

```

// Tuto06
// 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
//
// 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 du
rail de contact : A3
int Pin_etatRail = 3;

// 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;

// EtatVoie
// retourne l'état de la voie
//  voie_Occupee la voie est occupée
//  voie_Libre   la voie est libre
//  voie_STOP    la voie est en STOP (CSx ou MSII)
//
// la voie est déclarée occupée si au moins une valeur acquise sur
A3 est supérieur au seuilSignal
// la voie est déclarée vide si aucune valeur acquise sur A3 n'est
supérieure au seuilSignal

```

```
// dans ce cas, la voie est déclarée STOP si aucune détection de  
signalnumérique sur A2
```

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

```
int EtatVoie()  
{
```

```
    int n = msFiltre;
```

```
    int vr = 0;
```

```
    int vt = 0;
```

```
    int c = 0;
```

```
    // flag pour savoir si on a reçu un signal de traction  
    boolean traction = false;
```

```
    // 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);
```

```
        //Serial.print(vr);
```

```
        //Serial.print(" ");
```

```
        if (vt>1) {
```

```
            traction = true;
```

```
        }
```

```
        if (vr>seuilSignal) {
```

```
            c = c + 1;
```

```
        }
```

```
        if (c>nbFiltre) {
```

```
            // la voie est encore occupée
```

```
            // attends la fin du compteur
```

```
            delay(n);
```

```
            // 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) {

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

    return voie_Libre;
}

// pas de traction
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);

    pinMode(Pin_etatRail, INPUT_PULLUP);

    // 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);
}

```

```
// 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 éteinte)
            digitalWrite(relaisPin,HIGH);

            // led 13 éteinte - voie alimentée
            digitalWrite(LED_BUILTIN,HIGH);
            delay(25);
            digitalWrite(LED_BUILTIN,LOW);

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

        case voie_STOP:
            // relais COMMON - NO (led verte éteinte)
            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);  
Serial.println("");
```

```
}
```