CPP1	Algorithme sur ESP32 – Durée 2h.

G.COLIN

Objectifs : ⇨ Mettre en œuvre des programmes de base sur carte ESP32
⇨ Faire la correspondance entre un algorigramme et langage C

Ressources disponibles :

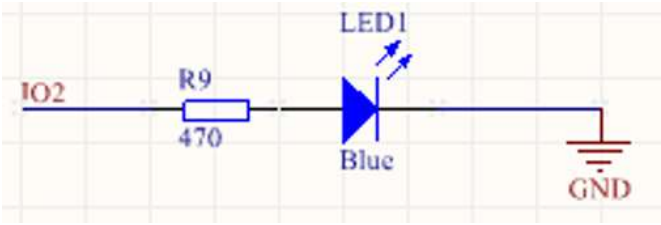
- Site Arduino
- Cours « Algorithme et Algorigramme » sur le site mede.fr/2tsse

Le développement se fait sous l'IDE Arduino. Choisir la carte NodeMCU-32S pour ce TP.

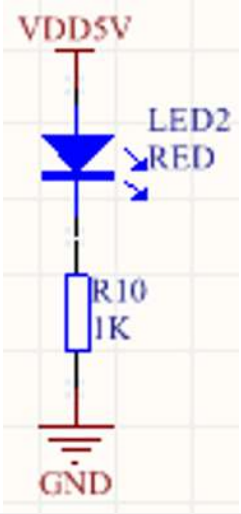
Type de carte: "NodeMCU-32S"
Upload Speed: "921600"
Flash Frequency: "80MHz"
Port: "COM13"

Les projets sont à sauvegarder dans l'espace de travail sous KWARTZ, sous répertoire Arduino_ESP32 (à créer).

La LED bleu sur carte NodeMCU-32S est câblée sur IO2.



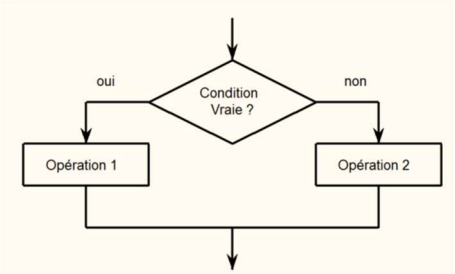
La LED rouge est câblée sur le 5V



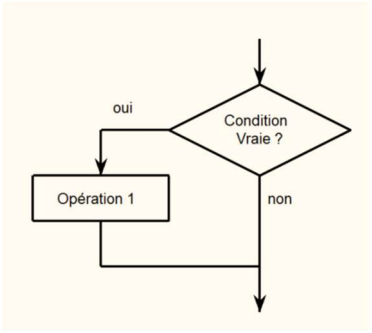
A la première utilisation de l'ESP32, il est nécessaire de suivre la procédure « Installation de la carte ESP32 dans l'IDE Arduino », donnée en annexe.

La structure IF…ELSE …

Algorithme	Algorigramme	Traduction en C

SI condition vraie
| Faire opération 1
SINON
| Faire opération 2
FIN
		if (condition)
		opération1 ;
		else
		opération2 ;

L'alternative Sinon peut ne pas exister :

Algorithme	Algorigramme	Traduction en C
SI condition vraie | Faire opération 1 FIN		if (condition) opération1 ;

Exercice :

```
#define led 2
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);
}
```

Algorithme de la boucle principale :

Lire la donnée reçue sur la liaison série du moniteur
SI donnée reçue = caractère 1
| Allumer led
FIN SI
SI donnée reçue = caractère 0
| Eteindre led
FIN SI

Remarques :

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

La fonction Serial.read() retourne l'octet reçu sur la liaison série du moniteur (UART0), elle retourne la valeur -1 si il n'y a pas de donnée reçue.

⇨ Mettre en œuvre le programme ci-dessus sur la carte ESP32

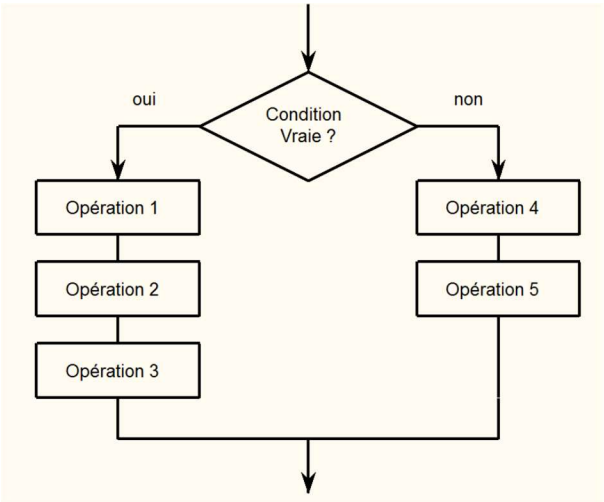
Pour envoyer une donnée (code ascii du caractère) sur le port série du moniteur, il faut entrer une valeur puis cliquer « envoyer », dans la fenêtre du moniteur série

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

⇨ 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
		if (condition) {
		opération1 ;
		opération2 ;
		opération3 ;
		}
		else {
		opération4 ;
		opération5 ;
		}

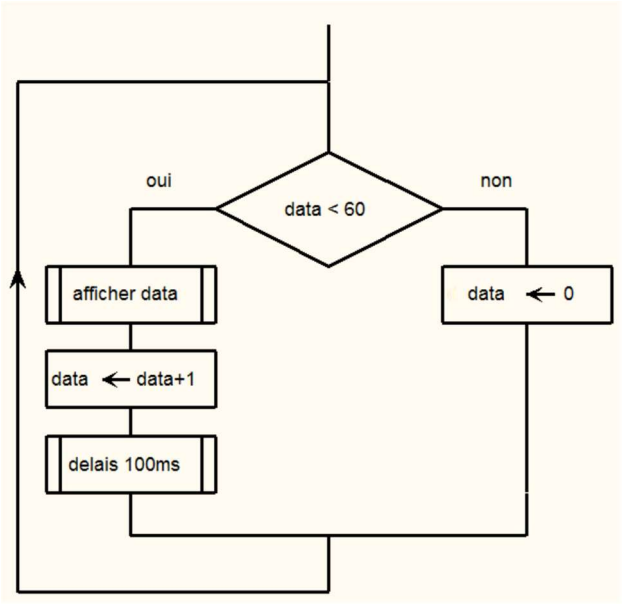
⇨ Tester le programme ci-dessous

```
#define led 2
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);
}
```

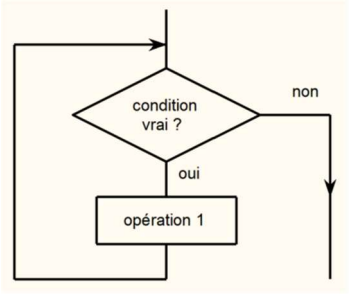
⇨ Modifier le programme pour, en plus, afficher « LED OFF » lors de l'envoie du caractère 0. Reporter la modification sur le **compte rendu.**

⇨ Tracer à main levée l'algorigramme de la boucle principale sur le compte rendu.

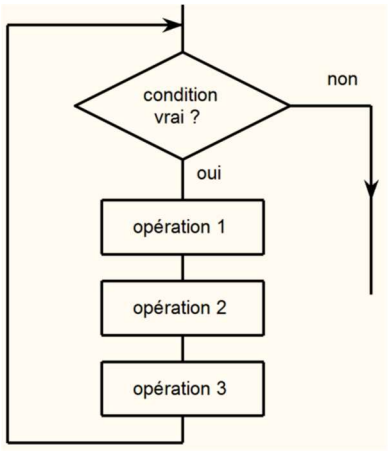
⇨ Proposer un programme qui affiche sur le moniteur série une valeur qui augmente d'une unité toutes les 100ms, et comprise entre 0 et 59 inclus. Dans cette partie, on demande d'utiliser l'instruction "IF…ELSE…".
L'organigramme de la fonction loop() est donné ci-dessous :



La structure TANT QUE

Algorithme	Algorigramme	Traduction en C
TANT QUE la condition est vraie | Faire opération 1 FIN		while (condition vraie) opération1 ;

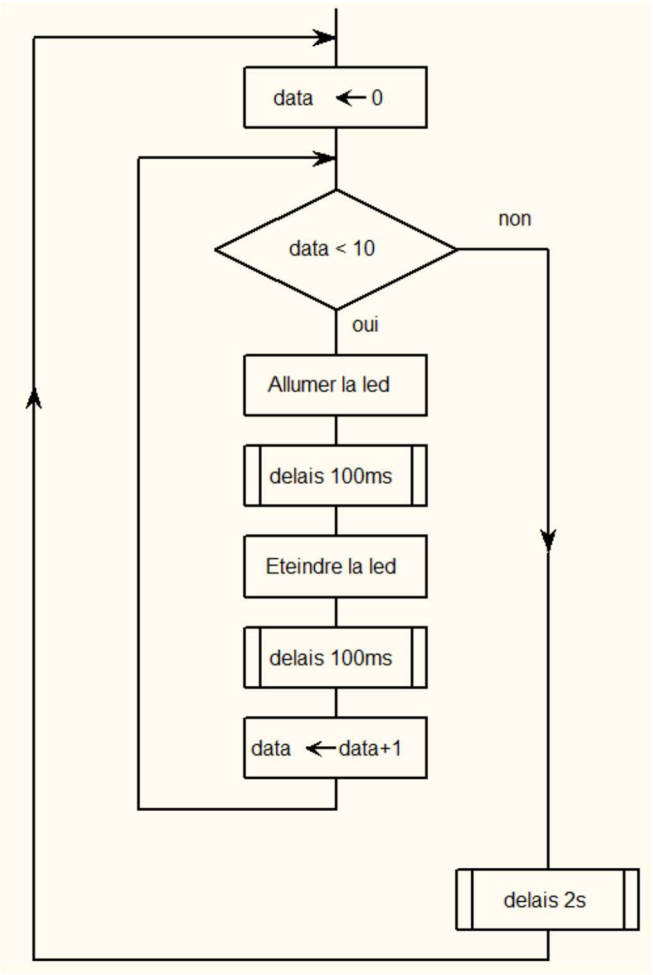
Algorithme	Algorigramme	Traduction en C

TANT QUE la condition est vraie
| Faire opération 1
| Faire opération 2
| Faire opération 3
FIN
		while (condition vraie) {
		opération 1 ;
		opération 2;
		opération 3 ;
		}

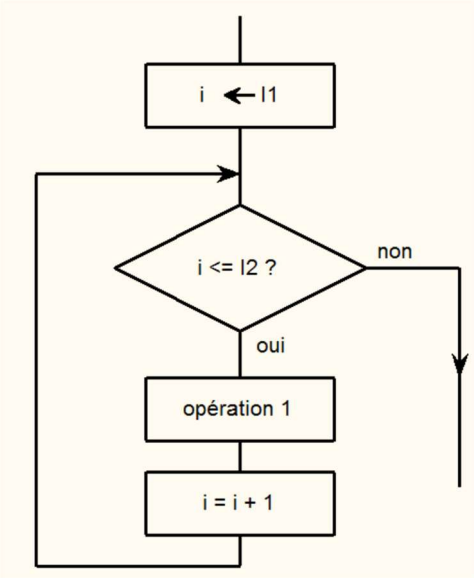
⇨ Proposer un programme sur carte ESP32 qui réalise le fonctionnement suivant, avec obligatoirement une boucle « while » :

- la led 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		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;
			}

⇨Reprendre le programme de l'exercice précédent (clignotement de la led 10 fois) en remplaçant la boucle while() par une boucle for.

La structure SWITCH … CASE

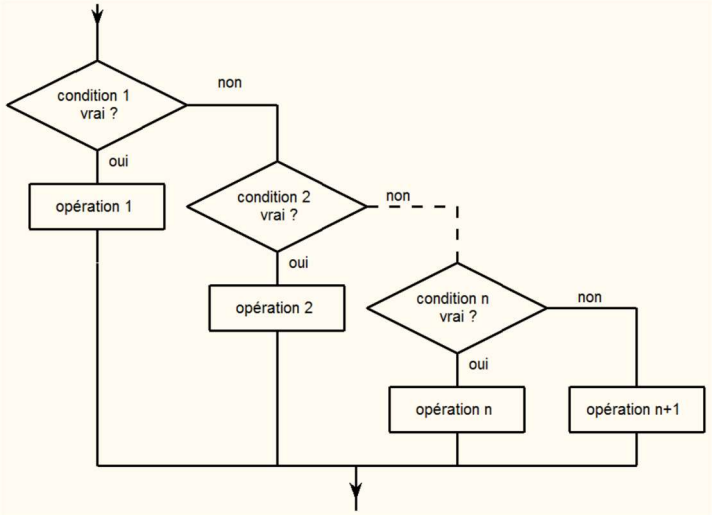
Algorithme

SELON CAS :
- cas 1 : faire opération 1
- cas 2 : faire opération 2
- ……
- cas n : faire opération n

AUTREMENT
- faire opération n+1

FIN

Algorigramme



Traduction en C

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

⇨ 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 :

```
#define led 2
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);
}
```

Modification de la variable temps

Clignotement en fonction de la valeur de temps