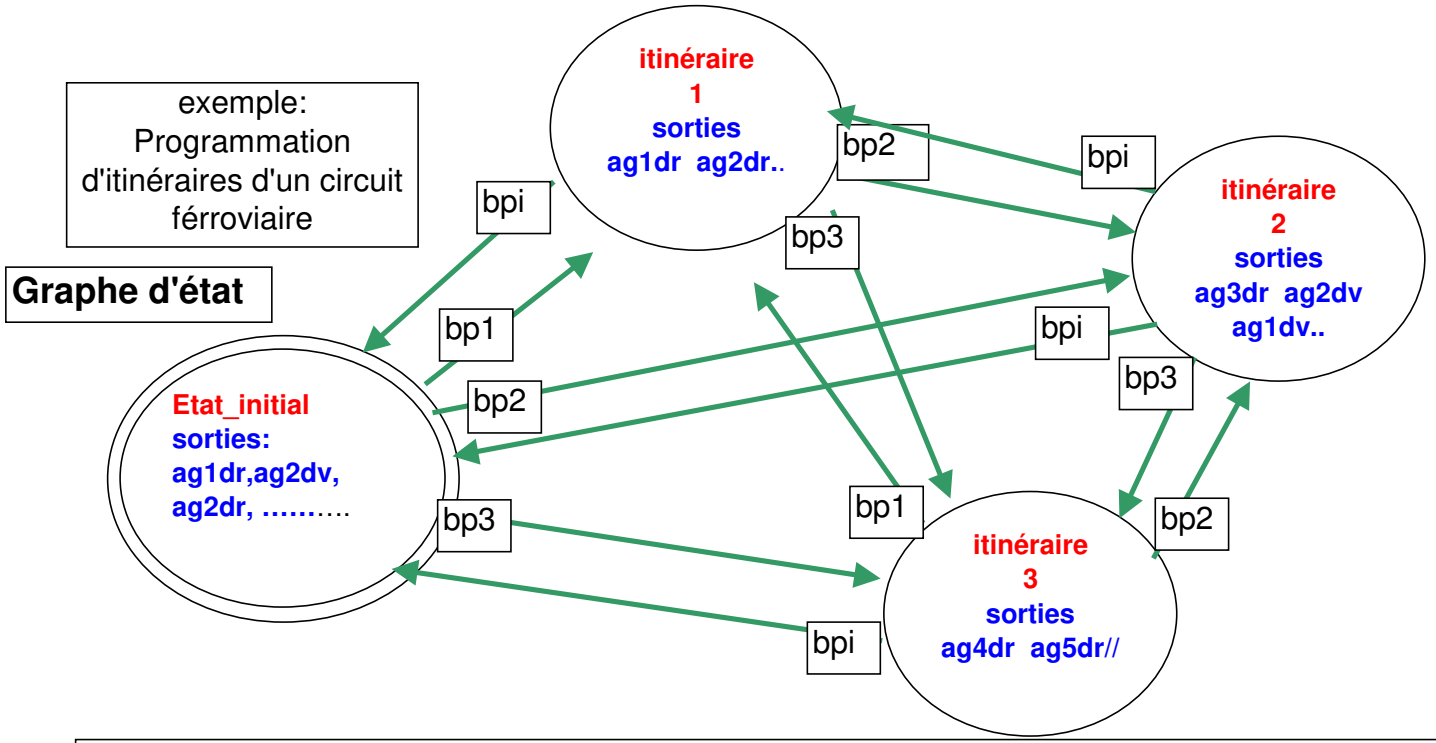
programmation par la méthode des machines d'état

d'après : https://www.youtube.com/watch?v=Ar1CYfRxs38&t=2st

Cette méthode permet de passer du dessin au langage de programmation .
Elle se décompose en 4 étapes
1/ énumérer les états (dans l'exemple ci dessous etat_intitial ,itineraire1,itineraire2,itrtinéraire3)
2/ programmer un sous programme par etat dans lequel on configure **les sorties**
3/ programmer un **switch case** (dans **loop**) qui permet de se positionner dans un sous programme
4/ programmer les **transitions** (passage d'un état vers un autre (if then dans chaque sous programme)

exemple:
Programmation
d'itinéraires d'un circuit
férroviaire

Graphe d'état

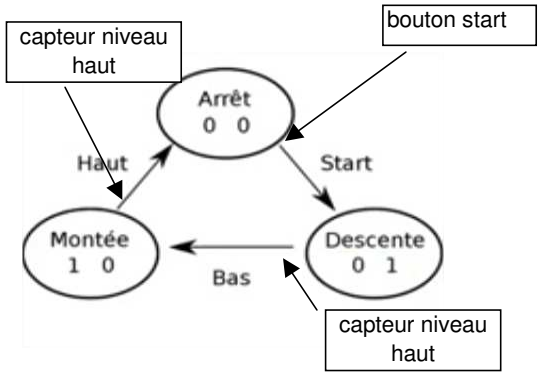


ce programme se composera de:
4 sous programmes: (Etat_intitial, itineraire1,, itineraire2, ,itineraire3)
3 tests **if then** dans chaque sous programme (exemple si bp2 est pressé executer itineraire 2)

Dans l'etat_initial on positionne tous les aiguillages
dans chaque etat on positionne les aiguillages qui correspondent à l' état souhaité
dans chaque etat des " **if then**" attendent que l'on presse un bouton poussoir pour passer à un autre état (cela pourrait être un délais, un évènement extérieur (capteur ..)

programmation de l'exemple https://www.youtube.com/watch?v=Ar1CYfRxs38&t=2st

Graphe d'état



0 0 , 01 , 1 0 : correspond à la commande de 2 relais
0 0 moteur à l'arret
01 moteur sens descente
10 moteur sens montée

```
 // j'ai adapté ce programm d'une vidéo qui explique la machine d'état

//https://www.youtube.com/watch?v=Ar1CYfRxs38&t=2st

 #define bpstart   3        // bouton de mise en route
 #define detecteurbas   4   //swtich niveau bas
 #define detecteurhaut   5  //swtich niveau haut
 // le moteur est comandé par 2 relais
 //ce qui permet d'inverser le sens (principe du pont en H )
 #define relaisM 11  //commande moteur
 #define relaisD 12  //commande moteur
              //relaisM=0,relaisD=0  moteur arret
             //relaisM=1,relaisD=0  moteur monte
              //relaisM=0,relaisD=1  moteur descend
```

```
enum {arret,descente, montee}; //il exite 3 etats qui corresponde à 1 arret,2à descendre ,3 à montee
                               // 1 correspond à arret ,2 à descente ,3 à montee

int etat = arret;          //la variable état doit donc  etre un nombre entier

//on teste si le bpstart est à appuyé
//(le bp est connecté entre l'arduino et la masse
//!digitalRead(bpstart) le ! signifie inverse
//on lit le bpstart si il est appyué start()= 0
//start() prendra la valeur 0  ou 1

int start()  {return !digitalRead(bpstart);}

// même principe que start()
int bas() {return !digitalRead(detecteurbas);
}

//même principe que start()
int haut() {return !digitalRead(detecteurhaut);}

void setup() { // on definit toutes  les entrées et les sorties
 pinMode  (bpstart , INPUT_PULLUP);
 pinMode  (detecteurbas, INPUT_PULLUP);
 pinMode  (detecteurhaut , INPUT_PULLUP);
 pinMode  (relaisM , OUTPUT);
 pinMode  (relaisD , OUTPUT);
 Serial.begin(9600); //affichage écran
 //on initialise l'état de depart
 etat = arret;
} // fin du setup

//programme principal
void loop() {

 switch (etat)
 {
  case arret:
  arreter();

    //configuration des sortie de l'eta arret
    //digitalWrite (relaisM, LOW);
    //digitalWrite (relaisD, LOW);
    //test des transitions ( bp ou des detecteurs)
    // port sortir de l'état arret

    if (start()) // si l'etat start est actif le moteur descent
```

```
    if (start()) // si l'etat start est actif le moteur descent

     {
      etat= descente;
      }
      break;       //on retourne au debut du switch

   case descente:
   descendre();
     if (bas())
    {
      etat = montee;
    }
   break;

   case montee:

    monter(); //configure les relais
     if (haut())
     {
       etat = arret;
     }
     break;
     }
  } // fin du loop
// sous programmes
 void monter()
 {
 Serial.print ("cycle  monter");
    Serial.print ("    ");
digitalWrite (relaisM,HIGH);
digitalWrite (relaisD, LOW);
 }

void descendre()
 {
   Serial.print ("cycle descendre");
  Serial.print("    ");
 digitalWrite (relaisM, LOW);
 digitalWrite (relaisD, HIGH);
 }
  void arreter()
 {
  Serial.print ("cycle arreter");
    Serial.print ("    ");
 digitalWrite (relaisM, LOW);
 digitalWrite (relaisD, LOW);
 }
```