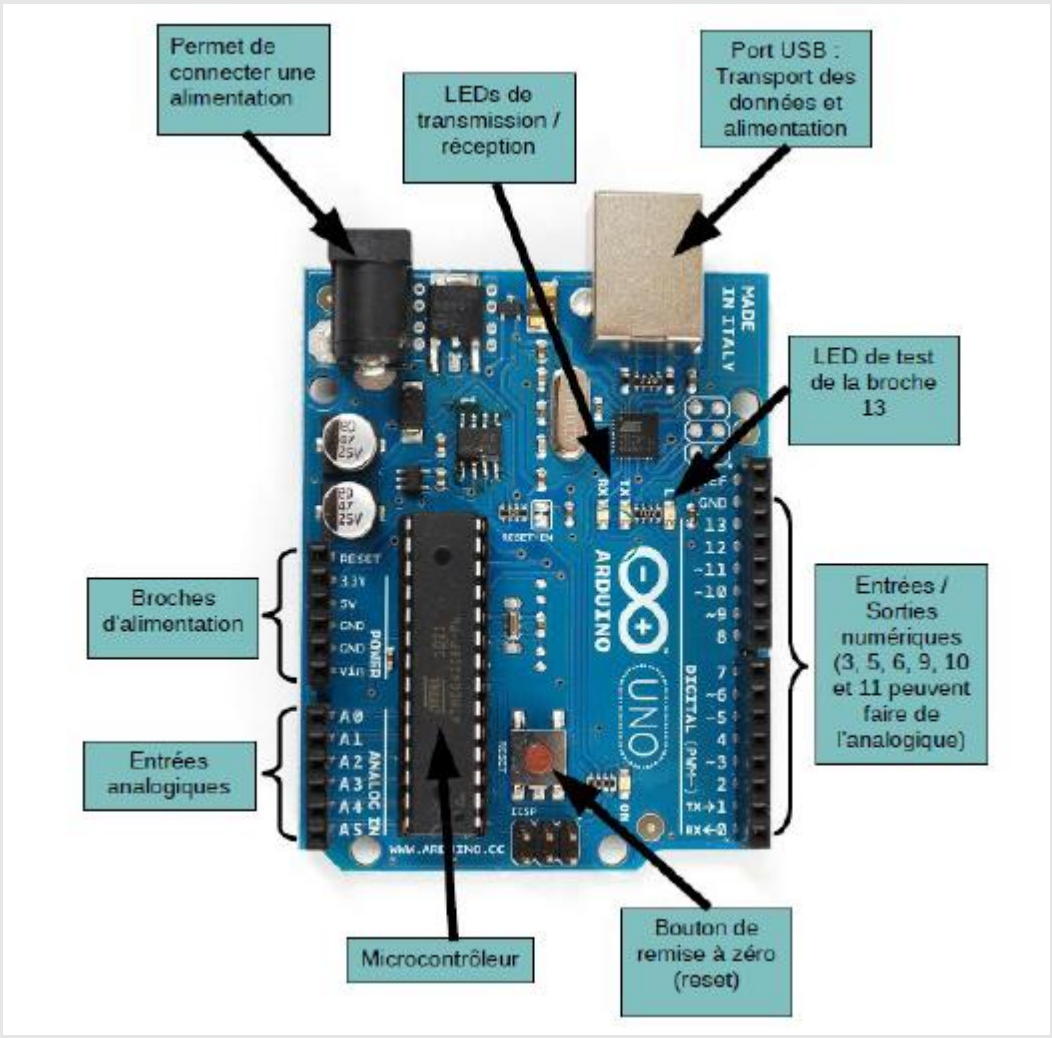
PRESENTATION DE LA CARTE ARDUINO ET DE SON INTERFACE LOGICIELLE

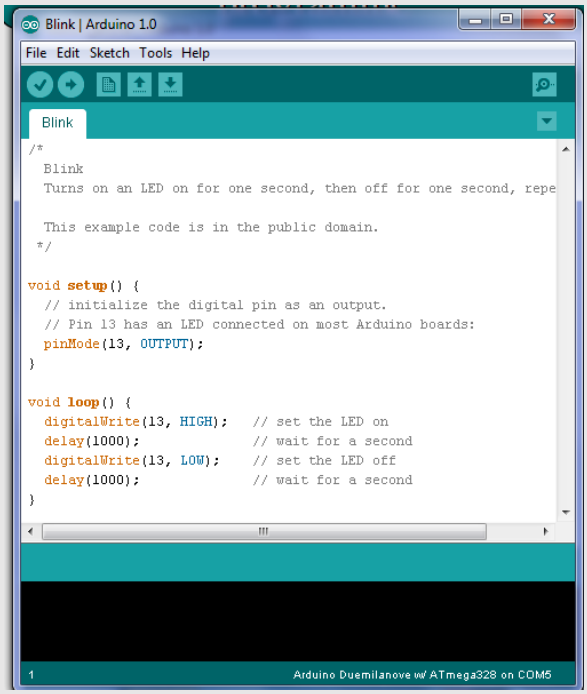
1. LA CARTE A MICROCONTROLEUR

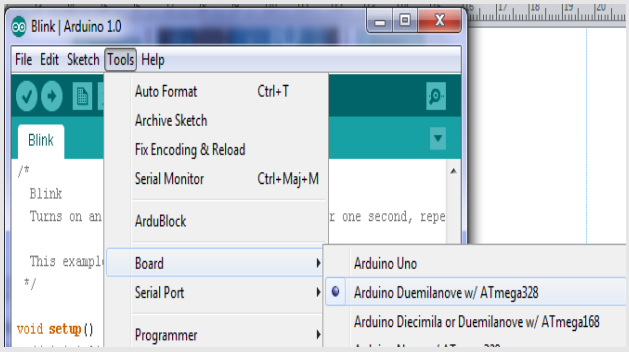
- Le rôle de la carte Arduino est de stocker un programme et de le faire fonctionner.
- La carte reçoit des informations analogiques ou numériques sur ses entrées.
- Le microcontrôleur traitera ces informations et les transmettra vers les sorties numériques.



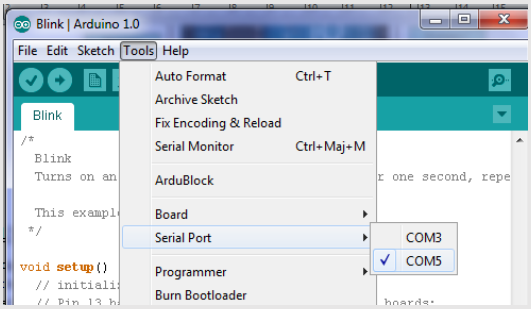
2. L'INTERFACE LOGICIELLE

Sur un ordinateur, le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte au travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information.



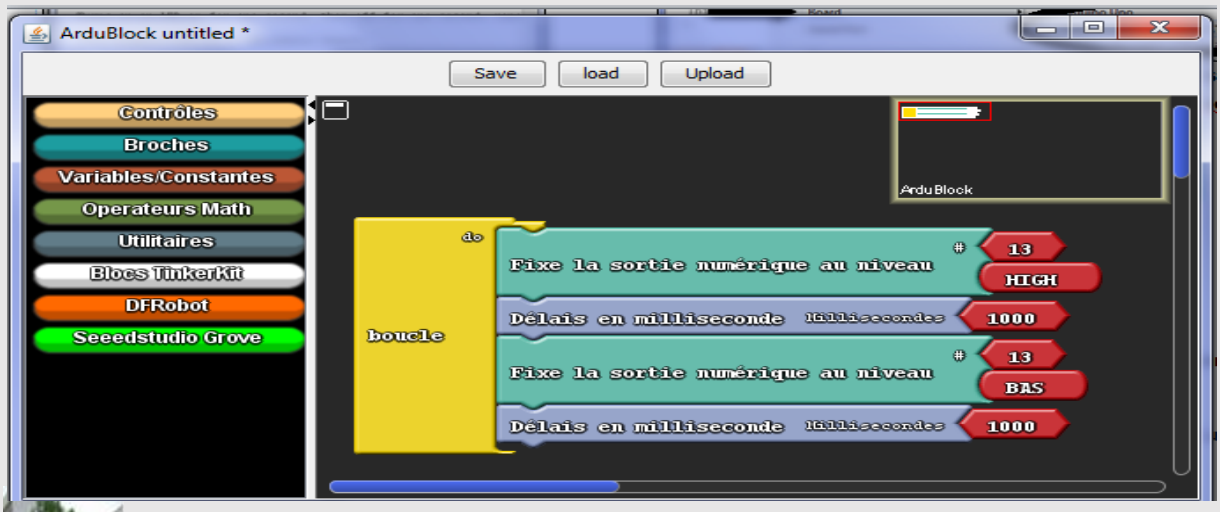


Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de carte (vérifier le nom de votre carte).



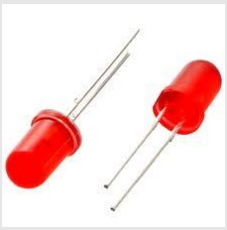
Il est nécessaire de sélectionner le port USB (le dernier de la liste quand la carte est branchée)

Pour simplifier la programmation, nous allons utiliser ARDUBLOCK qui est un inclus dans l'IDE arduino (menu tools) et qui permet de programmer avec des objets à la manière de SCRATCH.



Save permet de sauvegarder un document.
Load permet d'ouvrir un document existant.
Upload convertit le programme en langage C et le transfert dans le microcontrôleur de la carte Arduino.

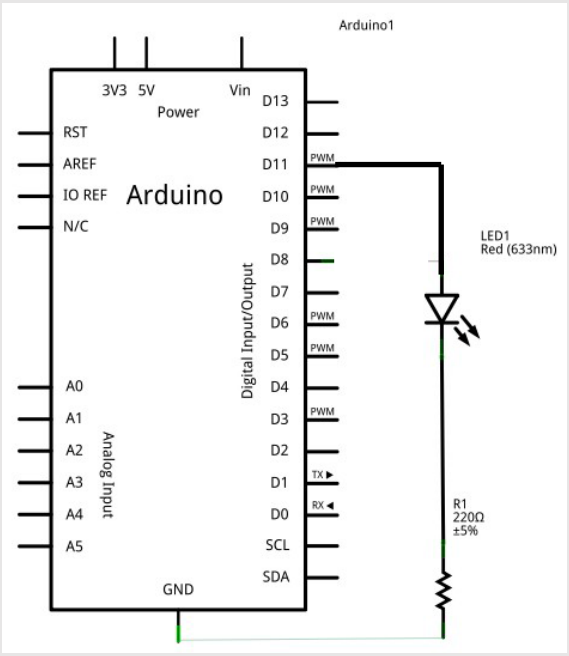
3. Faire clignoter une DEL



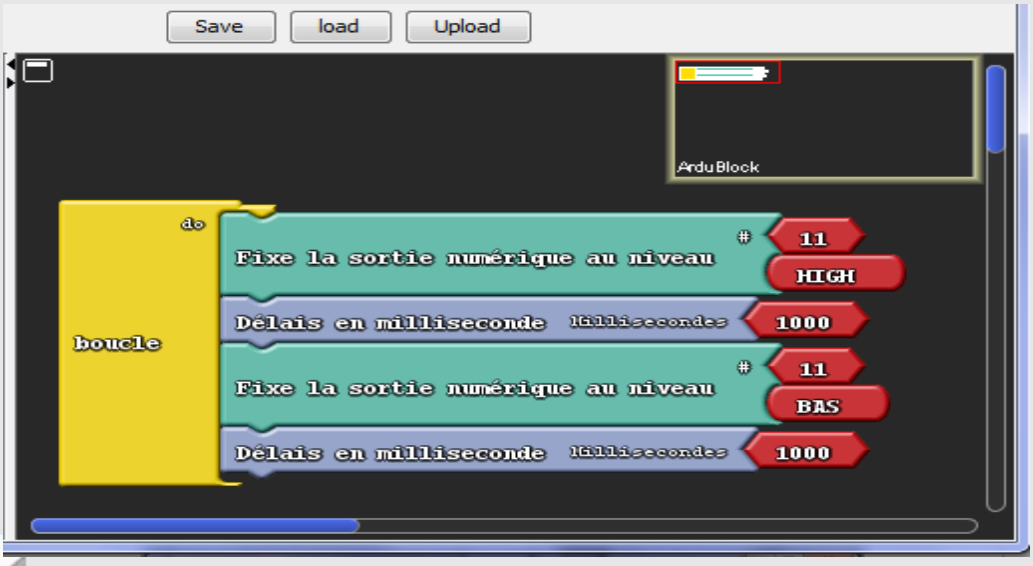
Identifier le composant à droite:……..
Quel est sa fonction? …….
Calculer la résistance R1 sachant que Vf=1.5V et on limite le courant à 15mA.

1) Réaliser le schéma suivant :

Les sorties Arduino limite le courant à 15mA. On peut supprimer la résistance R1.



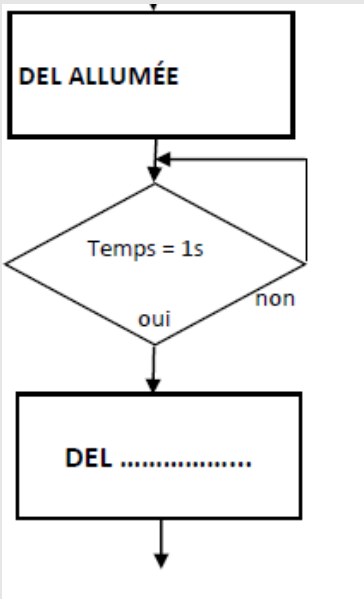
2) Recopier le programme ci-dessous .



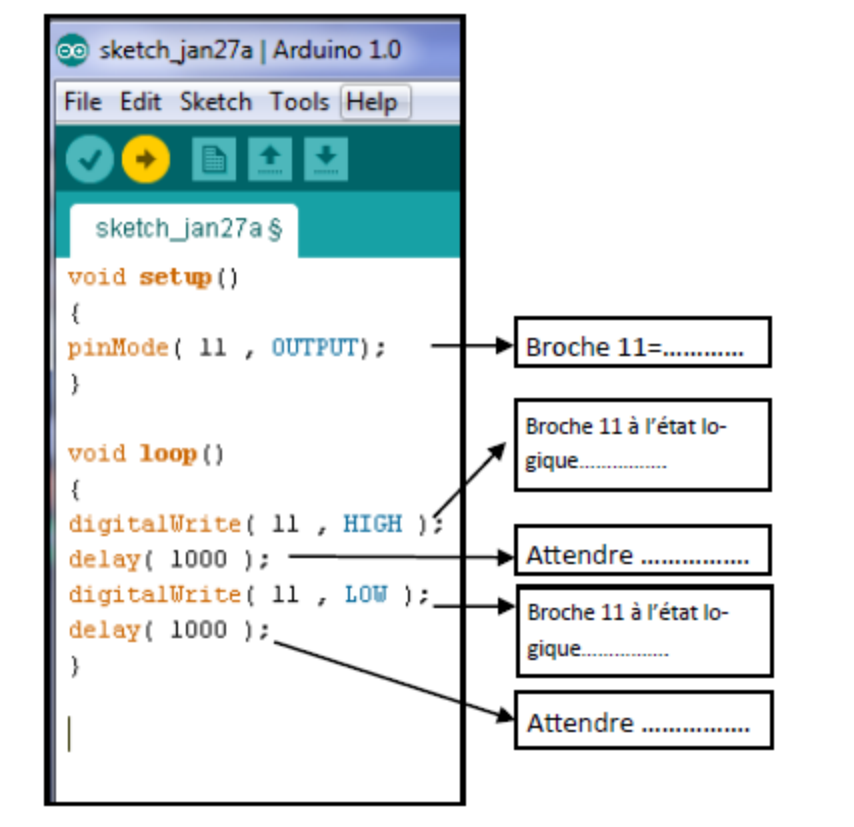
3) Upload

Que fait le programme: ……
Modifier le programme sur Ardublock pour faire clignoter 2 fois plus vite (délais en milliseconde =……)

4) Compléter le logigramme:



5) Lire le programme traduit en C et commenter:



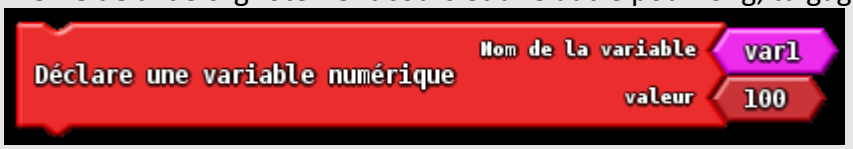
6) Une fois ce petit programme testé, tu peux essayer de diminuer le délai afin de voir quelle est la valeur à partir de laquelle l'œil ne voit plus la LED clignoter (persistance rétinienne). Donner la valeur.

7) A l'aide d'un oscilloscope relever l'oscillographe sur la broche D11.

   Donner la fréquence de ce signal ainsi que son amplitude.

8) Ensuite tu vas écrire le logigramme pour faire un SOS lumineux en langage Morse avec la LED (aide-toi d'Internet pour trouver les informations).

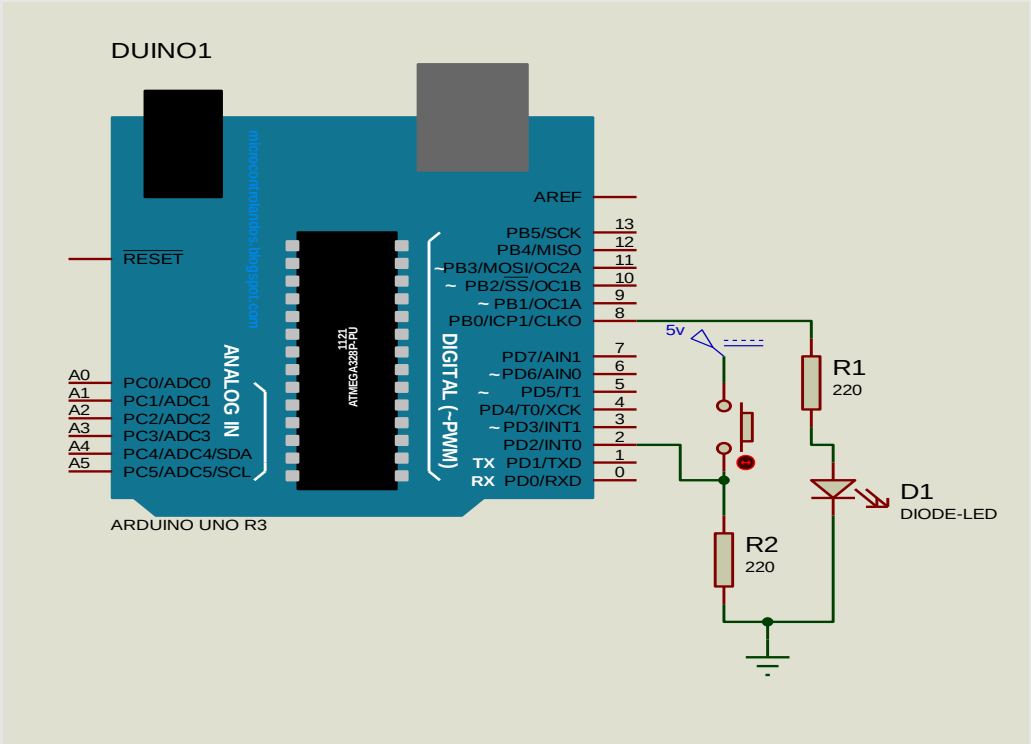
   Pour éviter de modifier plusieurs paramètres à chaque fois, utilise une variable pour fixer le délai de clignotement cours et une autre pour long, tu gagneras du temps !



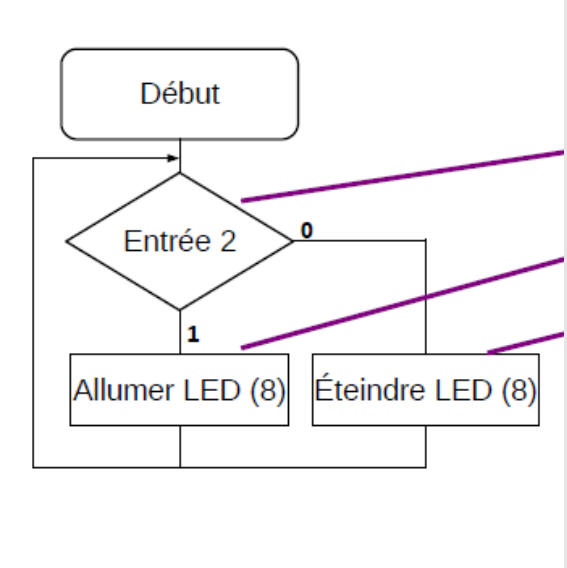
Tu peux aussi utiliser le bloc Répète pour effectuer plusieurs fois la/les même(s) opération(s).

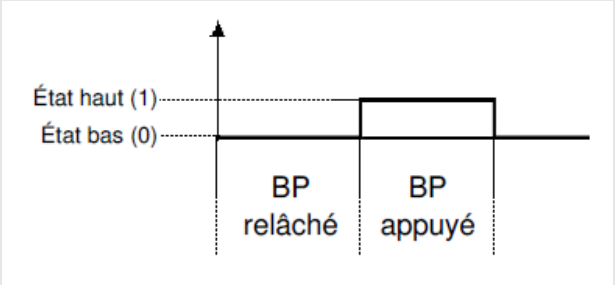
4. Le bouton poussoir (BP)

Schéma de branchement



Logigramme





1) Réalise le programme sous Ardublock.

2) Vérifie le bon fonctionnement du programme.

3) Maintenant réalise le programme suivant : j'appuie une fois sur le bouton poussoir, la LED s'allume (et reste allumée) ; j'appuie une deuxième fois sur le bouton poussoir, la LED s'éteint.

Il faut utiliser une variable : j'appuie une fois, la variable est mise à 1, j'appuie une deuxième fois, la variable est mise à 0.

4) Réalise le programme sous Ardublock.

5) Vérifie le bon fonctionnement du programme.

6) On veut maintenant réaliser une minuterie. Lors d'un appuie sur le bouton poussoir la led reste éclairée 10 secondes. Réaliser le programme et vérifier son fonctionnement.