

Établissement de la logique de gestion du canton 1 pour préparer l'écriture du code Arduino



The image shows a screenshot of the Arduino IDE interface. The window title is "code3 | Arduino 1.6.11". The code editor contains the following C++ code:

```
code3
/*
  Code 3 - Edurobot.ch, destiné à l'Arduino
  Objectif: faire un chenillard à 4 LEDs montées sur les broches 10 à 13
 */

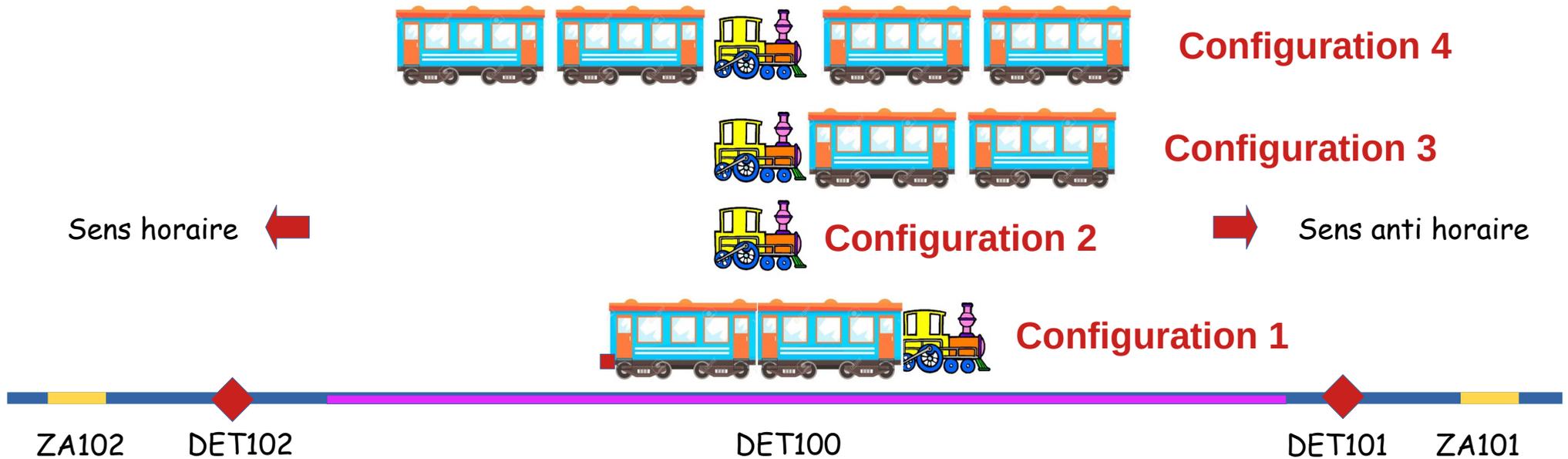
void setup()      // début de la fonction setup()
{
  pinMode(10, OUTPUT); // Initialise la broche 10 comme sortie
  pinMode(11, OUTPUT); // Initialise la broche 11 comme sortie
  pinMode(12, OUTPUT); // Initialise la broche 12 comme sortie
  pinMode(13, OUTPUT); // Initialise la broche 13 comme sortie

  Serial.begin(9600); // Ouvre le port série à 9600 bauds
} // fin de la fonction setup()

void loop()      // début de la fonction loop()
{
  digitalWrite(10, HIGH); // Met la broche 10 au niveau haut = allume la LED
  digitalWrite(11, LOW);  // Met la broche 11 au niveau bas = éteint la LED
  digitalWrite(12, LOW);  // Met la broche 12 au niveau bas = éteint la LED
  digitalWrite(13, LOW);  // Met la broche 13 au niveau bas = éteint la LED

  delay(100); // Pause de 100ms
}
```

At the bottom of the IDE, the status bar indicates "Arduino/Genuino Uno sur /dev/tty.usbmodem1421".



Typologie des trains sur ce canton (et sur les 7 autres du circuit) :

- 1) **Configuration 1**, train long avec un consommateur de courant aux deux extrémités (moteur et feux), pouvant être détecté sur DET 100 seul, ou sur DET100 et DET101 ensemble, ou sur DET100 et DET102 ensemble,
- 2) **Configuration 2**, locomotive haut le pied, autorail, draisine susceptibles de ne plus être détectés sur DET100 mais détectés sur DET 101 seul ou DET 102 seul,
- 3) **Configuration 3**, train long en mode « pousse » avec un seul consommateur de courant à une extrémité (moteur), pouvant être détecté sur DET 100 ou sur DET101 seul ou sur DET102 seul, avec le risque d'engager une partie du train sur un des aiguillages de sortie sans vérification de son positionnement correct,
- 4) **Configuration 4**, train avec un seul consommateur de courant au centre la rame, pouvant être détecté sur DET100 seul ou sur DET101 seul ou sur DET102 seul, avec le risque d'engager une partie du train sur un des aiguillages de sortie sans vérification de son positionnement correct.

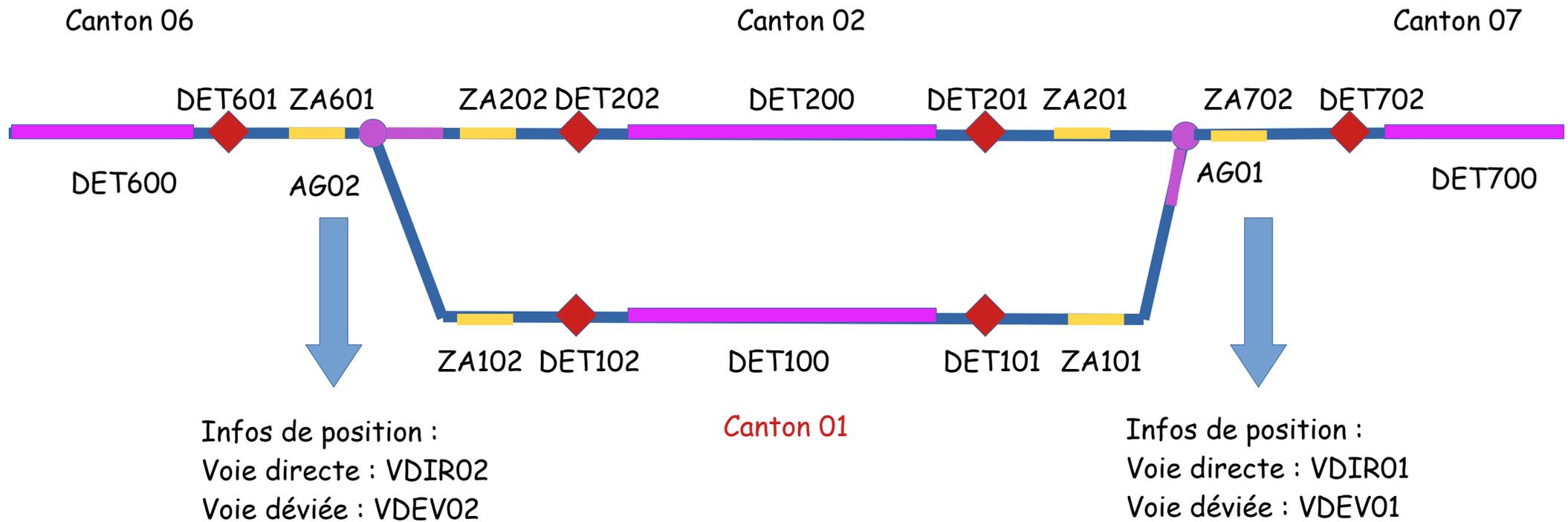
Type de configuration	Détection uniquement par la consommation de courant sur DET100, DET101, DET102	Détection par consommation de courant sur DET100 et par coupure d'un faisceau laser sur DET101 et DET102
Train long avec un consommateur de courant aux deux extrémités (moteur et feu de fin de convoi)	Couvre le besoin	Couvre le besoin
Locomotive haut le pied, autorail, draisine, etc. avec un seul consommateur de courant (moteur)	Ne couvre pas le besoin, risque de confusion sur la fin réelle d'occupation du canton	Couvre le besoin
Train long en mode « pousse » avec un seul consommateur de courant à une extrémité (moteur)	Ne couvre pas le besoin, risque de confusion sur la fin réelle d'occupation du canton et risque d'engagement de la rame sur l'aiguillage de sortie mal positionné	Couvre le besoin
Train avec un seul consommateur de courant (moteur) au centre la rame	Ne couvre pas le besoin, risque de confusion sur la fin réelle d'occupation du canton et risque d'engagement de la rame sur l'aiguillage de sortie mal positionné	Couvre le besoin



Détection de consommation de courant

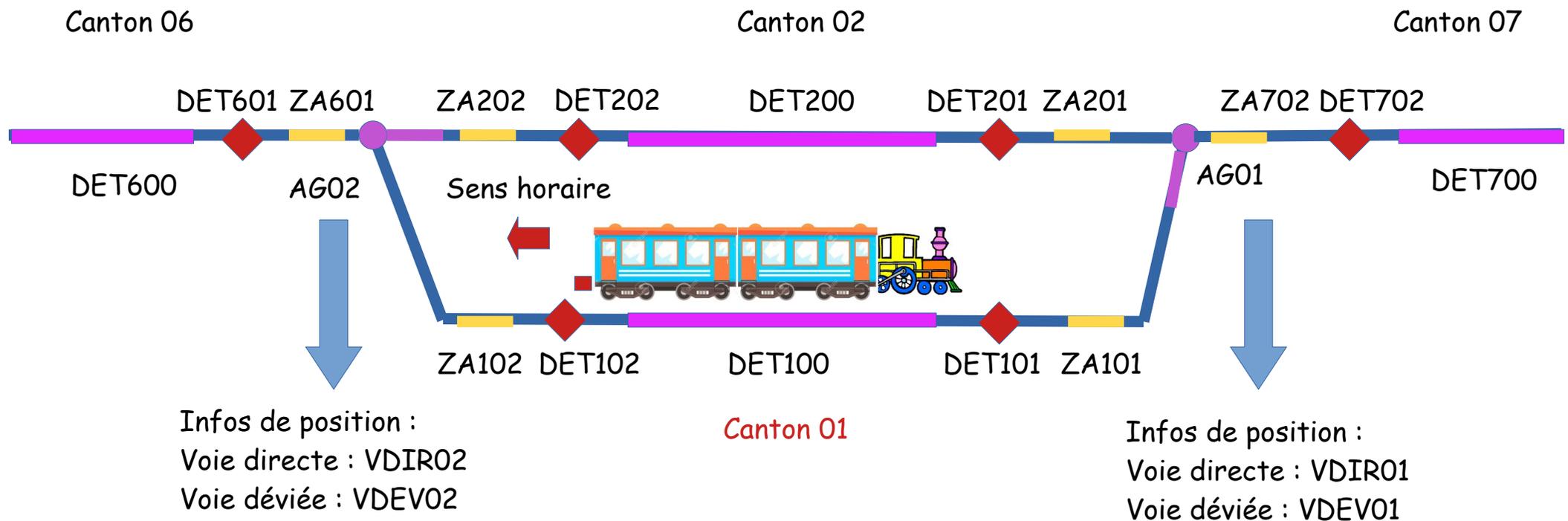


Détection laser avec capteur VL5310



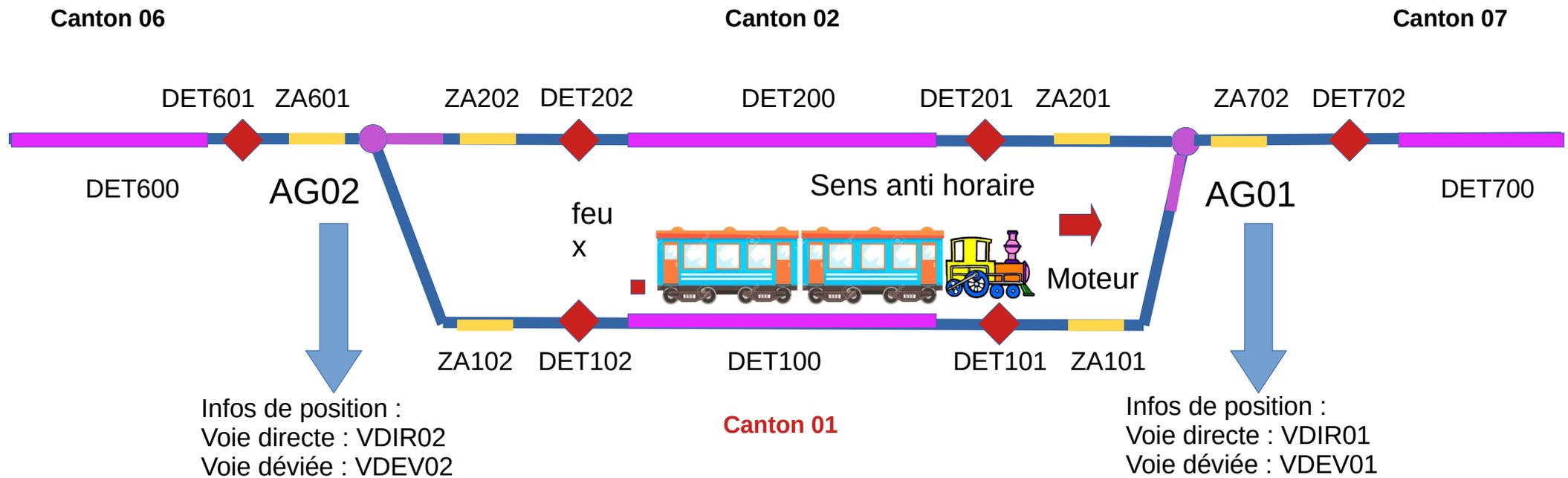
Cas 1 : absence de train

- Si $DET100 = 0$, si $DET101 = 0$, si $DET102 = 0$ alors il n'y a pas de train sur le canton 1,
- La variable d'état du canton $ETATCANTON1 = 0$ pour canton libre
- Les deux zones d'arrêt ZA101 et ZA102 sont alimentées pour permettre d'accepter le franchissement d'un train .



Cas 2 : train qui se déplace en sens horaire (en marche avant ou arrière, configuration du train de 1 à 4)

- Si $DET100 = 1$, et si $DET102 = 1$, alors il y a un train sur le canton 1 qui se déplace en sens horaire,
- La variable d'état du canton 1 devient $ETATCANTON1 = 2$ pour canton occupé,
- $ZA101 = 1$ l'alimentation de cette zone est aussitôt coupée (pour protéger ce train dans le cas où $AG01$ serait en voie déviée par oubli de l'opérateur humain),
- Si $VDIR02 = 0$ et si $VDEV02 = 1$, alors l'aiguillage $AG02$ est correctement positionné en voie déviée, sinon si $VDIR02 = 1$ et si $VDEV02 = 0$, l'aiguillage $AG02$ est positionné en voie directe, l'opérateur humain est averti au TCOE qu'il doit intervenir sur l'aiguillage $AG02$, et $ZA102 = 1$ l'alimentation de la zone d'arrêt est coupée, le train ne peut pas passer, jusqu'à ce que l'opérateur humain ait rectifié la position de l'aiguillage $AG02$,
- Si la variable d'état du canton 6 est $ETATCANTON6 = 0$ le canton 6 est libre, le train peut y accéder (sinon le train attend la libération du canton 6),
- Arduino active aussitôt toutes les zones d'arrêt $ZA301, ZA401, ZA501$ en sortie des cantons de la zone Nord pour empêcher l'accès au canton 6, le canton 6 est alors « réservé » avec $ETATCANTON6 = 1$ jusqu'au moment où l'arrivée du train sur $DET601$ puis $DET600$ le rend « occupé » avec $ETATCANTON6 = 2$



Cas 3 : train qui se déplace en sens anti horaire (en marche avant ou arrière, configuration du train de 1 à 4)

- Si $DET100 = 1$, et si $DET101 = 1$, alors il y a un train sur le canton 1 qui se déplace en sens anti horaire,
- La variable d'état du canton 1 devient $ETATCANTON1 = 2$ pour canton occupé,
- $ZA102 = 1$ l'alimentation de cette zone est aussitôt coupée (pour protéger ce train dans le cas où $AG02$ serait en voie déviée par oubli de l'opérateur humain),
- Si $VDIR01 = 0$ et si $VDEV01 = 1$, alors l'aiguillage $AG01$ est correctement positionné en voie déviée, sinon si $VDIR01 = 1$ et si $VDEV01 = 0$, l'aiguillage $AG01$ est positionné en voie directe, l'opérateur humain est averti au TCOE qu'il doit intervenir sur l'aiguillage $AG01$, et $ZA101 = 1$ l'alimentation de la zone d'arrêt est coupée, le train ne peut pas passer, jusqu'à ce que l'opérateur humain ait rectifié la position de l'aiguillage $AG01$,
- Si la variable d'état du canton 7 est $ETATCANTON7 = 0$ le canton 7 est libre, le train peut y accéder (sinon le train attend la libération du canton 7),
- Arduino active aussitôt toutes les zones d'arrêt $ZA302, ZA401, ZA502$ en sortie des cantons de la zone Nord pour empêcher l'accès au canton 7, le canton 7 est alors « réservé » avec $ETATCANTON7 = 1$ jusqu'au moment ou l'arrivée du train sur $DET 702$ puis $DET700$ le rend « occupé » avec $ETATCANTON7 = 2$