

Connected Flowers





Sommaire :

- Problématique
- Algorithme et programmes
- Lien entre les capteurs et l'interface
- Problèmes rencontrés
- Améliorations futur
- Conclusion



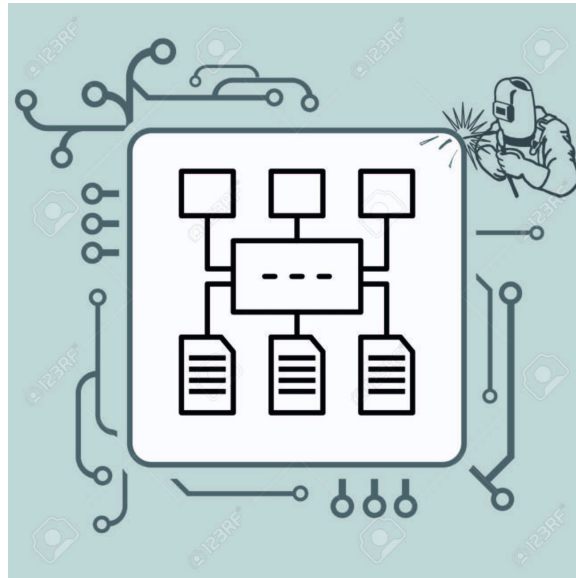
PROBLÉMATIQUE



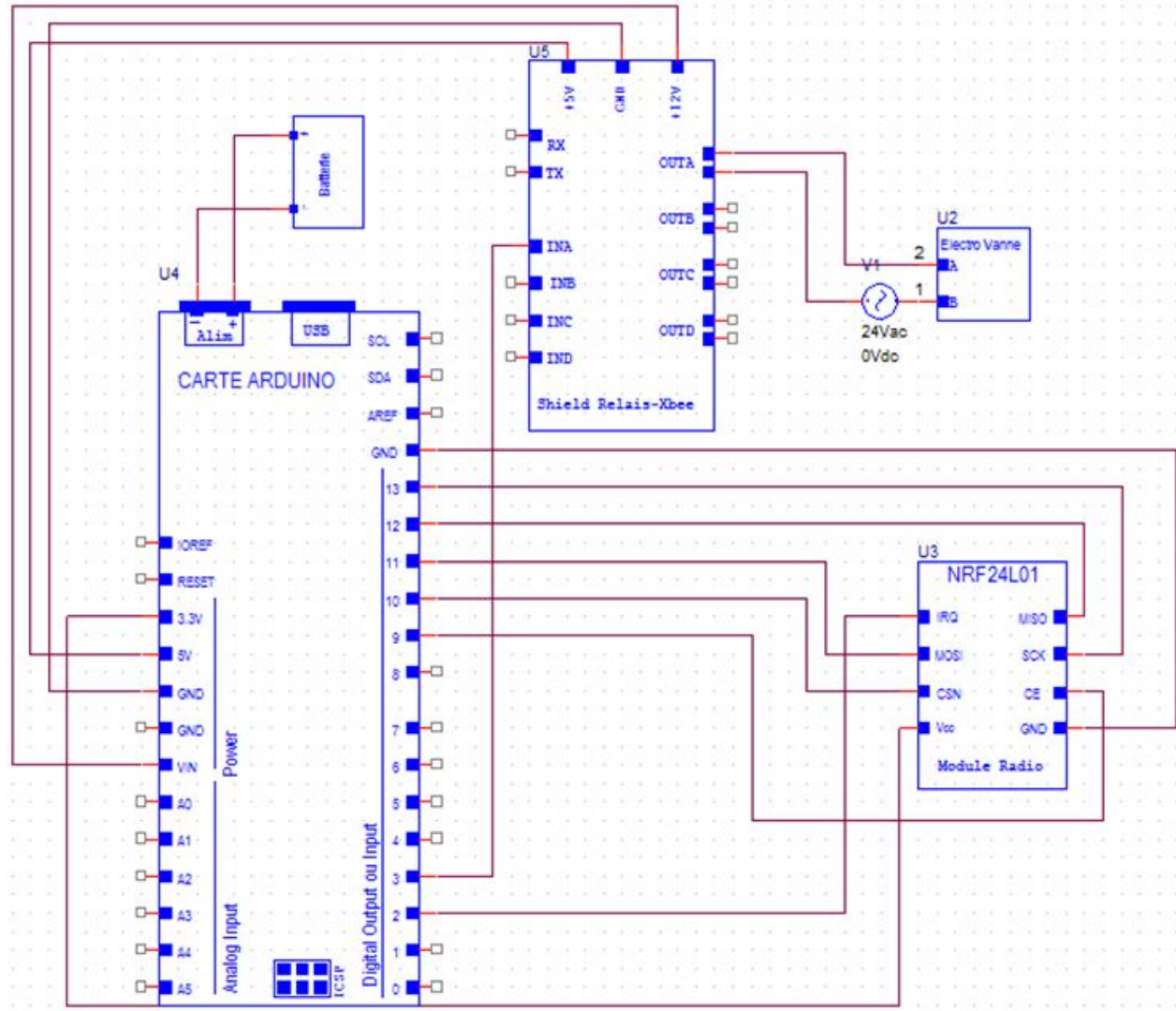


Notre but ici est de fournir un outil connecté utilisant une base de données de plusieurs plantes, permettant à des personnes non expérimentés de pouvoir faire pousser des plantes chez soi facilement.

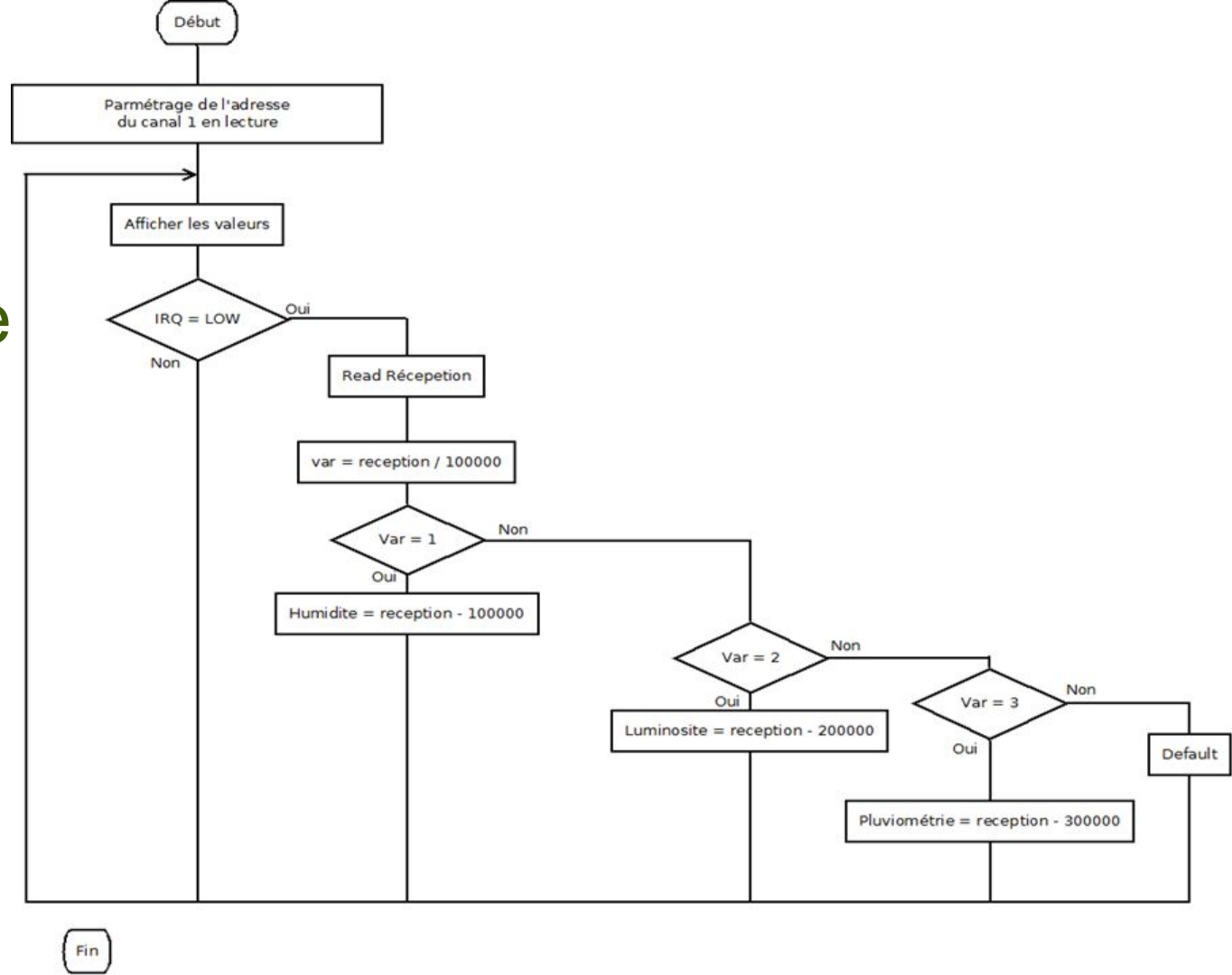
ALGORITHME ET PROGRAMMES



Câblage



Algorithme du NRF





Programme du NRF

```
#include <SPI.h> // librairie pour contrôler le bus SPI
#include <RF24.h> // librairie pour contrôler le module RF24
#define PIN_CS 10 // attribution de la broche 10 pour PIN_CS
#define PIN_CE 9 // attribution de la broche 9 pour PIN_CE
RF24 radio (PIN_CE, PIN_CS);

int pinIrqNrf24 = 2; //IRQ sur la broche 2 lance l'interruption 0

// inclusion des données et mise a 55 pour savoir si il y a une erreur
unsigned long int luminosite = 55;
unsigned long int humidite = 55;
unsigned long int pluvio = 55;
// mise en place de l'adresse utilisée
const uint64_t addrCanal = 0xF0F0F0F0C6;
//définition de la variable newValeur en false
boolean newValeur = false;

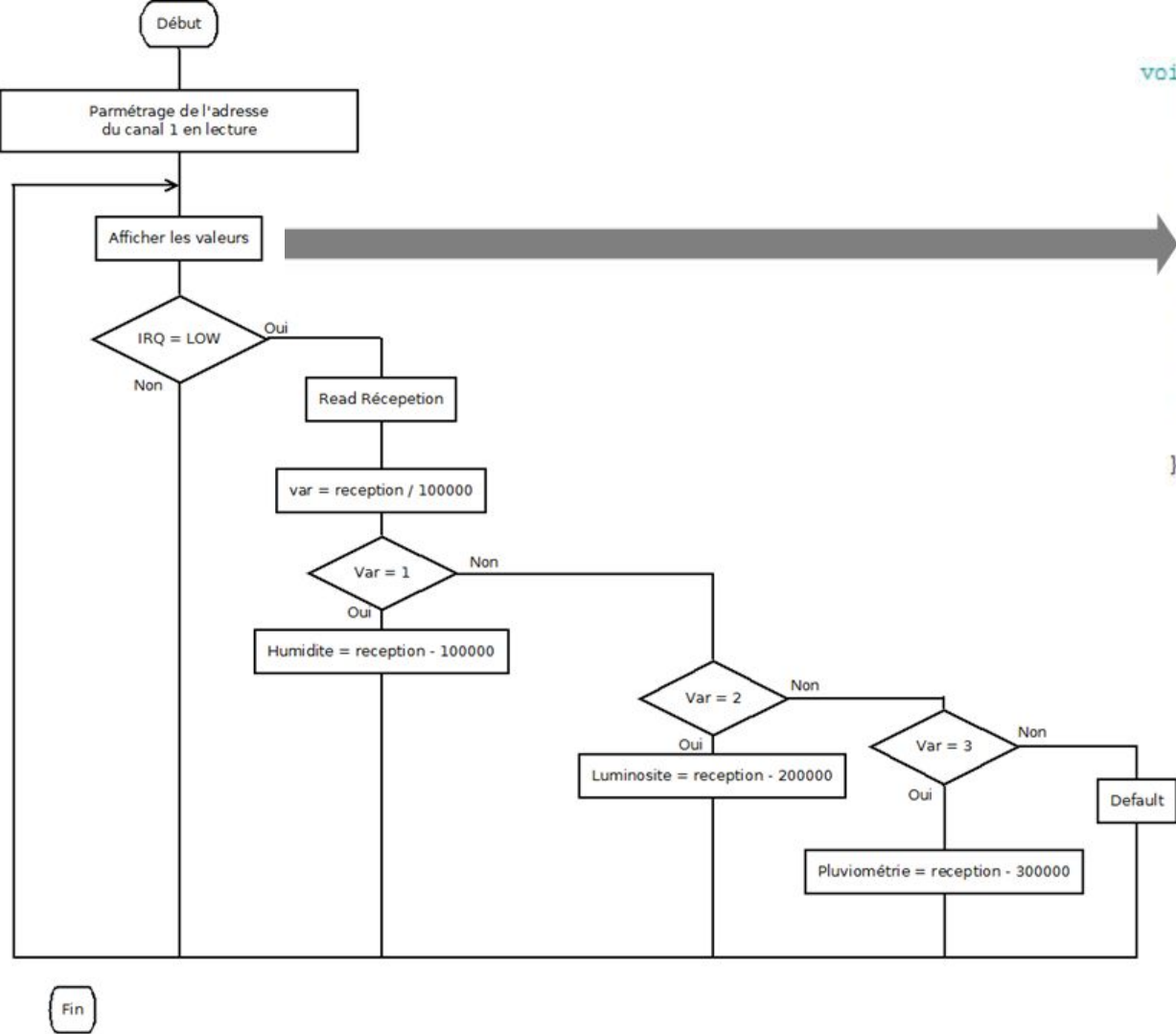
void setup() {
  Serial.begin(9600);
  // début du système radio
  radio.begin();

  radio.setPALevel(RF24_PA_LOW); //RF24_PA_HIGH consomme trop de courant

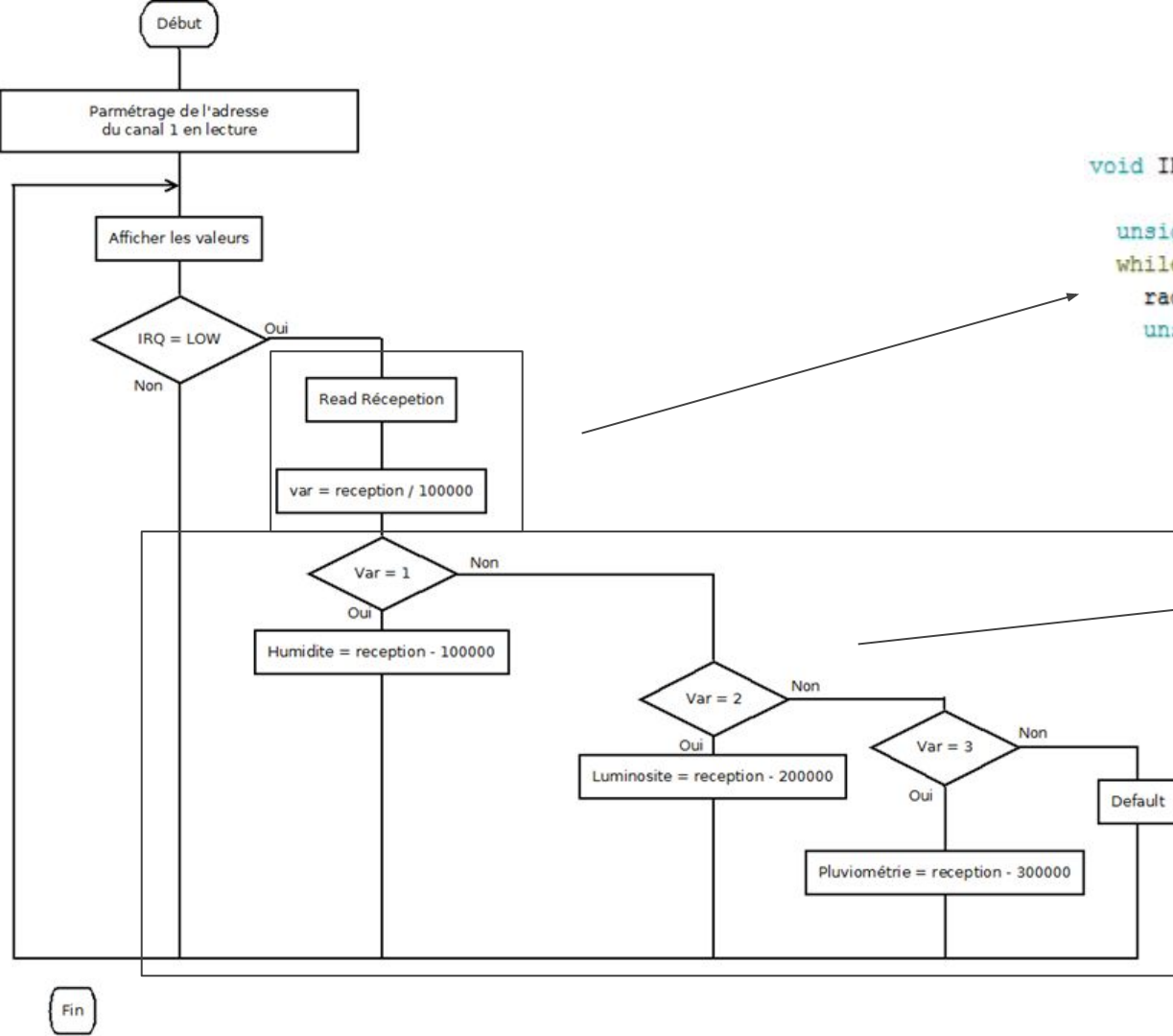
  radio.openReadingPipe(1, addrCanal); //définition de l'adresse canal 1 en lecture

  //Si la broche IRQ et en état bas nous utilisons le void irqRecep
  attachInterrupt(digitalPinToInterrupt(pinIrqNrf24), irqRecep, LOW);

  radio.startListening(); //En lecture
}
```

```
void loop() {  
    if(newValeur == true){  
        // affichage des données  
        Serial.print(F("Humidité : "));  
        Serial.println(humidite);  
        Serial.print(F("Luminosité : "));  
        Serial.println(luminosite);  
        Serial.print(F("Pluviométrie : "));  
        Serial.println(pluvio);  
        Serial.println( );  
        newValeur = false;  
    }  
}
```



```
void IRQ() {
```

```
    unsigned long int R;
    while(radio.available()){
        radio.read(&R, sizeof(unsigned long int));
        unsigned long int var = R / 100000;
```

```
    // début du switch et mise en place des 3 cases
```

```
    switch (var) {
        case 1:
            humidite = R - 100000;
```

```
        break;
```

```
        case 2:
            luminosite = R - 200000;
```

```
        break;
```

```
        case 3:
            pluvio = R - 300000;
        break;
```

```
    default:
        break;
```

Fin



LIEN ENTRE CAPTEURS ET INTERFACE



▼ ⚙ planteco usr
📄 nom : varchar(255)
📄 modepasse : varchar(255)
🔑 usr_id : int(11)

▼ ⚙ planteco usrplante
plante_id : int(11)
🔑 usrplante_id : int(11) *
usr_id : int(11)

▼ ⚙ planteco plante
📄 nom : varchar(255)
📄 categorie : varchar(255)
📄 description : varchar(255)
📄 photo : varchar(255)
minhumidite : int(11)
mintemperature : int(11)
luminosite : int(11)
📄 floraison : varchar(255)
🔑 plante_id : int(11)
maxtemperature : int(11)
maxhumidite : int(11)

▼ ⚙ planteco info
luminosite : int(11)
temperature : int(11)
humidite : int(11)




PROBLEME RENCONTRES





AMÉLIORATIONS FUTURE





```
const int out1 = 5;
const int out2 = 6;
const int out3 = 7;
const int out4 = 8;
```

```
void setup() {
  //initialisation des broches pour les relays
  pinMode(out1, OUTPUT);
  pinMode(out2, OUTPUT);
  pinMode(out3, OUTPUT);
  pinMode(out4, OUTPUT);
  Serial.begin(9600);
}
```

Déclarations des variables

Définitions des broches


```

const int out1 = 5;
const int out2 = 6;
const int out3 = 7;
const int out4 = 8;

void setup() {
  // put your setup code here, to run once:
  pinMode(out1, OUTPUT);
  pinMode(out2, OUTPUT);
  pinMode(out3, OUTPUT);
  pinMode(out4, OUTPUT);
  Serial.begin(9600);
}

void setRelay(int relay, int value)
{
  if(relay > 0 && relay < 5) digitalWrite ((relay+4),value);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("Relay 1: Enabled ... ");
  delay(1000);
  setRelay(1, 1);
  Serial.println("La vanne est ouverte");
  delay(1000);
  setRelay(1, 0);
  Serial.println("La vanne est fermée");
  Serial.println(" ");
}

```

Programmes relais
et vanne

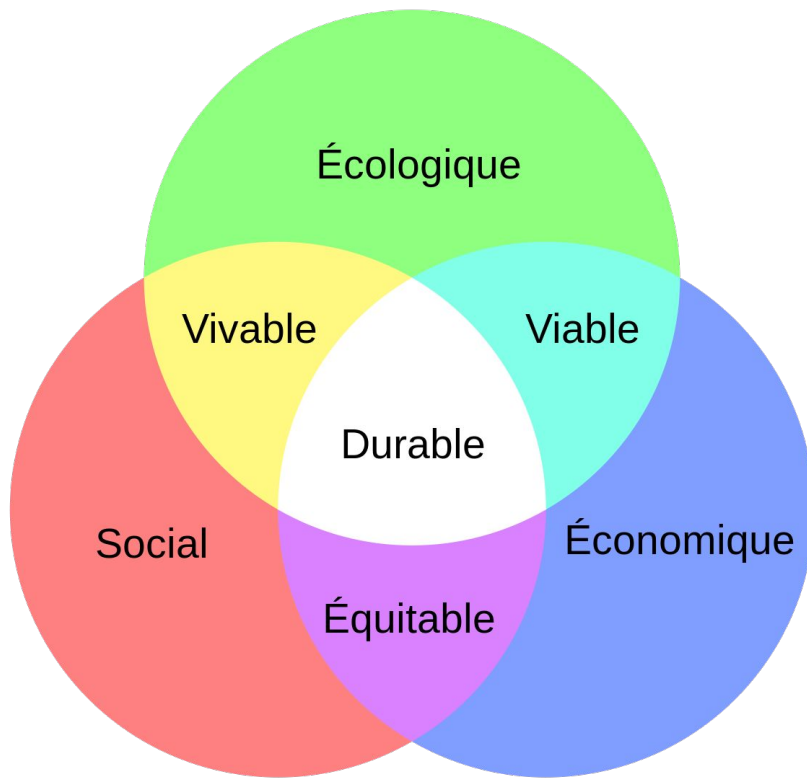


CONCLUSION





LE DÉVELOPPEMENT DURABLE





MERCI DE VOTRE ÉCOUTE

