Beaucoup de documents de présentation des cartes Arduino sont disponibles sur le web.

Celui-ci se limite au strict-minimum à connaître pour commencer à utiliser une carte Arduino.

Il s'adresse plutôt à des élèves de 3°, quelques connaissances en électricité sont souhaitables, ainsi que la pratique d'un logiciel de programmation par blocs, du type Scratch.

Les notions de capteurs et actionneurs doivent être acquises et quelques rappels préalables sur les notions de signal analogique (ou logique) peuvent être utiles.

Table des matières

1. Définition : Qu'est-ce que c'est ?

Une carte Arduino est une petite (5,33 x 6,85 cm) carte électronique équipée d'un **micro-contrôleur**. Le **micro-contrôleur** permet, à partir d'événements détectés par des **capteurs**, de programmer et commander des **actionneurs** ; la carte Arduino est donc une **interface programmable**.

Carte Arduino UNO

La carte Arduino la plus utilisée est la carte **Arduino Uno**.

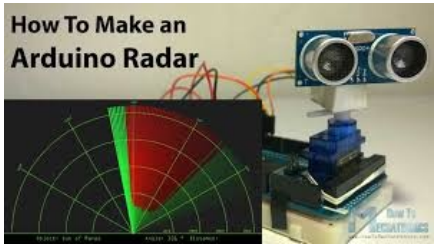
La conception matérielle (**schémas électroniques** et **typons**) est distribuée sous licence Creative Commons Attribution Share-Alike 2.5

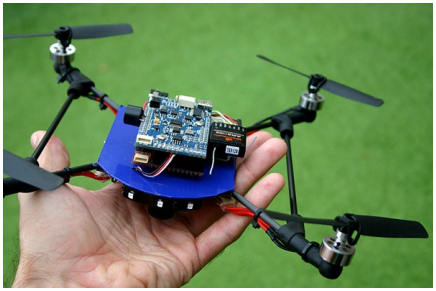
Le **code source** de l'environnement de programmation et les bibliothèques embarquées sont disponibles sous licence LGPL.

Une grande communauté d'amateurs et de passionnés contribuent à développer des applications et à les partager.

2. L'usage : Mais à quoi ça peut donc bien servir ?

Les possibilités d'utilisation sont infinies : SI on peut ...





détecter un ou plusieurs événements (variation de température, mouvement, présence, distance …)

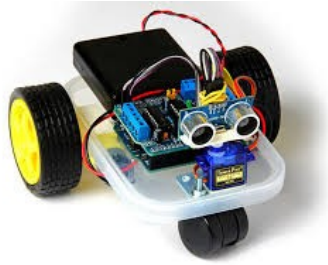
ET

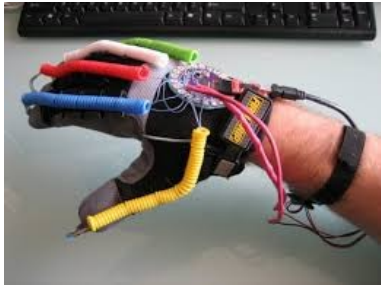
en fonction de ces événements,

agir sur le monde réel à l'aide d'actionneurs (résistances chauffantes, moteurs ...),

ALORS

on peut tout faire : bras de robot, régulation de température, effets lumineux, instruments de musique, systèmes d'alarmes, ...

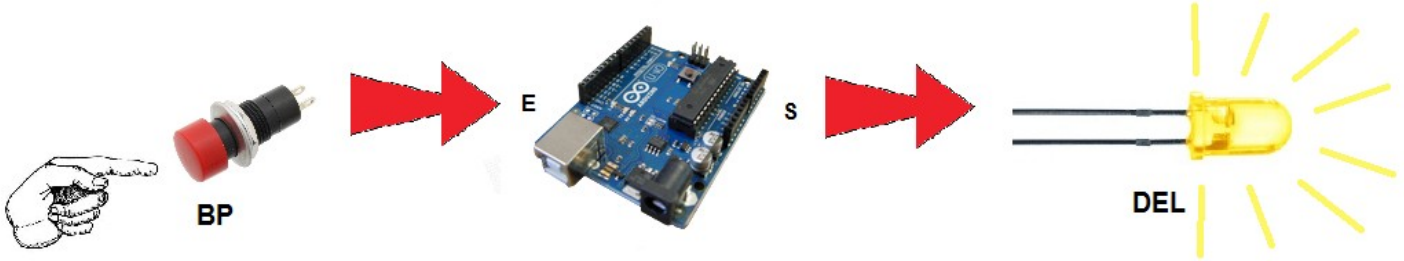




De nombreuses cartes additionnelles (« shields ») ont été créées afin d'enrichir les applications potentielles, les seules limites sont notre imagination, le temps, et le budget (quelques dizaines d'euros suffisent au départ pour s'équiper).

3. Comment on s'en sert ?

Supposons que l'on veuille faire clignoter une lampe (DEL) à l'aide d'un bouton poussoir (BP).



Nous connectons le bouton poussoir (BP) à une entrée (E) de la carte Arduino et notre lampe (diode lumineuse DEL) à une sortie (S) de la carte.

Mais nous devons également programmer la carte, de manière à ce que la diode lumineuse clignote.

Nous rédigeons le programme à l'aide d'un logiciel, installé sur un ordinateur.

Le programme doit sans cesse surveiller l'entrée connectée au bouton poussoir :

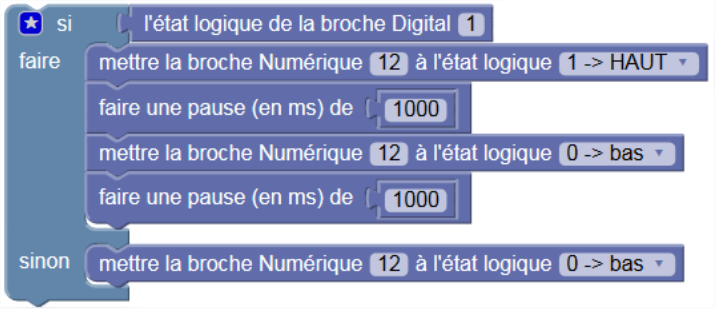
- si le bouton poussoir est appuyé il doit : allumer la diode, attendre 1 seconde, éteindre la diode, attendre 1 seconde, puis recommencer (allumer la diode, attendre 1 seconde …).
- si le bouton poussoir n'est pas (ou plus) appuyé, il doit éteindre la diode.

Il est possible de programmer le comportement de la carte Arduino de deux manières différentes :

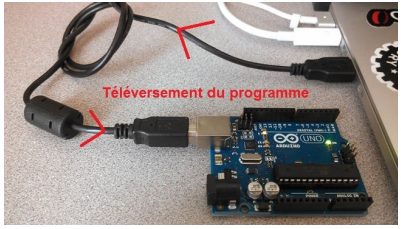
Programmation en langage C

```
void setup()
{
  pinMode(1, INPUT);
  pinMode(12, OUTPUT);
}
void loop()
{
  if (digitalRead(1)) {
    digitalWrite(12, HIGH);
    delay(1000);
    digitalWrite(12, LOW);
    delay(1000);
  } else {
    digitalWrite(12, LOW);
  }
}
```

Programmation par blocs



Une fois le programme créé, nous le **téléversons**, (=transférons) à la carte Arduino à l'aide d'un câble USB, comme celui d'une imprimante.



Nous pouvons alors retirer le câble USB : notre carte Arduino est autonome, elle nécessite alors toutefois une alimentation électrique (pile 9V).

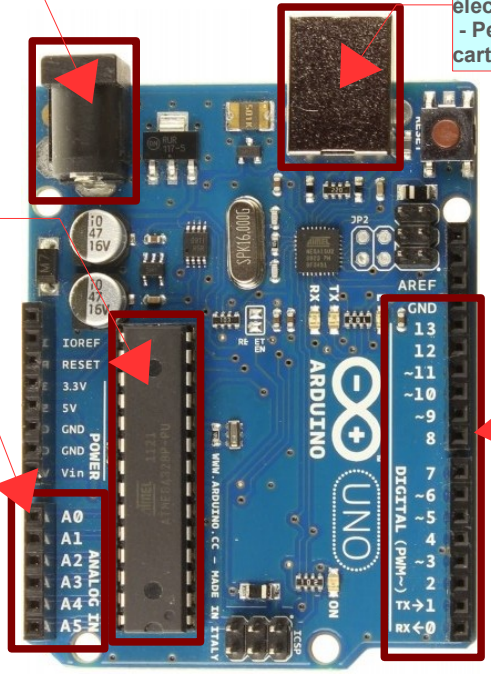
4. Zoom sur la carte

Prise jack :
- Permet de brancher une alimentation
(pile, batterie, adaptateur secteur, + au centre Vin 7 à 12 V).

Connexion USB (Universal Serial Bus):
- Permet d'alimenter la carte en énergie électrique (5V).
- Permet de téléverser le programme dans la carte.

Microcontrôleur :
- stocke le programme et l'exécute.

Entrées analogiques :
- Permet de brancher des capteurs et des détecteurs analogiques.



Entrées et sorties numériques (Digital) :
- Permet de brancher des actionneurs.
- Permet de brancher des détecteurs.

4.1. Les entrées/sorties numériques : de D0 à D13

Chacun des connecteurs D0 à D13 peut être **configuré par programmation en entrée ou en sortie**, nous pouvons donc avoir par exemple les connecteurs 2 et 3 configurés comme des entrées et les connecteurs 7, 8 et 9 configurés comme des sorties.

Il est par conséquent possible de connecter côte à côte des capteurs logiques (interrupteurs par exemple) aux connecteurs 2 et 3 et des actionneurs aux connecteurs 7, 8 et 9.

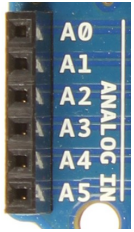
Les signaux véhiculés par ces connecteurs sont des signaux logiques, c'est-à-dire qu'ils ne peuvent prendre que deux états : HAUT (5 Volts) ou BAS (0 Volt), par rapport au connecteur de masse GND, qui lui est toujours, par définition, à 0 Volt.

Attention : les connecteurs ne peuvent pas fournir en sortie un courant supérieur à 40 mA, ce qui interdit de brancher directement un moteur sur une sortie logique.

Remarquez le signe **~ sur les connecteurs 3, 5, 6, 9 10 et 11**, nous verrons plus tard sa signification (**PWM**) et son importance.

Vocabulaire : on qualifie parfois ces entrées/sorties de numériques, de logiques ou de digitales, ces trois adjectifs sont ici considérés comme synonymes.

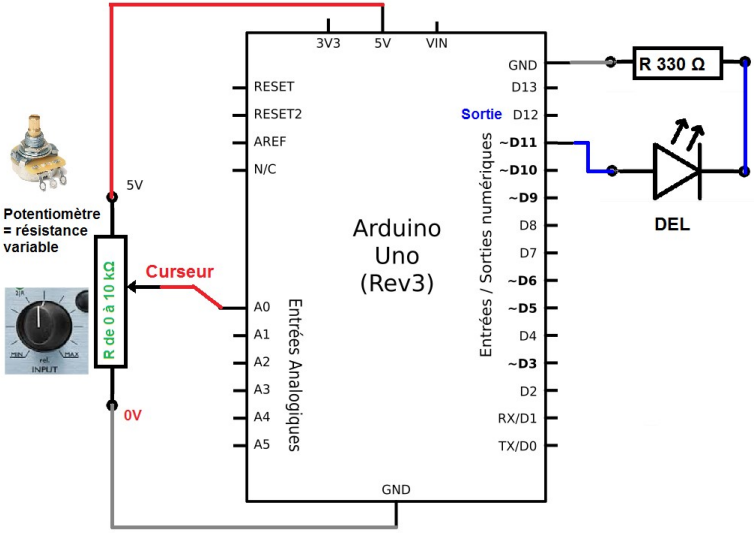
4.2. Les entrées analogiques A0 à A5



Contrairement aux entrées/sorties numériques qui ne peuvent prendre que deux états HAUT et BAS, ces six entrées peuvent admettre un millier de valeurs (1024 exactement) analogiques comprises entre 0 et 5 Volts.

Nous pourrons donc avoir des valeurs de tension précises à 5 mV près (≈ 5V/1024) .

Exemple d'utilisation d'une entrée analogique : faire varier la luminosité d'une DEL

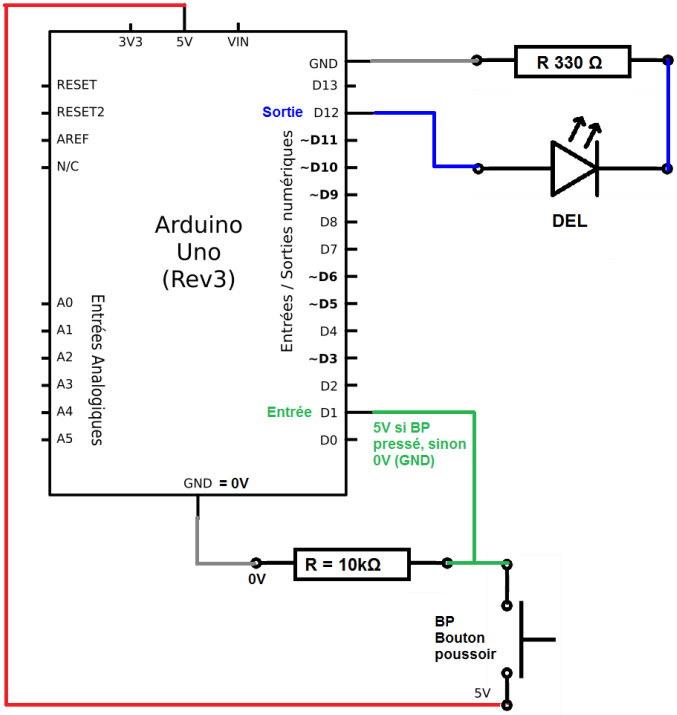


Si nous tournons le potentiomètre, le curseur se déplace d'un bout à l'autre de la résistance, et la tension de l'entrée A0 varie de 0 V à 5 V (0V en bas, 5V en haut sur le schéma).

Cette valeur analogique peut être ici utilisée par le programme pour faire varier la luminosité de la diode lumineuse.

On pourrait ainsi commander la vitesse de rotation d'un moteur ou la position d'un servo-moteur.

4.3. Mais revenons à notre clignotant (schéma structurel)



L' « entrée/sortie numérique » **D1** est utilisée en entrée (**INPUT**).

- Si le bouton poussoir est appuyé, l'entrée D1 est en contact avec le niveau 5V, c'est le niveau « Haut »

- Si le bouton poussoir n'est pas appuyé, l'entrée D1 n'est pas alimentée en tension, elle reste à 0V, c'est le niveau « Bas ».

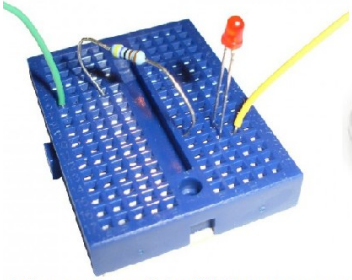
L' « entrée/sortie numérique » **D12** est utilisée en sortie (**OUTPUT**).

Pour allumer la diode (DEL), le programme alimente la sortie D12 en 5V, il la place en niveau « Haut ».

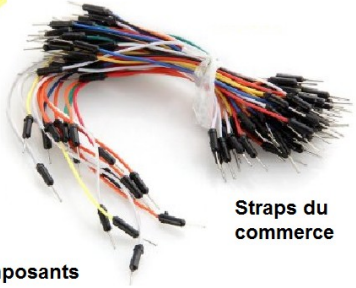
Notez la présence obligatoire de la résistance de 330 Ohms en série avec la Diode, elle limite le courant à une valeur acceptable (< 40 mA, voir page 7). En son absence, la carte Arduino ne résisterait pas.

5. Comment la connecter ? Straps et plaque d'essai

Une **plaque d'essai** permet de réaliser des montages électroniques sans soudure.
La plaque d'essai s'utilise avec des « **straps** » qui sont des fils de cuivre isolés, de longueur et couleur variables.
Plusieurs modèles existent, nous utiliserons des plaques d'essai comme celle représentée à droite.



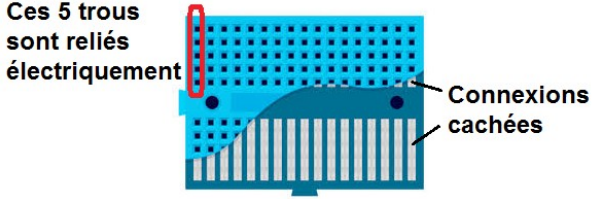
Plaque d'essai de 170 trous + composants et straps "faits maison"



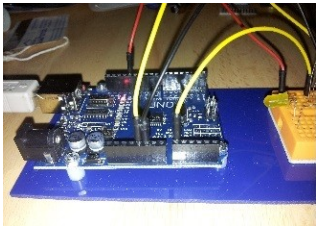
Straps du commerce

Mais comment le courant passe-t-il de la résistance à la diode lumineuse ?

La plaque d'essai comporte des **connexions cachées** comme sur le schéma ci-contre :
Chaque bande de cuivre met en contact 5 trous.
Les trous sont espacés exactement de 2,54 mm (un dixième de pouce).

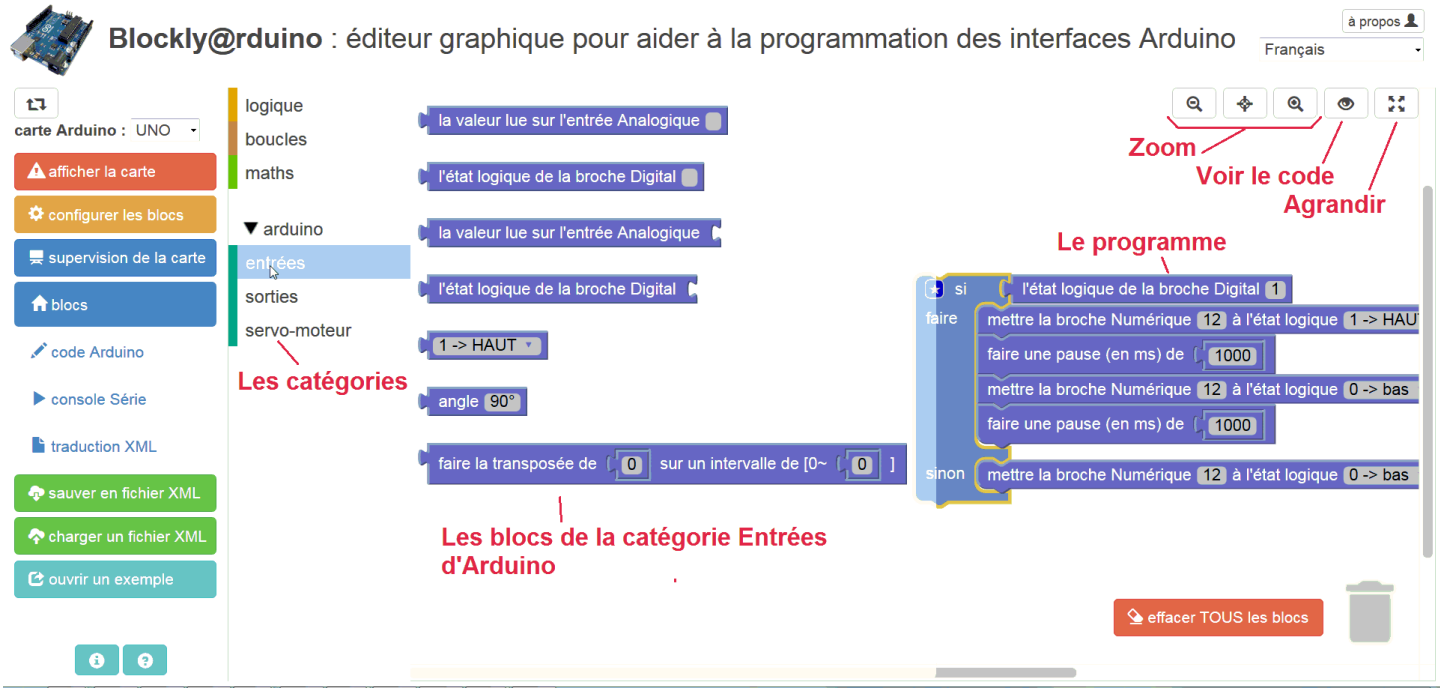


Ci-contre une carte arduino et une plaque d'essai reliées par des straps :



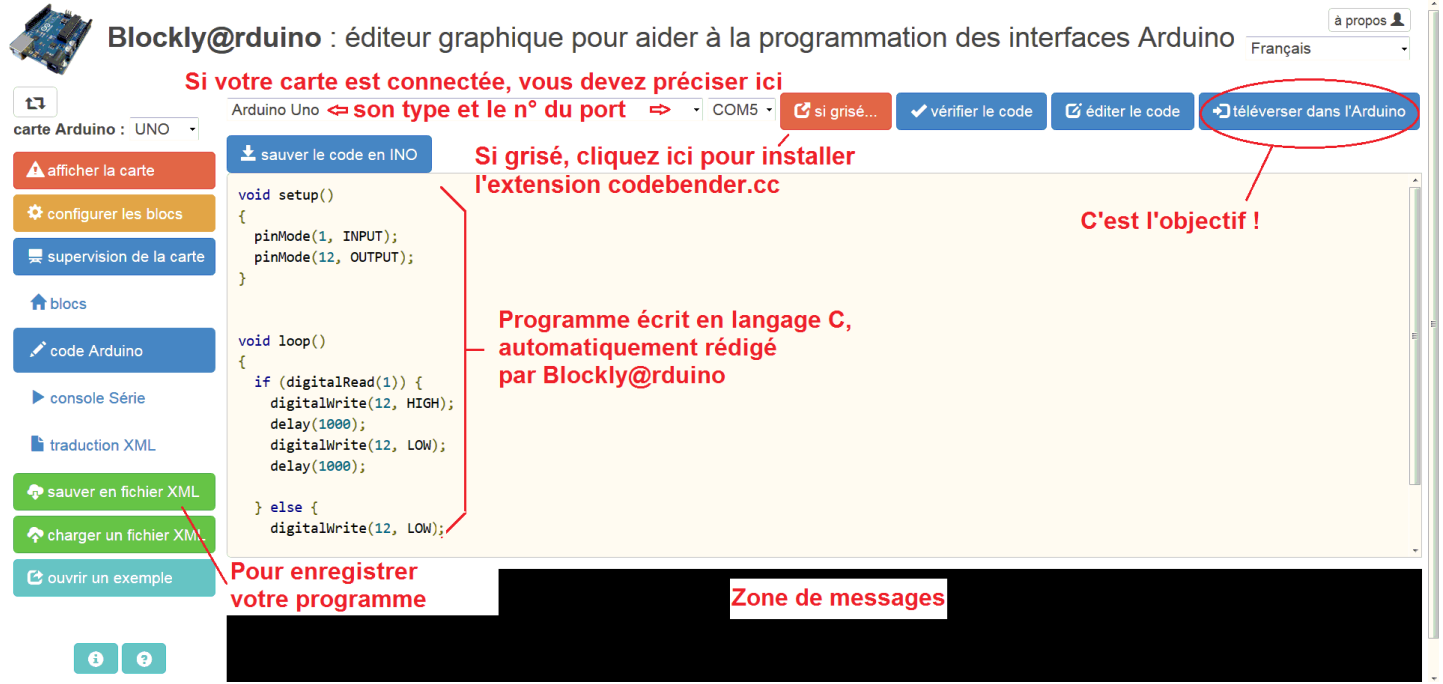
6. Comment la programmer ?

Nous utiliserons le logiciel en ligne Blockly@rduino, qui nécessite l'extension Codebender.cc de Firefox. Allez sur la page www.TechMania.fr et cliquez sur le logo :



Faites simplement glisser à droite de l'écran les blocs nécessaires à la réalisation du programme.

Fenêtre code Arduino



7. Un peu de méthode, ça ne peut pas faire de mal ...

Il suffit de se poser les bonnes questions dans le bon ordre et noter les réponses :

Le cahier des charges : Que doit faire mon application, quelles informations dois-je y entrer, quels ordres en sortie doit-elle produire ? Essayer de décrire par écrit ce que doit faire le système, et dans quelles conditions.

Les entrées : Quels sont mes capteurs ou détecteurs, sont-ils de type analogique ou numérique ?

- Quelles entrées ? A0 à A5 pour les capteurs analogiques, D0 à D13 pour les signaux numériques.

Les sorties : Quels sont les actionneurs dont j'ai besoin ?

- Quelles sorties ? De D0 à D13, en excluant celles déjà réservées aux entrées numériques.

Le matériel : De quels composants ai-je besoin ? Nous avons vu par exemple que les diodes lumineuses et les boutons-poussoir nécessitaient des résistances, mais différentes.

Le programme : Essayer de décrire sur le papier, sous forme de logigramme ou d'algorithme les grandes lignes du programme. Nommez les variables éventuelles (« Température » par exemple).

Passez sur Blockly@rduino et construisez votre programme, sans connecter la carte.

Après validation du programme, enregistrez-le.

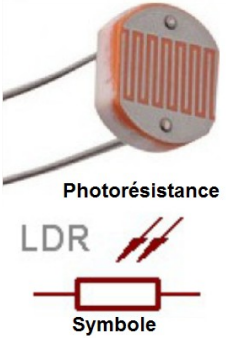
Les connexions : Insérez les composants sur la plaque d'essai, et reliez la plaque d'essai et la carte Arduino par des straps.

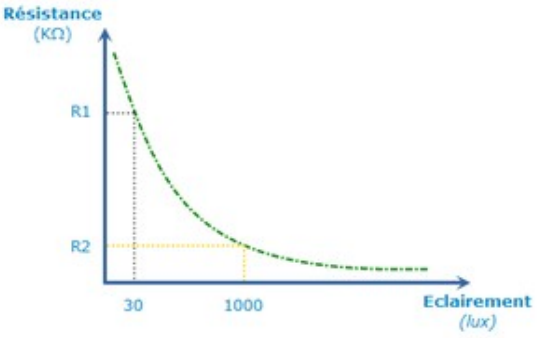
Après vérification du professeur :

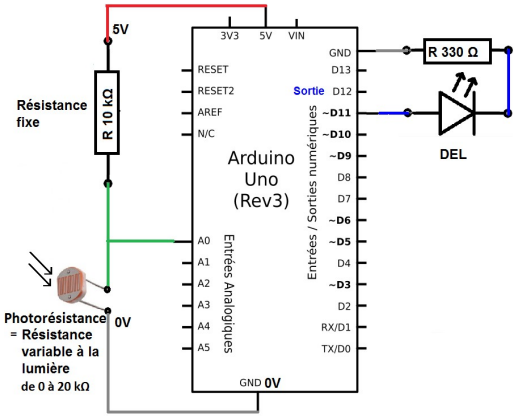
Connectez la carte Arduino à l'ordinateur, téléversez le programme et testez votre application.

8. Exemples d'applications simples

- Clignotant droit et gauche pour un véhicule : Faire clignoter une deuxième diode, en appuyant sur un deuxième bouton poussoir, les diodes doivent s'éteindre 0,5 secondes et rester allumées 1 s.
- Faire clignoter la deuxième diode à l'inverse de la première.
- Déclencher à l'aide du BP un signal lumineux de SOS · · · — — — · · · (points de 0,2 s, tirets de 1 s)
- Allumer une diode si la luminosité (l'éclairement) baisse. Cette application nécessite une photorésistance et une résistance de 10 kΩ).







La valeur de la photorésistance diminue quand l'éclairement augmente, (voir courbe ci-dessus) il faut donc remplacer le potentiomètre du montage de la page 8 par la photorésistance et une résistance de 10 kΩ. Allez, un peu d'aide : SI la valeur de l'entrée A0 est supérieure à un certain seuil (essayez la valeur 250 puis ajustez le seuil), ALORS allumez la diode lumineuse.