



# SIN

## SYNTHESE

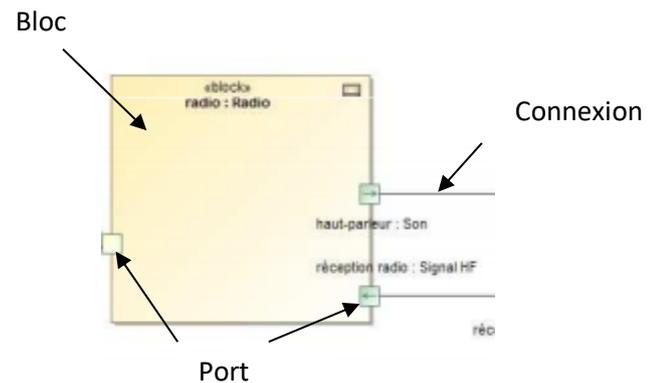


### Structure de programme Arduino

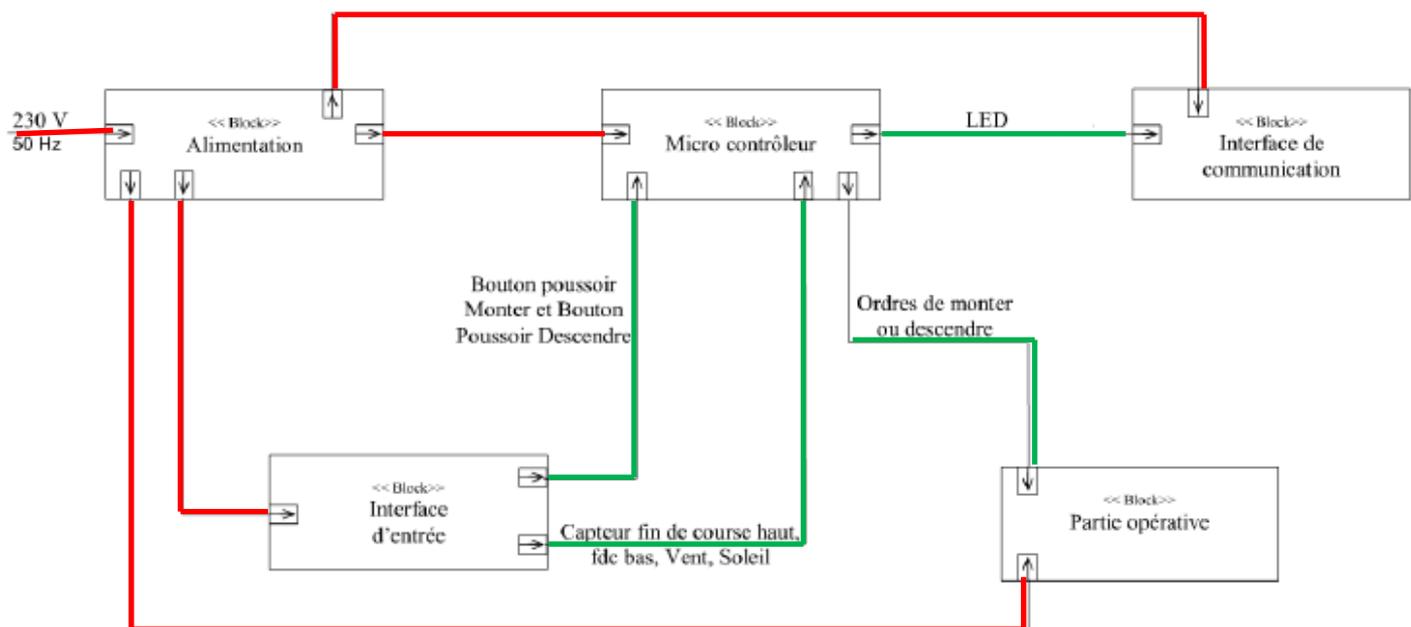
#### 1 Diagramme de bloc interne IBD du Store

Le diagramme de bloc interne IBD décrit la structure interne du système. Il permet de représenter les différentes parties du système, les ports, les connexions et les échanges entre les différentes parties du système.

- Repasser en **rouge** les flux d'énergie
- Repasser en **vert** les flux d'information



IBD Store :



#### 2 Programmation

Il existe différents types de langages de programmation des microcontrôleurs :

- Le langage **procédural** : **c**,...
- Les langages **orienté objet** : **Java**, **C#**...

Les cartes ARDUINO utilisent un **langage procédural et orienté objet en fonction de l'utilisation dérivé du C/C++**

### 3 Structure des programmes Arduino

Les programmes Arduino sont basés sur le langage C/C++ et peuvent être divisés en trois parties principales :

#### 1. **Déclaration des variables et constantes du programme**

Les **variables** sont des espaces mémoire que vous pouvez utiliser dans les programmes pour stocker des valeurs.

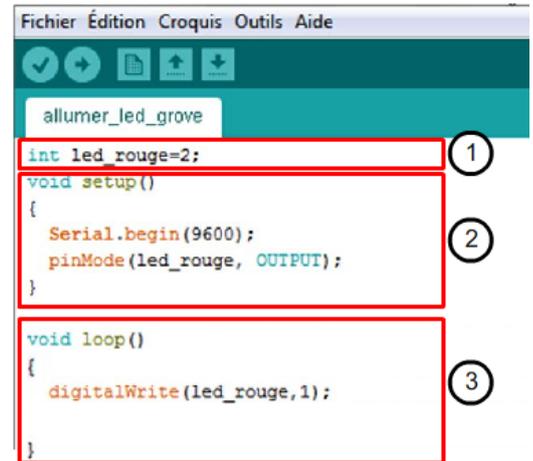
Les **constantes** sont des valeurs particulières ayant une signification spécifique.

#### 2. **La fonction d'initialisation de la carte : void setup()**

La fonction **setup()** est appelée une seule fois au démarrage du programme. Cette fonction est utilisée pour initialiser les variables, le sens des broches, les bibliothèques utilisées.

#### 3. **La fonction principale qui se répète à l'infini : void loop()**

La fonction **loop ()** (boucle en anglais) s'exécute en boucle sans fin, permettant à votre programme de fonctionner.



```
Fichier Édition Croquis Outils Aide
allumer_led_grove
int led_rouge=2;
void setup()
{
  Serial.begin(9600);
  pinMode(led_rouge, OUTPUT);
}
void loop()
{
  digitalWrite(led_rouge, 1);
}
```

### 4 Algorithmique

#### 4.1 Structure alternative (si alors sinon)

Elle permet de : **faire un choix**

Sa particularité est : **Elle a deux formes, si alors ou si alors sinon**

Sous Arduino, on utilise les mots clés : **if (condition) {...instructions; } else{...instructions;...}**

#### 4.2 Structures itératives Tant que ...faire

Elle permet de : **faire en boucle des opérations.**

Sa particularité est : **0...n elle peut ne jamais s'exécuter si la condition n'est jamais validée.**

Sous Arduino, on utilise les mots clés : **while (condition) { instructions;...}**

#### 4.3 Structures itératives Répéter...jusqu'à, Faire...tant que

Elle permet de : **faire en boucle des opérations**

Sa particularité est : **1...N elle s'exécute au moins 1 fois.**

Sous Arduino, on utilise les mots clés : **do {... instructions;...} while (condition);**

#### 4.4 Structure itérative Répéter n fois (pour faire)

Elle permet de : **faire en boucle des opérations**

Sa particularité est : **N est connu.**

Sous Arduino, on utilise les mots clés : **for (initialisation; condition; incrément) {... instructions;...}**

## 4.5 Exemple

### Cahier des charges :

Tant qu'un interrupteur marche est activé, une pompe se met en route en fonction d'un capteur.

