



## Principes de PROGRAMMATION

Les microcontrôleurs qui se trouvent sur les cartes électroniques de types PICAXE ou ARDUINO sont comparables à un microprocesseur d'ordinateur, capable donc de **gérer des instructions permettant le pilotage de petits systèmes techniques**.

Après les choix du langage ainsi que du logiciel de programmation, **la configuration** doit s'effectuer par deux paramétrages essentiels :

==> Le port de connexion (*quel USB sur le poste Informatique ?*)

==> L'identification du microcontrôleur présent (*8M2, 18M, ...sous PICAXE*) ou de la carte utilisée (*UNO, MEGA, LEONARDO, ...sous ARDUINO*)

### Une démarche : Algorithme-Organigramme-Bloc-Code

#### Étape 1

Pour décrire le fonctionnement attendu d'un système, on commence donc simplement à écrire chronologiquement ses conditions de fonctionnement et ses actions voulues.

On appelle cela **l'ALGORITHME**.

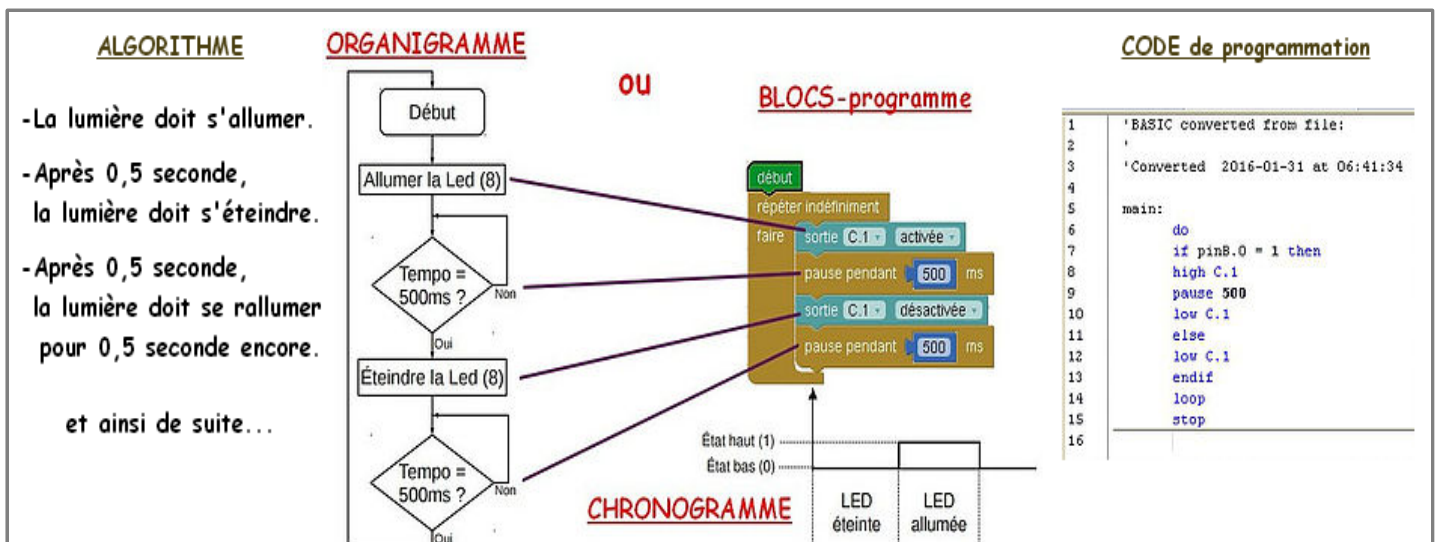
#### Étape 2

On traduit alors le tout en une **succession de CONDITIONS et d'ACTIONS** sous forme d'**ORGANIGRAMME** (Logigramme) ou bien de **BLOCS**.

#### Étape 3

Au téléversement du programme dans le circuit du système, le programme sera automatiquement **compilé en CODE** machine pour être exécuté.

Remarque complémentaire ==> On peut aussi schématiser le fonctionnement selon un **CHRONOGRAMME**.



## Les principes d'un LOGIGRAMME ou ORGANIGRAMME

-L'organigramme obéit à des règles d'écriture très simples :

Il débute toujours par une case début et il n'y a que **trois types de cases**.



Un ovale qui correspond au Début ou Fin (si fin il y a) de l'organigramme.



Correspond à une action à effectuer.



Correspond à une question à laquelle on peut répondre uniquement par oui ou par non.

Ressource ==> [http://technopujades.free.fr/picaxe/ressources\\_theorique/fiche-organigramme.pdf](http://technopujades.free.fr/picaxe/ressources_theorique/fiche-organigramme.pdf)

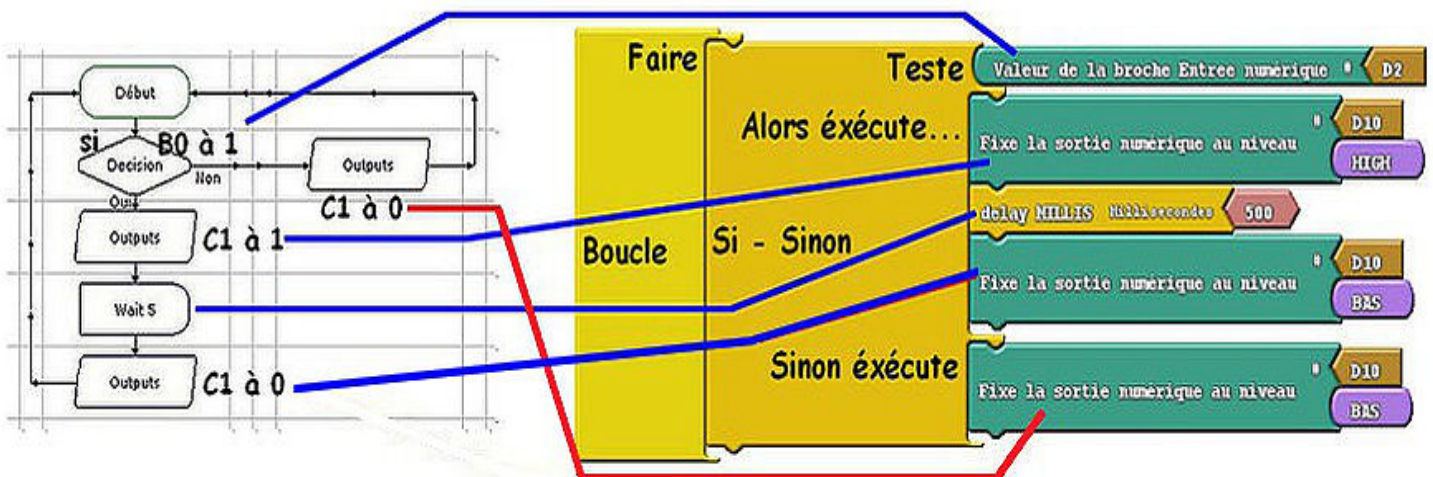
et plus officiellement selon la normalisation ==> [http://www.wikiwand.com/fr/Organigramme\\_de\\_programmation](http://www.wikiwand.com/fr/Organigramme_de_programmation)

## Exemple de correspondance :

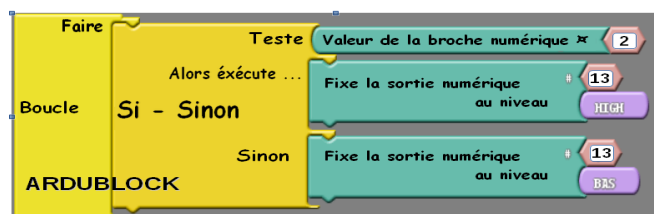
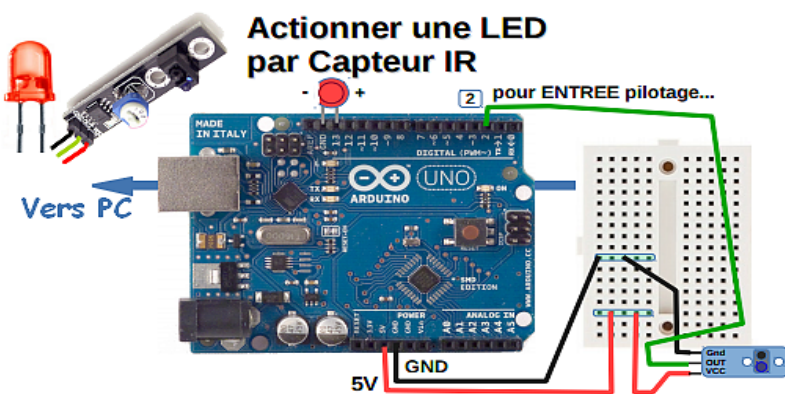
### Logigramme

*puis*

### BLOCS-Programme



(Remarque : La programmation par bloc est plus proche du langage commun et facilite la traduction de l'algorithme en Conditions-Actions)



```
void setup()
{
  pinMode( 2 , INPUT);
  pinMode( 13 , OUTPUT);
}

void loop()
{
  if (digitalRead(2))
  {
    digitalWrite( 13 , HIGH );
  }
  else
  {
    digitalWrite( 13 , LOW );
  }
}
```



### ALGORITHME

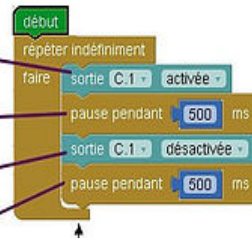
- La lumière doit s'allumer.
  - Après 0,5 seconde, la lumière doit s'éteindre.
  - Après 0,5 seconde, la lumière doit se rallumer pour 0,5 seconde encore.
- et ainsi de suite...

### ORGANIGRAMME

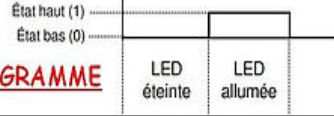


OU

### BLOCS-programme



### CHRONOGRAMME



### CODE de programmation

```
1 'BASIC converted from file:
2 '
3 'Converted 2016-01-31 at 06:41:34
4
5 main:
6   do
7     if pinB.0 = 1 then
8       high C.1
9       pause 500
10      low C.1
11     else
12       low C.1
13     endif
14   loop
15 stop
16
```

### ALGORITHME

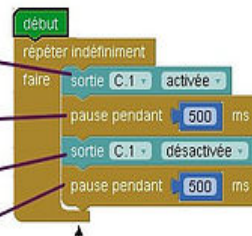
- La lumière doit s'allumer.
  - Après 0,5 seconde, la lumière doit s'éteindre.
  - Après 0,5 seconde, la lumière doit se rallumer pour 0,5 seconde encore.
- et ainsi de suite...

### ORGANIGRAMME

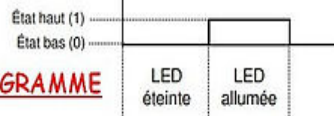


OU

### BLOCS-programme



### CHRONOGRAMME



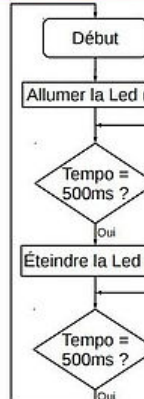
### CODE de programmation

```
1 'BASIC converted from file:
2 '
3 'Converted 2016-01-31 at 06:41:34
4
5 main:
6   do
7     if pinB.0 = 1 then
8       high C.1
9       pause 500
10      low C.1
11     else
12       low C.1
13     endif
14   loop
15 stop
16
```

### ALGORITHME

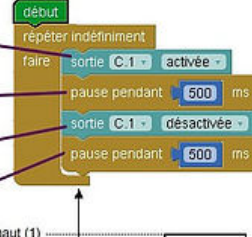
- La lumière doit s'allumer.
  - Après 0,5 seconde, la lumière doit s'éteindre.
  - Après 0,5 seconde, la lumière doit se rallumer pour 0,5 seconde encore.
- et ainsi de suite...

### ORGANIGRAMME

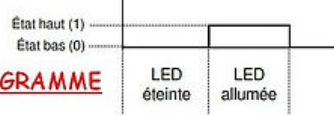


OU

### BLOCS-programme



### CHRONOGRAMME



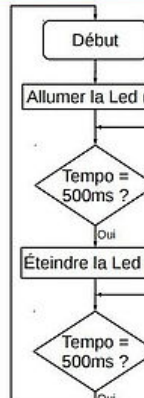
### CODE de programmation

```
1 'BASIC converted from file:
2 '
3 'Converted 2016-01-31 at 06:41:34
4
5 main:
6   do
7     if pinB.0 = 1 then
8       high C.1
9       pause 500
10      low C.1
11     else
12       low C.1
13     endif
14   loop
15 stop
16
```

### ALGORITHME

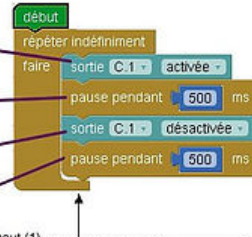
- La lumière doit s'allumer.
  - Après 0,5 seconde, la lumière doit s'éteindre.
  - Après 0,5 seconde, la lumière doit se rallumer pour 0,5 seconde encore.
- et ainsi de suite...

### ORGANIGRAMME

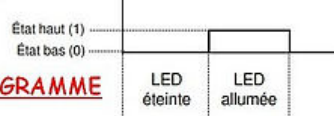


OU

### BLOCS-programme



### CHRONOGRAMME



### CODE de programmation

```
1 'BASIC converted from file:
2 '
3 'Converted 2016-01-31 at 06:41:34
4
5 main:
6   do
7     if pinB.0 = 1 then
8       high C.1
9       pause 500
10      low C.1
11     else
12       low C.1
13     endif
14   loop
15 stop
16
```