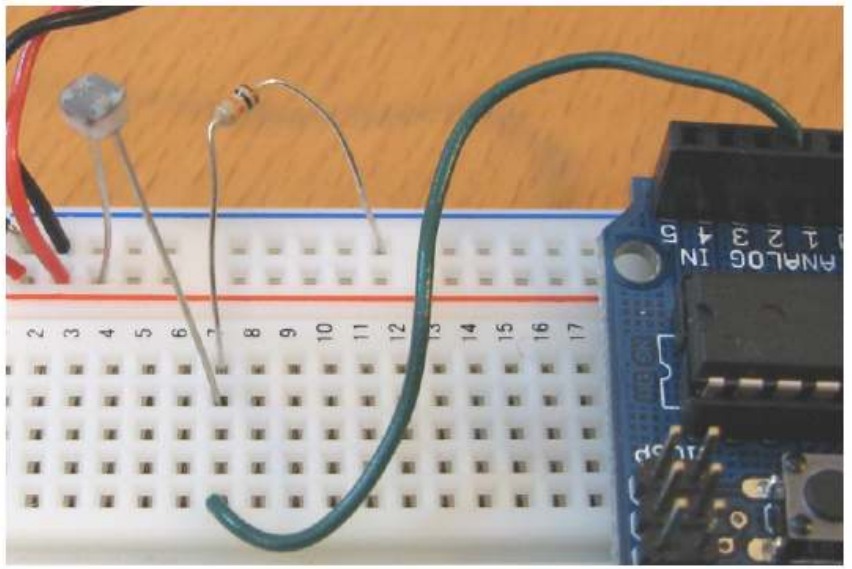
Il faut connecter la patte du milieu à une entrée analogique (A0 à A5)



Câbler une photorésistance

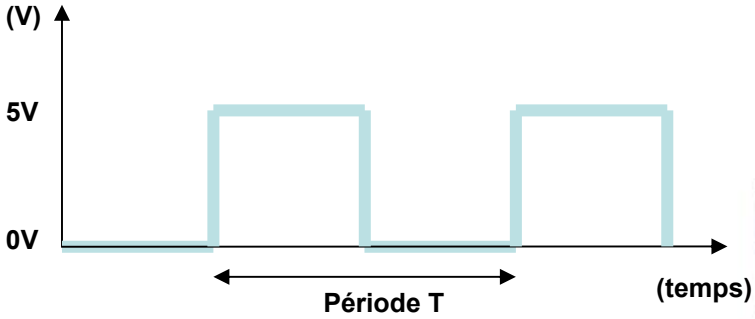
La mesure se fait sur entrée analogique. La résistance de la photorésistance diminue lorsque la lumière augmente.
On peut utiliser le même programme que celui du potentiomètre.



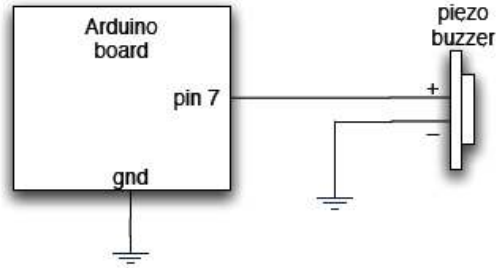


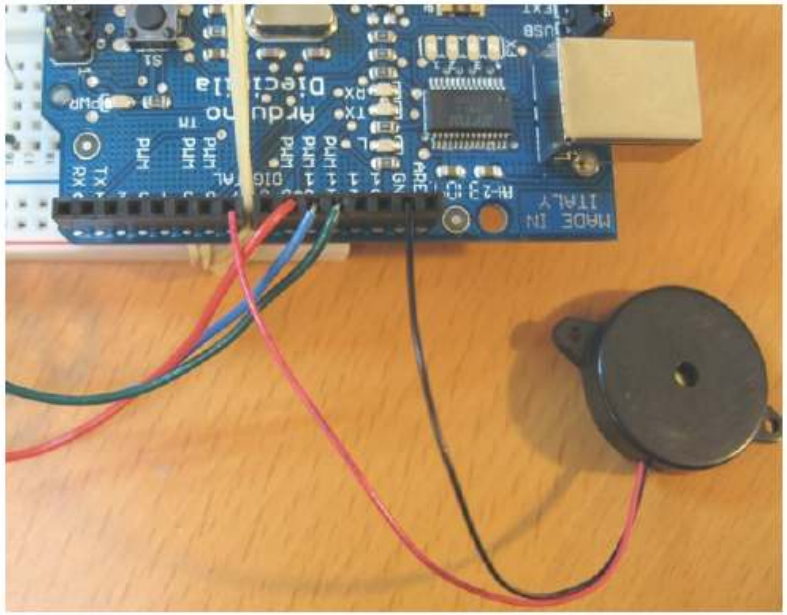
Câbler un buzzer

Le buzzer se câble sur une sortie numérique. On lui envoie alors un signal périodique dont on fait varier la fréquence en fonction de la note que l'on désire jouer. Exemple : le LA est un signal d'une fréquence f de 440 Hertz soit un signal qui varie 440 fois par seconde.



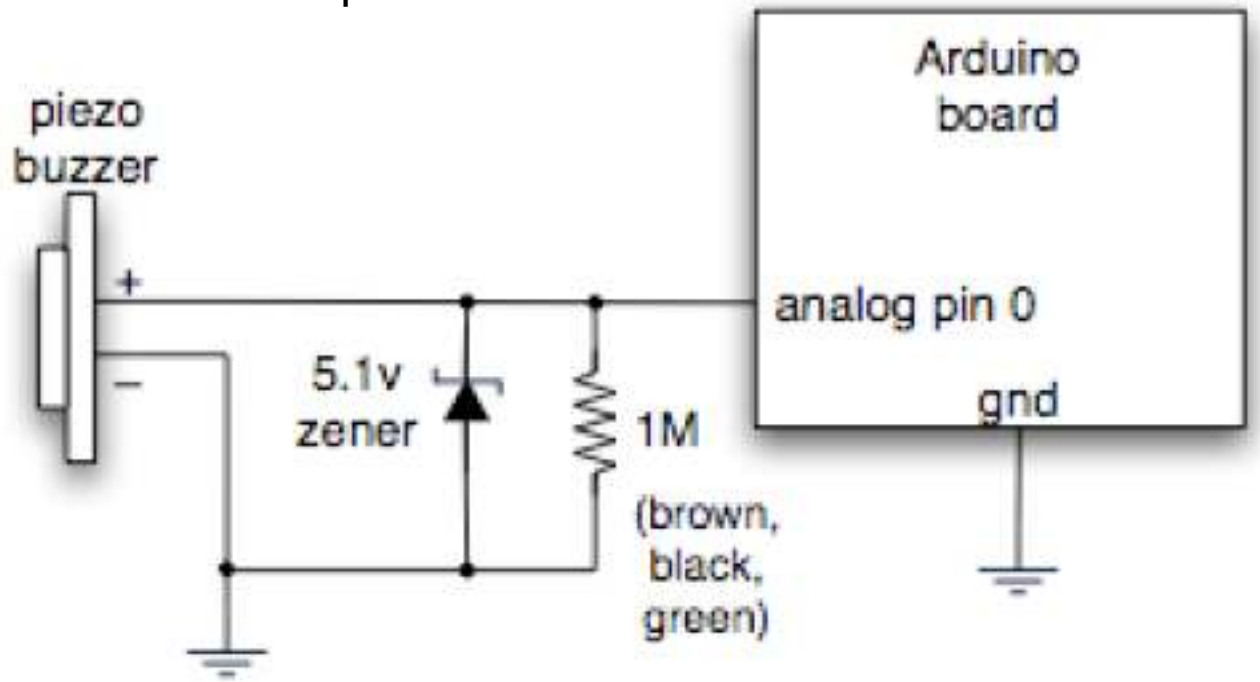
f = 1/T ; T=1/f ; f : fréquence; T : période.





Câbler un buzzer piezzo en capteur de frappe

L'entrée analogique 0 mesure la tension produite par le buzzer lorsqu'il vibre. Le buzzer fonctionne alors comme un microphone. Plus il vibre, plus la tension mesurée est grande.
A utiliser avec le programme « piezzo_capteur_de_frappe ».
La diode zener est optionnelle.



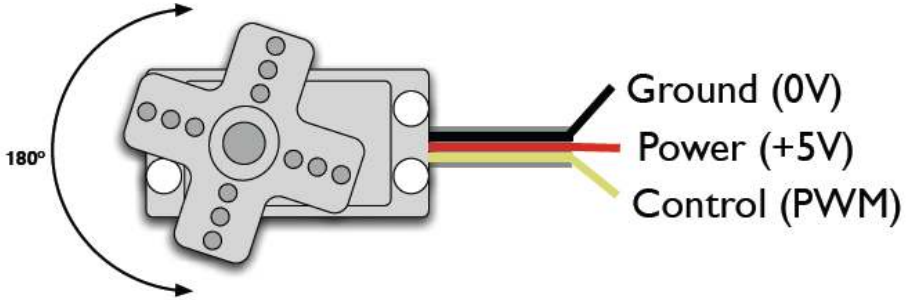
Câbler un servomoteur

On câble le servomoteur sur une des sortie numérique PWM.
Les sorties numériques PWM sont : D11, D10, D9, D6, D5.

On envoie un signal que l'on fait varier en fonction du sens et de la position désirée.

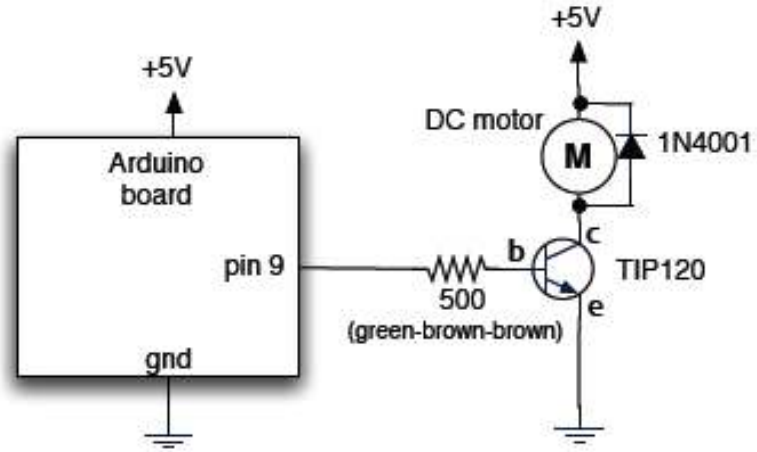
Un sous programme appelé servo.h doit être inclus dans le programme. Il est alors facile de le commander.

Le fil noir est connecté au 0V ou Gnd.
Le fil rouge est connecté au 5V.
Le fil jaune ou blanc est connecté à une sortie numérique PWM.



Câbler un moteur à courant continu

On utilise un transistor pour piloter le moteur.
Comme pour le buzzer, on envoie un signal dont la fréquence varie en fonction de la vitesse désirée.
La diode dite « de roue libre » permet d'évacuer le courant créer lorsque le moteur ralentie alors qu'il
n'est plus alimenter.
La tension d'alimentation du moteur peut être différente, 9V par exemple.



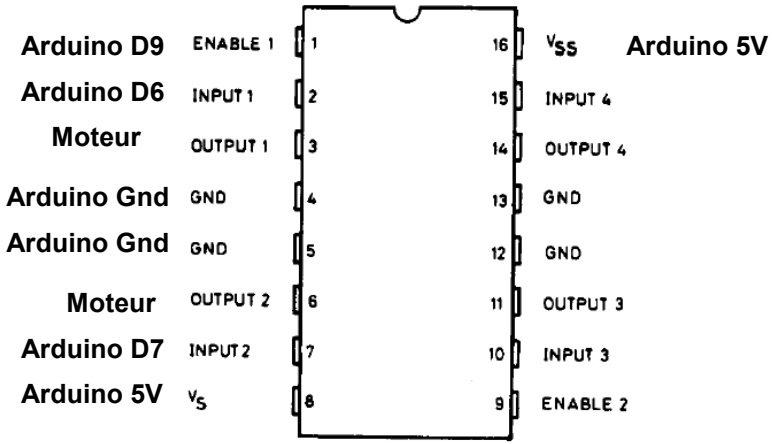
Câbler un moteur à courant continu avec le circuit intégré L293D

Le circuit intégré L293D nous permet de piloter 2 moteurs à courant continu.

La broche 1 à l'état haut permet de démarrer le moteur, à l'état bas l'éteint.

Les entrées input1 et input 2 permettent de choisir le sens de rotation du moteur selon les niveaux logique envoyés :

L293 Input1	**L293 input2**	**Sens du moteur**
« 1 »	« 0 »	horaire
« 0 »	« 1 »	antihoraire

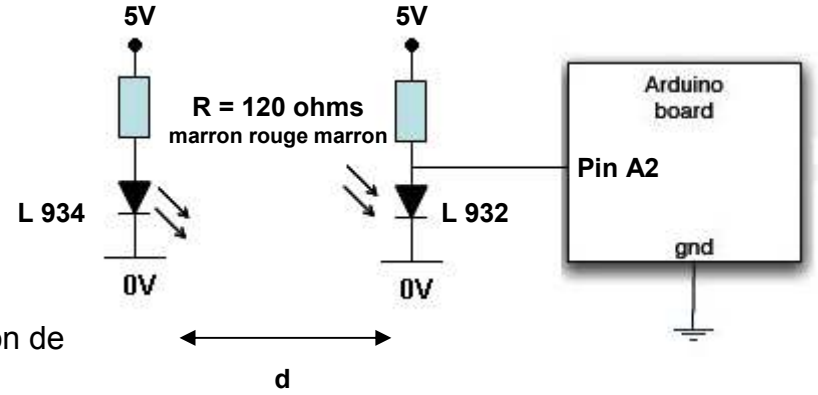


circuit intégré (CI), aussi appelé puce électronique, est un composant électronique reproduisant une ou plusieurs fonctions électroniques plus ou moins complexes, intégrant souvent plusieurs types de composants électroniques de base dans un volume réduit, rendant le circuit facile à mettre en œuvre. Il existe une très grande variété de ces composants divisés en deux grandes catégories : analogique et numérique."

Câbler une barrière infrarouge

La L.E.D. infrarouge émettrice (L934) émet une lumière visible au travers d'un appareil photo numérique.
La tension aux bornes de la L.E.D. réceptrice varie en fonction de la distance par rapport à la L.E.D. émettrice. Cette tension est maximale lorsque un objet coupe le faisceau lumineux.

Lorsque le faisceau lumineux est coupé, la broche analogique A2 enregistre le chiffre 1023, sinon elle enregistre un chiffre en fonction de la distance qui sépare les L.E.D..



d (cm)	0	1	2	3	4	5	6
Valeur en A2	319	450	727	885	950	980	990

Exemple de programme associé au montage

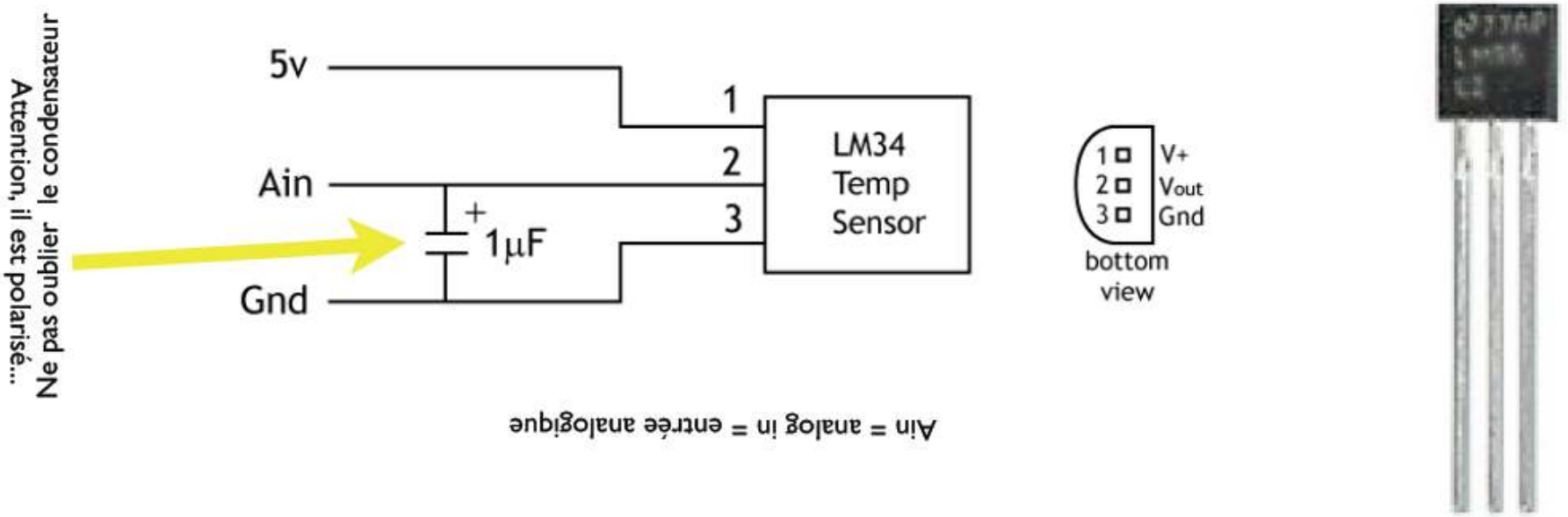
```
int led = 2;                    //la led réceptrice est sur la broche analogique 2
Int lecture = 0;                // variable qui stocke la valeur lue sur A2

void setup()
{}
void loop ()                    // se répète en boucle
{
 lecture=analogRead(led);       //lecture de la valeur lue
 If (lecture>=1000)             // comparaison de la valeur lue au seuil de détection
   {
    ACTION                      // si > ou = faire
   }
else                            // sinon faire
   { ACTION }
}
```

Attention : les deux L.E.D. sont identiques en apparence.

Câbler un capteur de température

La tension mesurée sur l'entrée analogique varie en fonction de la température.
Le programme « temperature_led » permet de l'utiliser.



Sources et ressources

http://todbot.com/blog/bionicarduino/ Tod E. Kurt

Centre de Ressources Art Sensitif
http://www.craslab.org
http://www.artsens.org

Livret Arduino en français par Jean-Noël Montagné, Centre de Ressources Art Sensitif,

http://www.pobot.org/

http://www.interface-z.com/

http://www.ladyada.net/learn/arduino/

http://www.arduino.cc/ Le site officiel