TSTI2D	ALGORITHME - ALGORIGRAMME Arduino suite	SIN

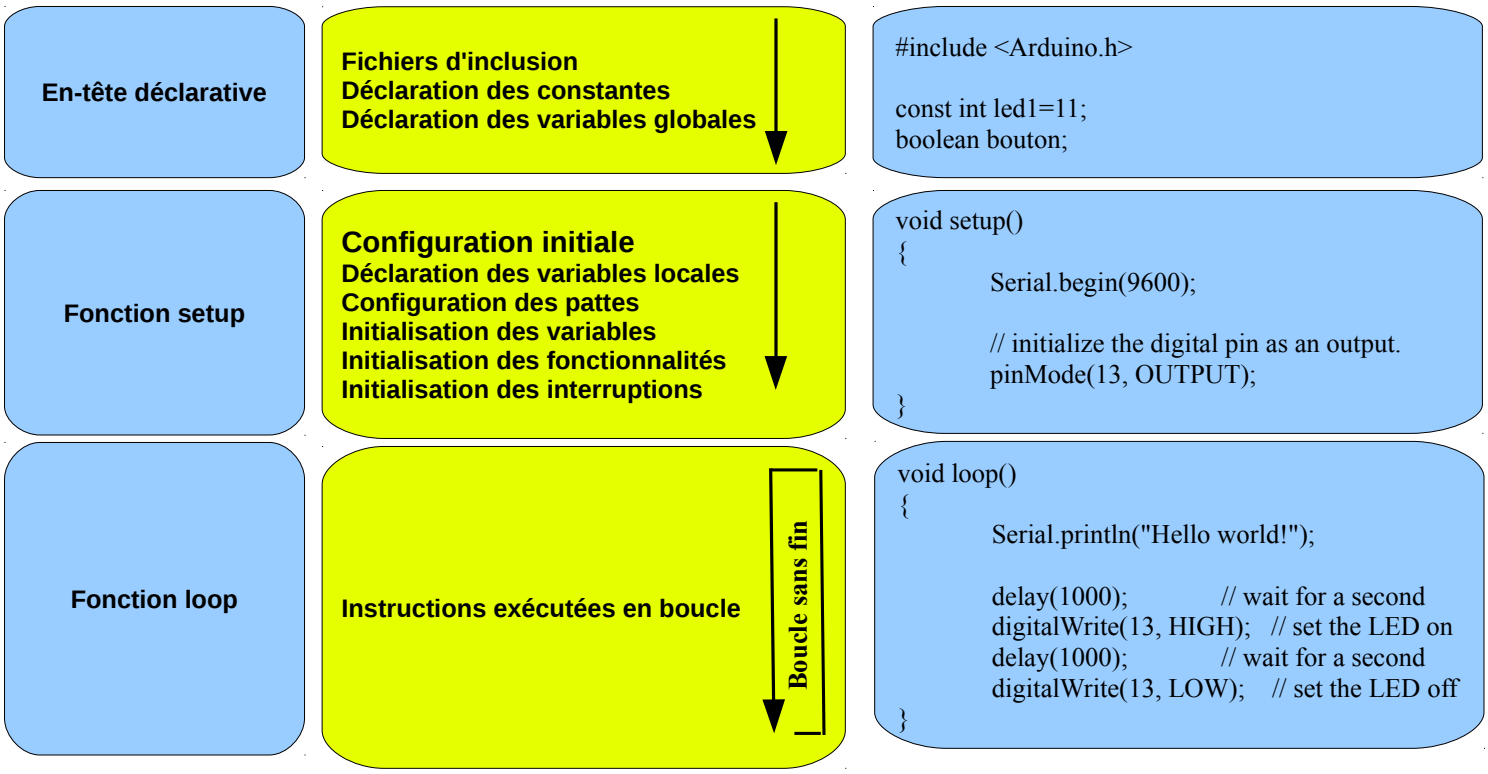
I. But

Etude pratique des structures algorithmiques avec arduino.

II. Structure des programmes dans arduino (Rappels)

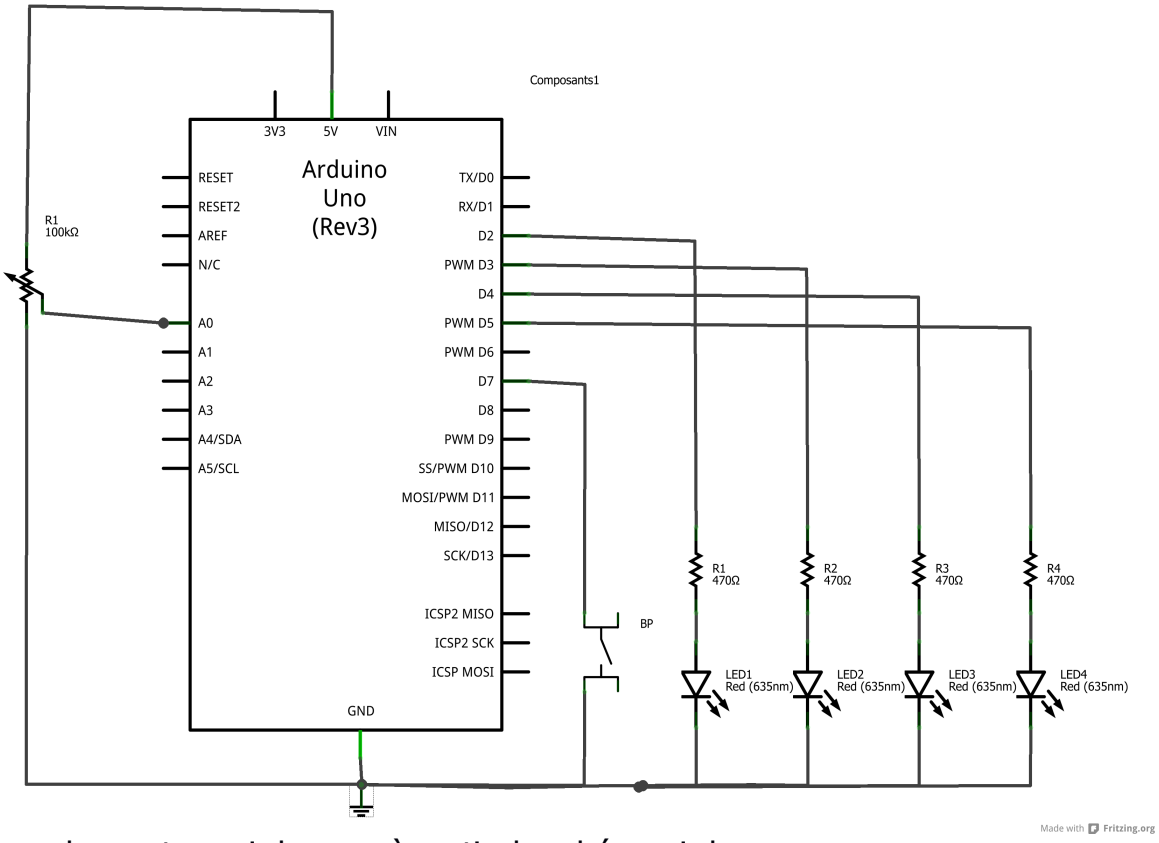
Un programme dans arduino se déroule de la manière suivante :
1. Prise en compte des instructions de la partie déclarative
2. Exécution de la partie configuration (*fonction setup( )*),
3. Exécution de la boucle sans fin (*fonction loop ( )*): le code compris dans la boucle sans fin est exécuté indéfiniment

Déroulement des programmes sur arduino

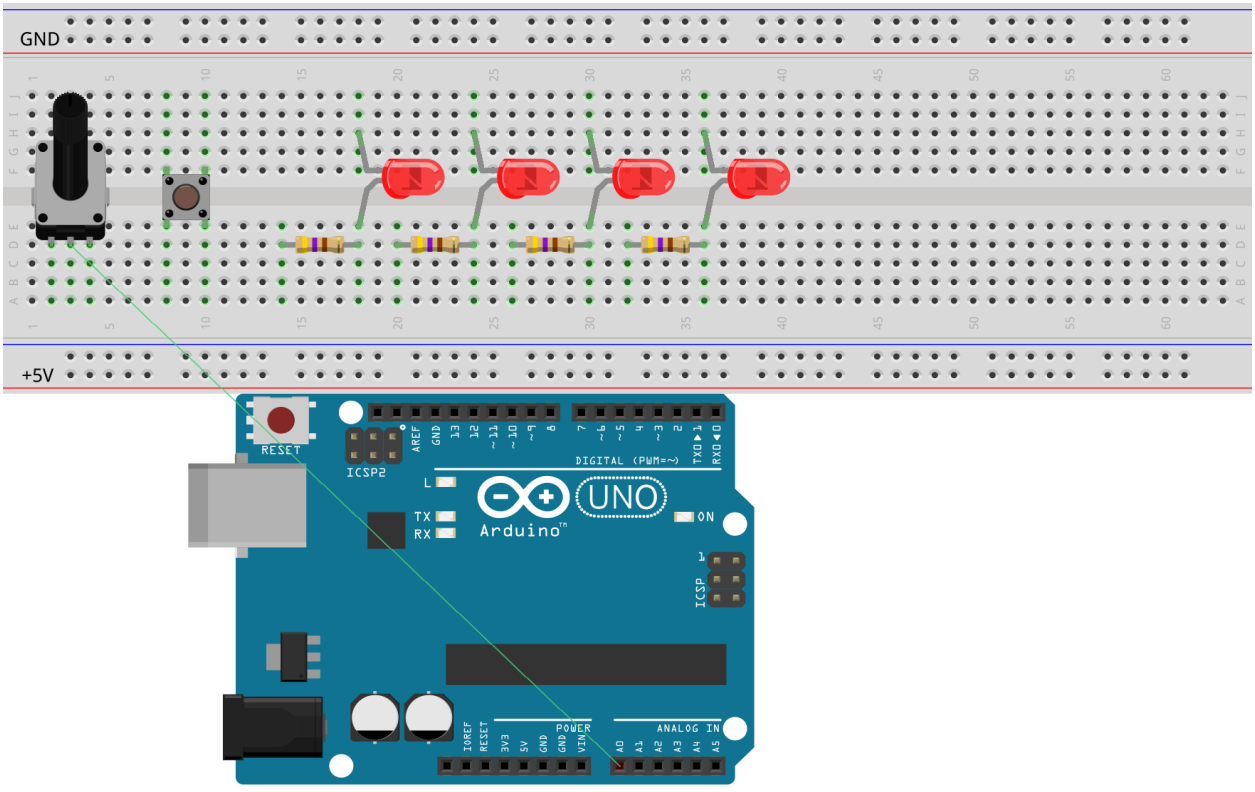


III. Montage

Pour étudier les principales structures des algorithmes, vous allez réaliser le schéma électrique suivant :



Compléter le schéma de montage ci-dessous à partir du schéma ci-dessus :

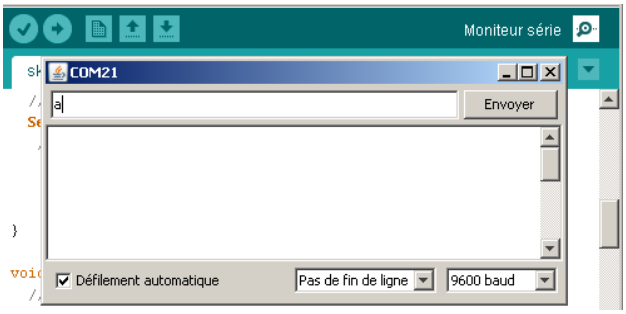


Réaliser le schéma ci-dessous.

IV. 4ème Algorithme : Structure de choix multiples

Un client souhaite commander les 4 DEL à l'aide du moniteur série du logiciel arduino.

Lorsqu'il envoie la lettre a par le moniteur série, il allume la led1 ;
b pour la led2 ;
c pour la led3 ;
b pour la led4 ;
Enfin, si il envoie une autre lettre, les 4 del s'éteignent.



Algorithme	Programme en C++

Constante led1 : entier =2 ~ déclaration de la del 1 ~
~ déclaration de la del 2 ~
~ déclaration de la del 3 ~
~ déclaration de la del 4 ~
	const int led1 = 2; //déclaration de la del 1
	//déclaration de la del 2
	//déclaration de la del 3 ~
	//déclaration de la del 4

Procédure setup()
Début
~ Configuration initiale ~
initialiser le port série de l'arduino à 9600baud
 	
initialiser led1 comme une sortie ~ del 1~
~ del 2~
~ del 3~
~ del 4~
Fin
	void setup()
	{
	//Configuration initiale:
	Serial.begin(9600); //initialise le port série
	pinMode(led1, OUTPUT); //del 1
	//del 2
	//del 3
	//del 4
	}

Procédure loop()
variable c : caractère
Début
~ exécutées en boucle ~
Si (données reçues sur port série)
Alors
c ← donnée reçue
 	
~ structure de choix multiples ~
Cas Ou c **Vaut**
'a' : allumer led1 ~ del 1~
'b' : ~ del 2~
'c' : ~ del 3~
'd' : ~ del 4~
autre : éteindre les led
FinCas ~
Fin Si
Fin
	void loop()
	{
	//exécutées en boucle
	
	// read the sensor:
	if (Serial.available() > 0)
	{
	char c = Serial.read();
	
	~ structure de choix multiples ~
	switch (c) {
	case 'a':
	digitalWrite(led1, HIGH); //led1
	break;
	case 'b':
	//led2
	break;
	case 'c':
	//led3
	break;
	case 'd':
	//led4
	break;

	default:
	// turn all the LEDs off:
	digitalWrite(led1 , LOW); //led1
	//led2
	//led3
	//led4
	} //fin switch
	} //fin si
	}

A l'aide de l'algorithme, compléter le programme en C++ de l'arduino.
Le faire vérifier par le professeur.

Tester le programme dans arduino.

Votre programme répond-il au cahier des charges :

V. 5ème Algorithme : structures *TANT QUE ... FAIRE*

Un client souhaite **faire** clignoter les 4 del **tant que** la tension aux bornes du potentiomètre est supérieure à 2,5V (512).

Remarque ; Arduino dispose en interne d'un module de « conversion analogique-numérique » qui permet d'obtenir une valeur entre 0 et 1023 correspondant au niveau de la tension entre 0 et 5V présente sur la broche. A0 à A5.

Une partie de l'algorithme est donné ci-dessous. Compléter le, pour répondre au cahier des charges.

Algorithme	**Programme en C++**

Constante led1 : entier =2 ~ déclaration de la del 1 ~
~ déclaration de la del 2 ~
~ déclaration de la del 3 ~
~ déclaration de la del 4 ~
	const int led1 = 2; //déclaration de la del 1
	//déclaration de la del 2
	//déclaration de la del 3
	//déclaration de la del 4

Variable mesure : entier ~ récupéra la mesure ~	int mesure ; //récupéra la mesure

Procédure setup()
Début
~ Configuration initiale ~
initialiser led1 comme une sortie ~ del 1~
~ del 2~
~ del 3~
~ del 4~
Fin
	void setup()
	{
	//Configuration initiale:
	//del 1
	//del 2
	//del 3
	//del 4
	}

Procédure loop()
Début
~ exécutées en boucle ~
	void loop()
	{
	//exécutées en boucle

mesure ← valeur de A0
 	
~ structure tant que faire ~
Tant Que mesure > 512
Faire
allumer led1 pendant 0,1s ~ del 1~
éteindre led1 ~ del 1~
 	
~ del 2~
~ del 2~
 	
~ del 3~
~ del 3~
 	
~ del 4~
~ del 4~
 	
FinFaire
Fin
	mesure = analogRead(A0);
	
	~ structure tant que faire ~
	while (mesure > 512)
	{
	digitalWrite(led1, HIGH); //led1
	delay(100); //led1
	digitalWrite(led1, LOW); //led1
	//led2
	//led2
	//led2
	//led3
	//led3
	//led3
	//led4
	//led4
	//led4
	} // fin while
	}

A l'aide de l'algorithme, compléter le programme en C++ pour l'arduino.
Le faire vérifier par le professeur.

Tester le programme dans arduino.

Votre programme répond-il au cahier des charges :

Remarques ;

- La boucle while (tant que faire) n'exécute les instructions de la boucle aucune fois si la condition est fausse au départ, à la différence de Do While (REPETER ... JUSQUA …) qui exécute les instructions 1 fois avant de tester la condition.

```
do      {
            actions
        }
while (condition)
```

- L'instruction while(1) réalise une boucle sans fin. Pratique si l'on veut stopper par exemple la boucle loop() après une seule exécution.

VI. 6ème Algorithme : structures POUR Indice ALLANT DE ... A .... FAIRE ...

Le client souhaite alimenter 4 DEL alternativement en utilisant que des boucles for :

```
// boucle incrémentant la variable i de 0 à 255, de 1 en 1
for (int i=0; i <= 255; i++){
    actions
}
```

Ecrire l'algorithme pour répondre au cahier des charges.

Algorithme	Programme en C++

Procédure setup()
Début
~ Configuration initiale ~
~ initialiser les bornes 2,3,4 et 5 en sortie digitale ~
~ structure Pour Indice Allant de ... à .... faire ~
Pour i **allant de 2 à 5**
Faire
initialiser la borne **i** en sortie digitale
FinFaire
Fin
 	
Procédure loop()
Début
~ exécutées en boucle ~
 	
~ structure Pour Indice Allant de ... à .... faire ~
Pour i **allant de 2 à 5**
Faire
mettre au niveau bas les bornes 2,3,4 et 5
mettre au niveau haut la borne **i**
pendant 0,3s
FinFaire
Fin
	void setup()
	{
	// initialize the LED pins:
	
	for (int i = 2; i < 6; i++) {
	pinMode(i , OUTPUT);
	
	} // fin for
	}
	
	void loop() {
	~ exécutées en boucle ~
	
	~ structure **Pour Indice Allant de ... à .... faire ~**
	
	}

A l'aide de l'algorithme, écrire le programme en C++.
Tester le programme dans arduino.
Le faire vérifier par le professeur.

Votre programme répond-il au cahier des charges?

VII. Conclusion