

Le langage C-ARDUINO

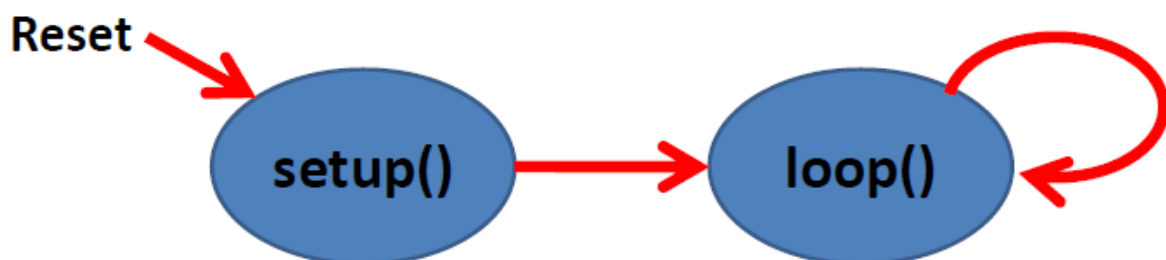
```
sketch_apr10a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Ici la configuration et l'initialisation

Ici le traitement

Le langage C-ARDUINO

- Au moment de lancement de l'application sous ARDUINO. Ce dernier lance le contenu de la procédure `setup()` en premier une seule fois puis lance le contenu de la procédure `loop()` infiniment.



Les Instructions de base

Les I/O numériques (digitales)

- La carte ARDUINO UNO possède 14 pins en I/O configurable individuellement. Numérotée de 0 à 13.
- Le C-ARDUINO utilise trois fonctions pour :
 - Configurer soit en entrée ou en sortie (`pinMode`).

- Communiquer soit en lecture ou en écriture (`digitalRead`, `digitalWrite`).
- Avec quelques constantes:
 - Pour la configuration (`INPUT`, `OUTPUT`).
 - Pour la communication (`LOW`, `HIGH`).
- Faites attention au langage C, c'est un langage qui respecte les lettres minuscules et majuscules .

La fonction `pinMode`:

- Syntaxe: `pinMode(pin, mode)`.
- pin: est une valeur entre 0 et 13 indique la pin qu'on va configurer.
- mode: est une valeur présente le sens de la pin (`INPUT` : pour configurer en entrée, `OUTPUT` : pour configurer en sortie).
- Exemple : configurer la pin 12 en entrée et la pin 7 en sortie.
- `pinMode (12, INPUT) ;`
- `pinMode (7, OUTPUT) ;`

```
Void setup() { pinMode(12, INPOUT);
               pinMode (7, OUTPUT) ;
}
```

La fonction `digitalWrite` :

- Syntaxe : `digitalWrite(pin, value)`.
- pin : est une valeur entre 0 et 13 indique la pin qu'on va écrire.
- value : est une valeur présente la valeur qu'on va écrire dans la pin (`LOW`: pour forcer à 0V, `HIGH`: pour forcer à 5V).
- Exemple : mettre à 1 la pin 4 et à 0 la pin 5.
- `digitalWrite (4, HIGH) ;`
- `digitalWrite (5, LOW) ;`

La fonction `digitalRead` :

- Syntaxe : `digitalRead(pin)`.
- pin : est une valeur entre 0 et 13 indique la pin qu'on va lire.
- Cette fonction retourne : soit `LOW` ou `HIGH` selon l'état de la pin au moment de la lecture.
- Exemple : lire l'état des pins 2 et 3.
- `Etat_pin2 = digitalRead (2) ;`
- `Etat_pin3 = digitalRead (3) ;`

Travail à Faire : donner schéma et programme pour les exercices suivant :

Exercice 1: lire d'état de la **pin 3** puis copier cet état dans la **pin 4**.

- Exercice 2: écrire un programme qui clignote deux LED reliées à la **pin 6** et **7** par une fréquence de **2 Hz**.

Note la commande **delay**(duré en mili_secondes) permet d avoir une temporisation.

Exemple : `delay(1000)` ; c est une tempo de 1000ms=1 seconde.

- Exercice 3: écrire un programme qui fait l'**émulation** d'une **bascule SR**, dont l'entrée **S** est la **pin 2**, l'entrée **R** est la **pin 3** et la sortie **Q** est la **pin 5**.

Exercice 4: écrire un programme qui fait le clignotement des **LED (pin 4, 5, 6 et 7)** **20 fois** par une fréquence de **1Hz**.

- Exercice 5: écrire un programme qui clignote la **LED (pin 5)** si le Poussoir **P1 (pin 2)** est appuyé.

- Exercice 6: Lorsqu'on presse sur le bouton Poussoir **P1 (pin 2)**, la LED **L1 (pin 4)** doit s'allumer durant une seconde. Ensuite la LED **L2 (pin 5)** doit aussi s'allumer (les deux seront allumées) durant une seconde. Finalement, seule **L2** sera allumée, également une seconde. Après, les 2 LED seront éteintes et le programme attendra de nouveau qu'on presse sur le bouton Poussoir .