

```
/* test d'un automate pour la gestion de la sortie d'un canton
avec une aiguille abordée par la pointe face à des cantons
soit occupés soit libres.
Francis Couvelaère, le 22 juin 2021, programme libre de droits
*/
#include <Wire.h>
#include<splash.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 4
Adafruit_SSD1306 display( OLED_RESET );

// Les constantes de l'état des cantons

// déclaration de la constante d'occupation du canton 13
const int canton_13 = 7; // fil vert arrivant optocoupleur cantons

// déclaration de la constante d'occupation du canton 9
const int canton_9 = 6; // fil bleu arrivant optocoupleur cantons

// déclaration de la constante d'occupation du canton 5
const int canton_5 = 5; // fil violet arrivant optocoupleur cantons

// Les variables de l'état des cantons

// déclaration de la variable d'état du canton 13
int occupation13 = 0 ;

// déclaration de la variable d'état du canton 9
int occupation9 = 0 ;

// déclaration de la variable d'état du canton 5
int occupation5 = 0 ;

// Les constantes de l'état de l'aiguillage

const int aiguillage_droit = 8; // fil jaune arrivant optocoupleur aiguillages
const int aiguillage_devie = 9; // fil orange arrivant optocoupleur

// Les variables de l'état de l'aiguillage

int aiguillage1droit = 1 ;
int aiguillage1devie = 0 ;
```

```

// La constante de l'état du feu en sortie du canton 13
const int feu_vert = 4; // fil gris au 5v si feu vert 0v si feu rouge

// La variable de l'état du feu en sortie du canton 13

int feuvert = 0 ;

// création du tableau pour les 5 valeurs nécessaires à l'automate du canton 13
int AUTOMATE13 [5];

// création du message au TCO
int message = 0;

void setup()
{
  // déclaration des broches sur l'Arduino Nano
  pinMode(canton_13, INPUT);
  pinMode(aiguillage_droit, INPUT);
  pinMode(aiguillage_devie, INPUT);
  pinMode(feux_vert, OUTPUT);

  // Initialisation de la communication I2C à l'adresse 0x3C vers l'écran OLED 0,96 .
  display.begin( SSD1306_SWITCHCAPVCC, 0x3C );
  display.clearDisplay();

  // message au TCO pour le démarrage avant le relevé des états sur le circuit

  display.clearDisplay();
  display.setTextSize( 1 );
  display.setTextColor( WHITE );
  display.setCursor( 6, 3 );
  display.println( "attente donnees" );
  display.drawRect( 0, 0, display.width() - 2, display.height() - 2, WHITE );
  display.display();
}

void loop()
{
  // on relève tous les états des variables et on remplit le tableau AUTOMATE 13
  AUTOMATE13[0] = digitalRead(7); // état canton 13
  AUTOMATE13[1] = digitalRead(6); // état canton 9
  AUTOMATE13[2] = digitalRead(5); // état canton 5
  AUTOMATE13[3] = digitalRead(8); // état droit aiguillage
  AUTOMATE13[4] = digitalRead(9); // état dévié aiguillage
}

```

```

// feu par défaut au vert
digitalWrite(4, HIGH);

// exploitation du contenu du tableau

// l'aiguillage est voie directe vers le canton 9 qui est occupé
if ((AUTOMATE13[3] == 1) && (AUTOMATE13[1] == 1)) {
  message = 1 ;
}
// averti au TCO l'opérateur change la position de l'aiguillage
if ((AUTOMATE13[2] == 0) && (AUTOMATE13[4] == 1)) {
  message = 2;
}

// les deux cantons devant le canton 13 sont occupés
if ((AUTOMATE13[1] == 1) && (AUTOMATE13[2] == 1)) {
  message = 3;
}
// Traitement des messages de l'état du canton 13

switch (message) {

  // le train est bloqué par le feu au rouge car le canton 9 devant lui est
  // occupé et l'aiguillage est en voie directe
  case 1 :
    // le feu passe au rouge
    digitalWrite (4, LOW);
    // on renseigne le TCO
    display.clearDisplay();
    display.setTextSize( 1 );
    display.setTextColor( WHITE );
    display.setCursor( 6, 3 );
    display.println( "Train bloqué" );
    display.setCursor( 6, 11 );
    display.println( "Solution changement" );
    display.setCursor( 6, 20 );
    display.println( "position aiguillage" );
    display.drawRect( 0, 0, display.width() - 2, display.height() - 2, WHITE );
    display.display();
    break;

```

```
// l'opérateur a changé au TCO la position de l'aiguillage en voie déviée vers le canton 5 qui est libre
```

```
case 2 :
```

```
    // le feu passe au vert
    digitalWrite (4, HIGH);
    // on renseigne le TCO
    display.clearDisplay();
    display.setTextSize( 1 );
    display.setTextColor( WHITE );
    display.setCursor( 6, 3 );
    display.println( "Voie libre" );
    display.setCursor( 6, 11 );
    display.println( "vers canton 5" );
    display.setCursor( 6, 20 );
    display.println( "feu vert en sortie" );
    display.drawRect( 0, 0, display.width() - 2, display.height() - 2, WHITE );
    display.display();
    break;
```

```
// le train est bloqué par le feu au rouge car les cantons 9 et 5 devant lui sont occupés
```

```
case 3 :
```

```
    // le feu passe au rouge
    digitalWrite (4, LOW);
    // on renseigne le TCO
    display.clearDisplay();
    display.setTextSize( 1 );
    display.setTextColor( WHITE );
    display.setCursor( 6, 3 );
    display.println( "Arret impose " );
    display.setCursor( 6, 11 );
    display.println( "En attente" );
    display.setCursor( 6, 20 );
    display.println( "liberation cantons" );
    display.drawRect( 0, 0, display.width() - 2, display.height() - 2, WHITE );
    display.display();
    break;
```

```
}
```

```
} // fin du programme
```