ARD_TP2	Arduino – Algorithme

G.COLIN

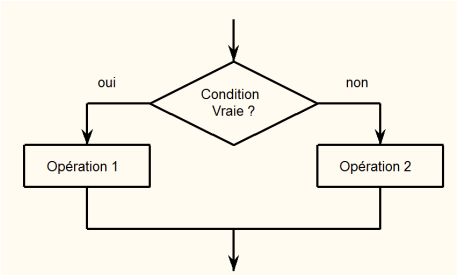
Objectifs : ⇨ Mettre en œuvre des programmes de base sur carte Arduino en s'aidant de ressources Internet
⇨ Faire la correspondance entre un algorigramme et langage C

Ressources disponibles :

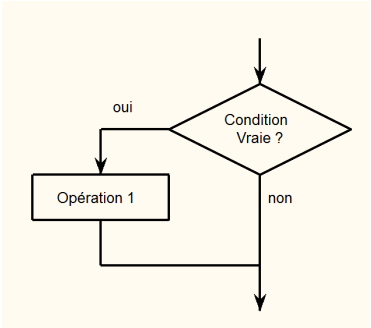
- Site en français : http://www.mon-club-elec.fr/ puis rubrique **La traduction Française de la référence Arduino**
- Cours « Algorithme et Algorigramme » sur le site mede.fr/1sti

La structure IF…ELSE …

Algorithme	Algorigramme	Traduction en C

- Si condition vraie
- Faire opération 1
Sinon
- Faire opération 2
-Fin si
		if (condition)
		opération1 ;
		else
		opération2 ;

L'alternative Sinon peut ne pas exister :

Algorithme	Algorigramme	Traduction en C
- Si condition vraie - Faire opération 1 -Fin si		if (condition) opération1 ;

Exercice :

```
int led = 13;
int data;
void setup()
{
        Serial.begin(115200);
         pinMode(led, OUTPUT);
         Serial.println("C'est parti!");
}
void loop()
{
        data=Serial.read();
         if (data=='1') digitalWrite(led, HIGH);
        if (data=='0') digitalWrite(led, LOW);
}
```

Rappel : '1' correspond au code ASCII du caractère 1 - '0' correspond au code ASCII du caractère 0

⇨ Mettre en œuvre le programme ci-dessus sur carte Arduino

Avec l'aide du site Arduino en français

1.1) Rappeler le rôle des instructions suivantes :
Serial.begin(115200); pinMode(led, OUTPUT); Serial.println("C'est parti!"); digitalWrite(led, HIGH);

1.2) Indiquer le rôle de la nouvelle instruction: **Serial.read();**

La condition == renvoie vrai s'il y a égalité.

1.3) De la même manière, indiquer la syntaxe pour les conditions suivantes :

- différent de
- inférieur à
- supérieur à
- inférieur ou égal à
- supérieur ou égal à

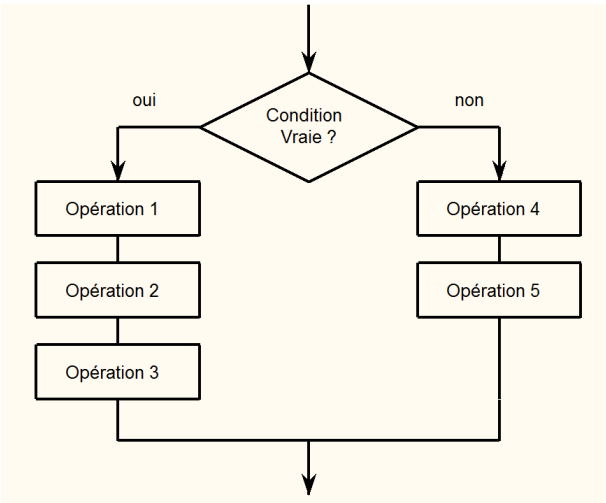
Pour envoyer une donnée (code ascii du caractère) sur le port série de la carte Arduino, il faut entrer une valeur puis cliquer « envoyer », dans la fenêtre du moniteur série

⇨ Tester le programme en envoyant le caractère 0, puis le caractère 1 et en observant la led jaune.

1.4) Décrire le comportement du programme.

Lorsqu'il y a plusieurs opérations à réaliser, on place des accolades { }

Algorithme	**Algorigramme**	**Traduction en C**

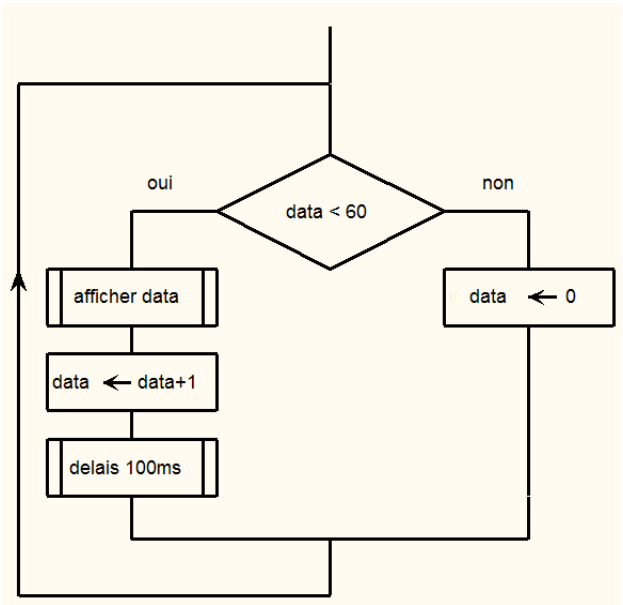
- Si condition vraie
- Faire opération 1
- Faire opération 2
- Faire opération 3
Sinon
- Faire opération 4
- Faire opération 5
-Fin si
		if (condition)
		{
		opération1 ;
		opération2 ;
		opération3 ;
		}
		else
		{
		opération4 ;
		opération5 ;
		}

⇨ Tester le programme ci-dessous

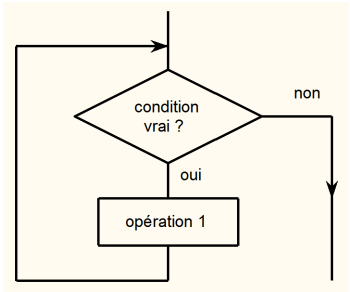
```
int led = 13;
int data;
void setup()
{
 Serial.begin(115200);
 pinMode(led, OUTPUT);
 Serial.println("C'est parti!");
}
void loop()
{
 data=Serial.read();
 if (data=='1')
  {
  digitalWrite(led, HIGH);
  Serial.println("LED ON");
  }
 if (data=='0') digitalWrite(led, LOW);
}
```

1.5) Modifier le programme pour, en plus, afficher « LED OFF » lors de l'envoie du caractère 0. Reporter la modification sur le compte rendu.

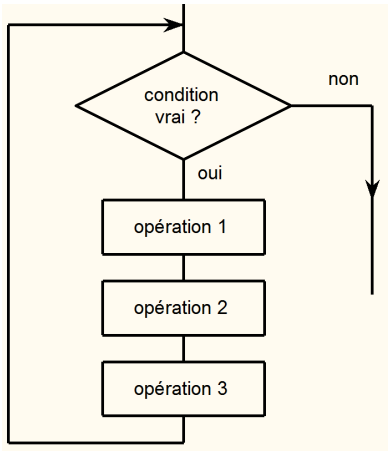
1.6) Proposer un programme qui affiche sur le moniteur série une valeur qui augmente d'une unité toutes les 100ms, et comprise entre 0 et 59 inclus. L'organigramme de la fonction loop() est donné ci-dessous :



La structure TANT QUE

Algorithme	Algorigramme	Traduction en C
- Tant que la condition est vraie - Faire opération 1 -Fin Tant que		while (condition vraie) opération1 ;

Algorithme	Algorigramme	Traduction en C

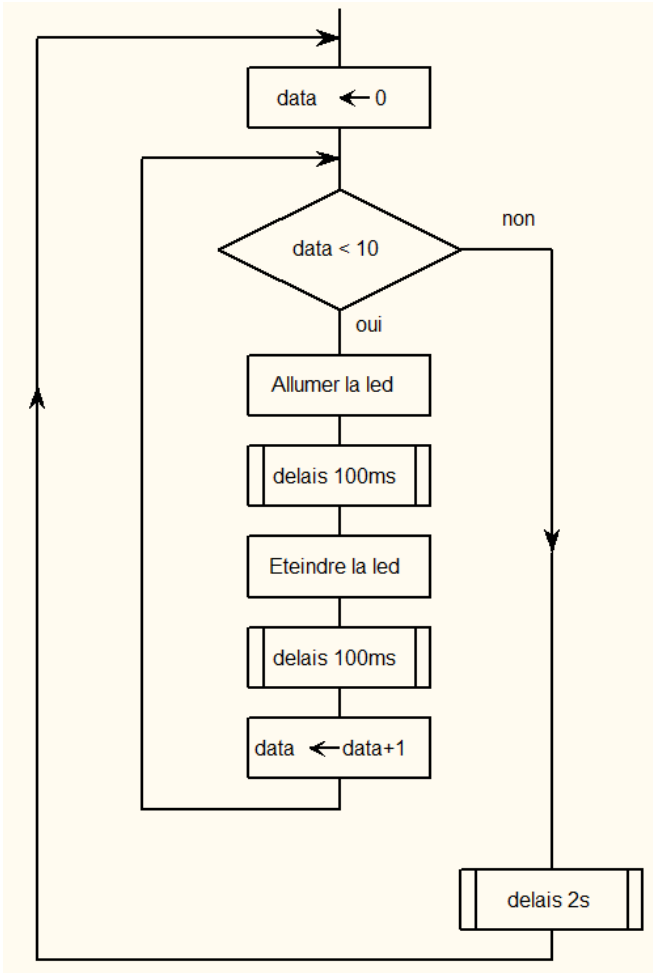
- Tant que la condition est vraie
- Faire opération 1
- Faire opération 2
- Faire opération 3
-Fin Tant que
		while (condition vraie)
		{
		opération 1 ;
		opération 2;
		opération 3 ;
		}

Exercice :

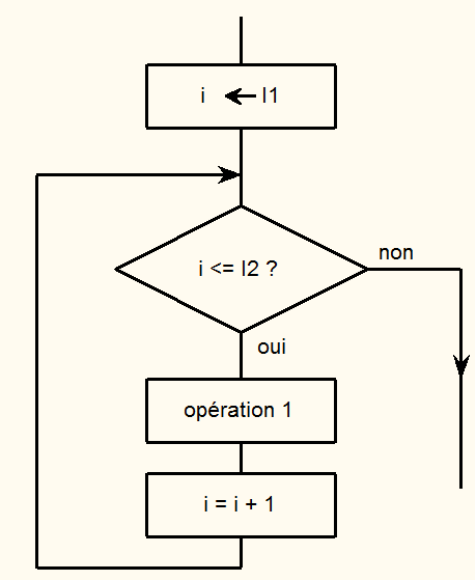
2.1) Proposer un programme sur carte ARDUINO qui réalise le fonctionnement suivant, avec obligatoirement une boucle « while » :

- la led jaune doit clignoter 10 fois toutes les 2 secondes. La fréquence de clignotement doit être de 5Hz.

L'organigramme de la fonction loop() d'ARDUINO est donné ci-dessous :



La structure FOR

Algorithme	Algorigramme	Traduction en C
- Pour i de I1 à I2 - Faire opération 1 -Fin pour		For (I = I1 ; i <= I2 ; i++) opération1 ;

Remarques: • comme précédemment, s'il y a plusieurs opérations à réaliser, on les met entre 2 accolades : { op1 ; op2 ;…}

Exemples : dans les 4 cas ci-dessous, la boucle est réalisée 7 fois

for (i=1 ; i<=7 ; i++)
{
operation1;
operation2;
operation3;
}
	for (i=0 ; i<7 ; i++)
	{
	operation1;
	operation2;
	operation3;
	}
		for (i=6 ; i<=12 ; i++)
		{
		operation1;
		operation2;
		operation3;
		}
			for (i=0 ; i<14 ; i=i+2)
			{
			operation1;
			operation2;
			operation3;
			}

3.1) Reprendre le programme de la question 2.1) en remplaçant la boucle while() par une boucle for.

La structure SWITCH … CASE

Algorithme

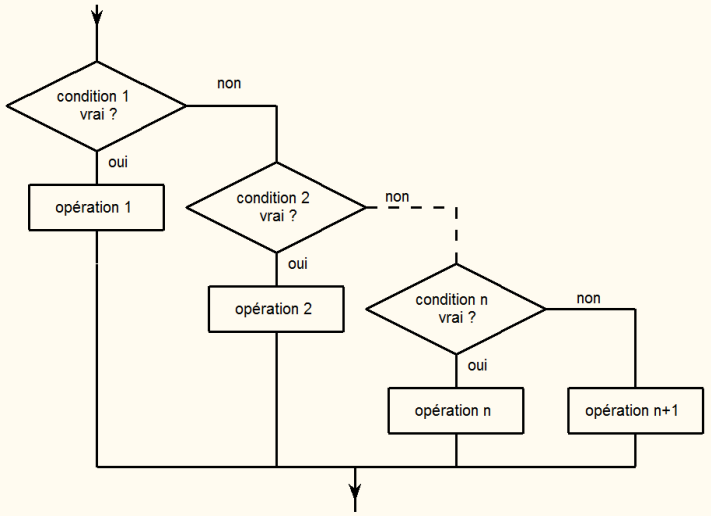
Selon cas :
- cas 1 : faire opération 1
- cas 2 : faire opération 2

……
- cas n : faire opération n

Autrement
- faire opération n+1

Fin cas

Algorigramme



Traduction en C

```
switch (variable)
{
        case 1: operation1;  break;
        case 2: operation2;  break;
        ……
        case n: operation_n;  break;
        default: operation_n+1;
}
```

Exercice:

4.1) Tester le programme ci-dessous et décrire son comportement.

```
int data;
void setup()
{
        Serial.begin(115200);
        Serial.println("C'est parti!");
}
void loop()
{
        data=Serial.read();
        switch(data)
        {
                case '1': Serial.println("one");break;
                case '2': Serial.println("two");break;
                case '3': Serial.println("three");break;
                case '4': Serial.println("four");break;
        }
}
```

4.2) Retirer une instruction break ; et tester. Décrire ce qui se passe.

4.3) Proposer un programme qui fait clignoter la led avec différente valeur de fréquence. Le délai est fixé par l'envoi d'un caractère ascii par le moniteur série. ('1'⇨100ms – '2'⇨200ms – '3 ' ⇨ 300ms – '4' ⇨ 400ms)

Une partie du programme est donné ci-dessous :

```
int led = 13;
int data;
int temps;
void setup()
{
        Serial.begin(115200);
        pinMode(led, OUTPUT);
        Serial.println("C'est parti!");
        temps=100;
}
void loop()
{
        data=Serial.read();
        switch(data)
        {



        }
        digitalWrite(led, HIGH);
        delay(temps);
        digitalWrite(led, LOW);
        delay(temps);
}
```