



Composants électronique avec le Diduino

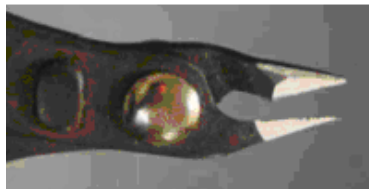
Comment interfacer, comment programmer

Les documents Diduino faciles à cliquer sont listés sous www.didel.com/Diduino.html

C1 Outils

Une pince coupante légère avec biseau plat est un complément utile pour préparer les fils de câblage.

Un fer à souder, pointe fine et réglable n'est pas indispensable, mais utile pour relier des composants spéciaux, allonger les fils, etc.



C2 Alimentation

Beaucoup trop à dire! USB nous facilite la vie avec une tension de 5V bien régulée et un courant max de 0.5 A. En principe protégé contre les court-circuits, mais il vaut mieux ne pas essayer. Les cartes Arduino ont un connecteur pour une alimentation 6-12V et on voit souvent une pile 9V anti-écologique alimentant ces cartes. Les cartes Diduino n'ont pas de connecteur pour alimentation 8-12V extérieur. Un connecteur Gabriel est prévu pour une alimentation 3.7 à 5V, mais il ne faut pas alimenter par cette prise quand USB est connecté.

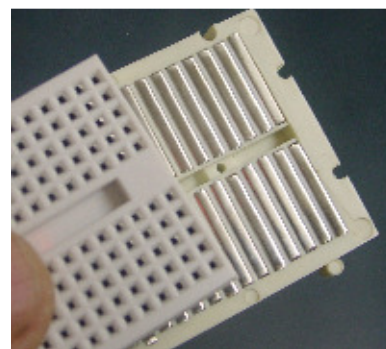
C3 Bloc d'expérimentation

Le bloc d'expérimentation (breadboard) permet d'insérer des composants et fils qui ont entre 0.5 et 0.7mm de diamètre. Plus petit, le contact n'est pas garanti. Plus gros, il peut être déformé et n'acceptera plus des fils de 0.5, 0.6, 0.7. Les composants du kit Diduino sont compatibles.

Les fils de 4 couleurs du kit ont un diamètre de 0.5mm (isolant 1.1mm) ne sont pas en cuivre, mais dans un alliage métallique qui donne plus de rigidité. Le dénudage se fait en pinçant juste assez avec la pince et en tirant pour dégager 5 mm d'isolant.

On fera des fils le plus court possible, en particulier les fils noir et rouge d'alimentation, pour faciliter la lisibilité du montage.

Attention, si vous collez la plaque en enlevant la feuille protectrice, vous aurez de la peine à récupérer la plaque pour la coller ailleurs, cela colle mieux côté ressorts!

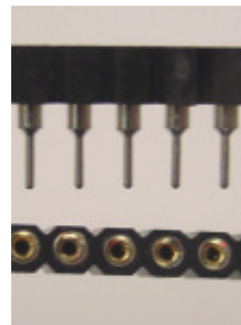
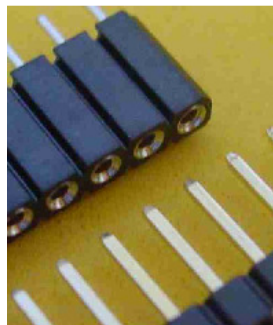


C4 Barrettes, connecteurs et fils

Les barrettes de connecteur au pas de 2.54mm existent dans 2 dimensions. N'utiliser que le 0.7mm comme connecteur. Les pins en 0.5mm sont trop fragile, mais les contacts femelle sont nécessaires pour les résistances et fils de câblage en 0.5mm.

Pour garantir des bons contacts de 0.6mm à 0.8mm, il faut exiger des contacts tulipe malgré leur prix.

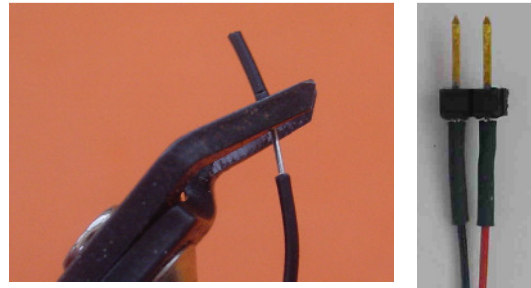
www.didel.com/kits/SavoirFaire.pdf



Les fils du kit, ne sont pas en cuivre, qui manquerait de rigidité, mais en acier doux étamé.

Ils sont facile à dénuder avec une pince bien coupante qui serre le fil de la bonne profondeur. Un coupe-angle rend le même service.

Les long fils doivent se faire avec des fils très souples, correctement protégés vers la soudure par de la gaine thermo-rétractable ou de la colle chaude (cf.SavoirFaire)



On peut acheter pour moins de 10 CHF un assortiment de 50 fils chinois avec des pins de 0.5mm. Les fils tiennent mal (0.55mm serait idéal) et sont trop longs.

Pour un montage clair, il faut absolument câbler les alimentations et connexions proches avec des fils rigides.



C5 Electronique - notions supposées connues

Nous aurons peu l'occasion de calculer, mais il faut connaître les unités suivantes et le loi d'Ohm. Courant I exprimé en milliAmpères (mA). Tension V exprimé en Volts (V). Résistance R en Ohm (Ω) et kiloOhm ($k\Omega$ ou simplement k). Loi d'Ohm: $R = U / I$.

C6 Résistances

Les valeurs des résistance sont codées par des anneaux de couleur. La progression est logarithmique: 1, 2.2, 4.7, 10 Des valeurs intermédiaires existent, mais pour nos applications on peut se limiter à quelques valeurs:

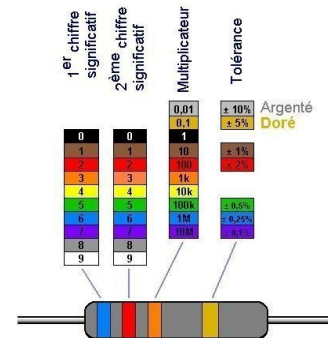
220 Ohm quand un veut "beaucoup" de courant (10-20 mA) (buzzer)

1 kOhm pour un courant de 3 à 5 mA (Leds très lumineuses)

4k7 pour un courant faible inférieur à 1mA (Leds)

22k si le courant demandé est négligeable, il faut simplement imposer une tension.

On comprendra mieux dans les explications qui suivent.



C7 Condensateurs

Autour d'un microcontrôleur, les condensateurs sont utilisés comme filtre pour écrêter les impulsions de courant de quelques nanosecondes lors des commutations de signaux internes. L'utilisation analogique d'un condensateur pour créer des retards n'a pas de sens quand on a un processeur pour gérer les délais.

C8 Leds

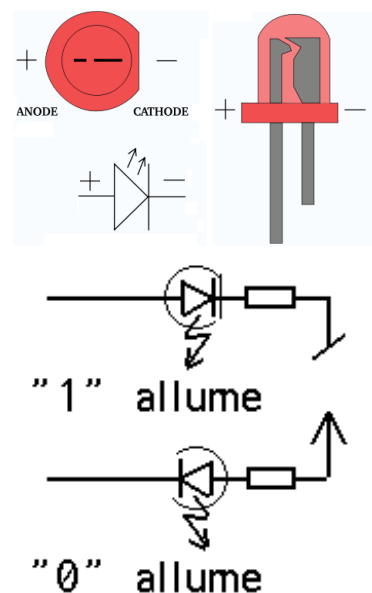
Les Leds (Light emitting diodes) ou Del (diodes électroluminescentes) sont des diodes, donc le courant ne passe que dans un sens, de l'anode (+, patte longue) vers la cathode (-, patte courte). Dans ce sens, la résistance est très faible et il faut limiter le courant avec une résistance en série qui fixe l'intensité lumineuse. Naturellement, la résistance peut être devant ou derrière la Led.

Pour allumer une Led, le processeur peut soit envoyer du courant en imposant un 1 sur sa sortie (retour du courant par la masse), soit pomper le courant envoyé depuis l'alimentation à travers la LED.

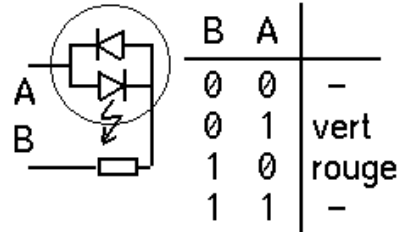
Les ingénieurs électroniciens préfèrent allumer avec un "zéro", car les transistors N qui tirent le courant vers la masse sont meilleurs que les transistors P qui envoient du courant depuis l'alimentation.

La tension minimale qui allume une Led est de 1.2 (Led rouge) à 3V (Led blanche). L'efficacité lumineuse aussi est très différente: la résistance en série est de 470 à 10 kOhm.

Si on a suffisamment de tension, on peut mettre deux ou plusieurs Leds en série. Par exemple avec 12V on peut mettre 3 diodes blanches en série et avoir 3V de chute de tension dans la résistance pour limiter le courant (calcul précis selon les données du fabricant).



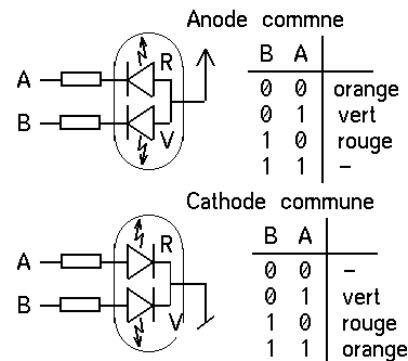
Une Led bicolor peut se faire avec deux diodes tête-bêche, ou s'acheter dans un boîtier de diode. Une résistance suffit. Très pratique pour tester une sortie d'ampli moteur bidirectionnel et faire la mise au point de la commande moteurs sans s'encombrer.



On peut aussi acheter des Leds bicolors à 3 fils, cathode commune ou anode commune (préférée).

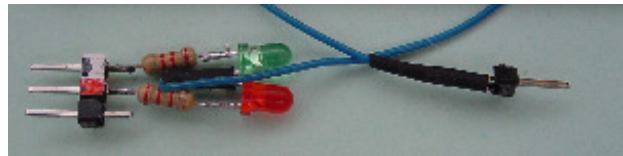
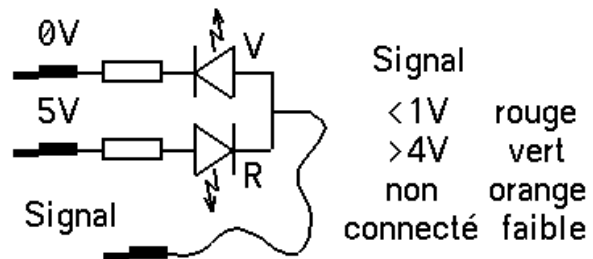
L'intérêt est que si l'on active les deux Leds, on a la couleur mélangée (orange). On voit des schémas avec une seule résistance sur l'anode commune pour une Led à 3 fils. Cela exclut le mélange des couleurs, puisque la Led qui a la tension de seuil la plus faible prendra tout le courant.

Les Leds tricolores sont similaires avec une diode de plus dans le boîtier. Le fil commun est plus long.



Avec deux Leds on fait un testeur très commode pour vérifier l'état d'une pin. La Led rouge est liée au +. Si le point commun est au - elle s'allume. La Led verte est au - Une tension supérieurs à 3-4V l'allume. En l'air, ou sur une entrée, les deux diodes sont allumées faiblement.

Il ne faut pas utiliser des diodes à haut rendement, donc haute tension de seuil si on veut avoir le mode couleurs mélangées pour une entrée flottante.



La programmation des Leds est abondamment documentée depuis le programme blink du début jusqu'aux cubes pleins de diodes.

Pour avoir sur une sortie une intensité variable, Arduino a la fonction `analogWrite()` ; sur quelques sorties, avec comme paramètres une valeur de 0 à 255. Le programme ci-contre, vu dans le Cours02 (2.9) allume et éteint la LED progressivement. Si on secoue la carte très rapidement, on voit des traits lumineux, puisque la proportionnalité est faite par PWM (voir sous moteurs) avec une fréquence de ~500 Hz.

```
//LedVarie.ino
const int Led3=3;
void setup()
{
  pinMode (Led3, OUTPUT);
}
void loop ()
{
  for (int i=0 ; i<256 ; i++ )
  {
    analogWrite(Led3,i);
    delay (20) ;
  }
  for (int i=255 ; i>0 ; i-- )
  {
    analogWrite(Led3,i);
    delay (20) ;
  }
}
```

Utiliser une sortie de microcontrôleur par Led est limitatif. On câble les Leds en tableau avec sélection des lignes et des colonnes, et en utilisant des circuits série prévu pour cette application. Voir www.bricobot.ch/docs/Leds.pdf ou www.didel.com/kits/Leds.pdf .

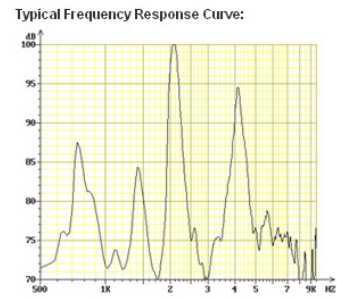
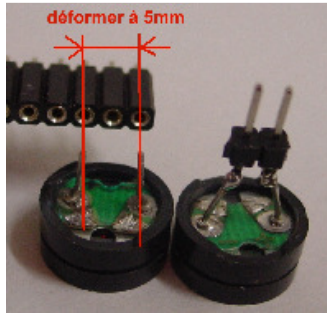
C9 Buzzer

Le buzzer (transducteur électroacoustique) est une membrane attirée par une bobine dont il est important de connaître la résistance. Il peut être piezo-électrique, donc c'est un condensateur. Dans les deux cas, c'est un haut-parleur de mauvaise qualité.

A ne pas confondre avec un buzzer actif, que l'on doit appeler vibreur, qui contient un oscillateur qui fait vibrer la membrane quand on applique la tension.

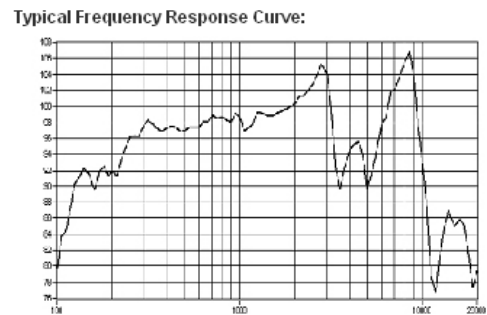
La courbe de réponse est lamentable; si vous jouez la gamme, les notes sont difficilement reconnaissables à cause de leur différence d'intensité!

Programmer un buzzer est comme clignoter une Led (Cours 1.12). Comme avec la fonction `delay()`; le processeur est complètement accaparé par un programme jouant des sons avec des `delayMicroseconds()`; Faire une tâche en parallèle perturbe le son, qui n'est déjà pas fameux. Programmer des sons et notes de musiques est toutefois un bon exercice pour comprendre les fonctions et tableaux (Cours x.x). Voir aussi www.didel.com/diduino/Musique.pdf (en préparation)



Exercice: diminuer lentement la période pour suivre la variation d'intensité. Trois poussoirs permettent d'arrêter en affichant la valeur sur le terminal, repartir en augmentant ou diminuant la période.

Un haut-parleur de 40mm a une meilleure courbe de réponse, mais sa résistance de 3.6 Ohm ne permet pas de le brancher sur le processeur sans passer par un ampli. Un transistor est la solution simple; un pont est plus efficace (section 15).



Arduino a 3 ordres pour générer des sons, selon 2 approches différentes. Dans un cas on lance un son et on l'arrête par une autre instruction, après un délai.

`tone (pin, frequencyHz) ;`

`noTone(pin) ;`

Dans l'autre cas, on donne la durée du son dans un 3e paramètre

`tone (pin, frequencyHz, duration) ;`

Plus de détails sous www.didel.com/diduino/Cours01.pdf

[Suite www.didel.com/diduino/Composants02.pdf](http://www.didel.com/diduino/Composants02.pdf)