

Matériel nécessaire :

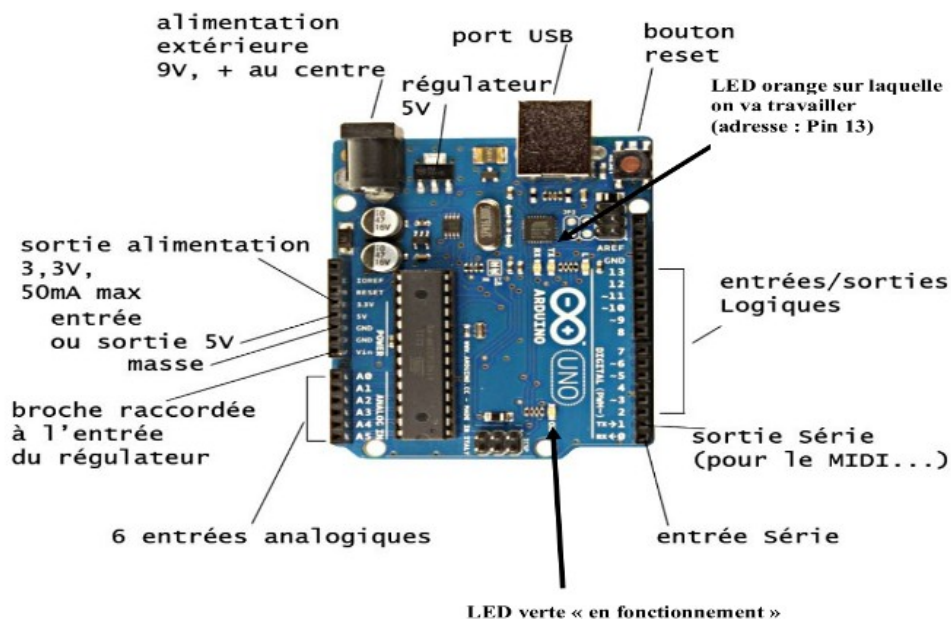
- * - Un module Arduino UNO
- * - Un cordon USB

Les connexions à réaliser : connectez le cordon USB entre l'ordinateur et la carte Arduino.

I. Découverte de la carte arduino

La carte Arduino Uno que nous allons utiliser est une sorte de petit ordinateur. Il possède un microcontrôleur (microprocesseur + mémoire+ interface de gestion des entrées/sorties).

La carte Arduino UNO :



Il possède donc des entrées / sorties logiques (« digital » en anglais), des entrées analogiques (0 à 5V), des moyens de communication série et bien sûr un résonateur à quartz d'une fréquence de 16MHz.

Premiers programmes

Ressources disponibles :

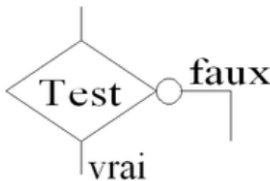
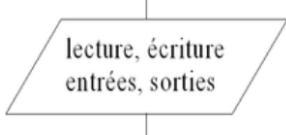
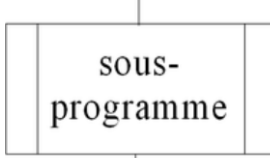
Vous disposez de différentes ressources concernant Arduino: classeurs, documents numériques ([language Arduino.pdf](#)) et le site internet officiel <http://arduino.cc/> sur l'onglet « reference" notamment (attention à ne pas faire de traduction automatique de la page car les logiciels de traduction ne fonctionnent pas bien avec des explications technologiques).

L'algorithmie :

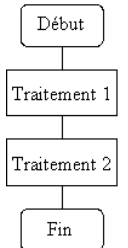
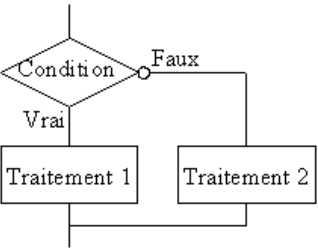
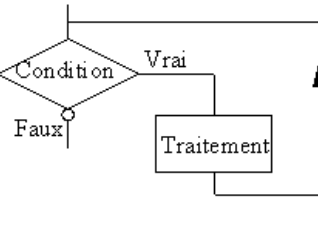
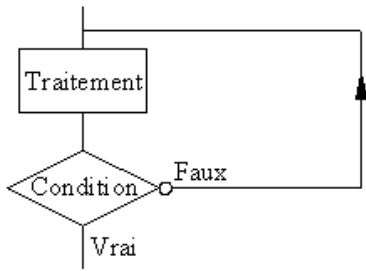
L'algorithme est une suite d'instructions, qui une fois exécutée correctement, conduit à un résultat donné. Pour comprendre, expliquer, analyser un programme nous allons utiliser l'algorithmie sous forme graphique (organigramme) plus facilement compréhensible quand la problématique est simple.

S1 : Découverte du système Arduino

Norme :

Symboles normalisés	Commentaires:
	<p>Les tests ou branchements conditionnels :</p> <p>La pointe vers le haut est l'entrée du test, la pointe avec le rond est le résultat du test lorsqu'il est faux, la pointe vers le bas est le résultat du test lorsqu'il est vrai.</p>
	<p>Mise à disposition d'une information à traiter ou enregistrement d'une information traitée.</p>
	<p>Appel de sous programme (appelé aussi « fonction » sous arduino)</p>

Quelques structures algorithmiques:

Séquence linéaire	Séquence alternative "si...alors...sinon"	Séquence répétitive "tant que...faire..."	Séquence répétitive "répéter...jusqu'à..."
			
<p>Début "Traitement1" "Traitement2" Fin</p>	<p>Si "condition" alors "Traitement 1" sinon "Traitement 2" Fin si</p>	<p>Tant que "condition" faire "traitement" Fin tant que</p>	<p>Répéter "traitement" jusqu'à "condition"</p>

Structure d'un programme Arduino:

La structure de base d'un programme Arduino (sketch) :

variables (déclaration des variables)

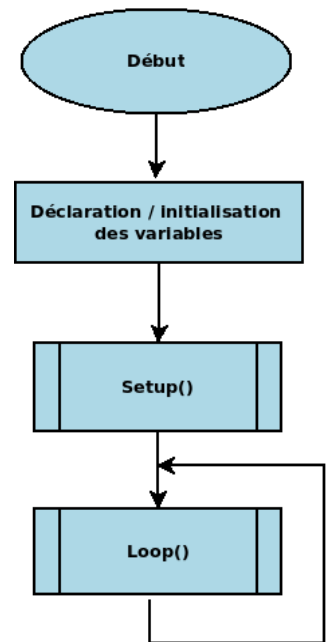
void setup() {

Commandes d'initialisation (entrées, sorties, ...)}

void loop() {

Instructions ;

}



- **A savoir :**

- La structure void setup() n'est exécutée qu'une seule fois.
- La structure void loop() est exécuté indéfiniment (Loop = boucle)
- La ligne de commentaires : // ... tout texte après les deux slashes est un commentaire.
- Le bloc de commentaires : /* */ tout texte entre les deux étoiles est un commentaire, le texte peut s'étendre sur plusieurs lignes.

Les premiers programmes simples

a) Programme N°1 : clignotement d'une led

Après avoir lancé le logiciel Arduino : Écrivez (ou copiez) le programme (sketch) suivant en respectant très exactement la mise en page :

S1 : Découverte du système Arduino

/* début du programme

Nom : Cligno_LED13

Fonctionnement : Commande sans arrêt le clignotement d'une LED. Les temps ON (active ou allumée) et OFF (inactive ou éteinte) sont identiques : 0.1s. La LED celle connectée sur la broche 13.
*/

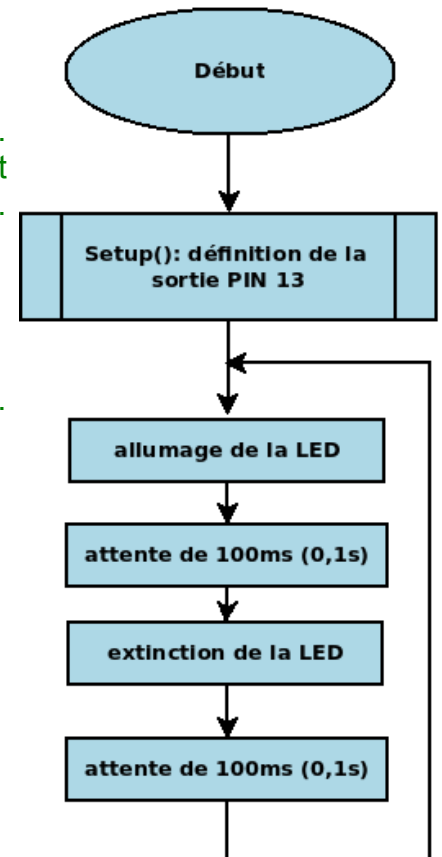
```
void setup()
```

```
{  
  pinMode(13, OUTPUT); // la broche 13 est définie en sortie.  
}
```

```
void loop()
```

```
{  
  digitalWrite(13, HIGH); // allume la LED  
  delay(100); // attend en ms (ici 100)  
  digitalWrite(13, LOW); // éteint la LED  
  delay(100); // attend en ms  
}
```

```
//fin du programme//
```



Ce programme fait clignoter la LED du module Arduino toutes les 0.2 secondes (ON-OFF). L'expression ON-OFF représente la période. Ici la durée ON est égale à la durée OFF (0,1 s).

L'instruction pinMode

pinMode (n° de broche, [OUTPUT ou INPUT]) ; déclare la broche « n° de broche » en entrée ou en sortie. Cela ne concerne que les entrées numériques (appelées aussi Tout Ou Rien. En anglais: digital)

L'instruction digitalWrite:

digitalWrite (n° de broche, [HIGH ou LOW]) met un niveau haut (1) ou bas (0) sur la broche désignée. Il s'agit bien d'une sortie logique appelée aussi TOR (Tout Ou Rien) ou digital (en anglais).

L'instruction delay ():

Delay (100) : le microcontrôleur de la carte Arduino attend 100 millisecondes.

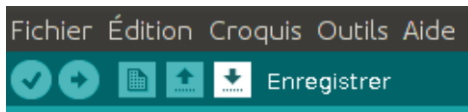
On remarque:



- que les broches numériques ("Digital Pin" en anglais) peuvent être soit des entrées soit des sorties.
- quand on veut savoir ce que fait une instruction on se reporte au site internet Arduino qui est la seule source fiable (site en anglais)

S1 : Découverte du système Arduino

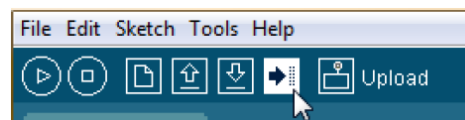
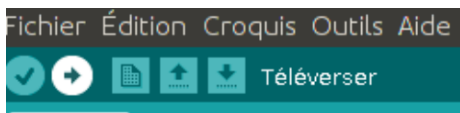
1. Sauvegardez ce programme sous le nom : Clign_LED13 à l'aide de l'icône représenté en blanc ci-dessous (exemple sur les versions anglaise et française):



2. Allez dans le menu « croquis » et choisir «vérifier/compiler » (ou CTRL R)

Cette fonction permet de vérifier qu'il n'y a pas d'erreurs de programmation (erreurs de syntaxe)

3. S'il n'y a pas d'erreurs, chargez (Upload ou téléverser) le programme dans la carte Arduino à l'aide de l'icône représentée en blanc ci-dessous :



Le programme s'exécute dès que le chargement est effectué. Un problème de connexion ? Vérifier le bon choix du port de communication (menu « outil » puis « port série »).

Le driver USB qui était demandé lors de la connexion de la carte Arduino au PC est-il installé ?

Si oui, allez dans le menu Tools- Serial Port, choisissez le bon port COM.

4. La LED clignote ? Oui, c'est gagné. Autrement appelez le professeur et expliquez lui clairement le problème auquel vous êtes confronté.
5. Si vous deviez traduite en français la fonction « digitalWrite » comment l'appelleriez-vous ?
6. Si vous deviez traduite en français la fonction « delay » comment l'appelleriez-vous ?
7. Modifiez maintenant le programme pour que la LED reste allumée pendant 1s et éteinte pendant 2s.
 - Pour constater le changement, n'oubliez pas de recharger le programme dans la carte Arduino après l'avoir modifié. Chronométrez !
 - Faire valider par le professeur

Remarque : à chaque modification du programme enregistrer celui-ci dans votre répertoire.

8. Modifiez maintenant le programme pour que la LED reste allumée pendant 1s, éteinte pendant 2s puis se rallume 4 secondes, s'éteigne 2s, se rallume 2 secondes et s'éteigne 5 secondes.
 - Pour constater le changement, n'oubliez pas de recharger le programme dans la carte Arduino après l'avoir modifié. Chronométrez !
 - Faire valider par le professeur