```
// Tuto07
// Exemple pour le forum 3rails / Julie Dumortier / Licence GPL
//
// extrait du programme Fosse v1.5
//   afficher l'état de la voie sur le rail contact : STOP, libre ou
occupé
//    sortie D3 -> relais (LOW = voie libre, HIGH = voie occupée)
//    sortie D13 -> led interne (LOW = voie alimentée, HIGH = voie
STOP) EXPERIMENTAL
//   utiliser un afficheur 7 segments 4 digits
//
// Retrouvez ce tutoriel sur le lien : https://forum.3rails.
fr/t/une-debutante-dans-le-decor-ep6-quelques-automatismes-arduino-na
no/18361
// version 14 mai 2021 : l'état de la voie est prise directement sur
le signal

// défini la broche (pin) utilisée pour commander le relais : D3
int relaisPin = 3;

// défini la broche (pin) utilisée pour lire la valeur analogique :
A3
int Pin_etatRail = 3;

#include <Arduino.h>
#include <TM1637Display.h>

#define CLK   9
#define DIO   8

TM1637Display display(CLK, DIO);
uint8_t segments[] = {0xff, 0xff, 0xff, 0xff};

// variable pour stocker le seuil de déclenchement
int seuilSignal = 384;

// variable pour compter les déclenchements
int nbFiltre = 16;

// on va filter le signal bas pendant 192 ms pour décider que la
voie est libre
int msFiltre = 192;
```

```
// Les différents états possibles pour la voie sur le rail contact
//  voie_Occupee la voie est occupée
//  voie_Libre   la voie est libre
//  voie_STOP    la voie est en STOP (CSx ou MSII)

#define voie_STOP    99
#define voie_Occupee  0
#define voie_Libre    1

// AfficheEtatVal
//  affiche l'état et une valeur
//

void AfficheEtatVal(int etat,int val)
{
  // 1er segment : Etat
  switch (etat) {
    case voie_Libre:
      segments[0] = SEG_F | SEG_E | SEG_D;
      break;

    case voie_Occupee:
      segments[0] = SEG_C | SEG_E | SEG_D | SEG_G;
      break;

    case voie_STOP:
      segments[0] = SEG_A | SEG_C | SEG_D | SEG_F | SEG_G;
      break;

    default:
      display.clear();
      break;
  }
  segments[1] = display.encodeDigit((val / 100) % 10);
  segments[2] = display.encodeDigit((val / 10) % 10);
  segments[3] = display.encodeDigit(val % 10);
  display.setSegments(segments);
}

// EtatVoie
// retourne l'état de la voie

int EtatVoie()
```

```
{
  int n = msFiltre;
  int c = 0;

  int vr = 0;
  int vt = 0;

  // flag pour savoir si on a recu un signal de traction
  int traction = 0;

  // tant que le signal est bas on continue
  while (n>0) {
    // lit les deux entrées : rail contact et rail normal
    vr = analogRead(Pin_etatRail);
    vt = vr; // analogRead(Pin_etatTraction);

    if (vt>1) {
      traction = vt;
    }

    if (vr>seuilSignal) {
      c = c + 1;
    }

    if (c>nbFiltre) {
      // la voie est encore occupée

      // attends la fin du compteur-> la fonction EtatVoie() est à
temps d'exécution constant (ce délai peut être supprimé)
      delay(n);

      // met à jour l'afficheur
      AfficheEtatVal(voie_Occupee,vr);

      // et retourne l'état d'occupation
      return voie_Occupee;
    }

    // attends 1 ms
    delay(1);

    // et boucle
    n = n - 1;
```

```
  }

  // on a eu un signal de traction
  if (traction>0) {

    // le rail contact est resté bas --> la voie est libre
    Serial.println("voie libre");

    // met à jour l'afficheur
    AfficheEtatVal(voie_Libre,traction);
    return voie_Libre;
  }

  // pas de traction

  // met à jour l'afficheur
  AfficheEtatVal(voie_STOP,0);

  return voie_STOP;
}

// code executé une seule fois au démarrage du module (ou après un
reset)
void setup() {

  // programme la sortie digitale D3 en sortie
  // le In du relais est connecté à cette sortie D3
  pinMode(relaisPin, OUTPUT);

  // Led 13 interne : rouge si STOP détecté, éteinte sinon
  pinMode(LED_BUILTIN, OUTPUT);

  // ouvre le port série (console de l'outil) avec la vitesse 57600
bauds
  // attention que le paramètre sur la console soit bien 57600 !
  Serial.begin(57600);

  // relais COMMON - NC (led verte allumée)
  digitalWrite(relaisPin,LOW);

  // regle l'intensité de l'affichage
  display.setBrightness(0x0f);
  display.clear();
```

```
}

// code executé en permanence (une boucle)
void loop() {

  int etat;

  etat = EtatVoie();

  switch (etat) {
    case voie_Libre:
      // relais COMMON - NC (led verte allumée)
      digitalWrite(relaisPin,LOW);

      // led 13 éteinte - voie alimentée
      digitalWrite(LED_BUILTIN,LOW);

      Serial.println("Voie libre");
      break;

    case voie_Occupee:
      // relais COMMON - NO (led verte eteinte)
      digitalWrite(relaisPin,HIGH);

      // led 13 éteinte - voie alimentée
      digitalWrite(LED_BUILTIN,LOW);

      Serial.println("Voie occupée");
      break;

    case voie_STOP: // EXPERIMENTAL
      // relais COMMON - NO (led verte eteinte)
      digitalWrite(relaisPin,HIGH);

      // led 13 allumée - voie STOP
      digitalWrite(LED_BUILTIN,HIGH);

      Serial.println("Voie STOP");
      break;

    default:
      Serial.print("Voie état inconnu = ");
      Serial.println(etat);
```

```
      break;
  }

  delay(250);

}
```