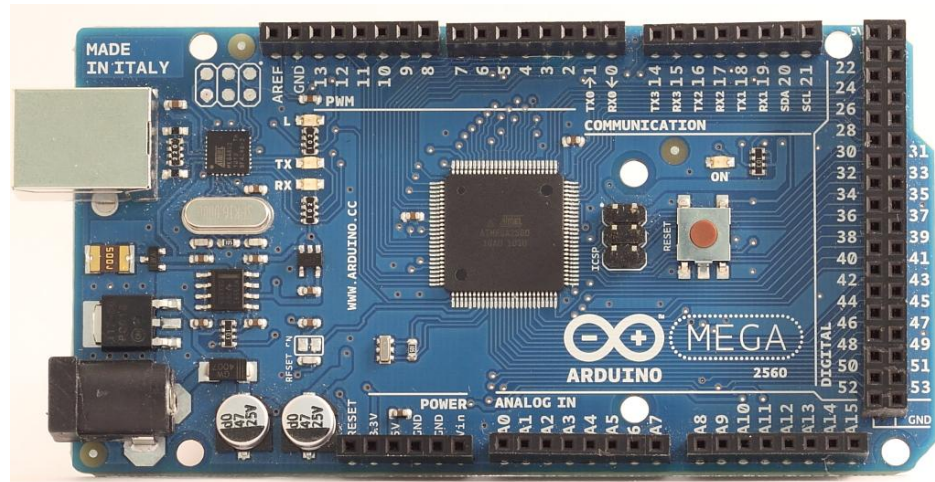


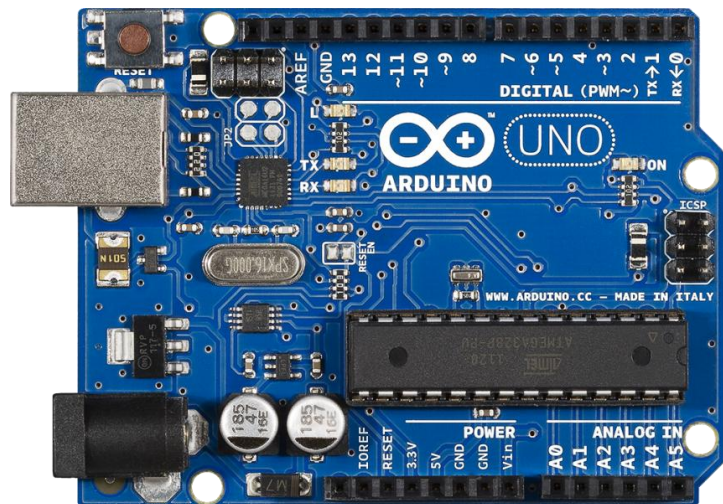
ARDUINO : Sorties numériques ou logiques

Les connecteurs "DIGITAL" :

Entourer les connecteurs notés DIGITAL sur la carte Arduino MEGA 2560.



Entourer les connecteurs notés DIGITAL sur la carte Arduino UNO.



I. Vocabulaire utilisé en électronique :

Donner les traductions de (pour une utilisation en électronique/électricité) :

Anglais	Français
Digital	numérique
Output	
Input	
High (état logique ou niveau logique)	
Low (état logique ou niveau logique)	
Pin	

Alimentation de la carte : la carte est alimentée (via le cordon USB) en 0V - 5V.

ARDUINO : Sorties numériques ou logiques

Compléter le tableau ci-dessous :

	Tension correspondante (0V ou 5V)
Etat logique ou niveau logique haut (High) :	
Etat logique ou niveau logique bas (Low) :	

II. Programmation d'une carte Arduino :

Demander une carte Arduino ainsi qu'un câble USB (voir le professeur).



Lancer le logiciel Arduino , sélectionner la bonne carte (c'est comme le port salut, c'est écrit dessus) à l'aide du MD (Menu Déroulant) « Outils » puis « Type de carte ». Sélectionner le port pour que la communication entre le PC et la carte puisse avoir lieu (MD « Outils » puis « Port série », prendre le 3^{ème} COM de la liste).

II. 1. Allumage et extinction d'une led :

Sélectionner dans le MD « Exemples » puis « 01.Basics » le fichier « Blink ».

Donner la traduction de blink :

--

Téléverser le programme (bouton en dessous de la barre des MD) dans la carte Arduino.

Observer (sur la carte !!) ce qui se passe (vous pouvez aussi vous aider du programme) :

Faire quelques modifications (à votre convenance) pour modifier le jeu de lumière. Expliquer en 2/3 lignes vos modifications et conséquences sur la led :

Le programme "blink" est le suivant :

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

ARDUINO : Sorties numériques ou logiques

Rappeler la signification des symboles / et */ :*

Donner la traduction de : Turns on an LED on for one second, then off for one second, repeatedly.

a) Fonction setup :

Donner la traduction de : the setup function runs once when you press reset or power the board

Donner la traduction de : initialize digital pin 13 as an output.

En déduire la traduction de : pinMode(13, OUTPUT);

b) Fonction loop :

Donner la traduction de : the loop function runs over and over again forever

Donner la traduction de turn the LED on (HIGH is the voltage level)

Donner la traduction de turn the LED off by making the voltage LOW

En déduire le niveau logique permettant d'allumer une LED :

Compléter le tableau ci-dessous :

	Etat de la led "L" (allumée ou éteinte)
Sortie 13 à l'état haut (HIGH)	
Sortie 13 à l'état bas (BAS)	

Donner la traduction de : write

ARDUINO : Sorties numériques ou logiques

En déduire la signification de :

<code>digitalWrite(13, HIGH);</code>	<code>digitalWrite(13, LOW);</code>
--------------------------------------	-------------------------------------

Donner la traduction de wait for a second :

--

Donner la traduction de :Delay

--

En déduire la signification de (préciser l'unité) : delay(1000);

--

c) Réécriture du programme :

Ce programme est encore modeste, seule la broche 13 est utilisée. On pourrait imaginer qu'il y en ait d'autres et que le programme soit beaucoup plus long. Il devient alors difficile de se souvenir à quoi correspond telle ou telle broche. Il vaut mieux alors donner un nom au numéro de la broche en créant une équivalence :

```
#define led 13 // déclaration d'une équivalence avec un nom "pertinent" (ne pas mettre de ;)
```

La ligne (`#define led 13 // déclaration d'une équivalence avec un nom "pertinent"`) est à mettre après les commentaires `*/` et avant le `void setup()`

Effectuer les modifications ci-dessus et remplacer tous les nombres 13 par la variable led (avec le #define, le logiciel associe tous les noms led à 13)

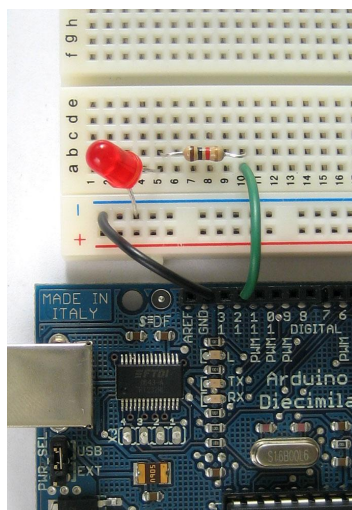
```
pinMode(led, OUTPUT); // la led est en sortie
```

De plus dans un souci de lisibilité et de compréhension, mettre les accolades des fonctions `setup` et `loop` alignées (voir le fichier présentation des cartes arduino). Vérifier le bon fonctionnement de votre programme.

III. Nouveau programme :

On souhaite commander une led comme la photo ci-dessous :

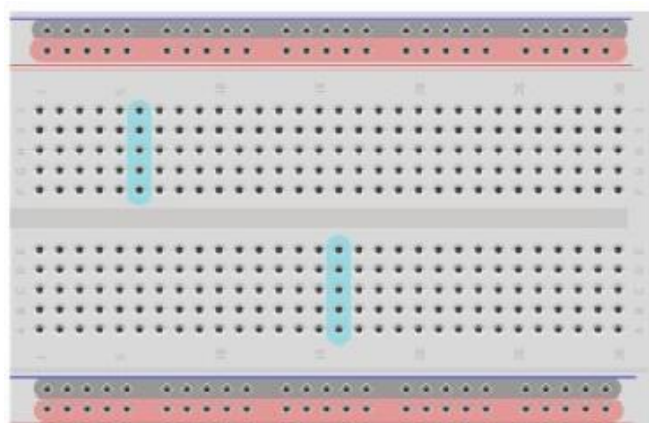
ARDUINO : Sorties numériques ou logiques



Il existe un outil très pratique lorsque l'on fait ses débuts en électronique ou lorsque l'on veut tester rapidement/facilement un montage. Cet accessoire s'appelle une **breadboard** (littéralement : planche à pain, techniquement : plaque d'essai sans soudure). Pour faire simple, c'est une plaque pleine de trous !

a) Principe de la breadboard ou labdec (marque répandue) :

Certes la plaque est pleine de trous, mais pas de manière innocente ! En effet, la plupart d'entre eux sont reliés. Voici un petit schéma rapide qui va aider à la compréhension.



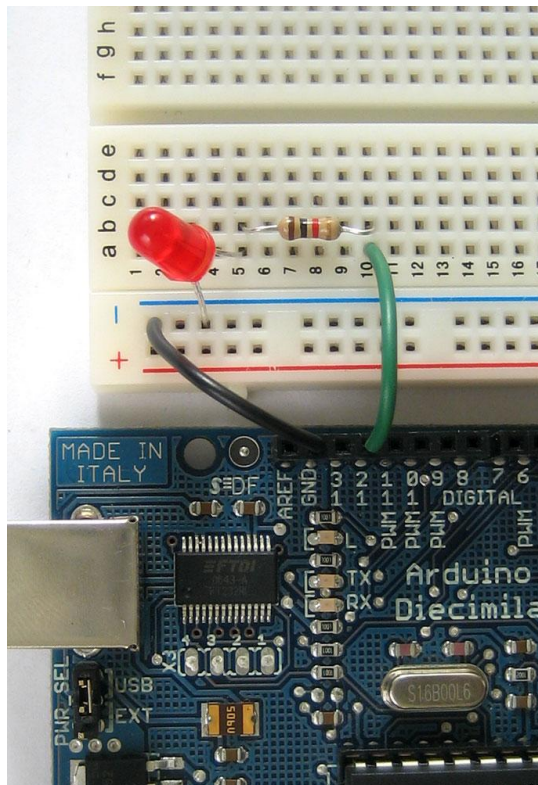
Les zones rouges et noires correspondent à l'alimentation (remarquer les traits horizontaux de couleur **rouge** et **bleu** qui rappellent les connexions). Par convention, les zones en **noir** représente la masse (**0V**) et les zones en **rouge** représente l'alimentation (**+5V**). Habituellement tous les trous d'une même **ligne** sont reliés sur cette zone (cf ci-dessus). Ainsi, vous avez une ligne d'alimentation parcourant tout le long de la carte.

Les zones en bleu ciel sont reliées entre elles par **colonne**. Ainsi, tous les trous sur une même colonne sont reliés entre eux. En revanche, chaque colonne est distincte.

Dernier point, un espace coupe la carte en deux de manière symétrique. Cette espace coupe aussi la liaison des colonnes. Ainsi, sur le dessin ci-dessus on peut voir que chaque colonne possède 5 trous reliés entre eux. Cet espace au milieu est normalisé et doit faire la largeur des circuits intégrés standards. En posant un circuit intégré à cheval au milieu, chaque patte de ce dernier se retrouve donc sur une colonne, isolée de la précédente et de la suivante.

Revenons à notre application :

ARDUINO : Sorties numériques ou logiques



Comment appelle t on les pattes d'une led ?

Comment les reconnaître ?

Noter sur l'image le nom des pattes et le méplat.

L'autre composant est une résistance.

A quelle broche de la carte Arduino est-elle reliée ?

La diode est relié à la résistance et à une broche notée GND de la carte Arduino.

Que signifie GND ?

Quelle est la valeur de la tension GND ?

Justifier la couleur du fil relié à GND :

Réaliser le montage ci-dessus (respecter les couleurs) sur plaque labdec (penser à modifier le programme) et valider le fonctionnement.

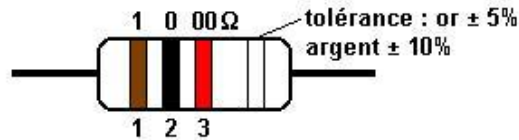
La résistance :

Quel est le nom de l'unité de la résistivité d'une résistance ?

Quel est le symbole de l'unité de la résistivité d'une résistance ?

Les valeurs des résistances se déterminent comme suit :

ARDUINO : Sorties numériques ou logiques



1 ère bague 1 er chiffre	2 ème bague 2 ème chiffre	3 ème bague multiplicateur
noir	0	$\times 1$
marron 1	1	$\times 10$
rouge 2	2	$\times 100$
orange 3	3	$\times 1000$
jaune 4	4	$\times 10000$
vert 5	5	$\times 100000$
bleu 6	6	$\times 1000000$
violet 7	7	
gris 8	8	
blanc 9	9	

Quelle est la valeur de la résistance sur la photo ci-dessus ?

Quelle est la valeur de la résistance donnée par le professeur ?

Quel appareil permet de mesurer la valeur d'une résistance ?

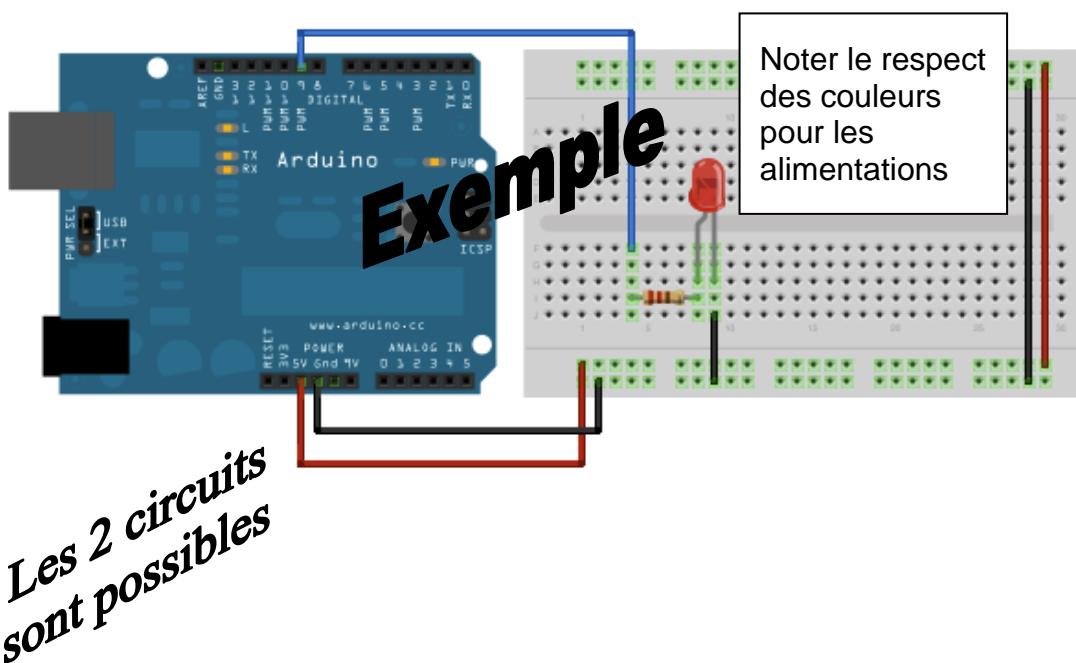
Vérifier votre valeur à l'aide cet appareil :

Quel est le rôle d'une résistance dans un circuit électrique ?

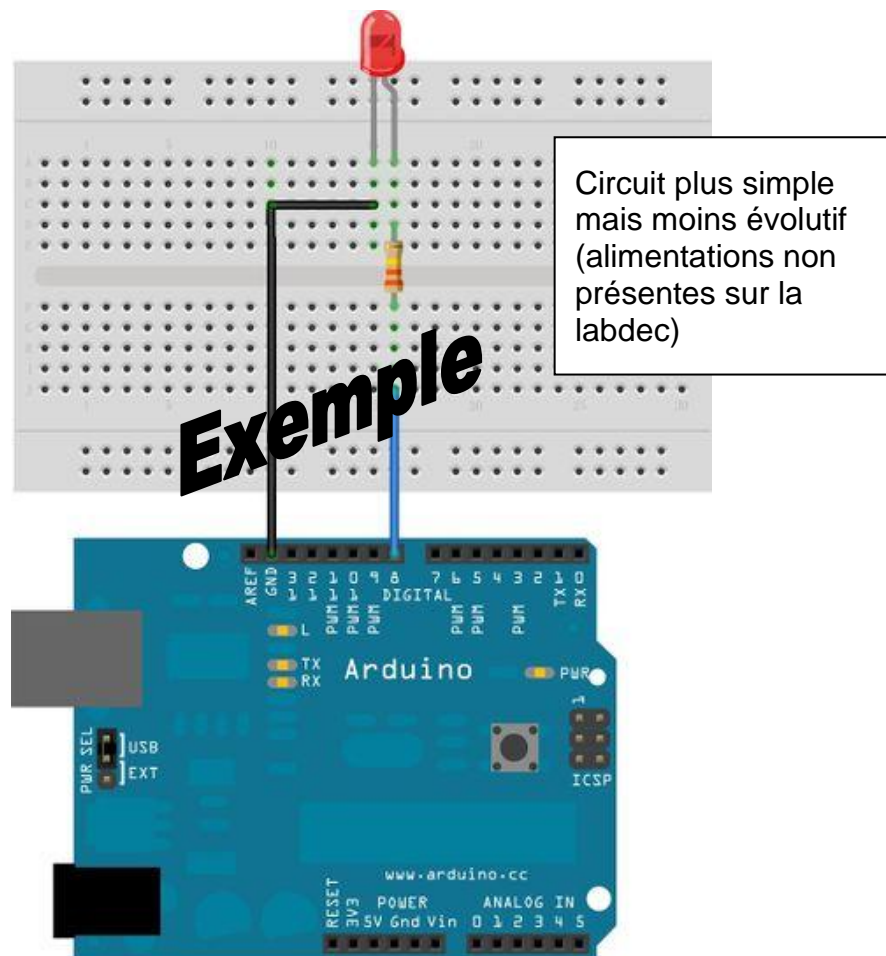
IV. Saisie du circuit :

Le logiciel fritzing permet de faire des circuits de manière simple et intuitive. Voici 2 exemples de circuit sous fritzing.

fritzing



ARDUINO : Sorties numériques ou logiques



L'alimentation **+5V** doit toujours être avec un fil **rouge**

La masse **0V** doit toujours être avec un fil **noir**

Lancer le logiciel fritzing (voir le fichier aide fritzing).

Refaire le circuit (que vous avez réalisé sur la labdec !!) sous fritzing en respectant la couleur de la led, la valeur de la résistance, le type de carte Arduino et la sortie correspondante.

V. Autre montage :

Essayer de faire clignoter 2 ou 3 leds en même temps : modifier le programme en conséquence.

Insérer votre programme et votre circuit sous fritzing (onglet Platine d'essai) :

Essayer de faire clignoter 2 ou 3 leds cycliquement : modifier le programme en conséquence.

Insérer votre programme et votre circuit sous fritzing (onglet Platine d'essai) :

ARDUINO : Sorties numériques ou logiques

VI. A retenir pour les sorties numériques ou logiques (DIGITAL) :

VI. 1. 1) Fonction pinMode() pour la configuration des broches en entrée ou en sortie

La fonction pinMode() permet de configurer une broche (pin en anglais) en entrée ou sortie

pinMode(numBroche, typeBroche);

numBroche = numéro de la broche de 1 à ... voir la carte Arduino

typeBroche = INPUT ou OUTPUT // entrée ou sortie

Exemple d'utilisation :

pinMode(13, OUTPUT); //la broche 13 est en sortie

#define pin_led 13 // déclaration d'une équivalence

pinMode(pin_led, OUTPUT); // pin_led est associé à la broche 13 qui est en sortie

VI. 2. 2) Fonction digitalWrite() pour écrire en sortie un niveau logique (DIGITAL) : 0 ou 1

La fonction digitalWrite() permet d'appliquer (d'écrire) un niveau logique sur une broche auparavant configurée en sortie :

digitalWrite(numBroche, etat);

numBroche = numéro de la broche de 1 à ... voir la carte Arduino

etat = LOW ou 0, HIGH ou 1

Exemple d'utilisation :

digitalWrite(13, HIGH);//niveau haut sur la broche 13 qui doit être configurée en sortie

digitalWrite(pin_led, HIGH);//niveau haut sur la broche 13 (led) qui doit être configurée en sortie

digitalWrite(13, 1);//niveau haut sur la broche 13 qui doit être configurée en sortie

digitalWrite(pin_led, 1);//niveau haut sur la broche 13 (led) qui doit être configurée en sortie