

# L'Architecture LOCODUINO

Vers une architecture ouverte et modulaire pour la commande automatisée du réseau avec l'Arduino

Jean-Luc Béchenec

Christophe Bobille

Dominique Bultez

Thierry Paris

Christian Bézanger

Hubert Delanghe

Pierre Lecouffe



[www.locodduino.org](http://www.locodduino.org)

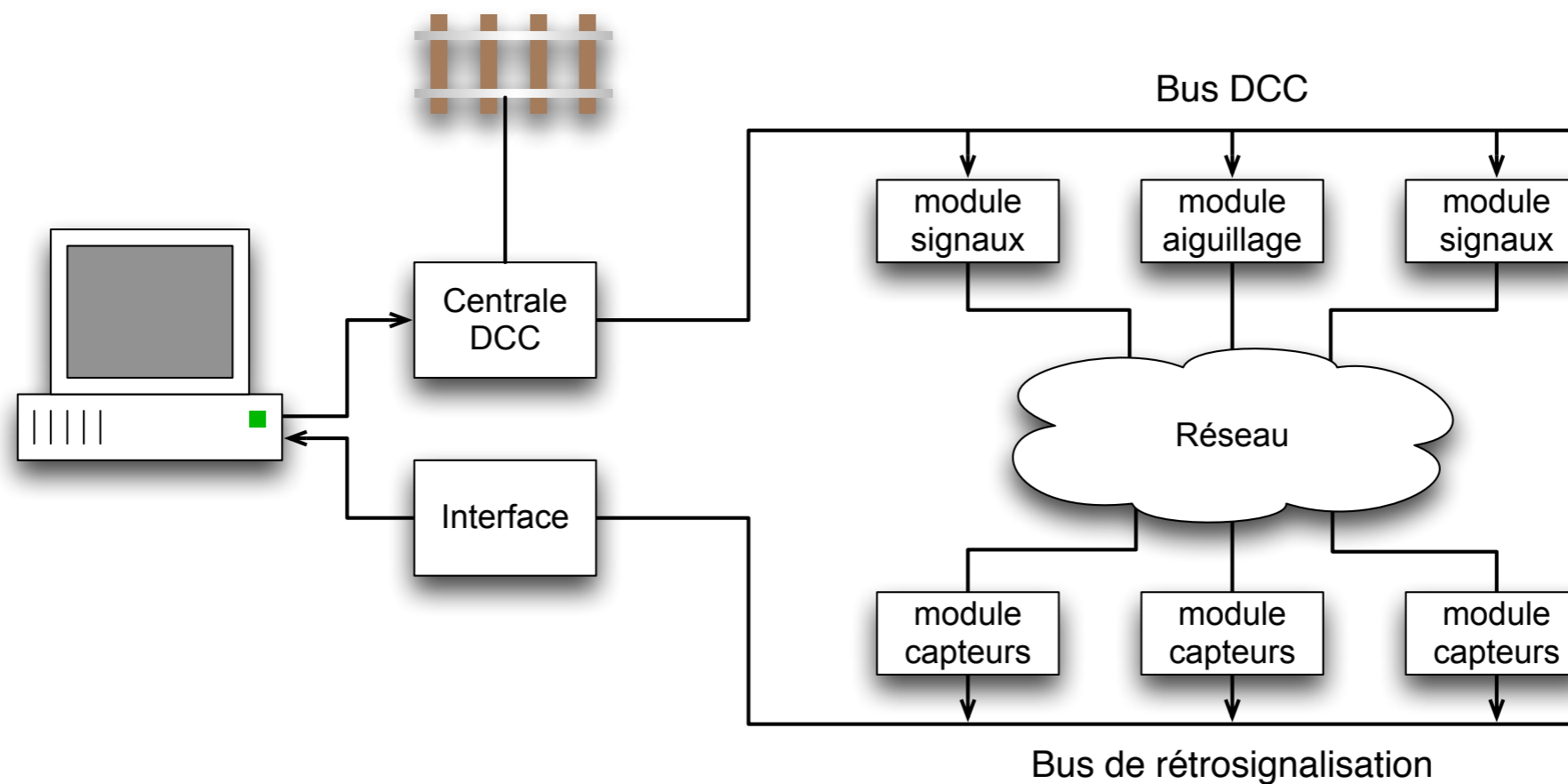


# Sommaire

1. Architecture électronique et informatique avec l'Arduino
2. Le Locoduinodrome
3. La carte Satellite V1
4. La carte Satellite V2

# Architecture classique

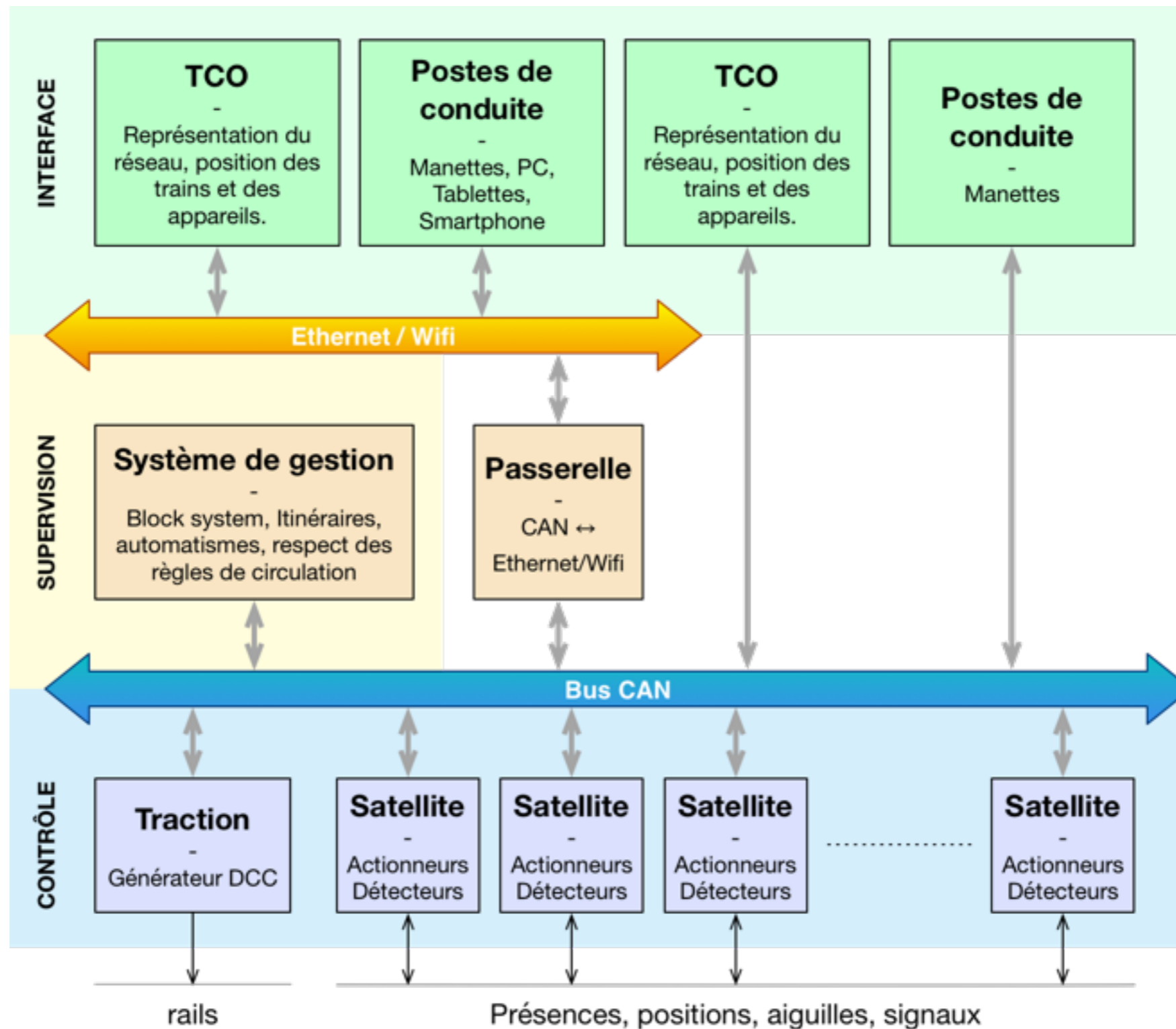
- Classiquement
  - Ordinateur *general purpose*
  - Bus DCC pour la commande
  - Bus pour la rétro-signalisation (S88, RS, CAN, ...)



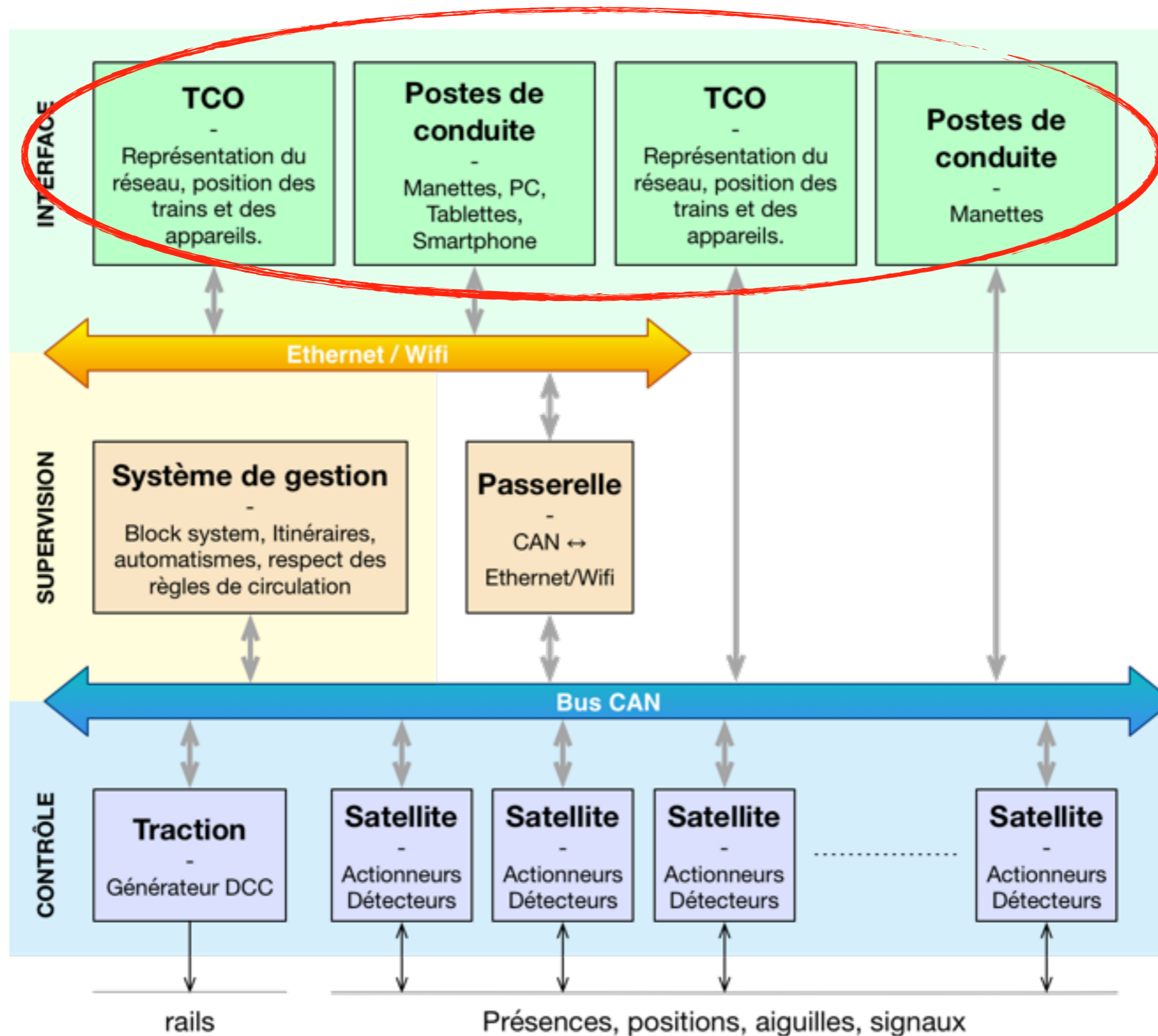
# Architecture classique

- Les modules sont mono-fonction
  - sous-utilisation des modules
  - ou câblage abondant (quand c'est possible)
- Les modules sont sur des bus séparés
  - actionneurs sur le DCC
  - capteurs sur un bus de rétro-signalisation

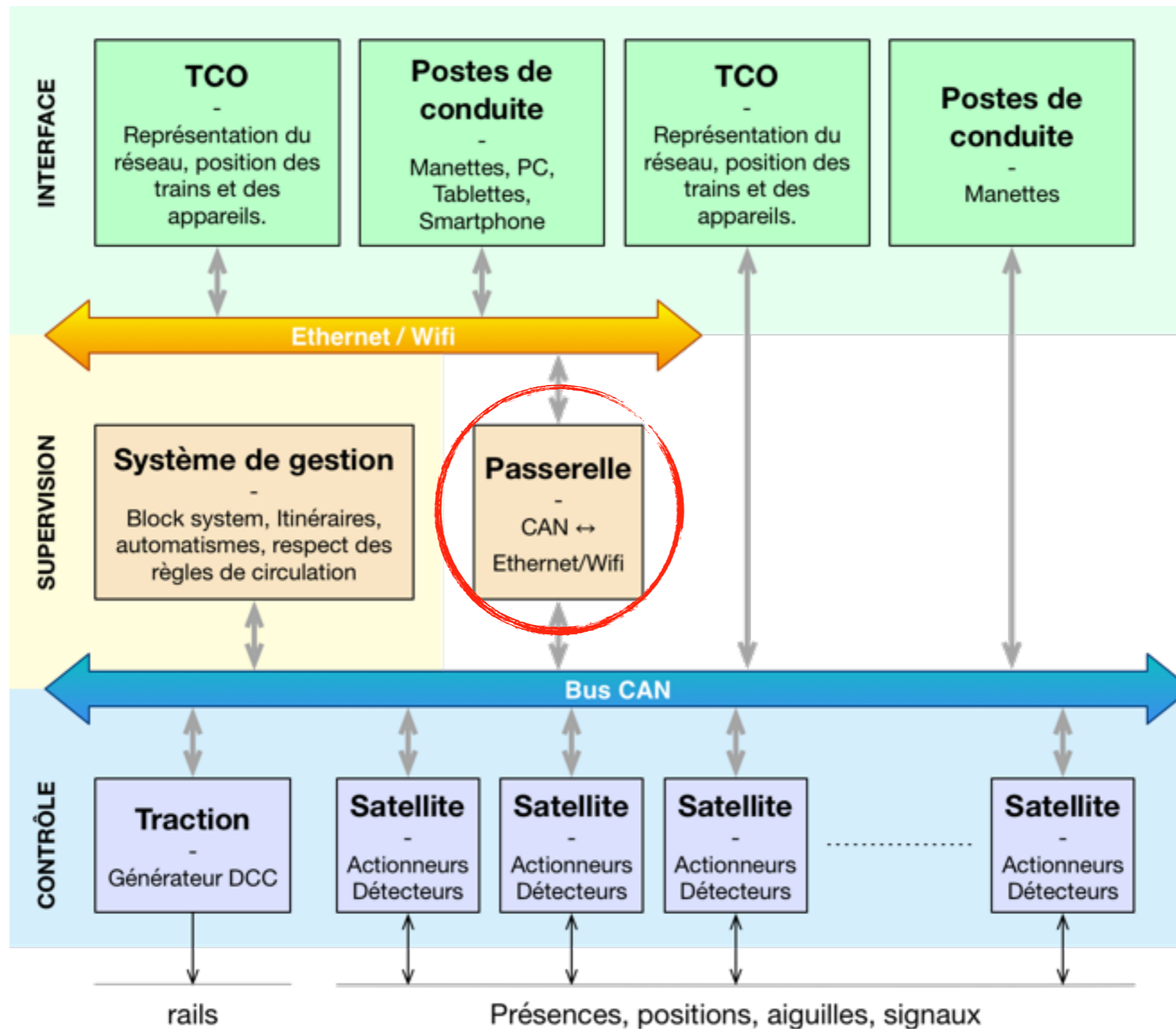
# Architecture Locoduino



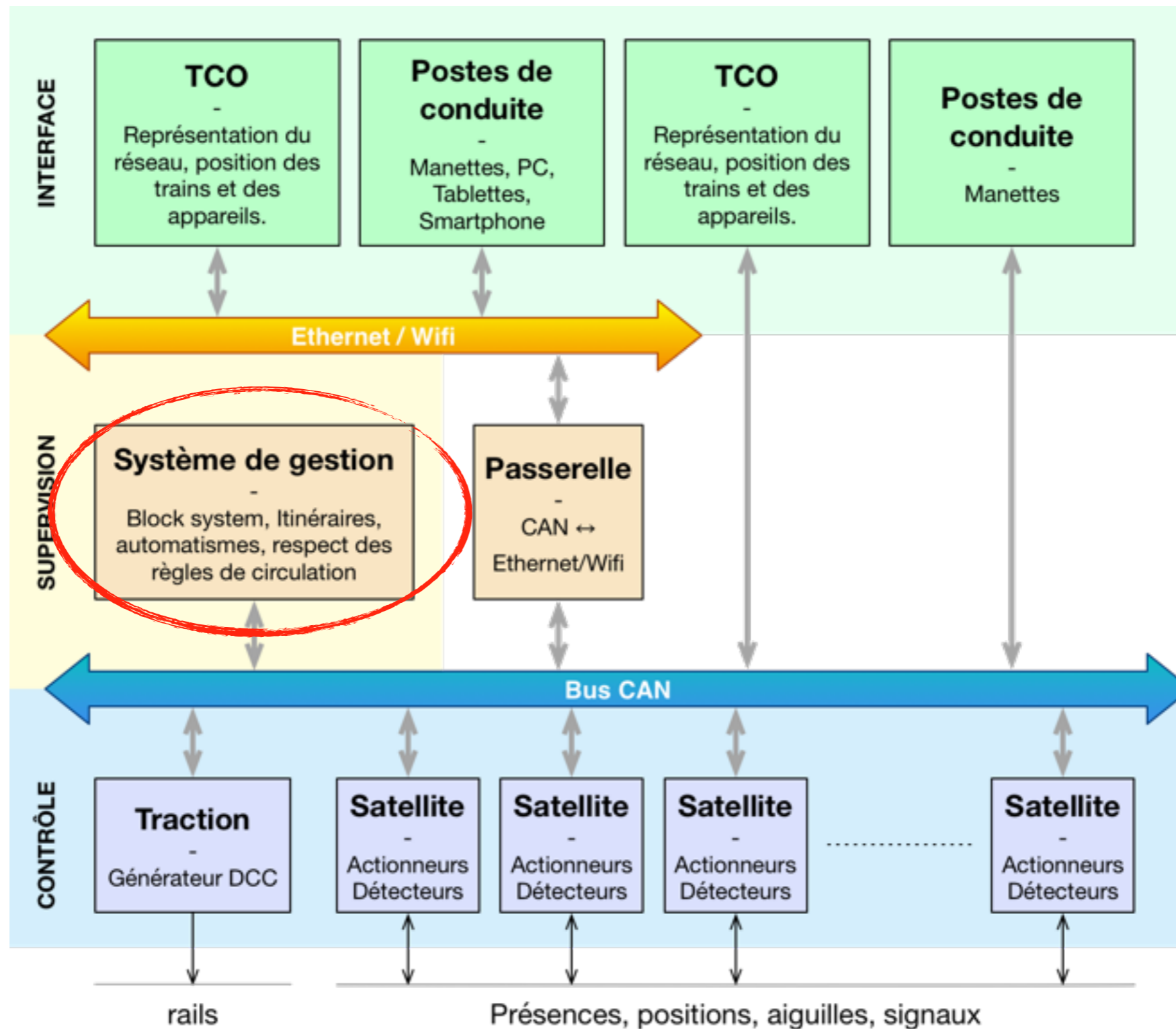
# Architecture Locoduiño



# Architecture Locoduiño

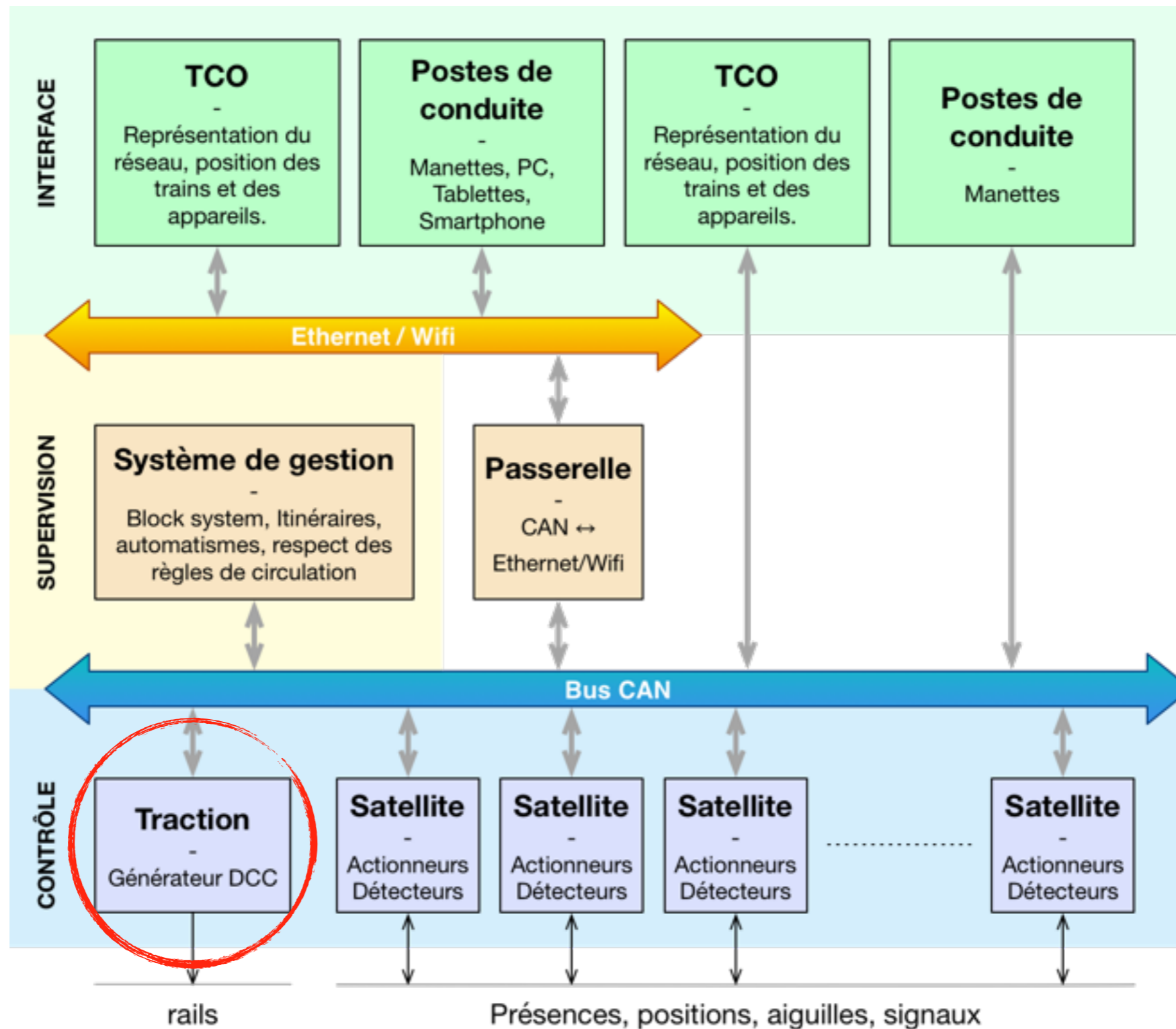


# Architecture Locoduiño

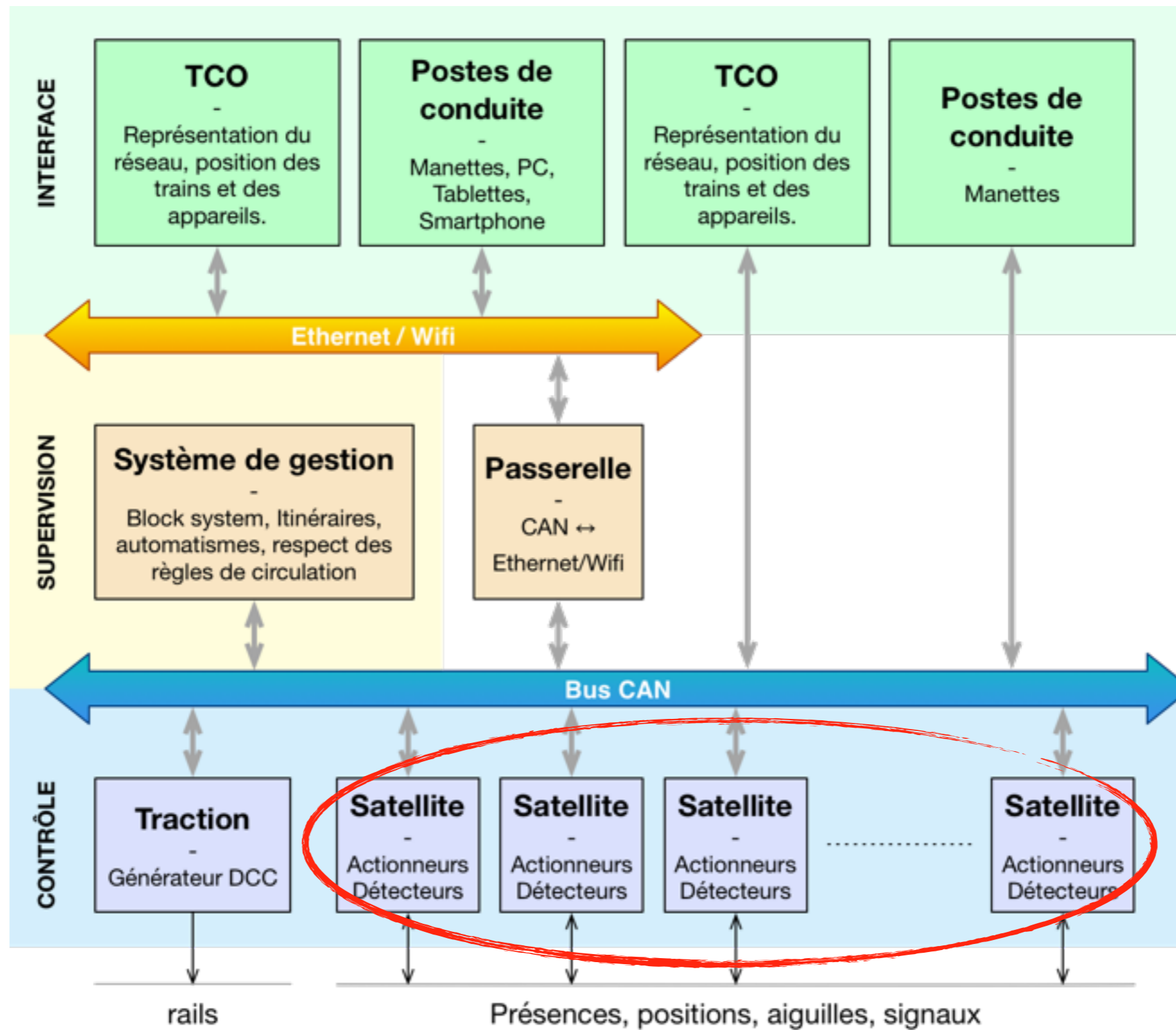




# Architecture Locoduino



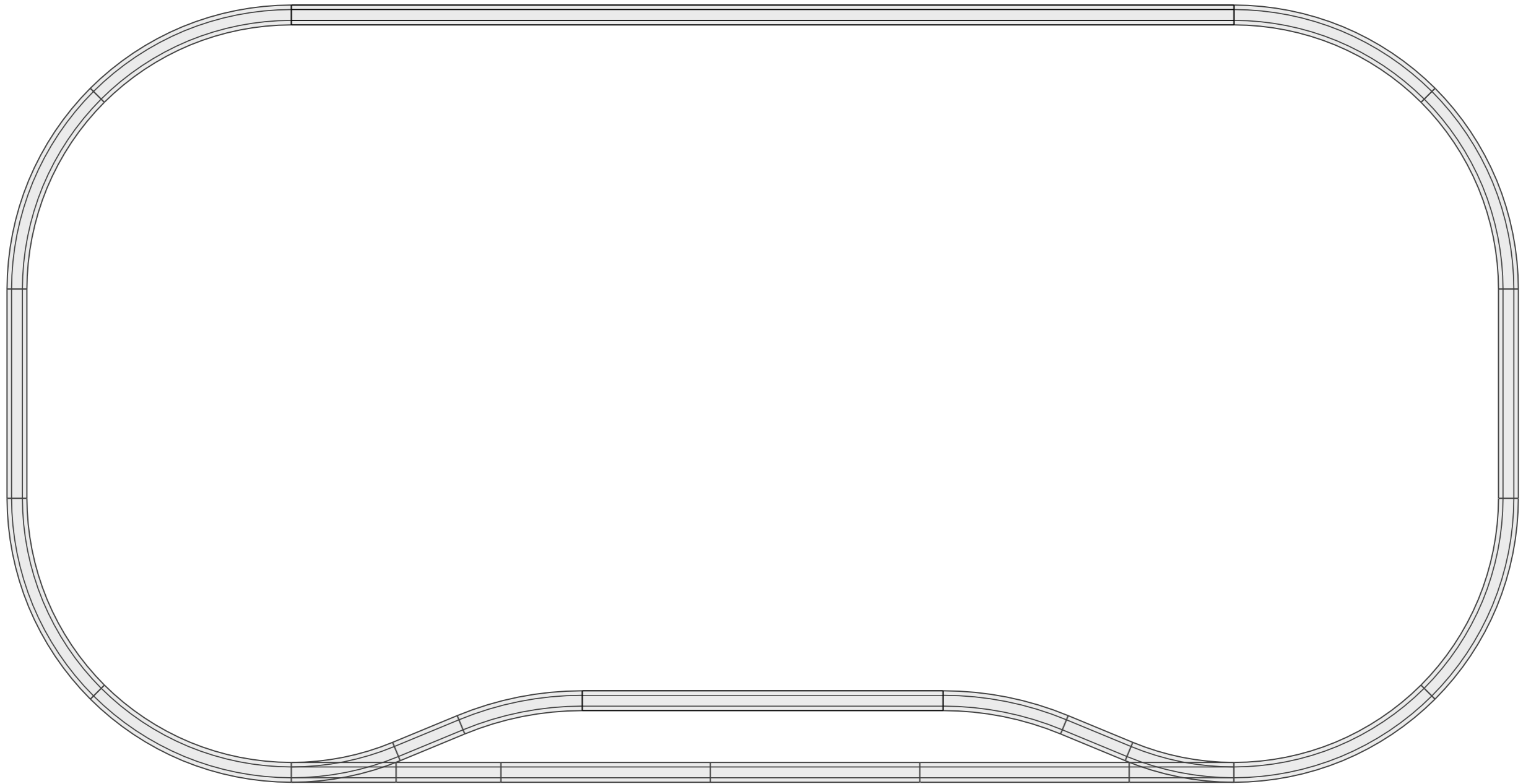
# Architecture Locoduiño



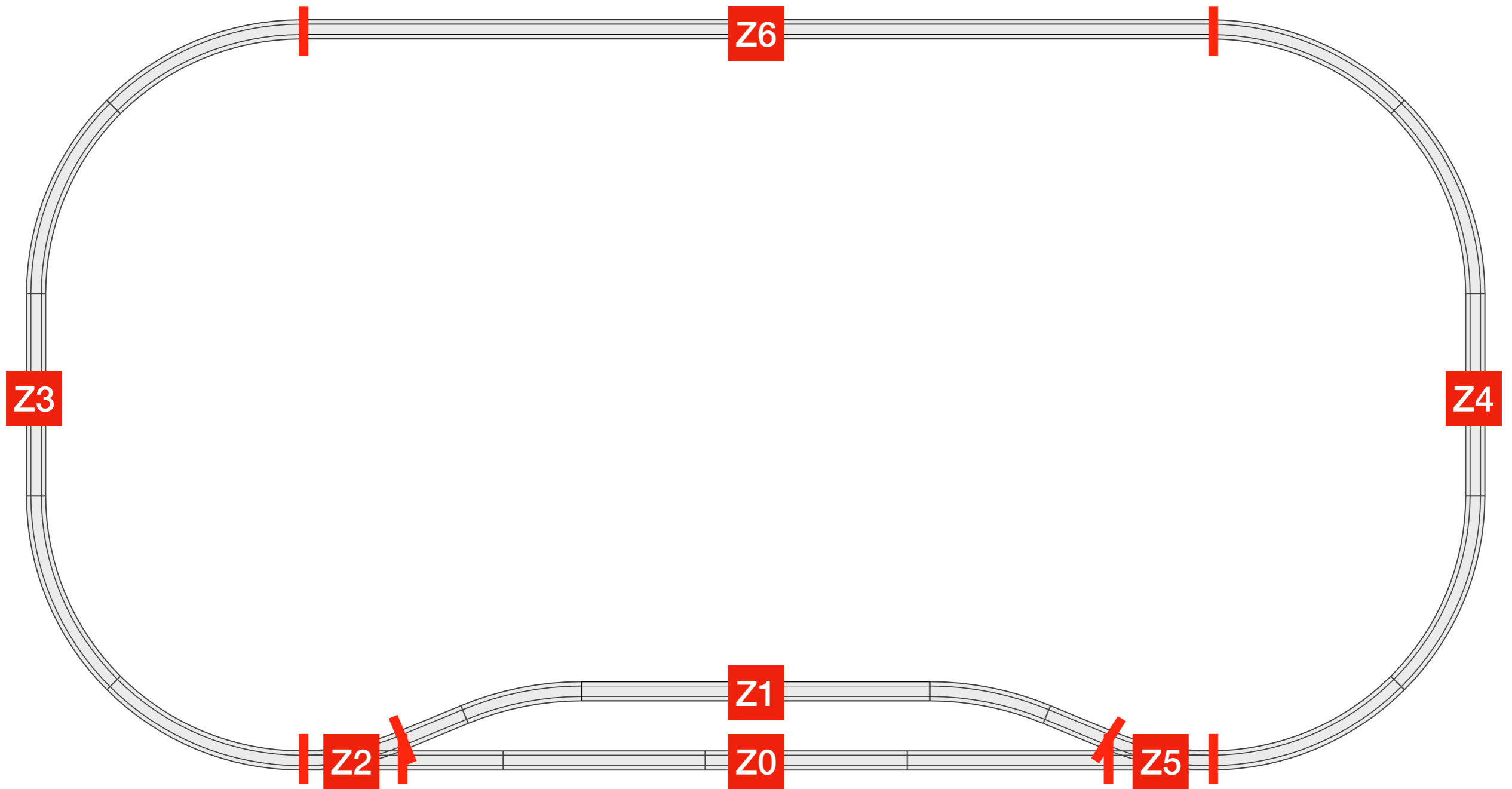
# Solution explorée

- **Modules multifonctions** et un **bus unique** de communication
  - modules configurables
  - minimisation du câblage.
  - utilisation du bus CAN

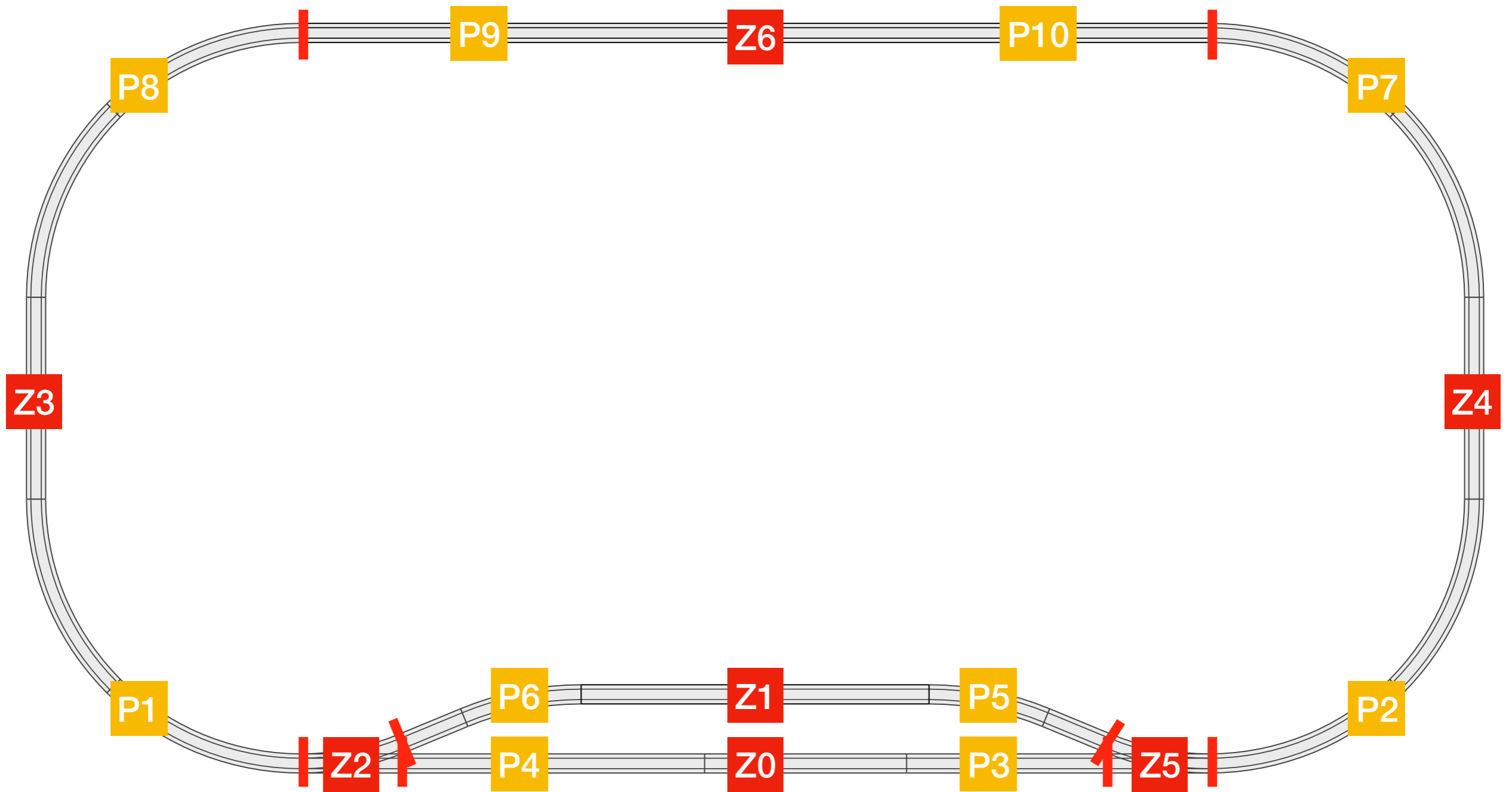
# Le Locoduinodrome



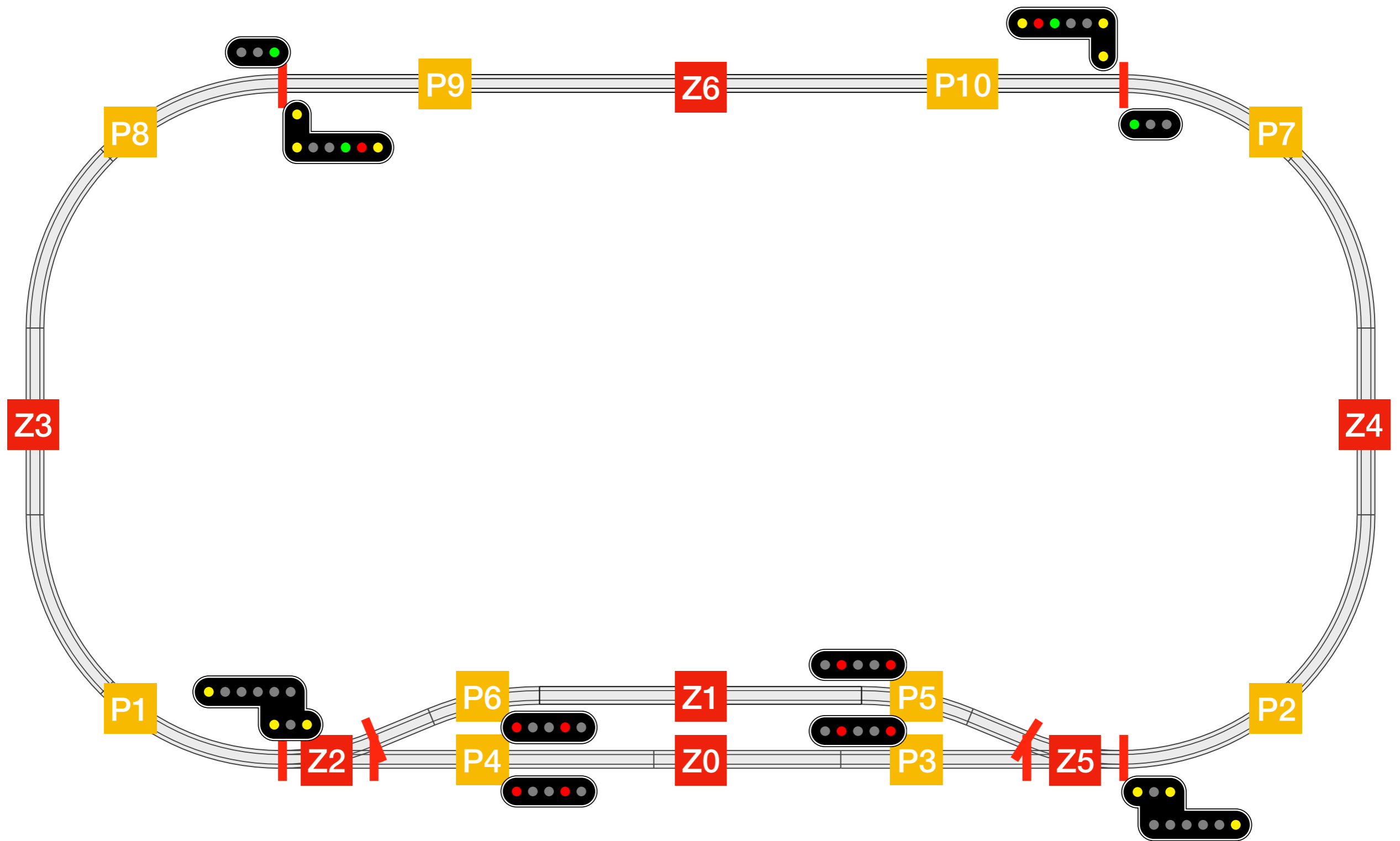
# Le Locoduinodrome



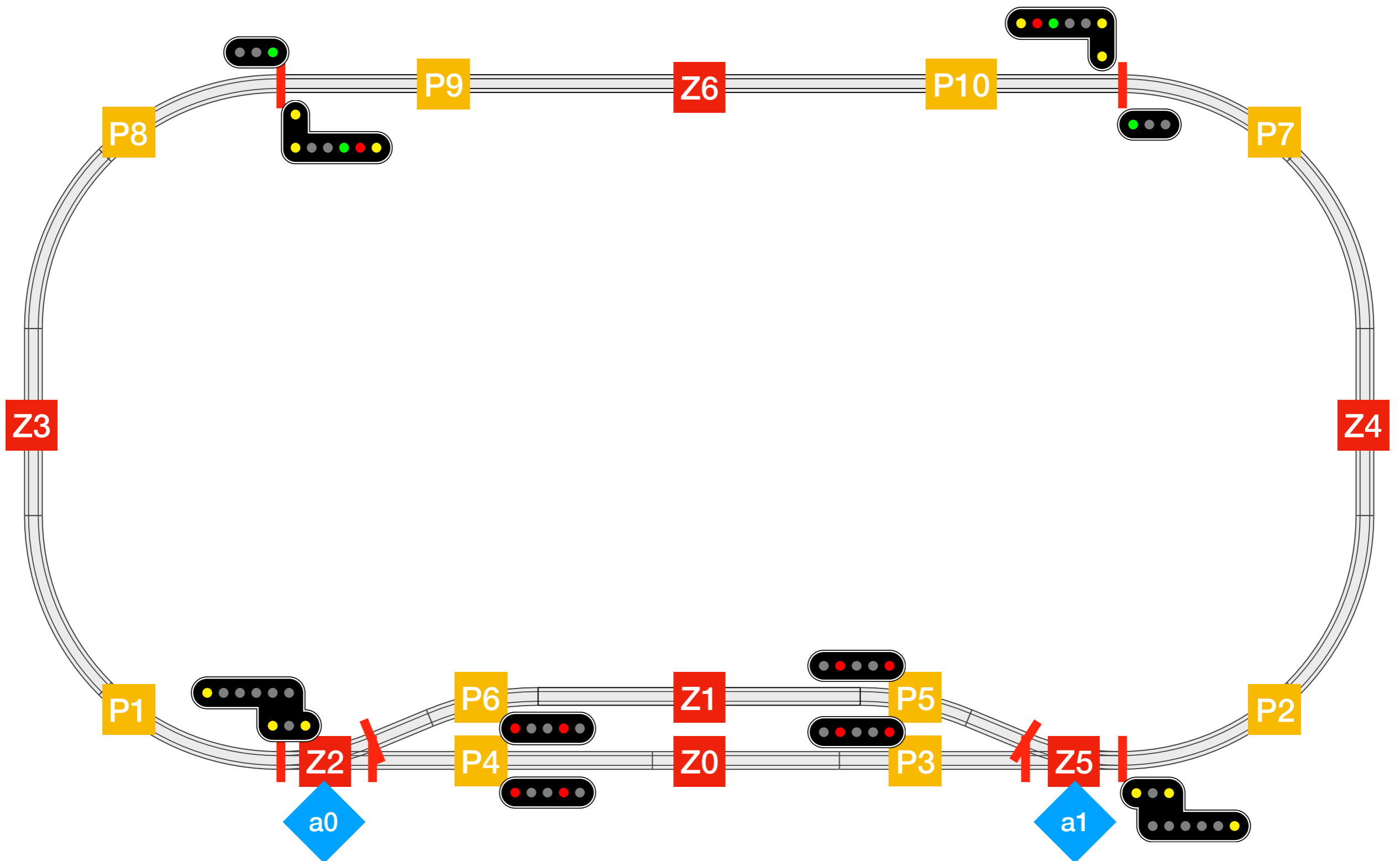
# Le Locoduinodrome



# Le Locoduinodrome



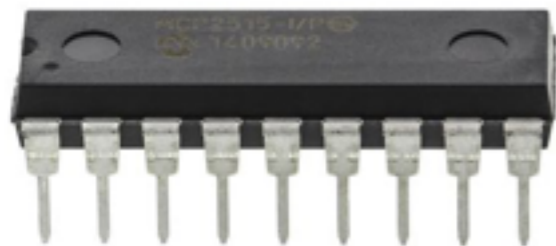
# Le Locoduinodrome



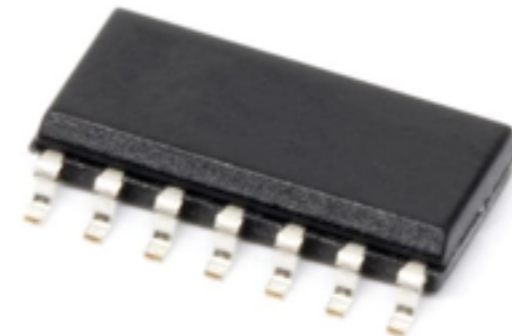


# Bus CAN

- largement utilisé dans l'industrie (automobile, avionique, ...)
- utilisé dans le modélisme ferroviaire (MERC, ZIMO)
- bus **multi-maître asynchrone**
- **temps réel** : les temps de transmission sont déterministes et peuvent être calculés à priori
- **priorité** entre les messages
- **détection d'erreur**.  
Probabilité d'une erreur non détectée  $< 4,7 \times 10^{11}$
- vitesse jusqu'à **1Mb/s**
- **contrôleurs autonomes** ne nécessitant que très peu de logiciel

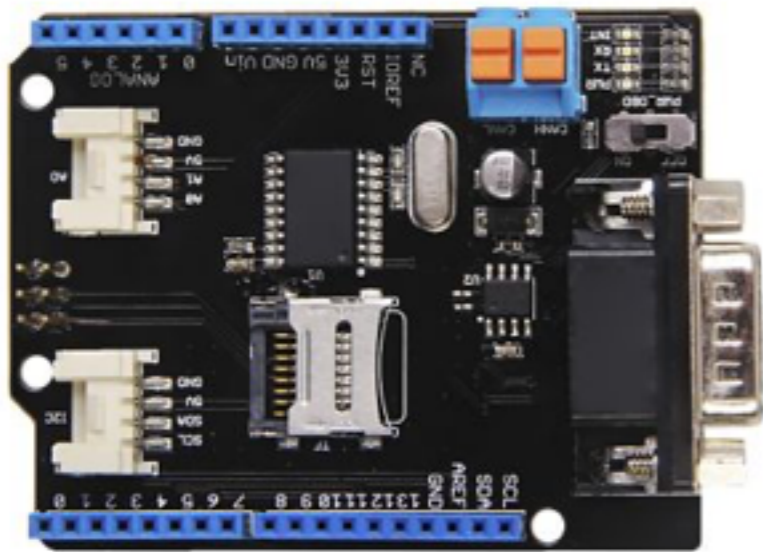


MCP2515



MCP2517FD

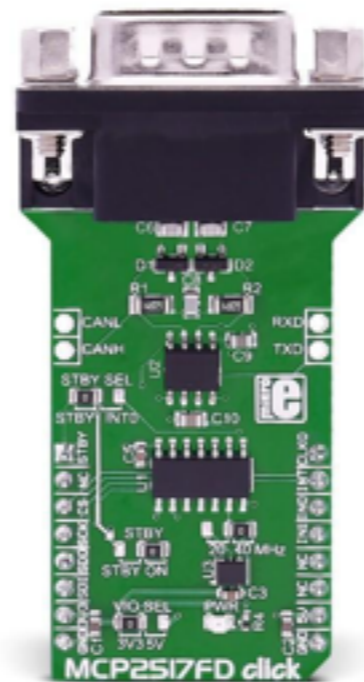
# Bus CAN et Arduino



CAN Bus Shield et module  
Groove de Seeedstudio  
(MCP2515)

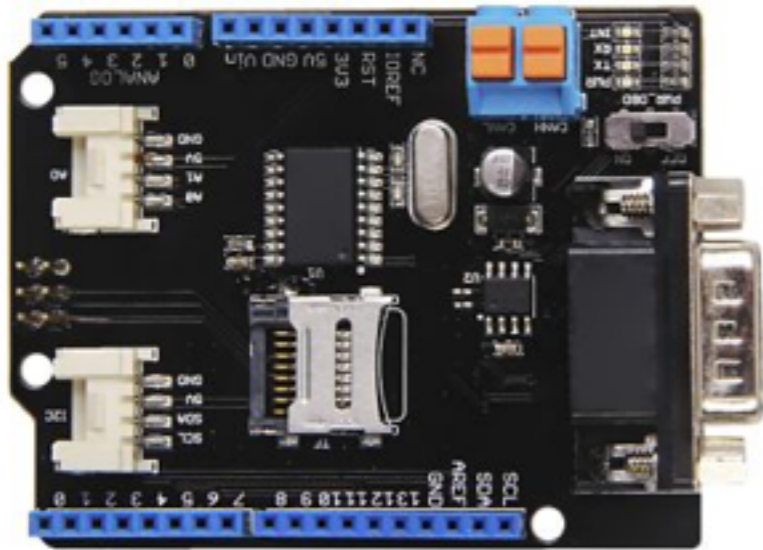


Module MCP2515  
(eBay)



Module MCP2517  
(Mikroelektronika)

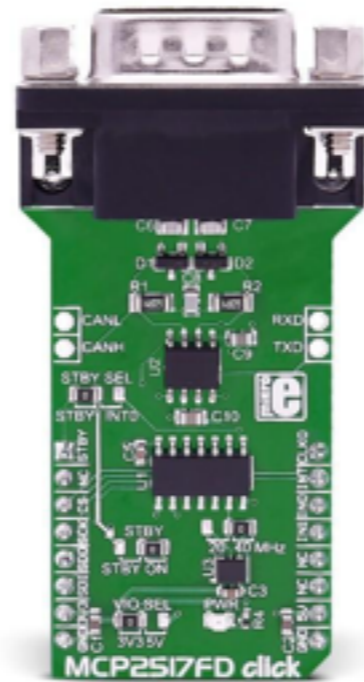
# Bus CAN et Arduino



CAN Bus Shield et module  
Groove de Seeedstudio  
(MCP2515)



Module MCP2515  
(eBay)



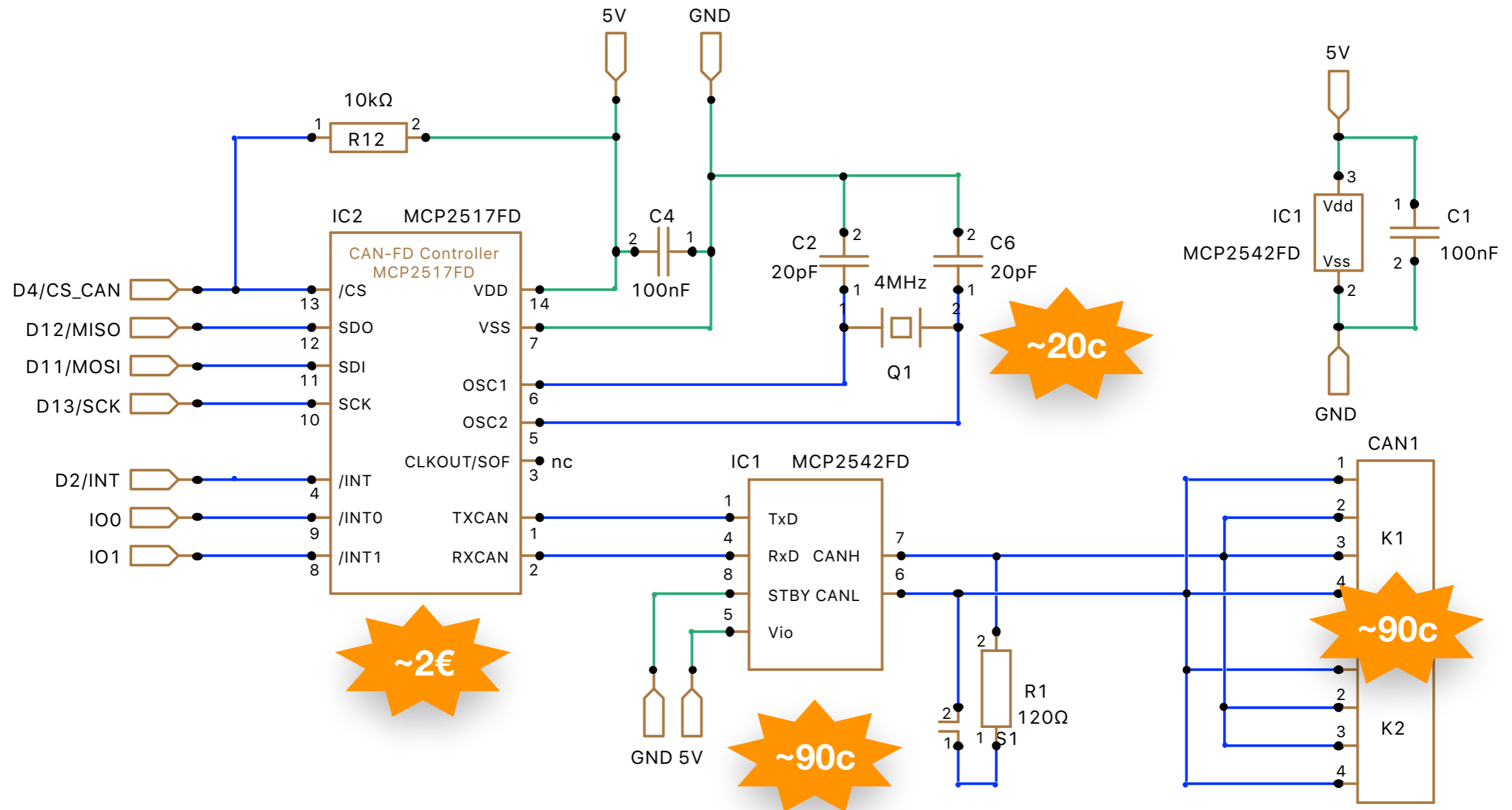
Module MCP2517  
(Mikroelektronika)

Arduino (like)  
équipés  
nativement :

- Due
- Teensy 3.x
- Nucleo
- ...

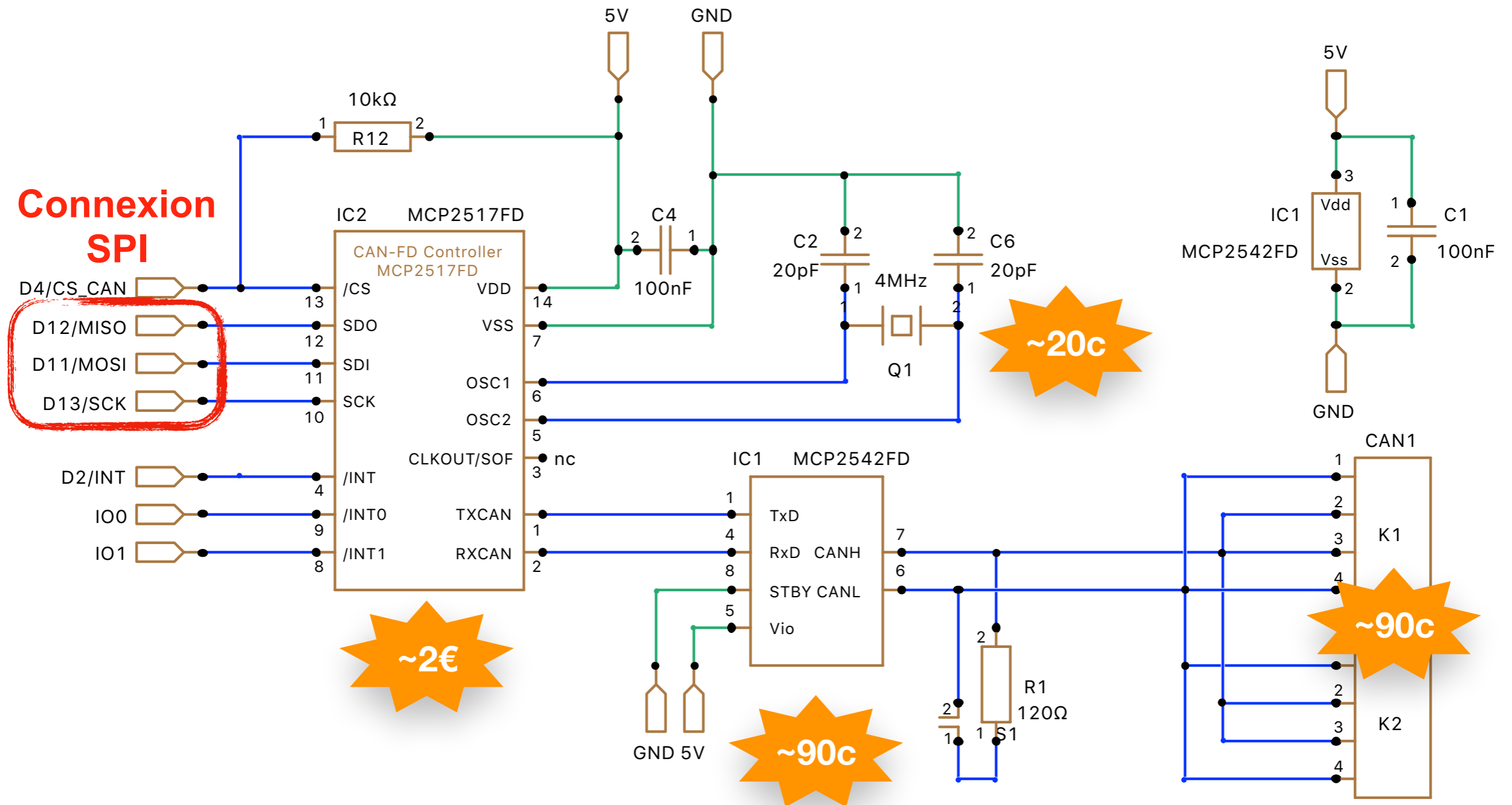
# Bus CAN et Arduino

- Réaliser sa propre interface CAN



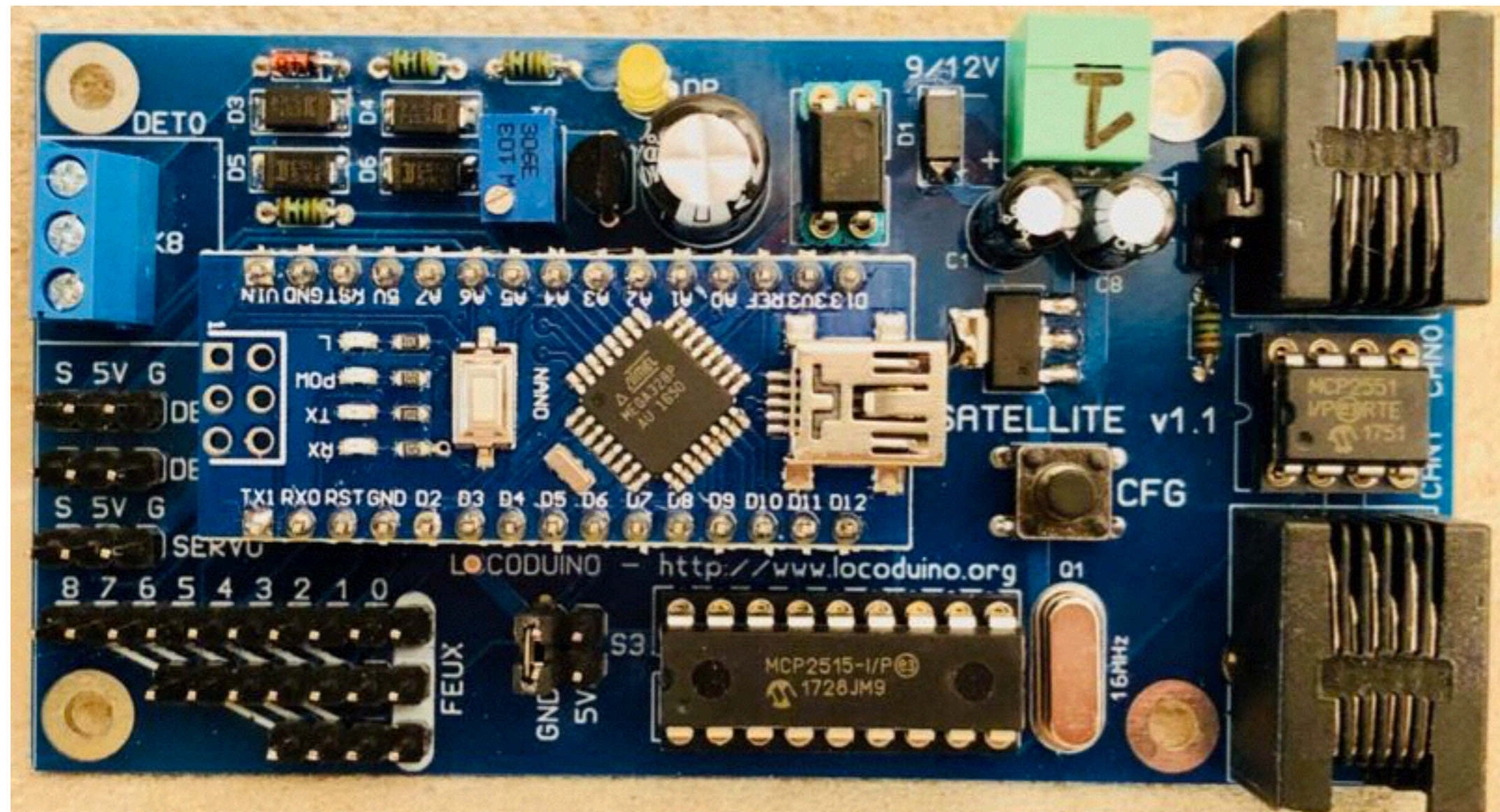
# Bus CAN et Arduino

- Réaliser sa propre interface CAN



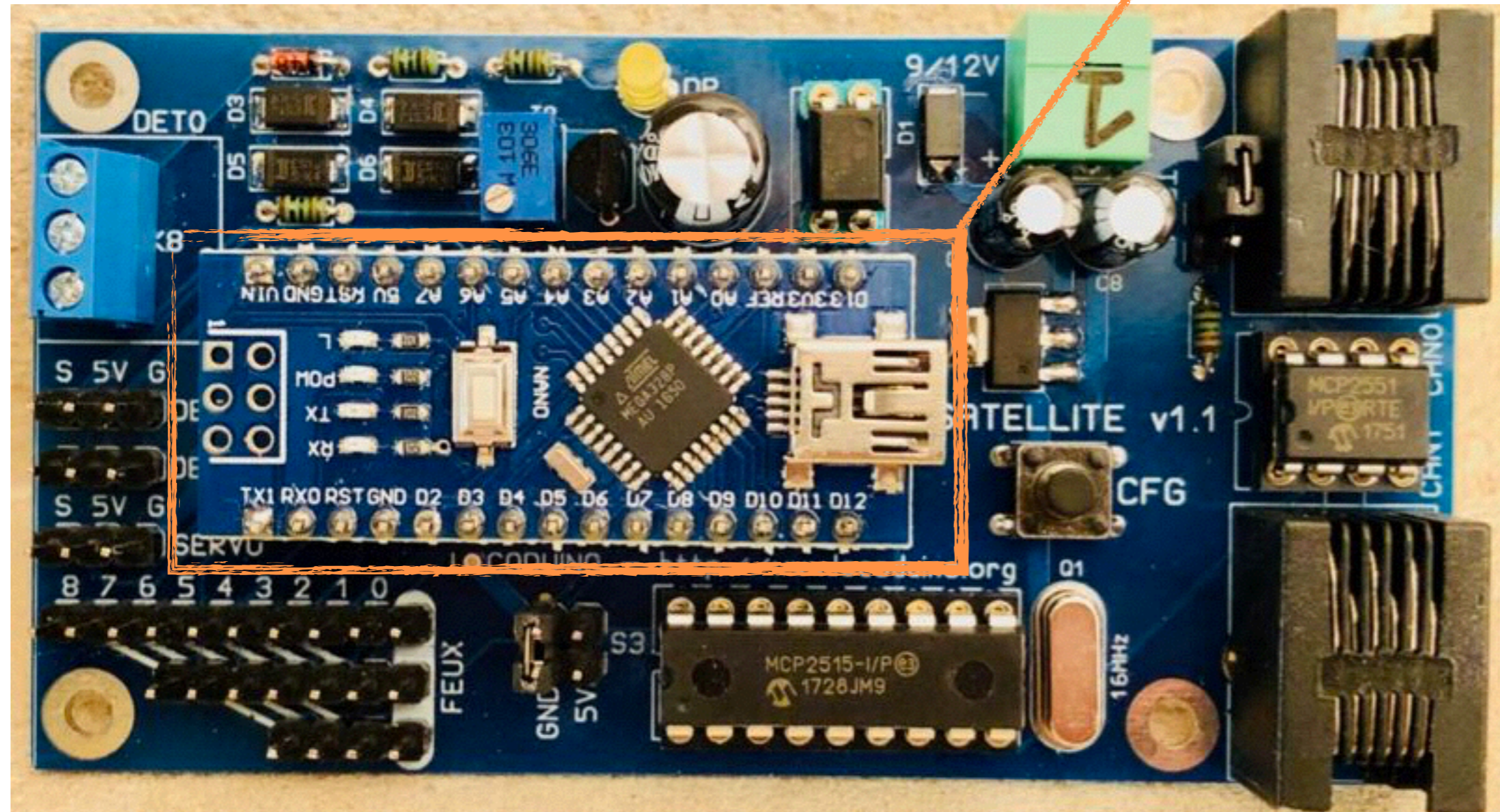
# La carte Satellite V1

- Carte universelle architecturée autour d'un Arduino Nano



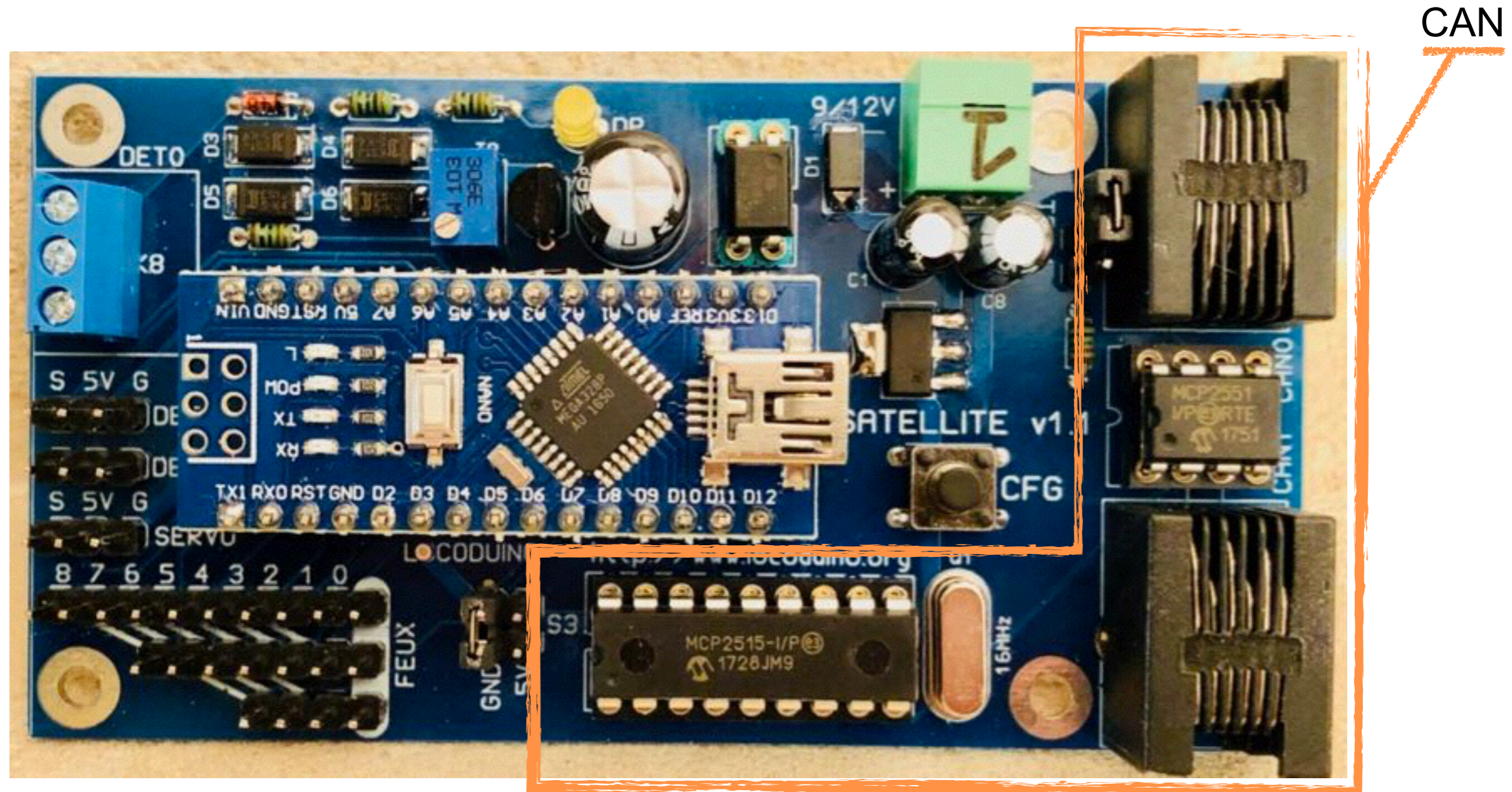
# La carte Satellite V1

- Carte universelle architecturée autour d'un Arduino Nano



# La carte Satellite V1

- Carte universelle architecturée autour d'un Arduino Nano



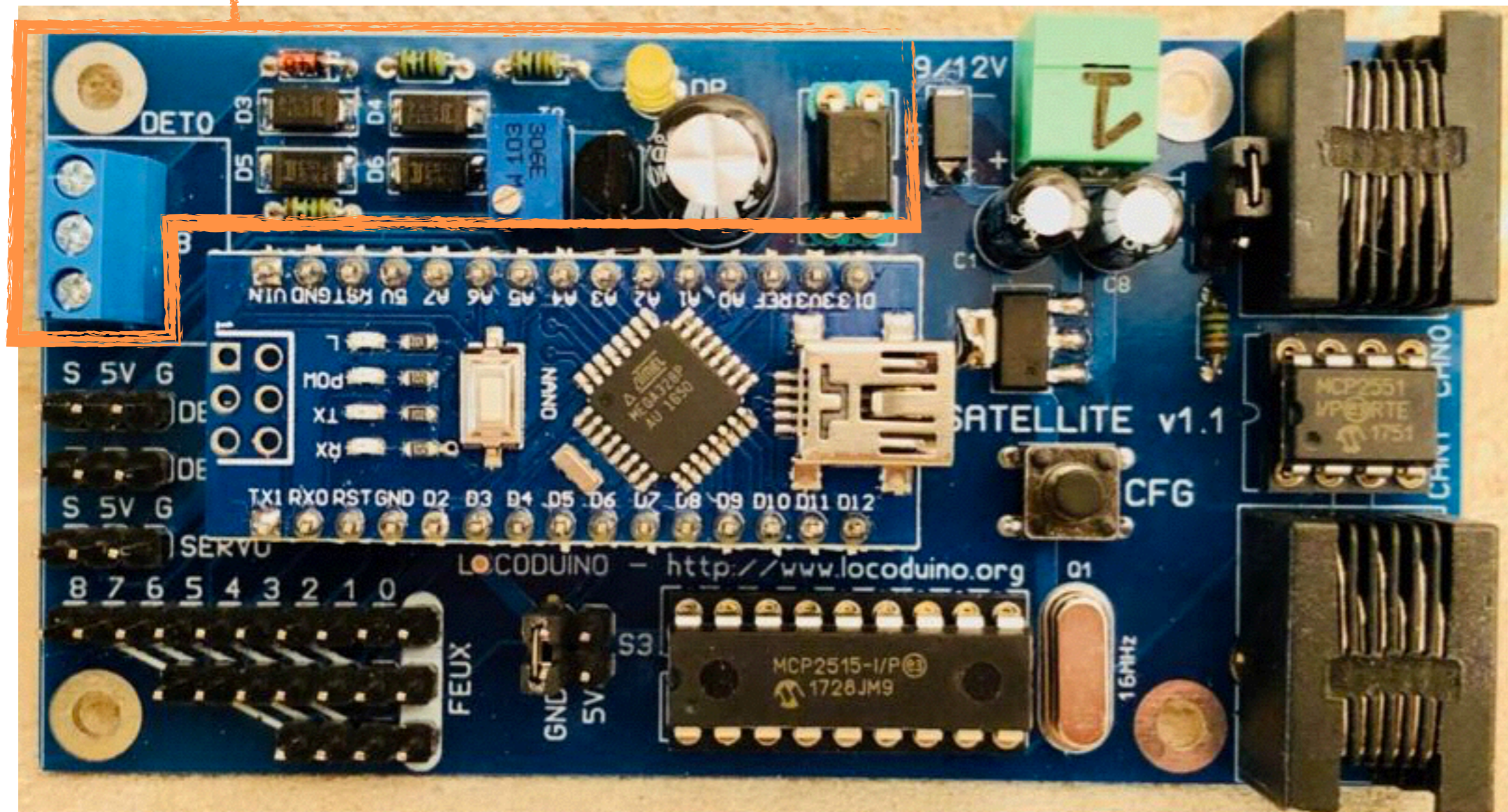
CAN



# La carte Satellite V1

- Carte universelle architecturée autour d'un Arduino Nano

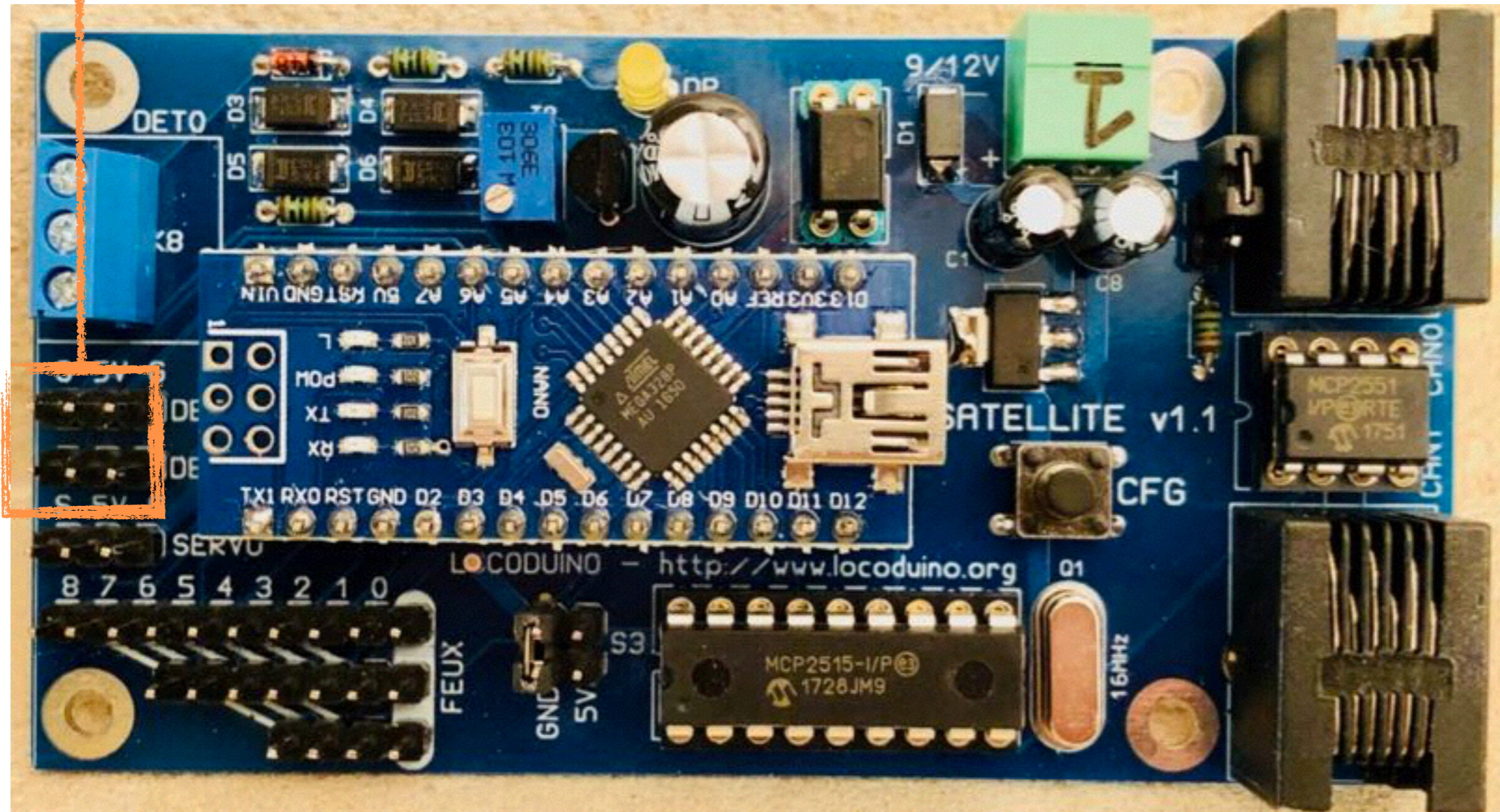
Détection de présence



# La carte Satellite V1

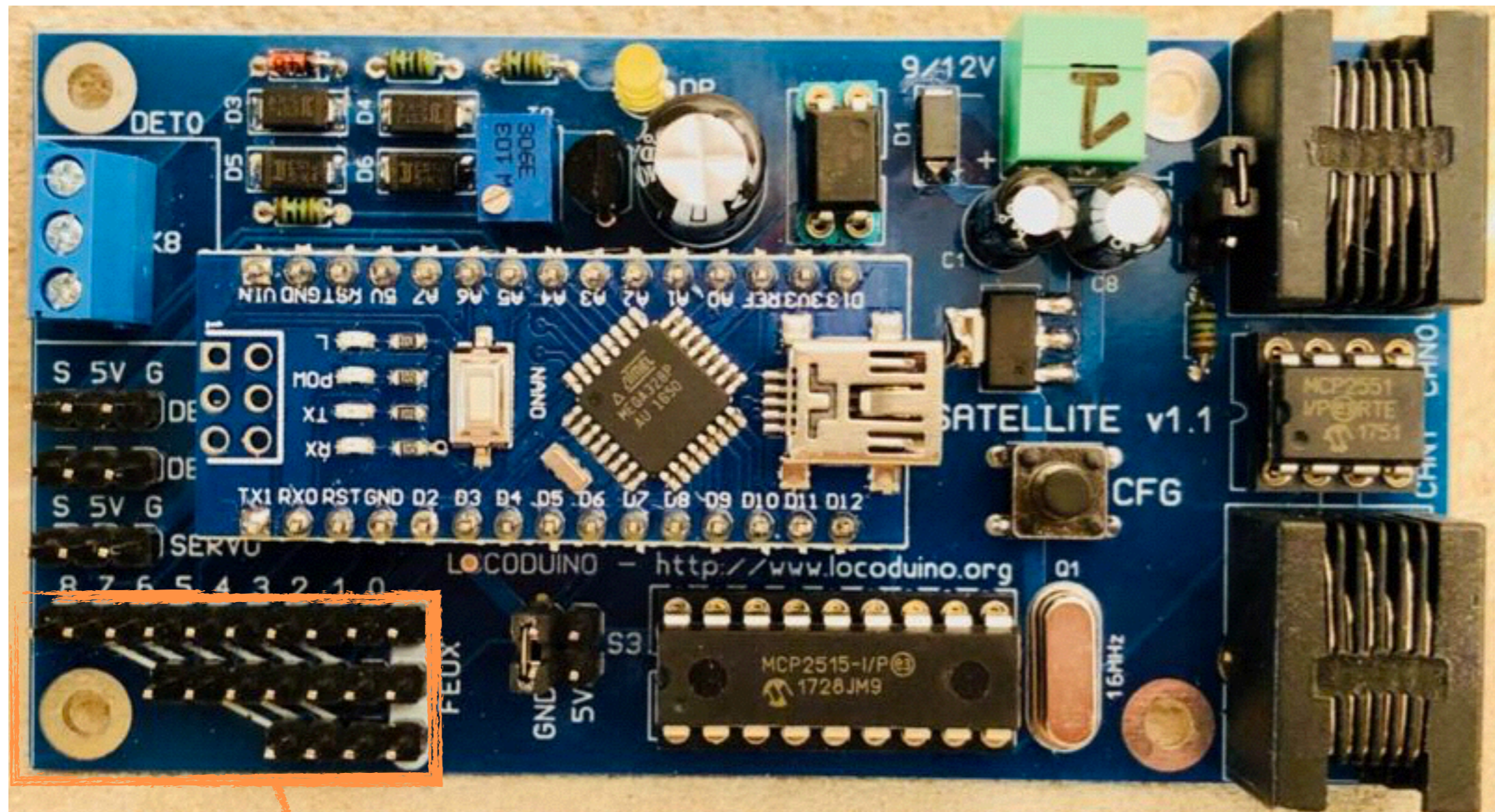
- Carte universelle architecturée autour d'un Arduino Nano

2 détections  
ponctuelles



# La carte Satellite V1

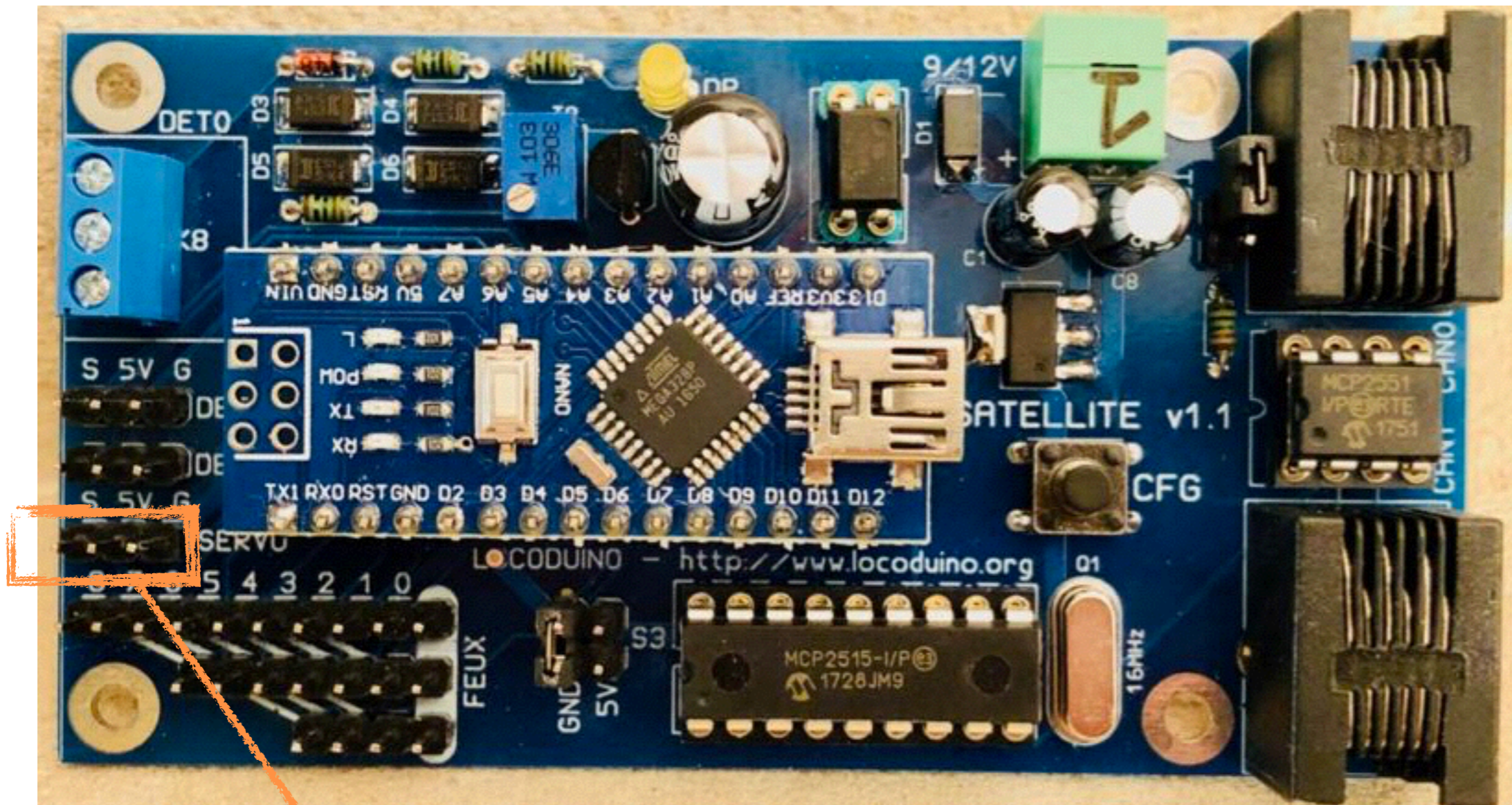
- Carte universelle architecturée autour d'un Arduino Nano



Connecteurs feux

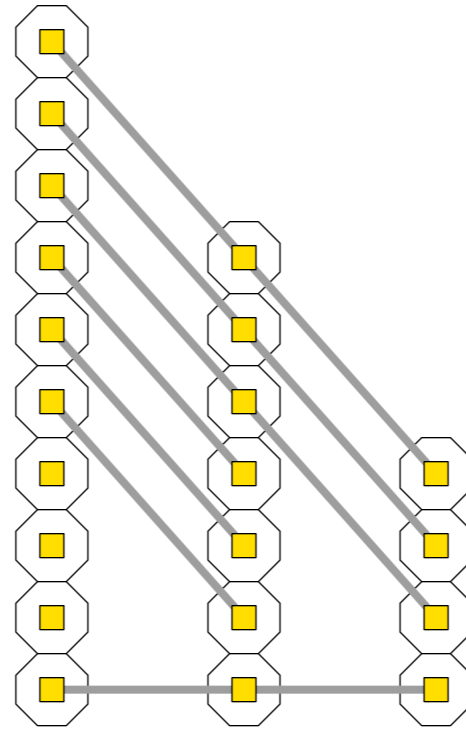
# La carte Satellite V1

- Carte universelle architecturée autour d'un Arduino Nano

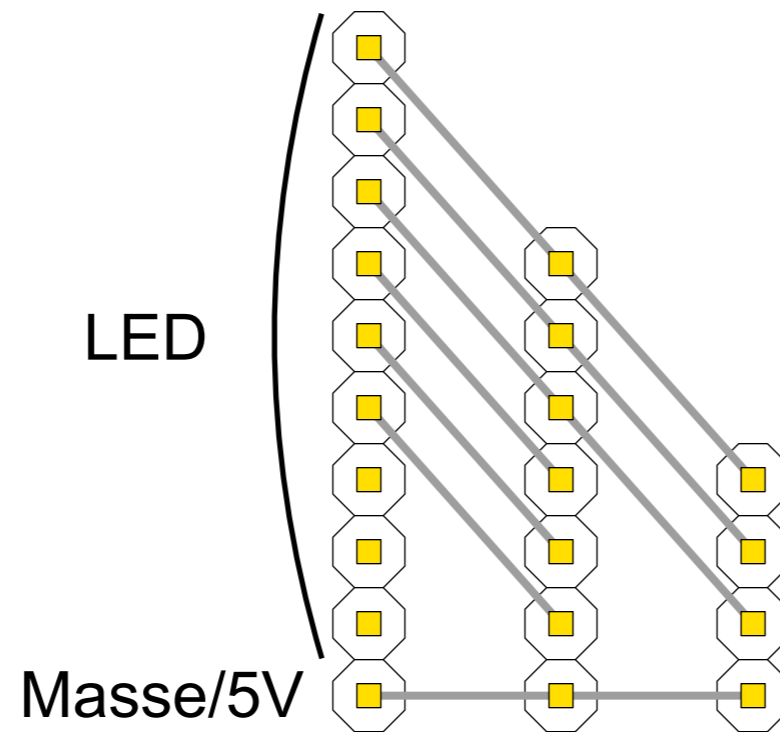


Connecteur Servo

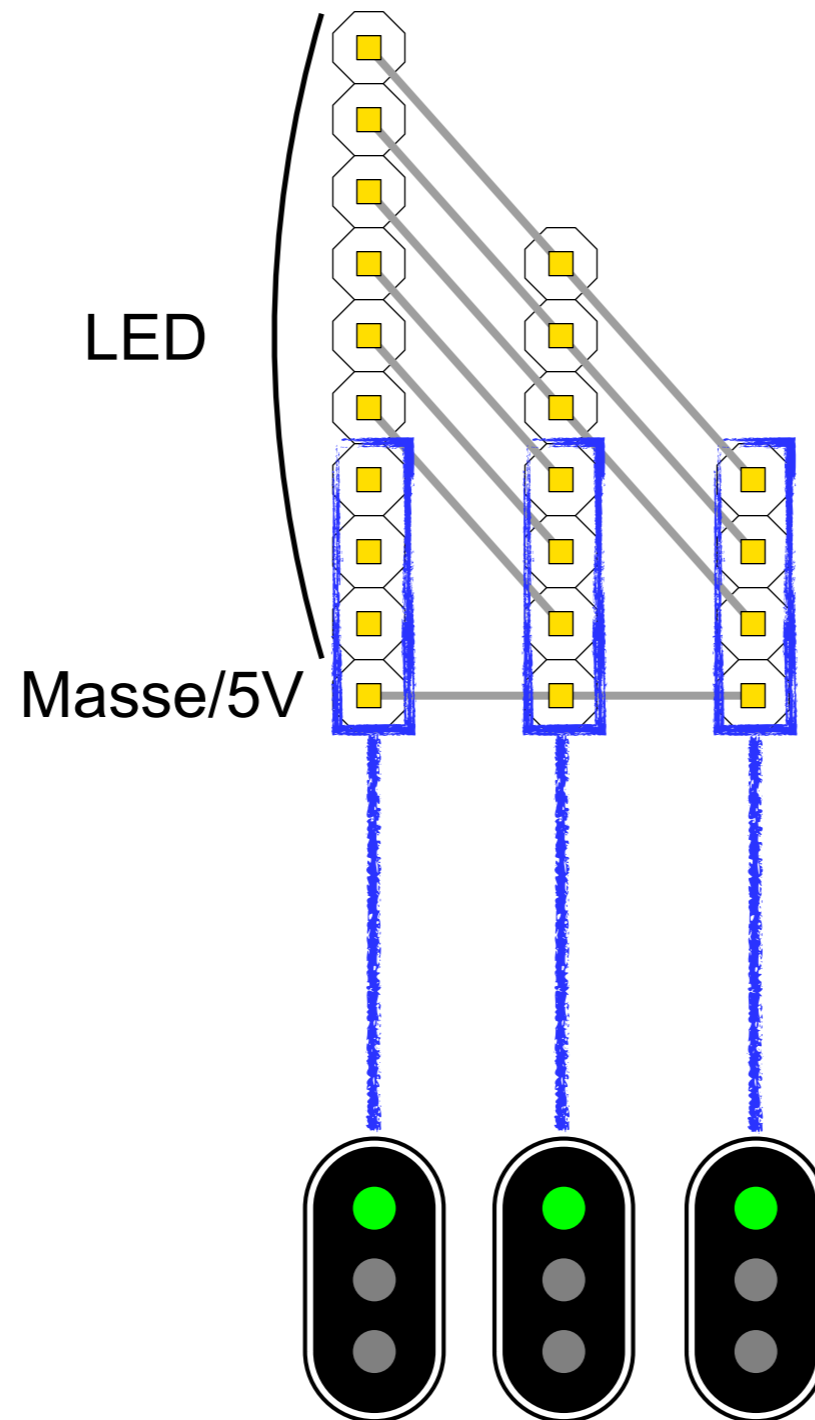
# Les connecteurs de feux



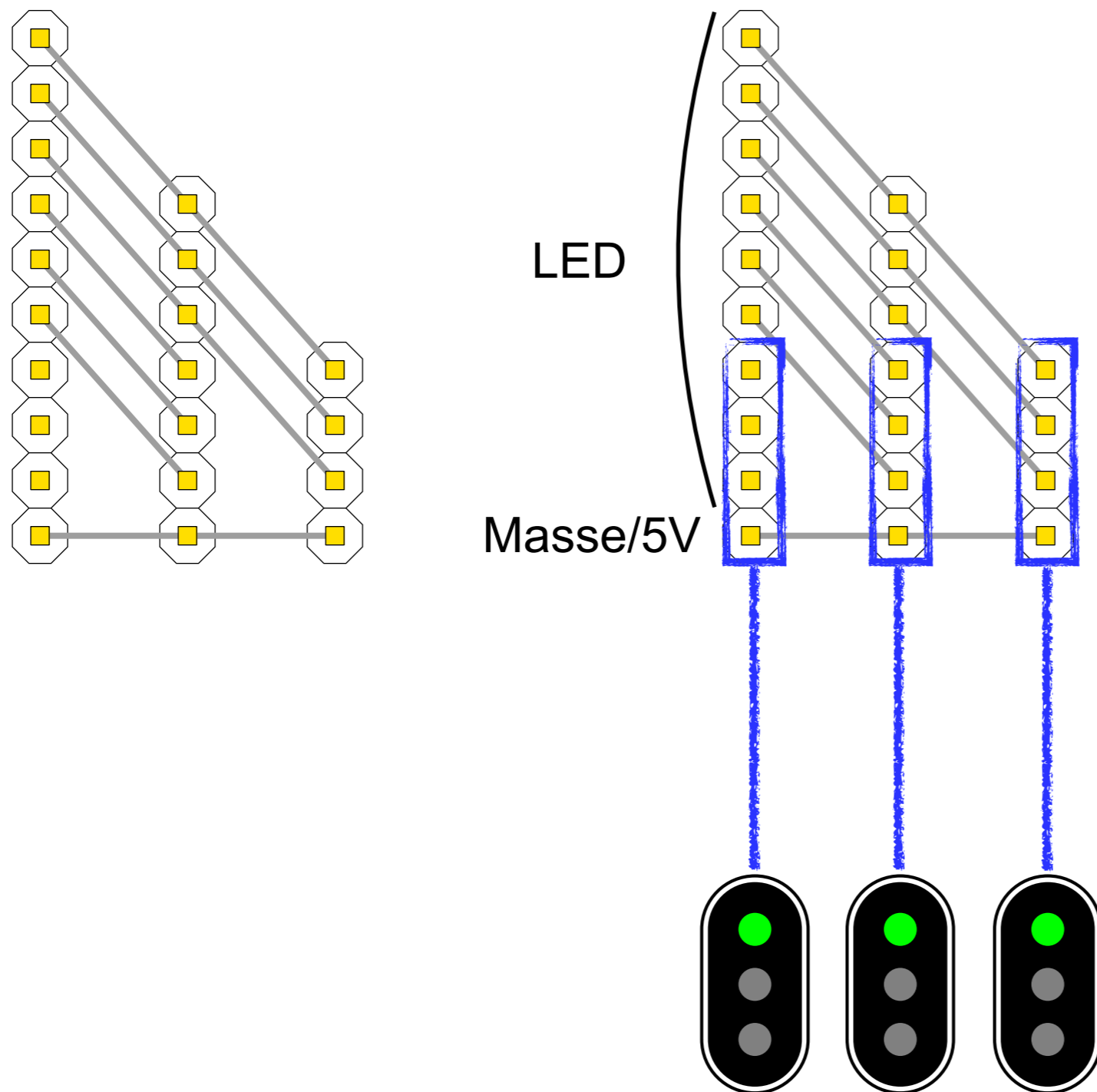
# Les connecteurs de feux



# Les connecteurs de feux

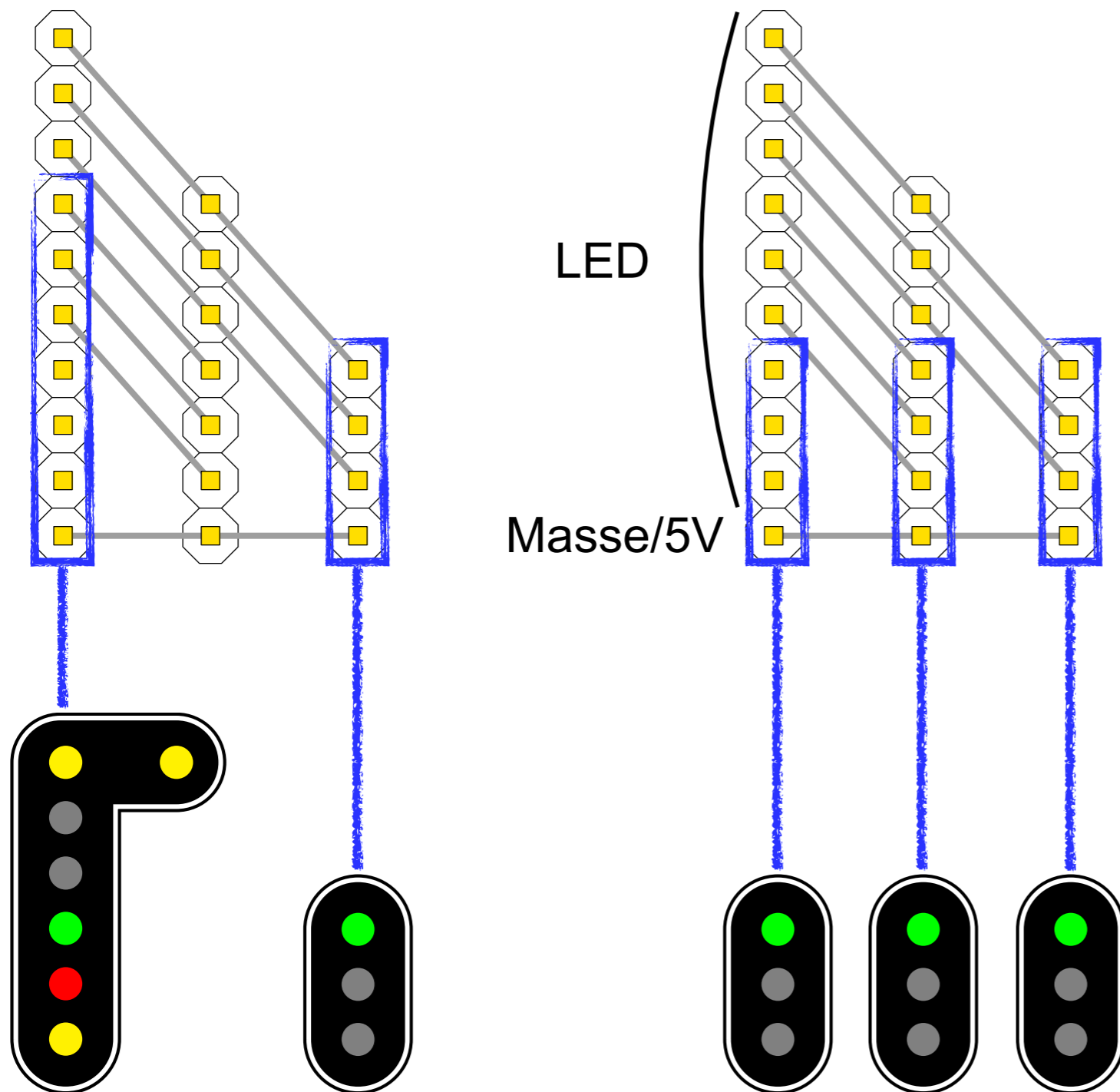


# Les connecteurs de feux

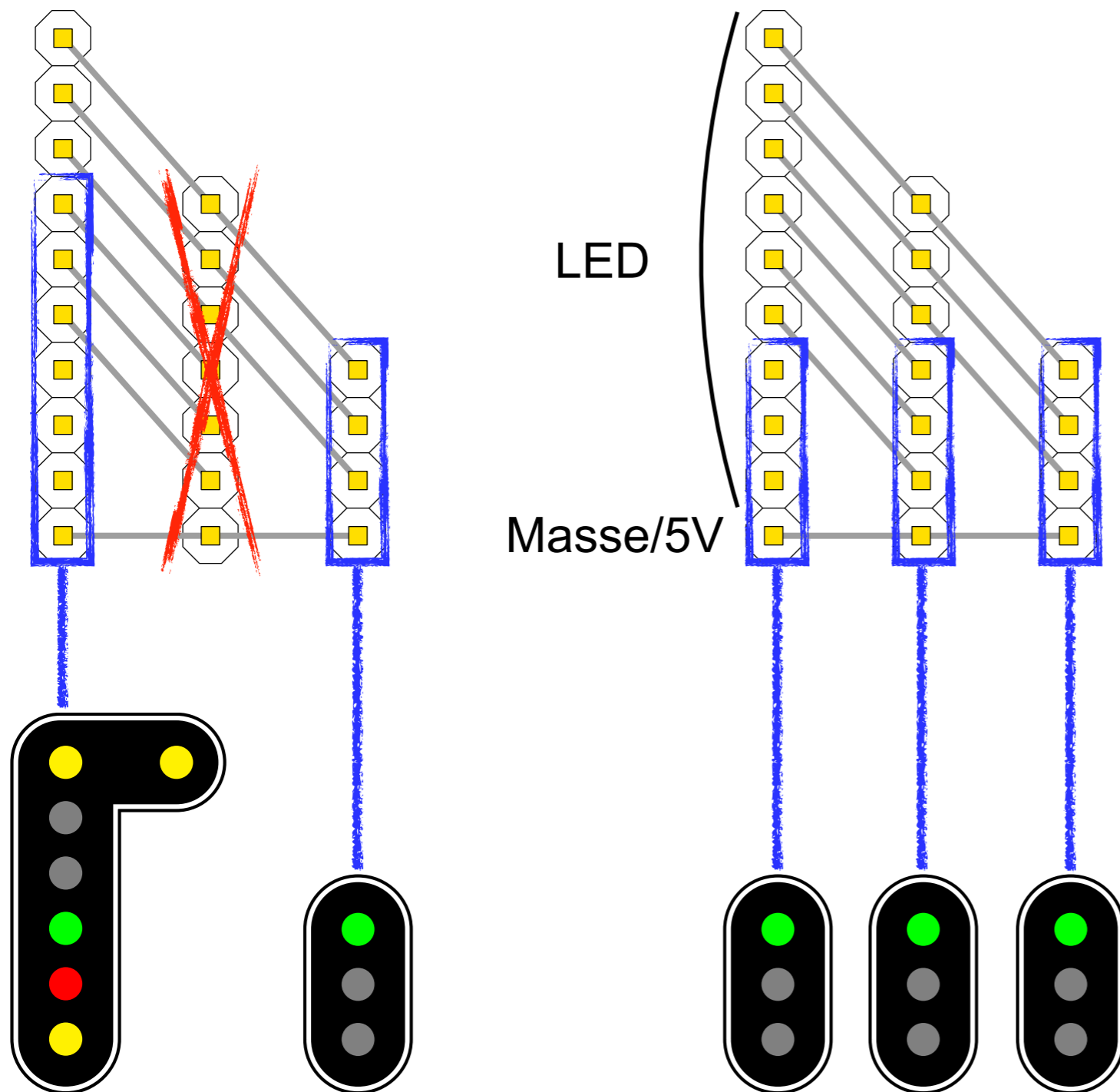




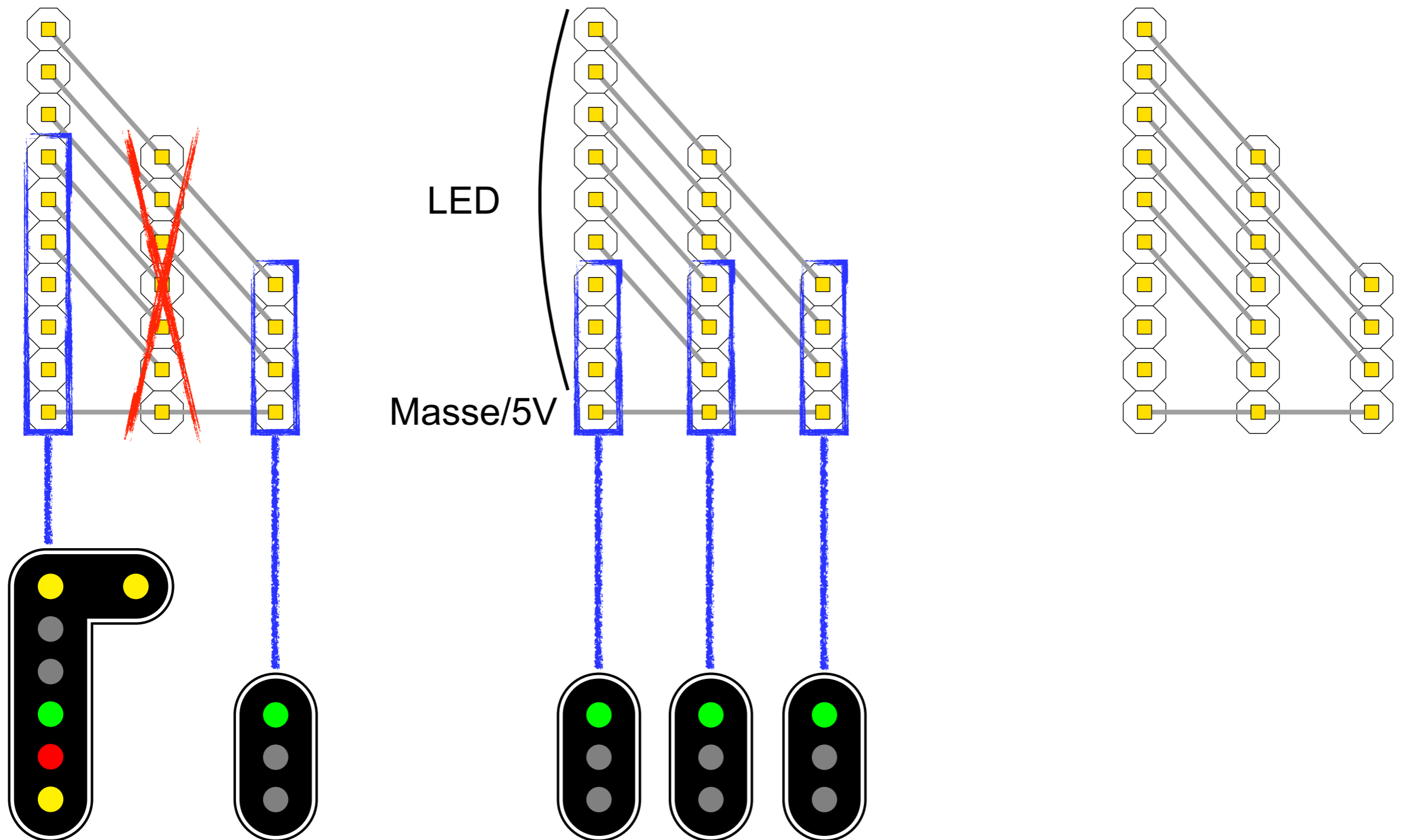
# Les connecteurs de feux



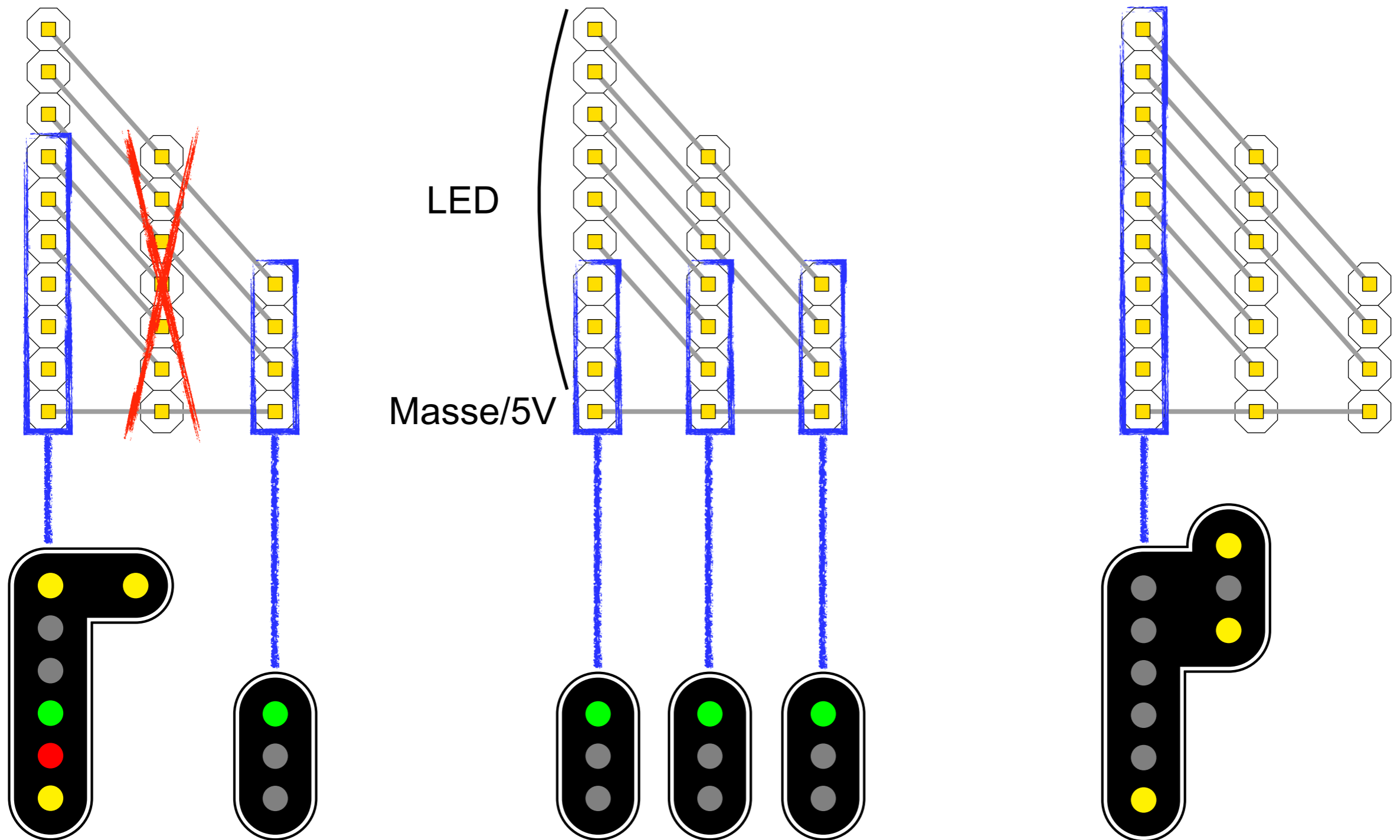
# Les connecteurs de feux



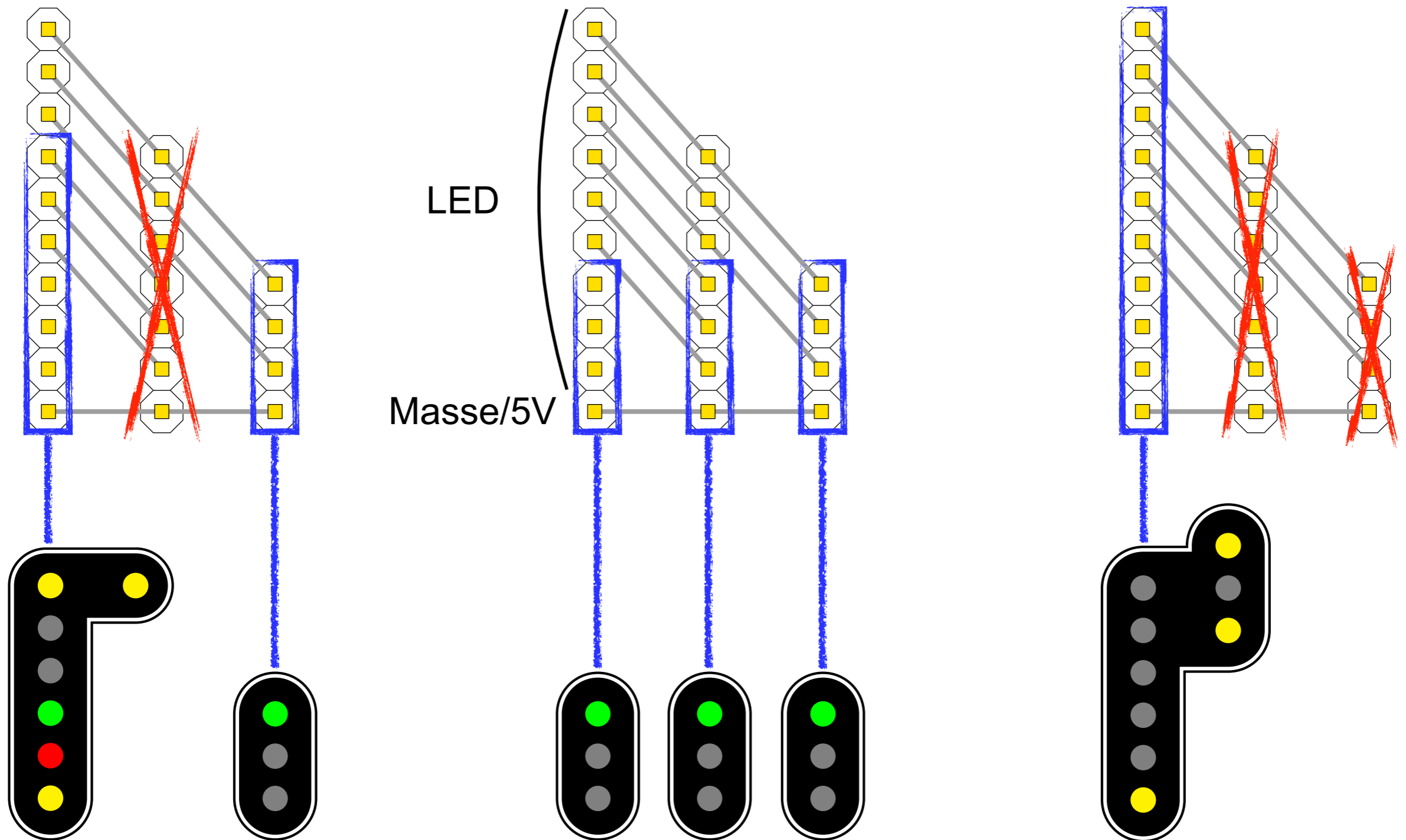
# Les connecteurs de feux



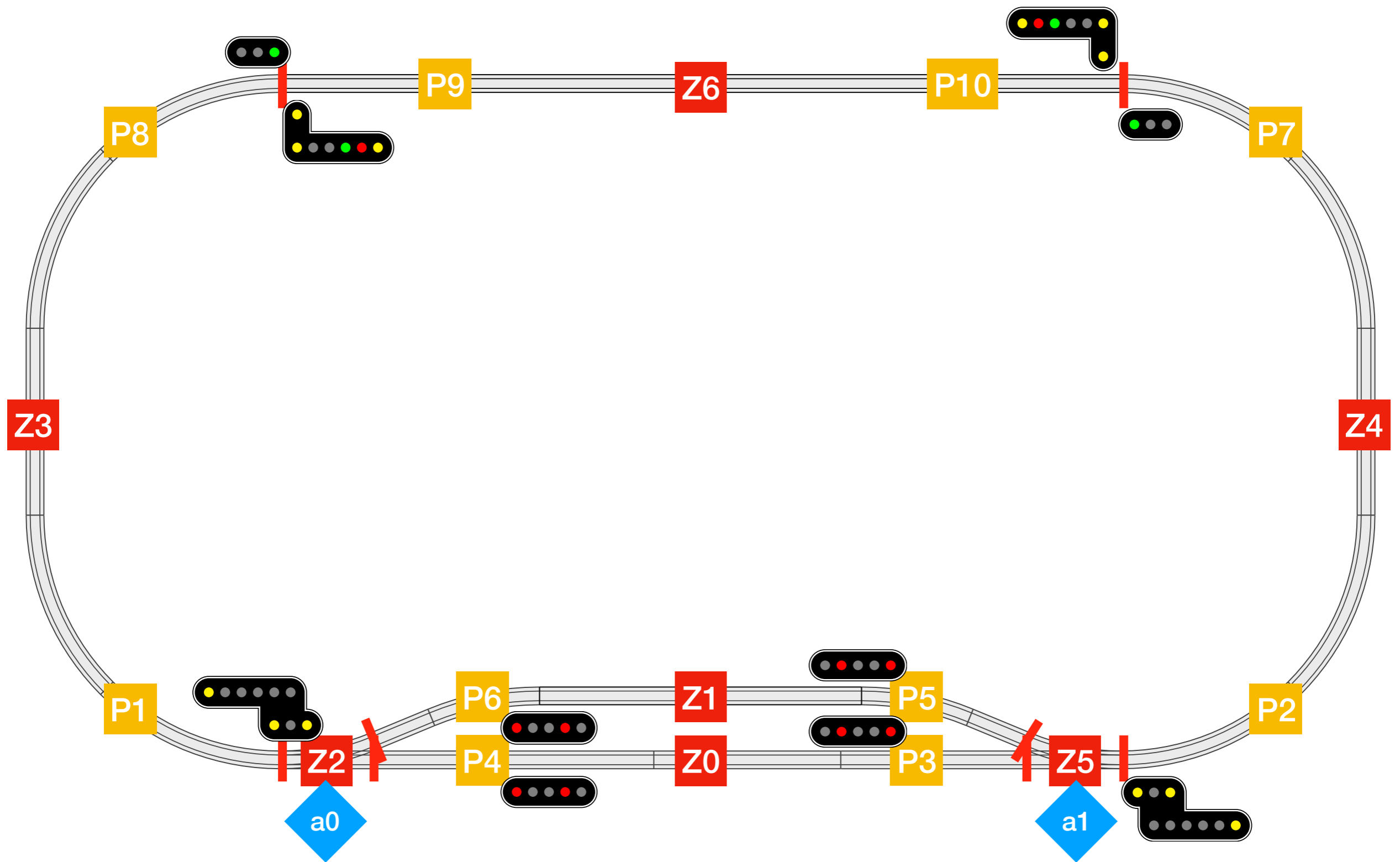
# Les connecteurs de feux



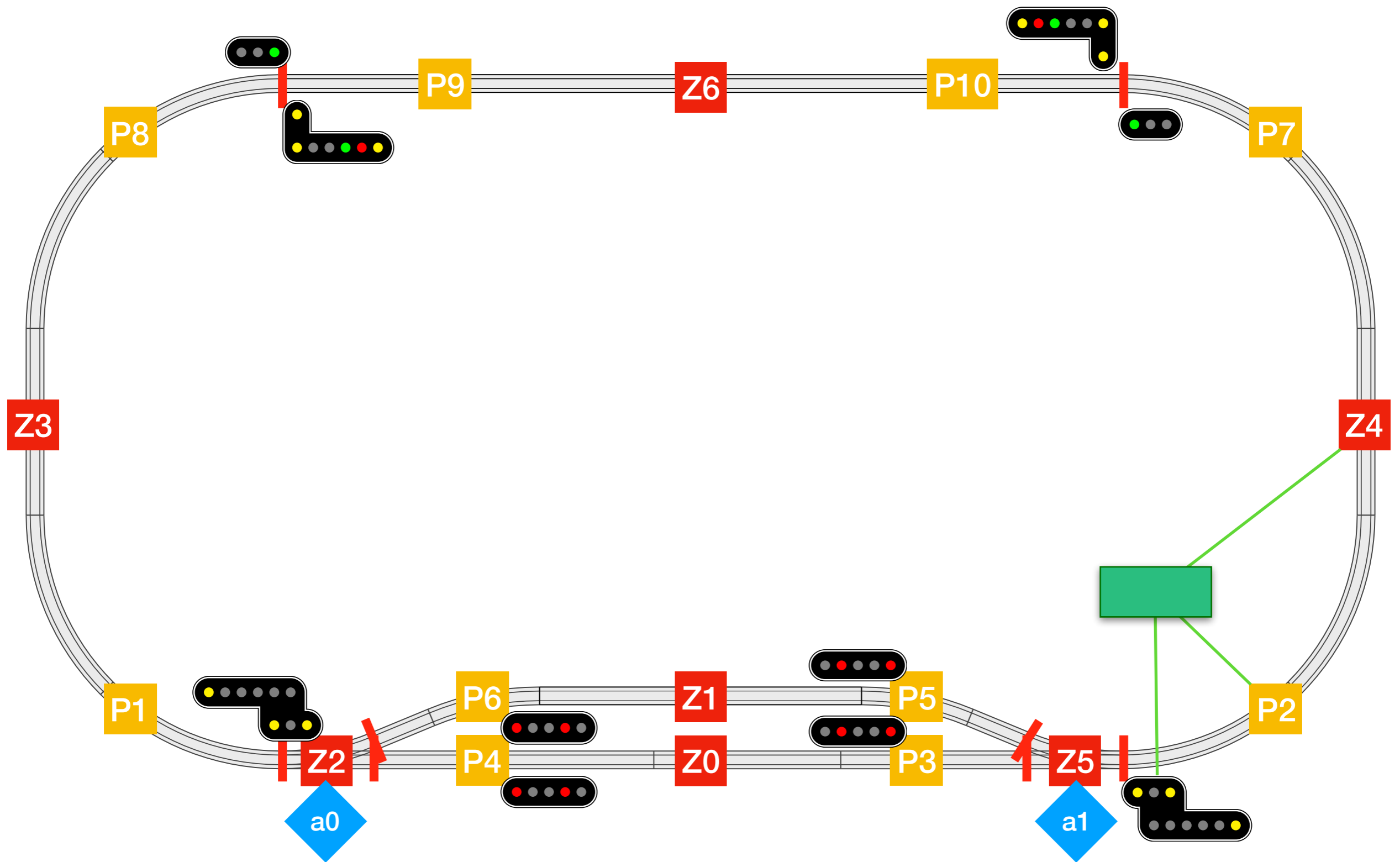
# Les connecteurs de feux



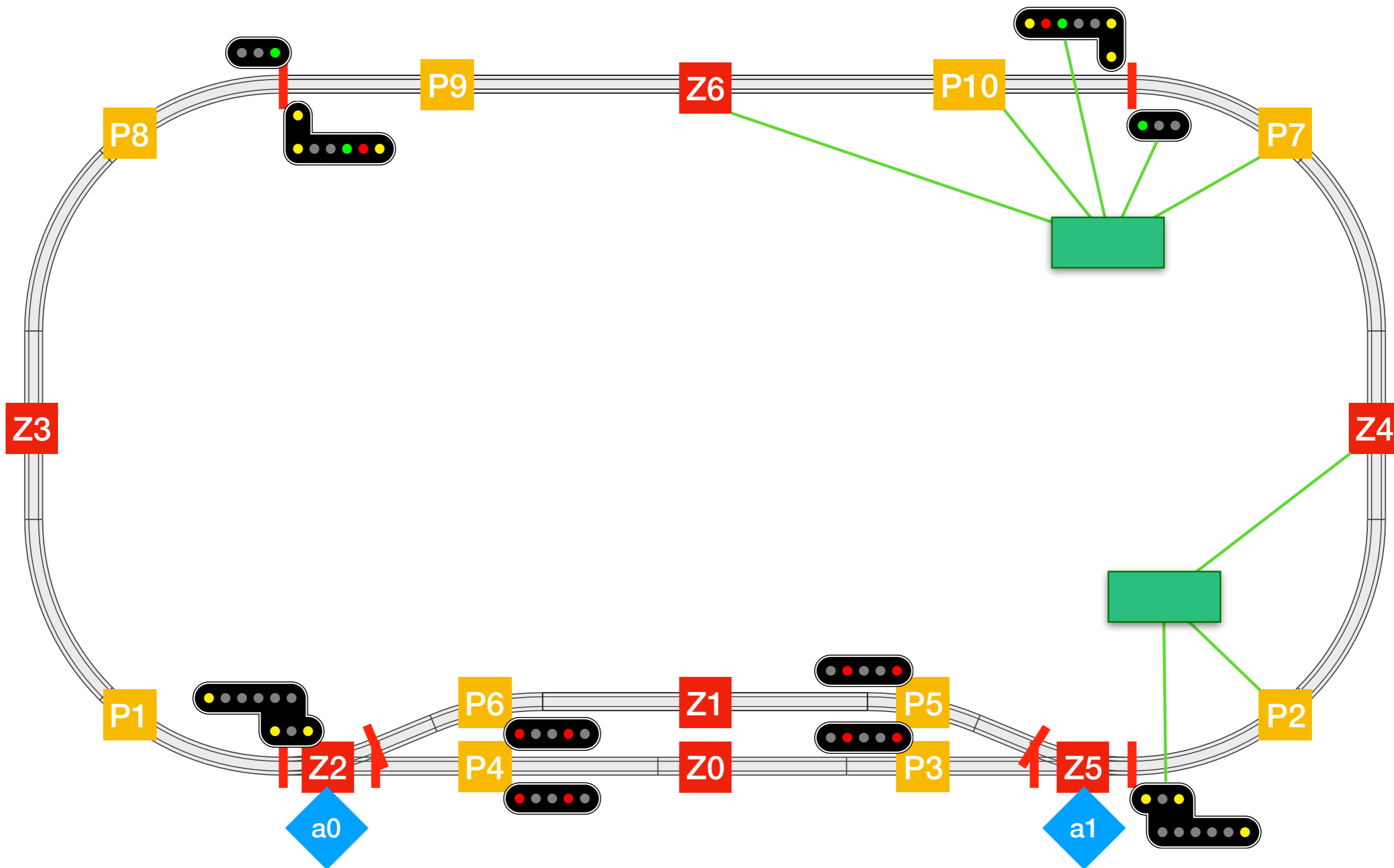
# Pilotage du Locoduinodrome



# Pilotage du Locoduinodrome

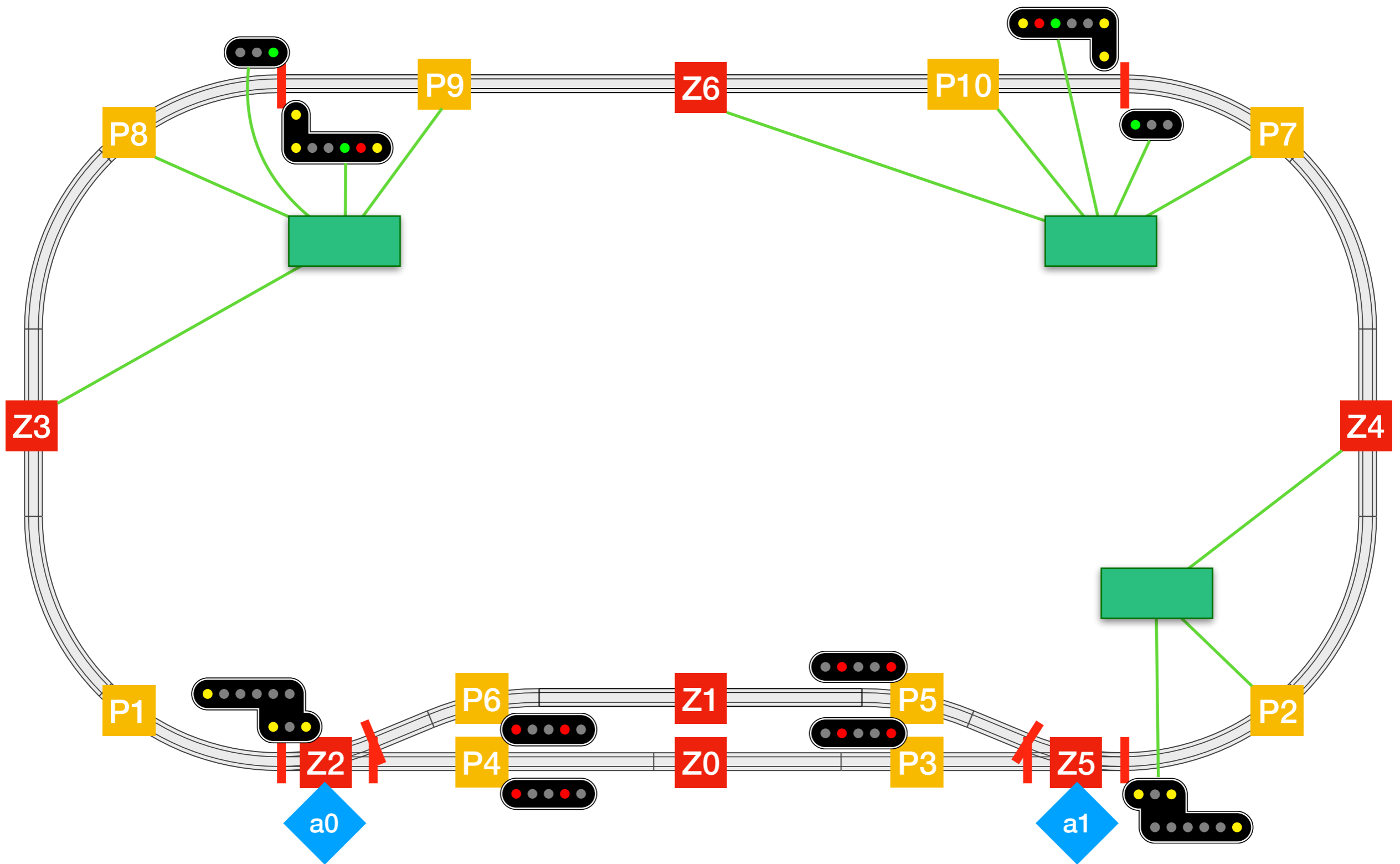


# Pilotage du Locoduinodrome

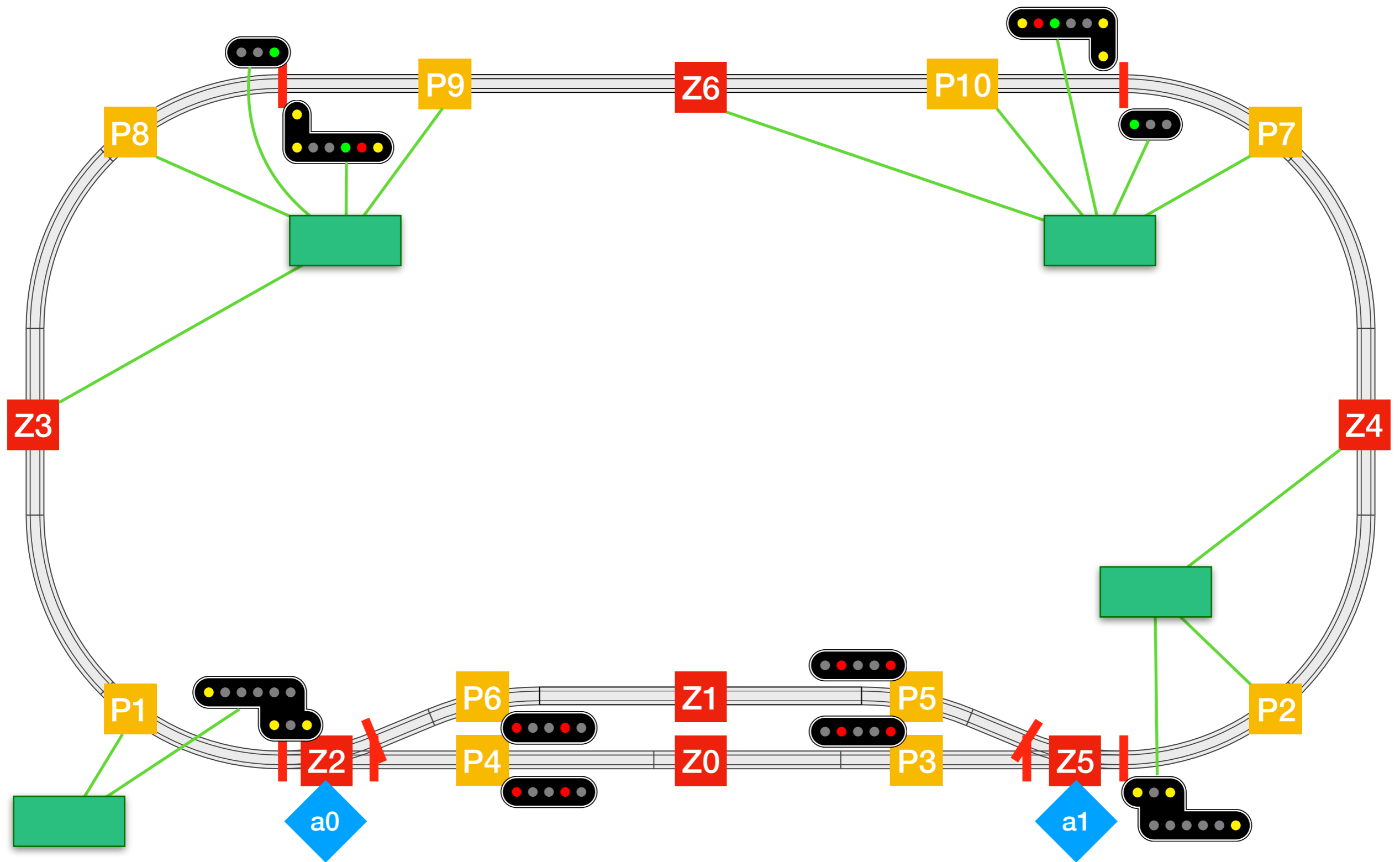




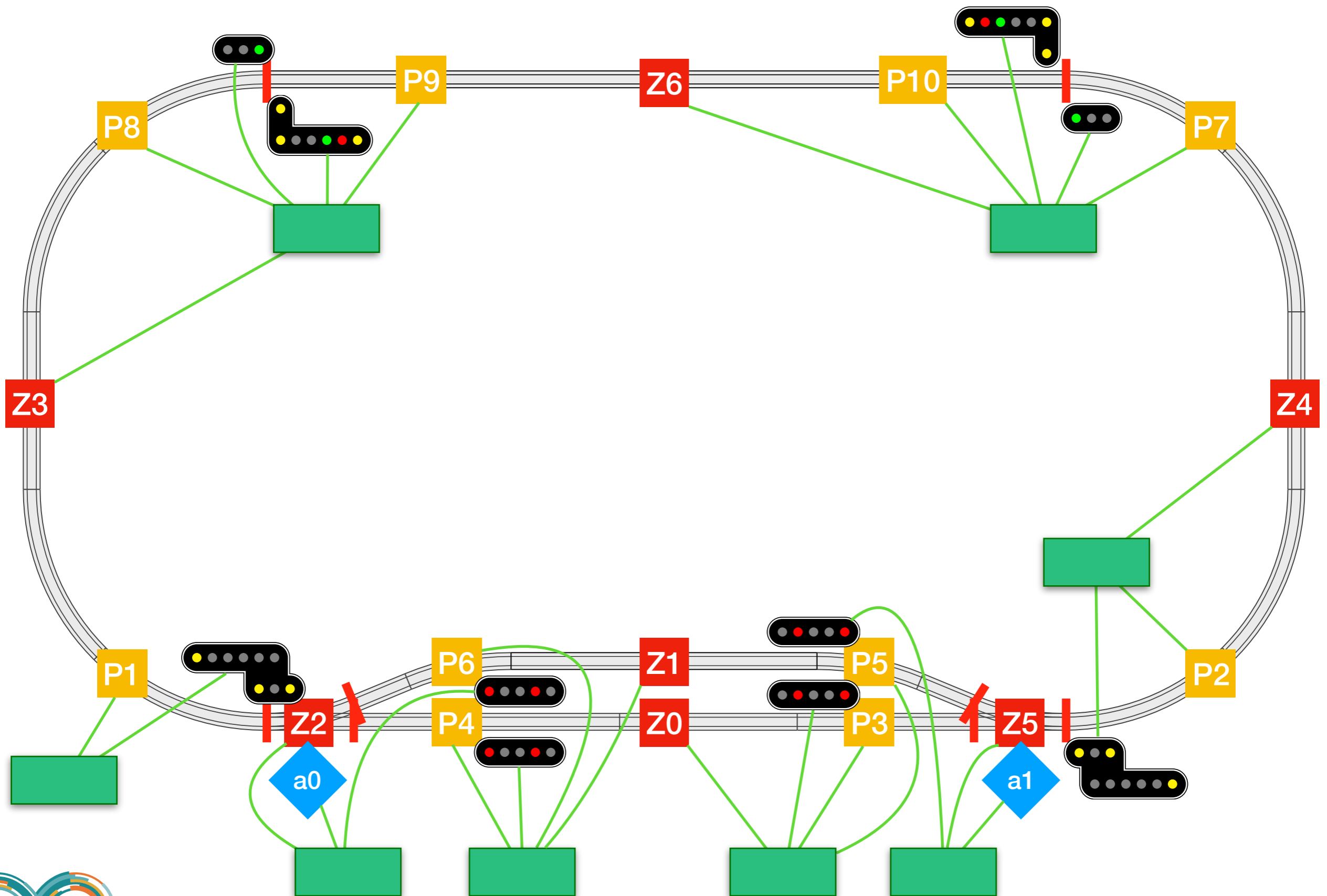
# Pilotage du Locoduinodrome



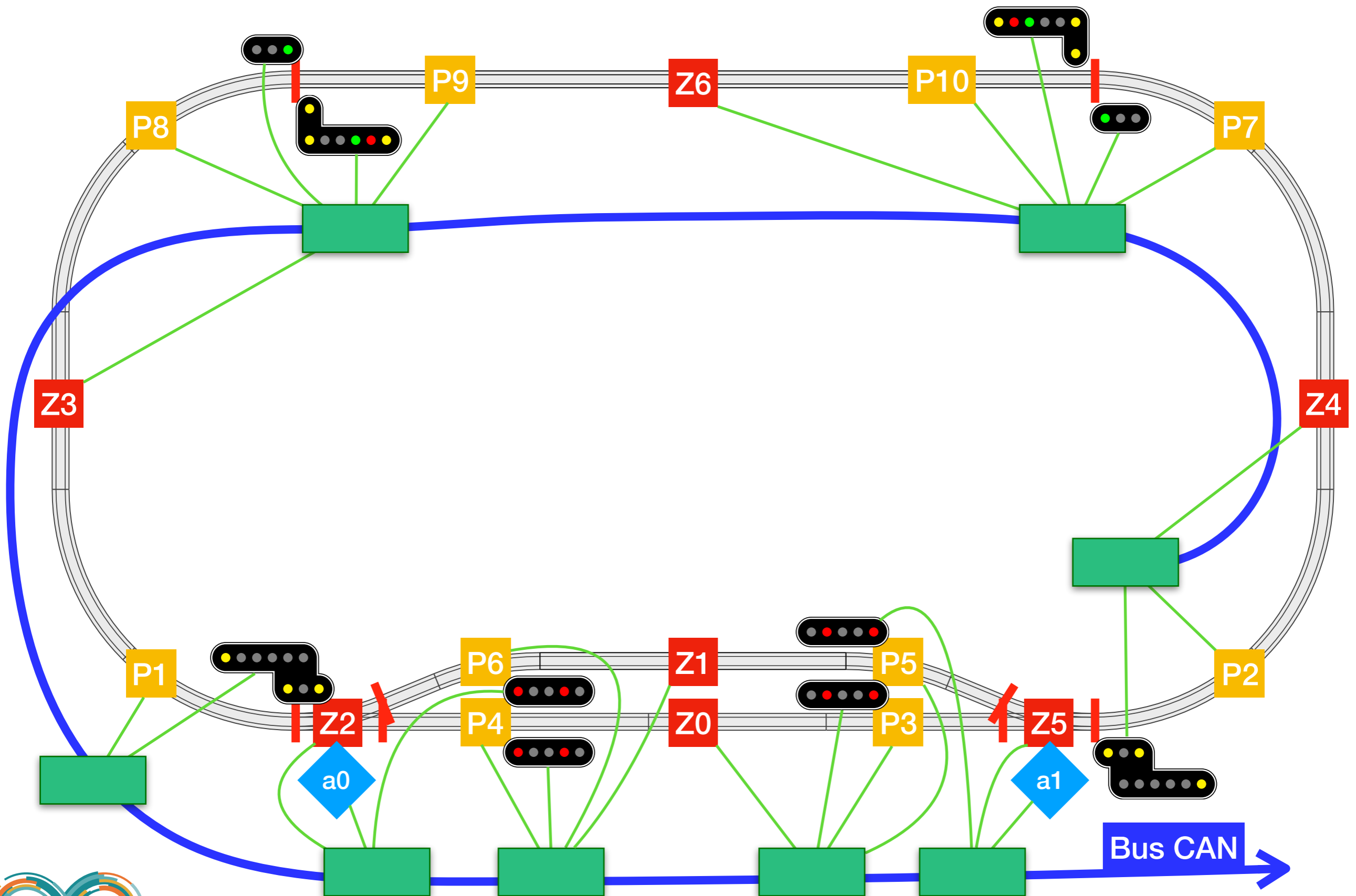
# Pilotage du Locoduinodrome



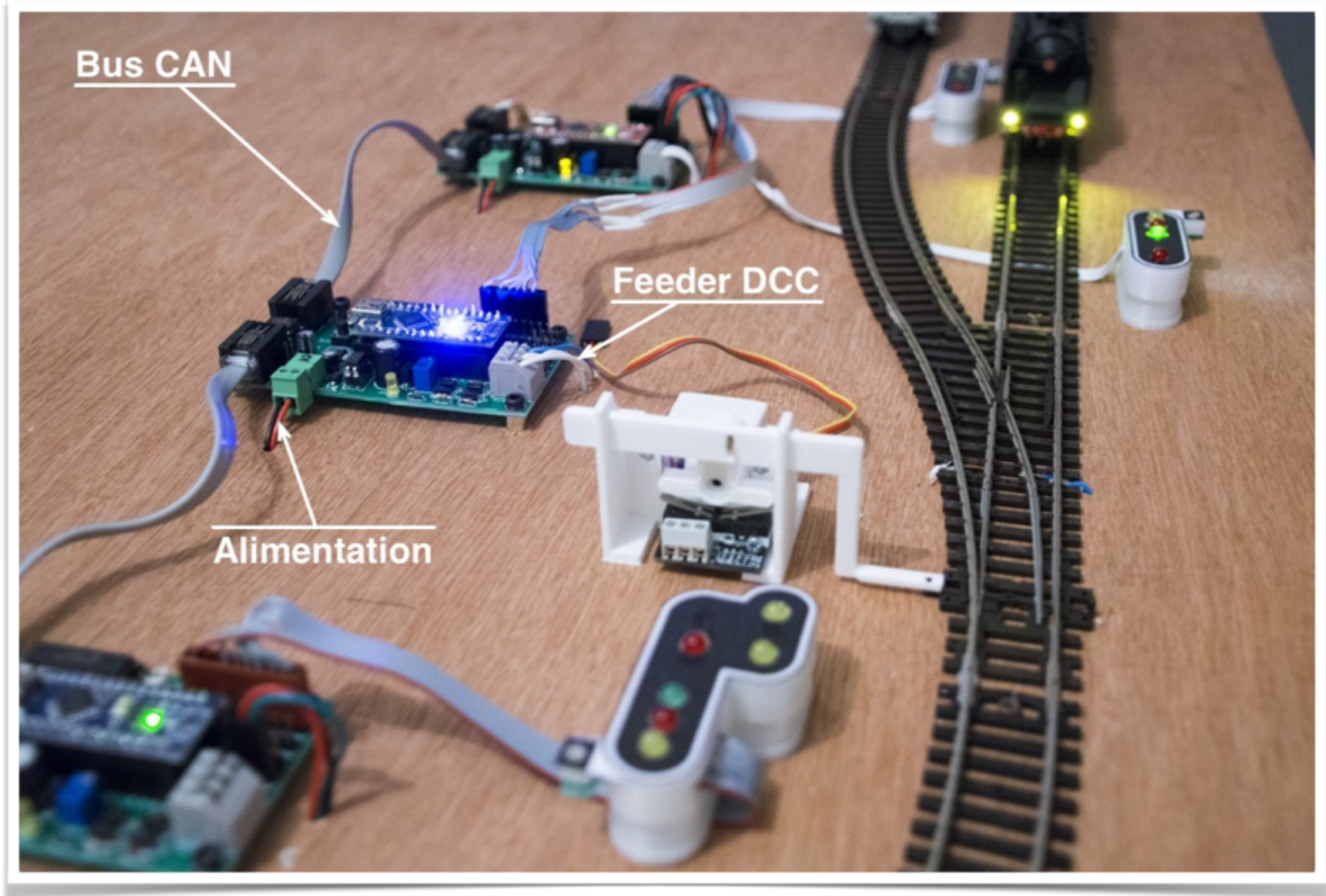
# Pilotage du Locoduinodrome



# Pilotage du Locoduinodrome



# Le Locoduinodrome



# Messagerie CAN

- **Événements vs états** : deux modes sont possibles
  - **événements** : les messages correspondent à des ordres ou des changements d'état.
  - **états** : les messages correspondent à la diffusion de l'état partiel du système par chacune de ses composantes.

# Messagerie CAN

- **Événements vs états** : deux modes sont possibles
  - **événements** : les messages correspondent à des changements d'état.
  - **états** : les messages correspondent à la diffusion de l'état partiel du système par chacune de ses composantes.

Perte de message ?  
Redémarrage d'une carte ?

# Messagerie CAN

- **Événements vs états** : deux modes sont possibles
  - **événements** : les messages correspondent à des ordres ou des changements d'état.
  - **états** : les messages correspondent à la diffusion d'un état partiel du système par chacune de ses composantes.

Perte de message  
ou redémarrage  
sans importance



# Messagerie CAN

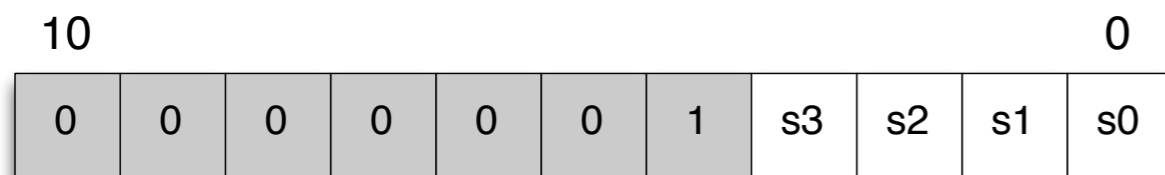
- **Événements vs états** : deux modes sont possibles
  - **événements** : les messages correspondent à des ordres ou des changements d'état.
  - **états** : les messages correspondent à la diffusion d'un état partiel du système par chacune de ses composantes.
- Les messages échangés sont des messages d'état :
  - états souhaités pour les LED et le servo
  - états actuels des détections
  - transmission périodique

Perte de message  
ou redémarrage  
sans importance

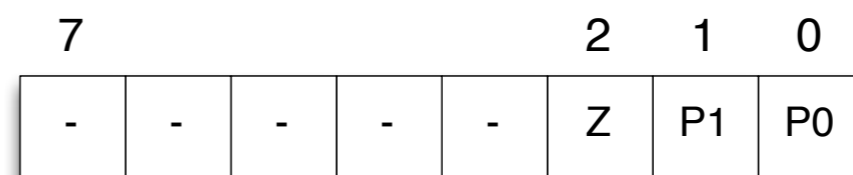
# Messagerie CAN

- Message émis par un satellite.

identifiant



donnée



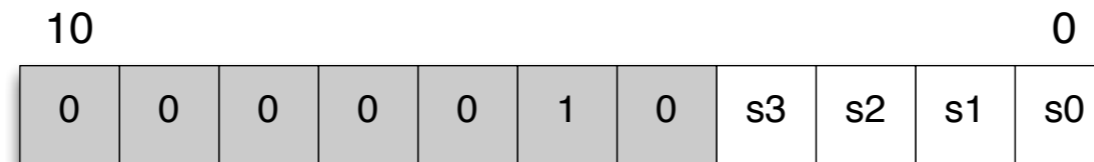
Z	État
0	Zone libre
1	Zone occupée

Px	État
0	Barrière libre
1	Barrière coupée

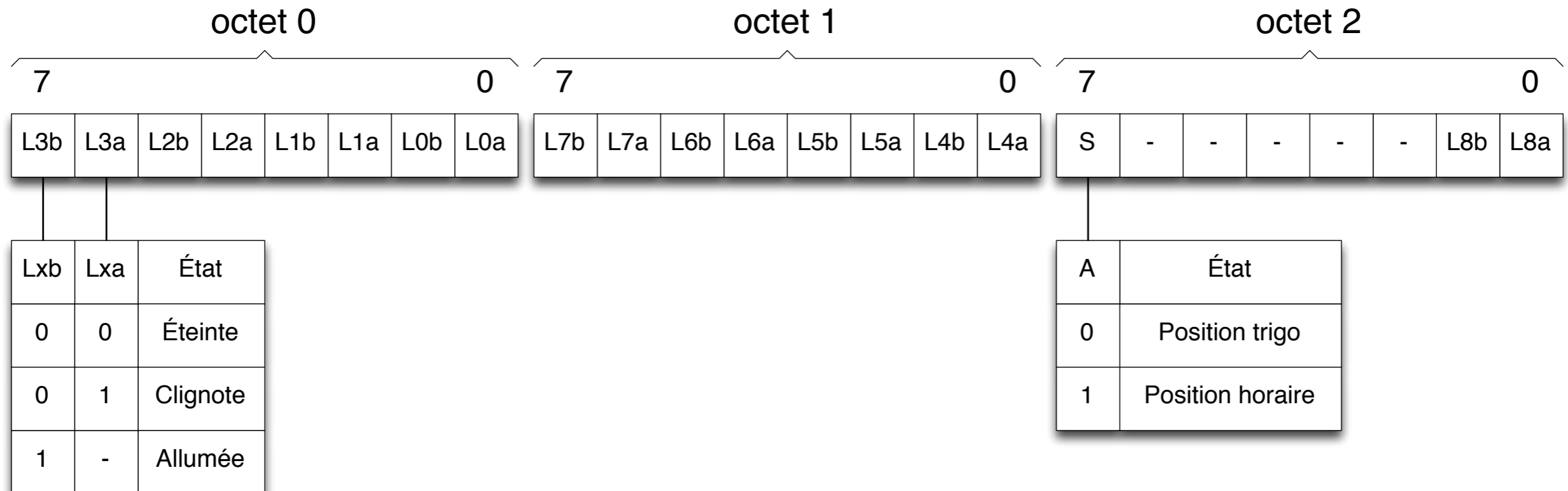
# Messagerie CAN

- Message reçu par un Satellite

identifiant



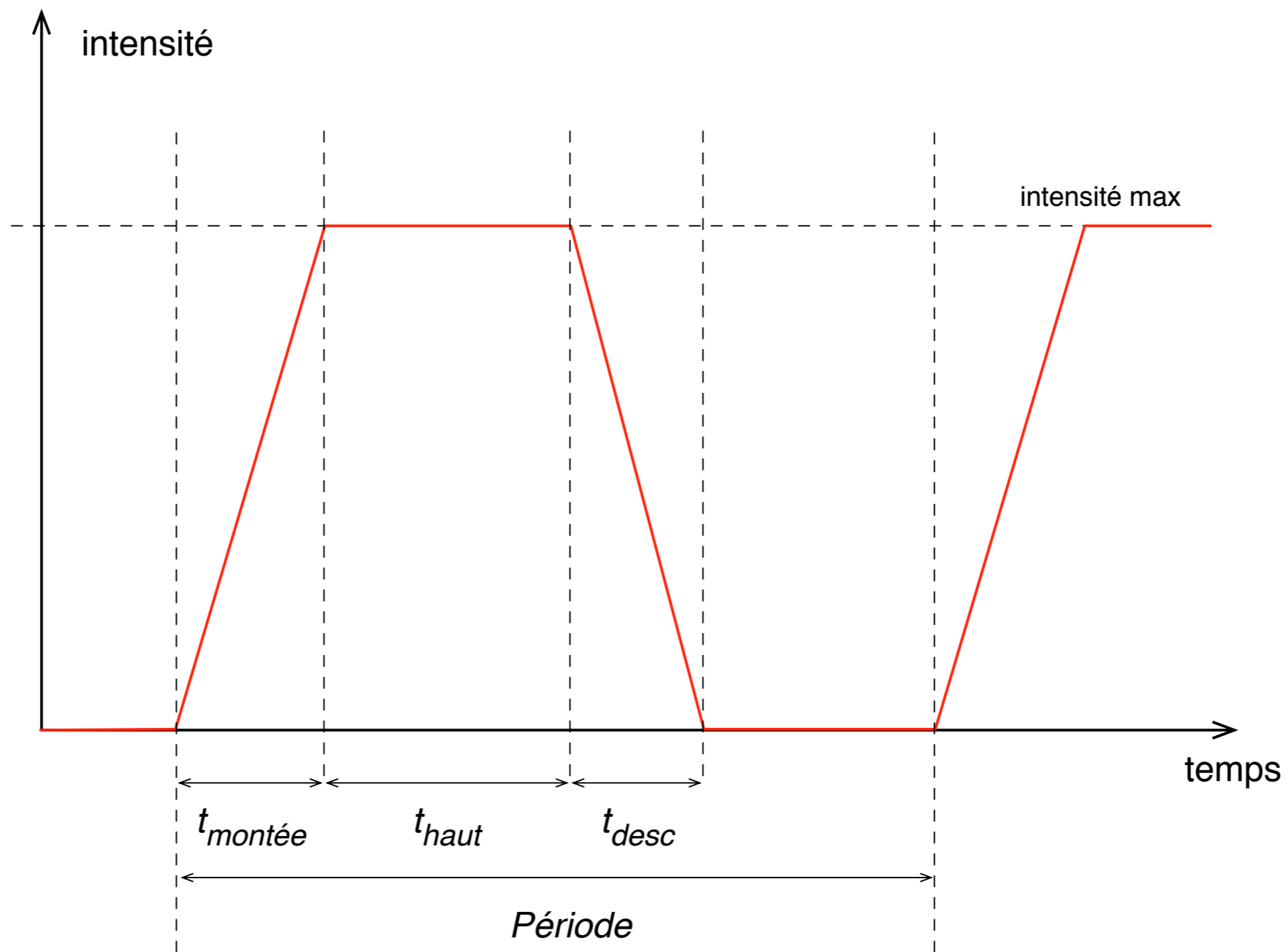
donnée





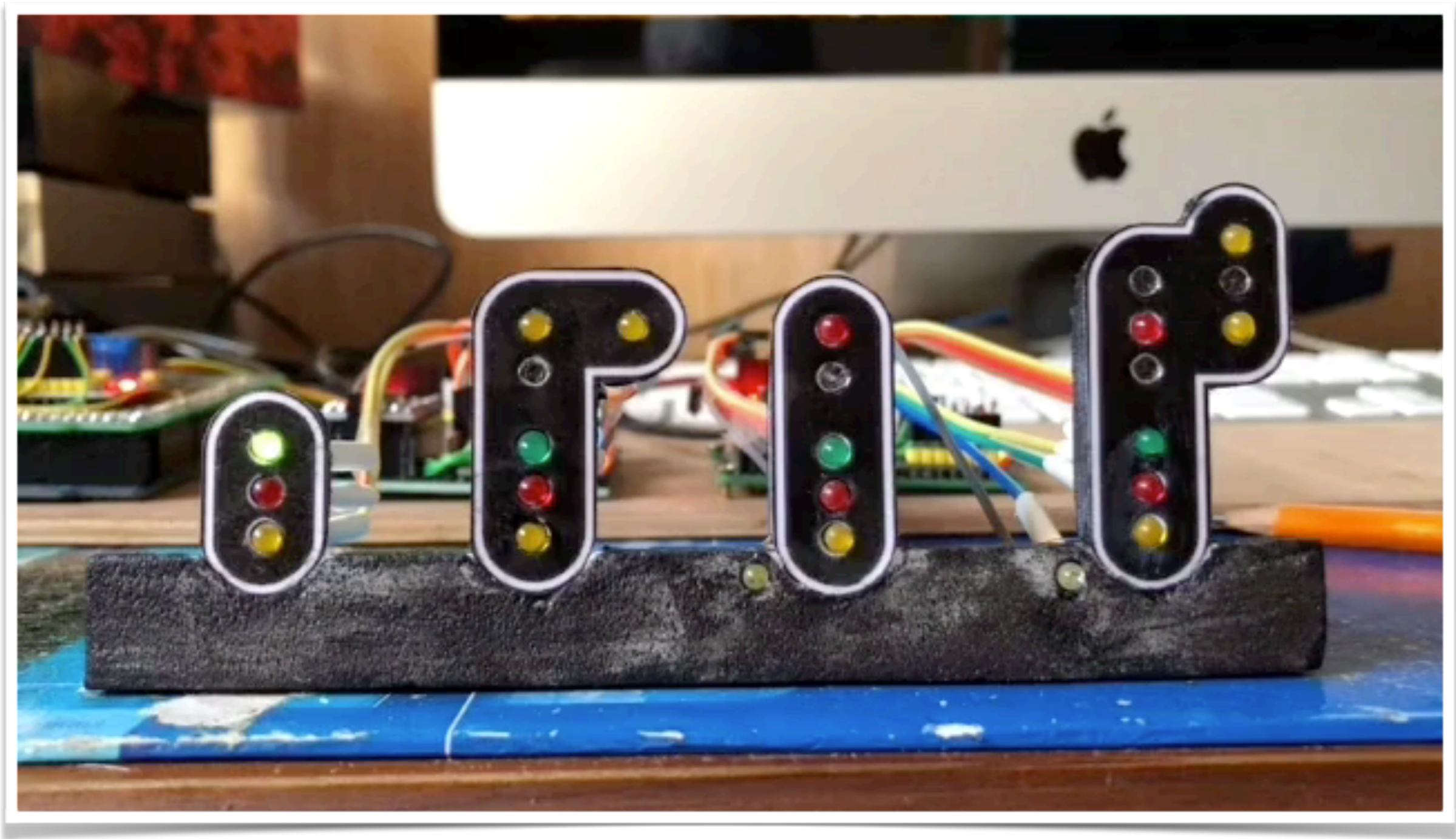
# Logiciel - les feux

- Bibliothèque Locoduino *LightDimmer*



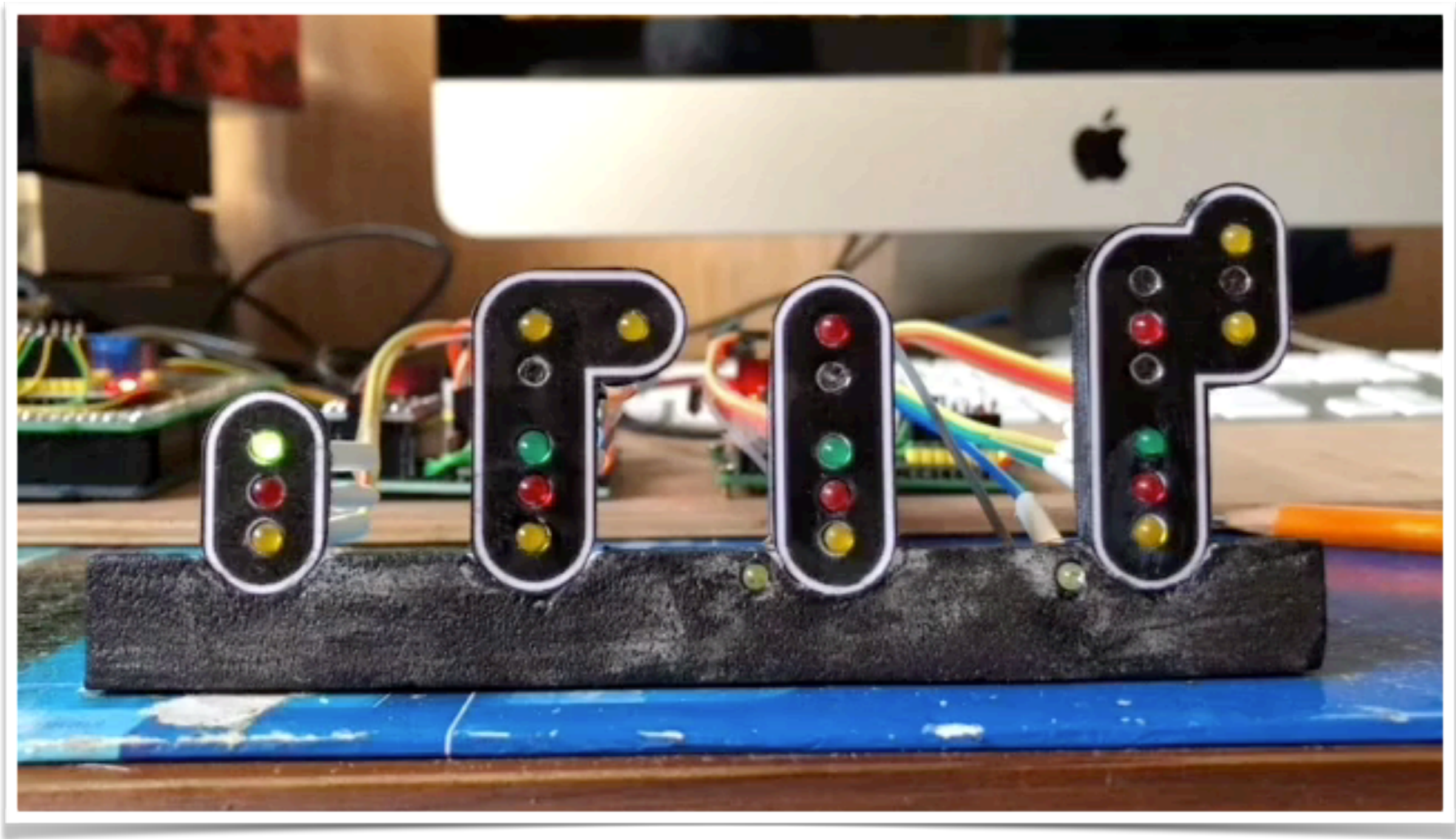
# Logiciel - les feux

- Bibliothèque Locoduino *LightDimmer*



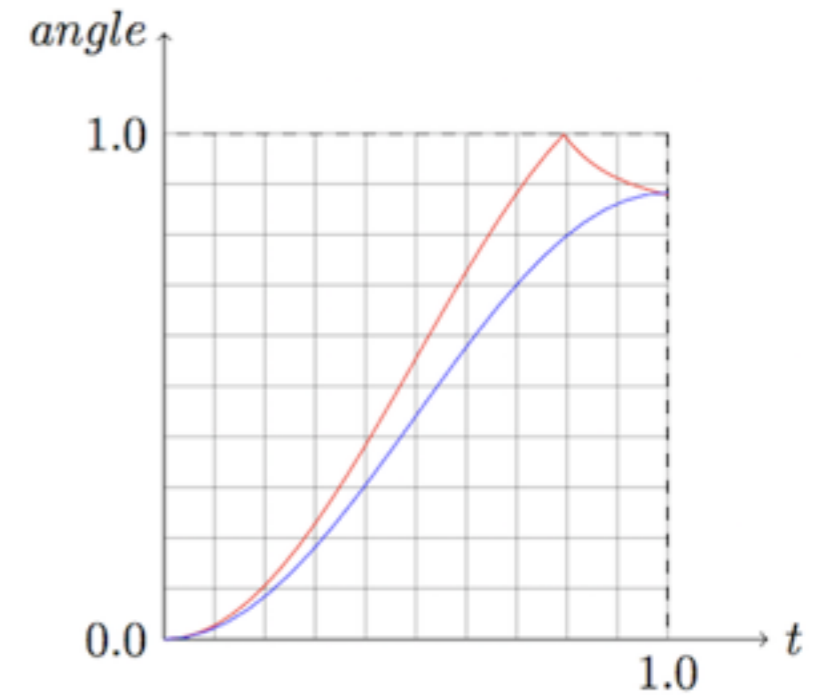
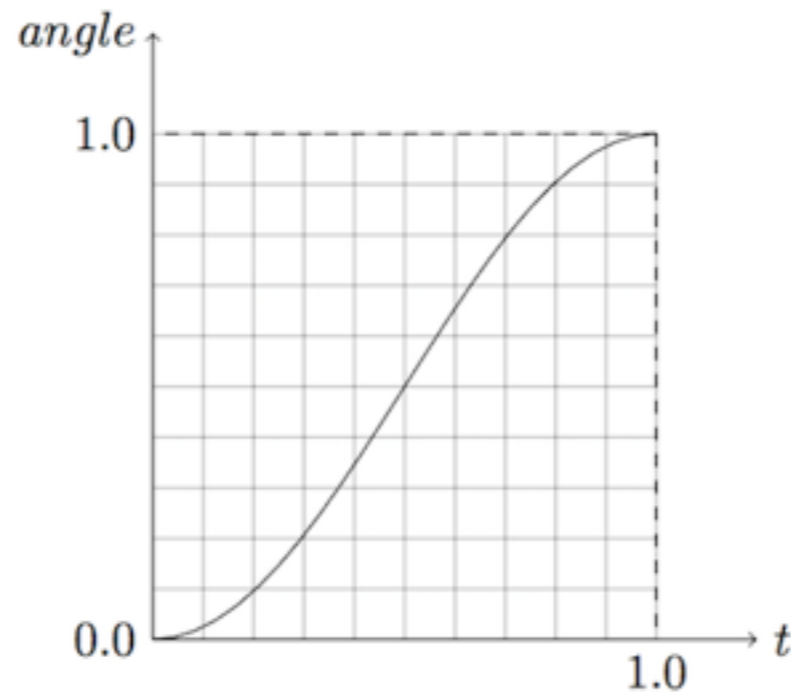
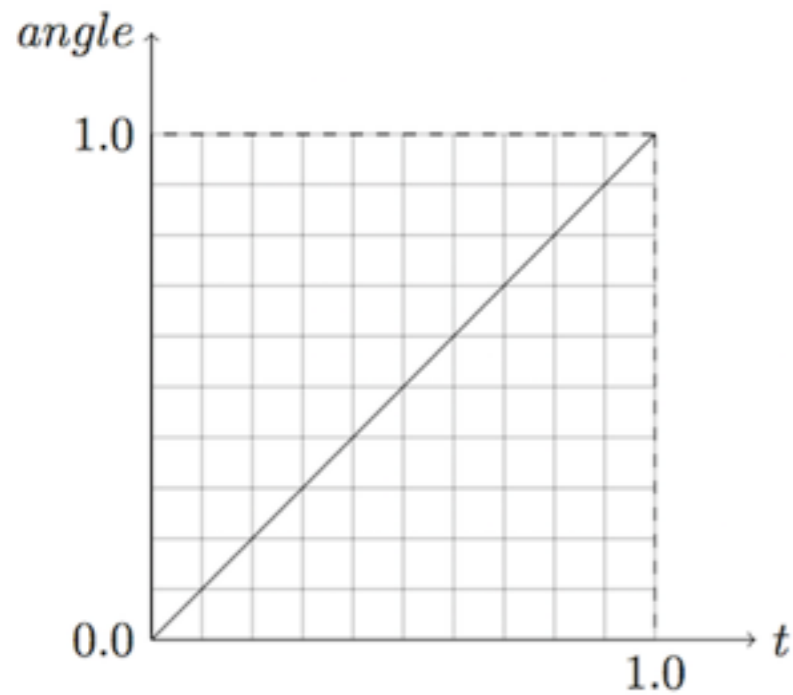
# Logiciel - les feux

- Bibliothèque Locoduino *LightDimmer*



# Logiciel - les servos

- Bibliothèque Locoduino *SlowMotionServo*





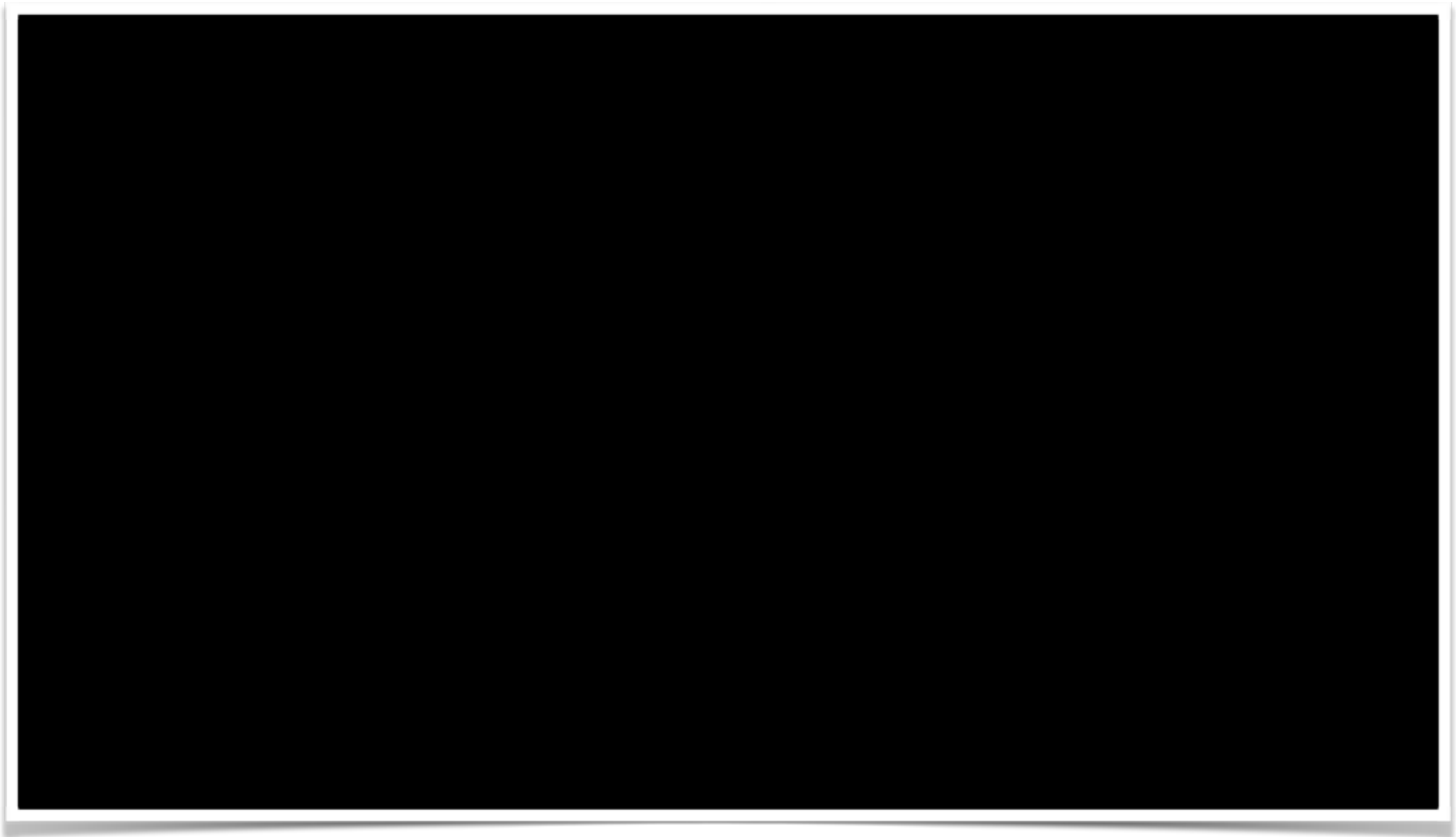
# Logiciel - les servos

- Bibliothèque Locoduino *SlowMotionServo*



# Logiciel - les servos

- Bibliothèque Locoduino *SlowMotionServo*



# Evolution : la carte Satellite V2

- Manque de souplesse de la Satellite V1
  - Assignation fixe des broches de l'Arduino aux fonctions (détection, LED, servo)
  - Détecteur d'occupation unique
  - Sous-utilisation du matériel
- Carte Satellite V2
  - Assignation des broches de l'Arduino à plusieurs fonctions
  - Configuration pour choisir une fonction par broche
  - Suppression du détecteur d'occupation
  - Système de cartes d'extension pour de l'électronique additionnelle

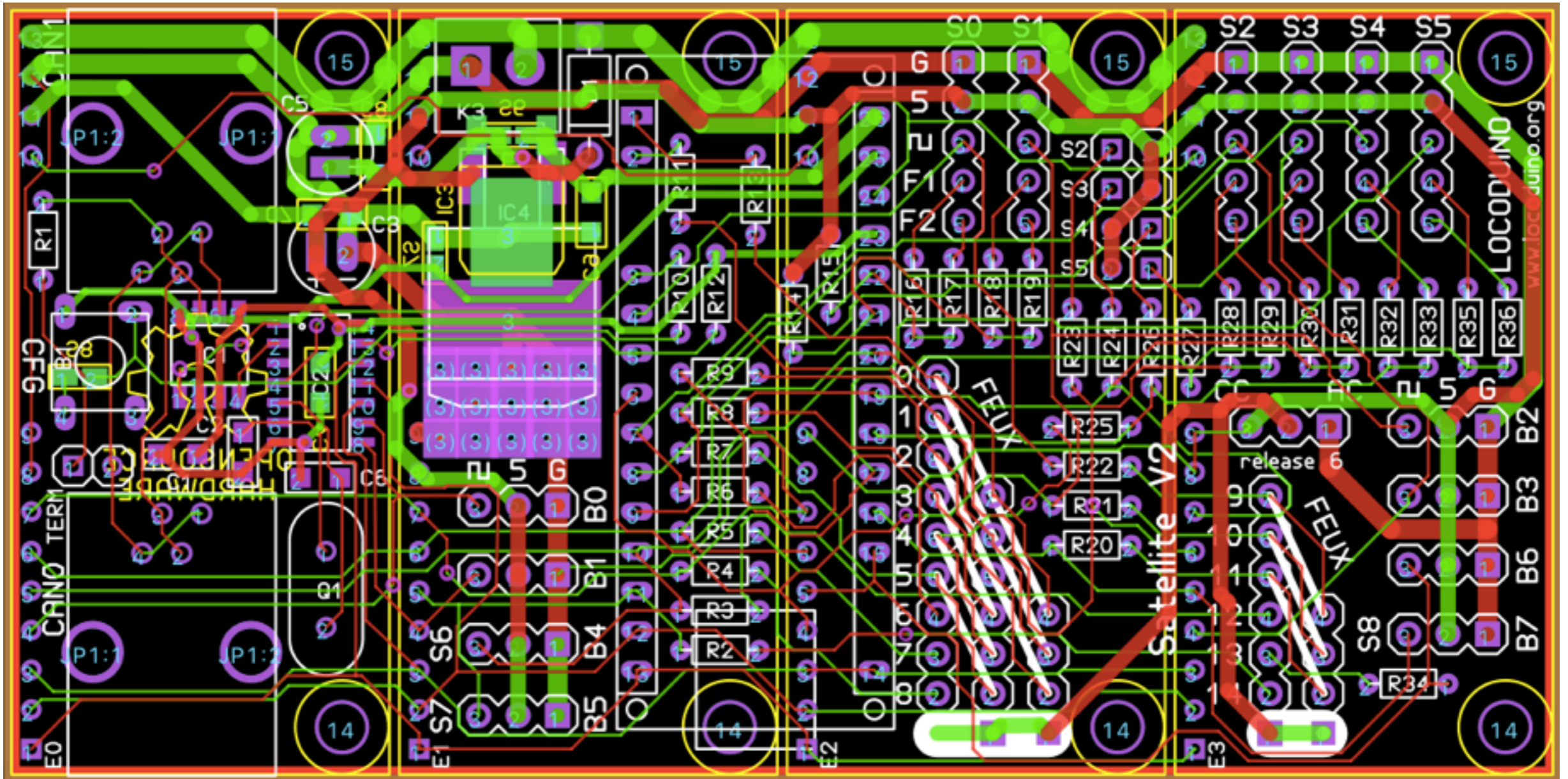
# Fonctions de la carte Satellite V2

- La carte Satellite V2 permet de connecter :
  - jusqu'à **6 servomoteurs** avec détection de fin de course;
  - jusqu'à **3 servomoteurs** sans détection de fin de course;
  - jusqu'à **15 LED** pour la signalisation;
  - jusqu'à **8 détecteurs**;
  - jusqu'à **4 cartes d'extension**.

Evidemment on ne peut pas avoir simultanément 9 servomoteurs, 15 LED et 8 détecteurs.

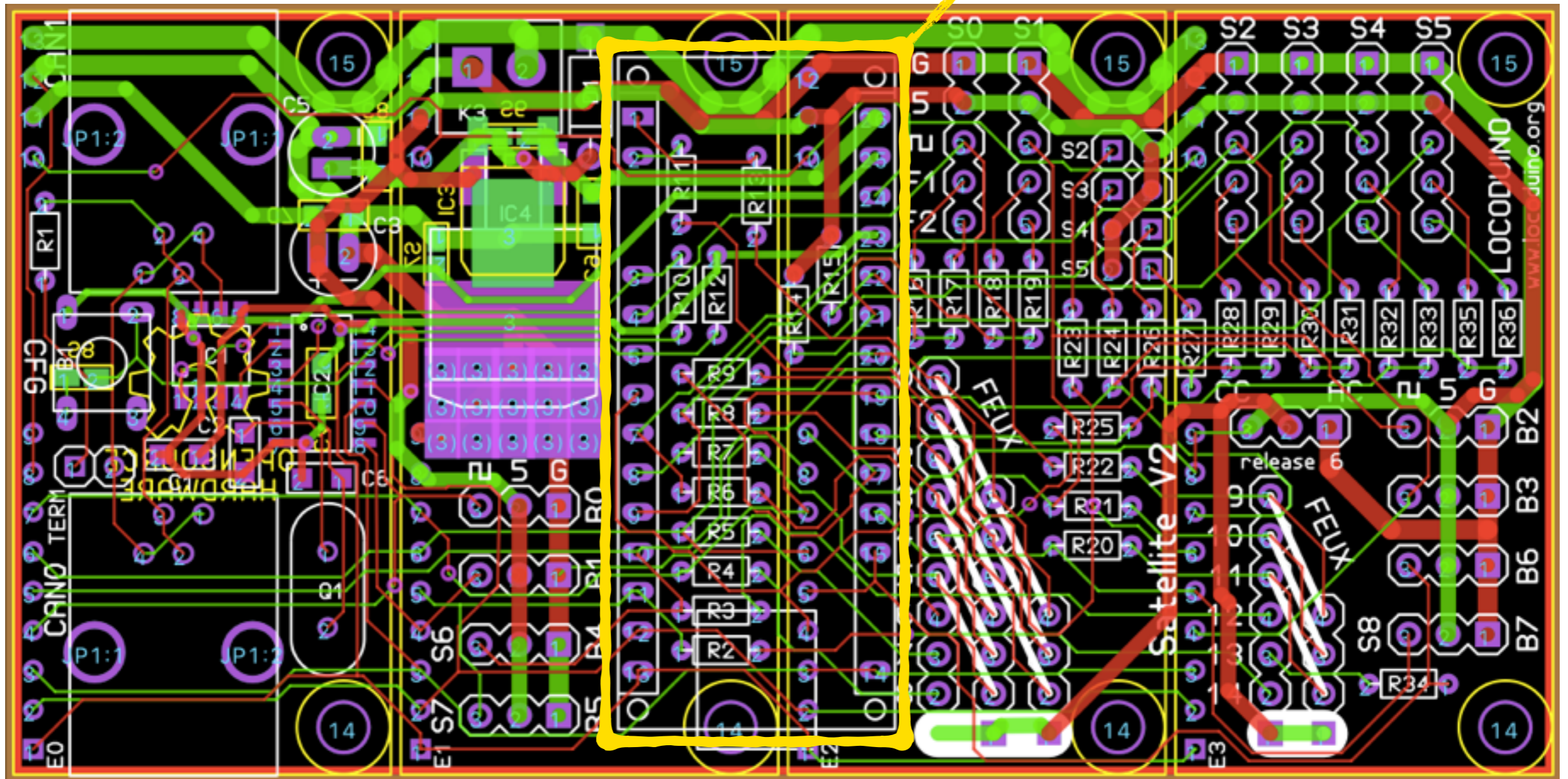
La carte Satellite V2 intègre un contrôleur CAN SPI MC2517FD de Microchip.

# Implantation de carte Satellite V2

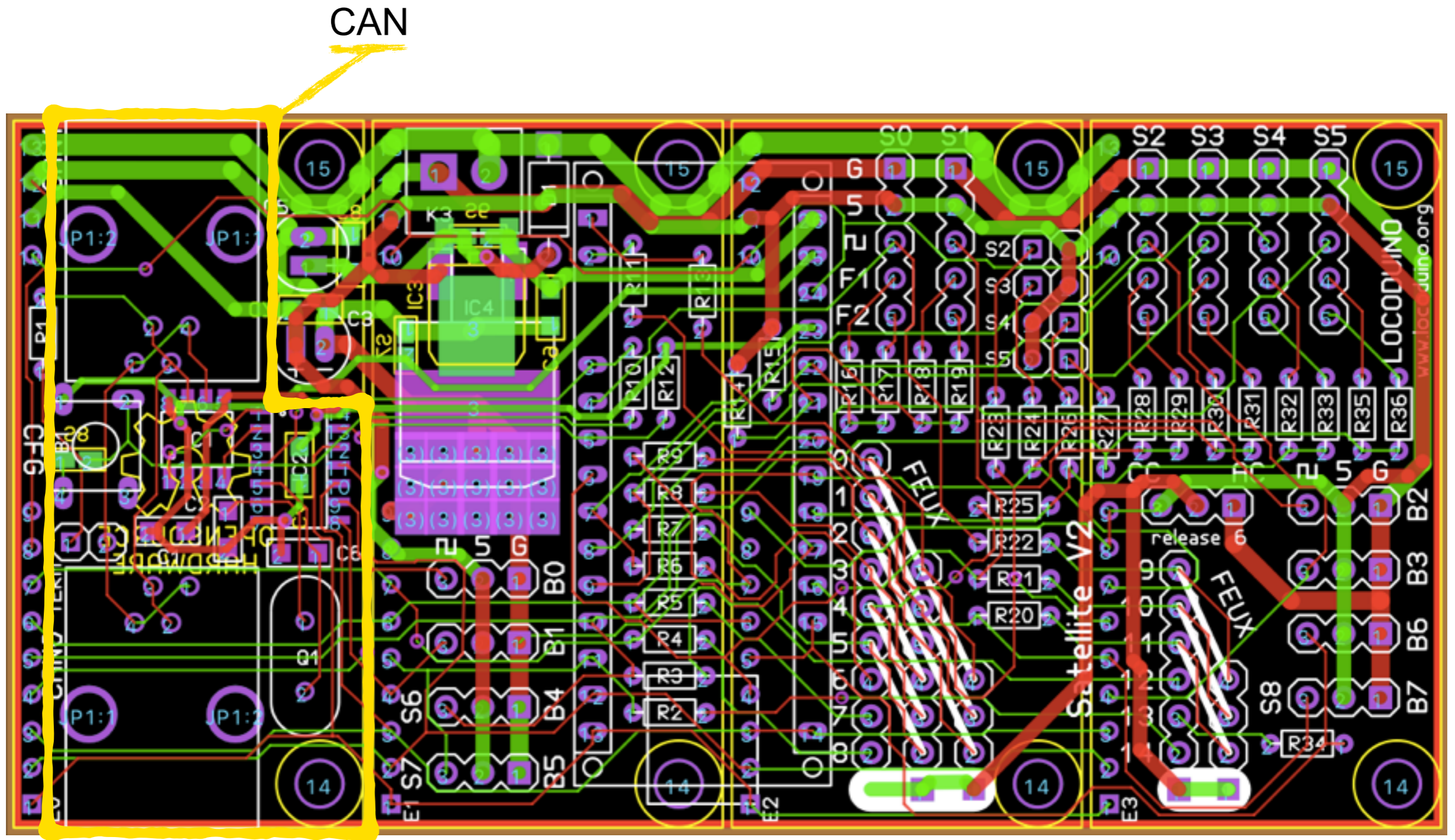


# Implantation de carte Satellite V2

Arduino Nano

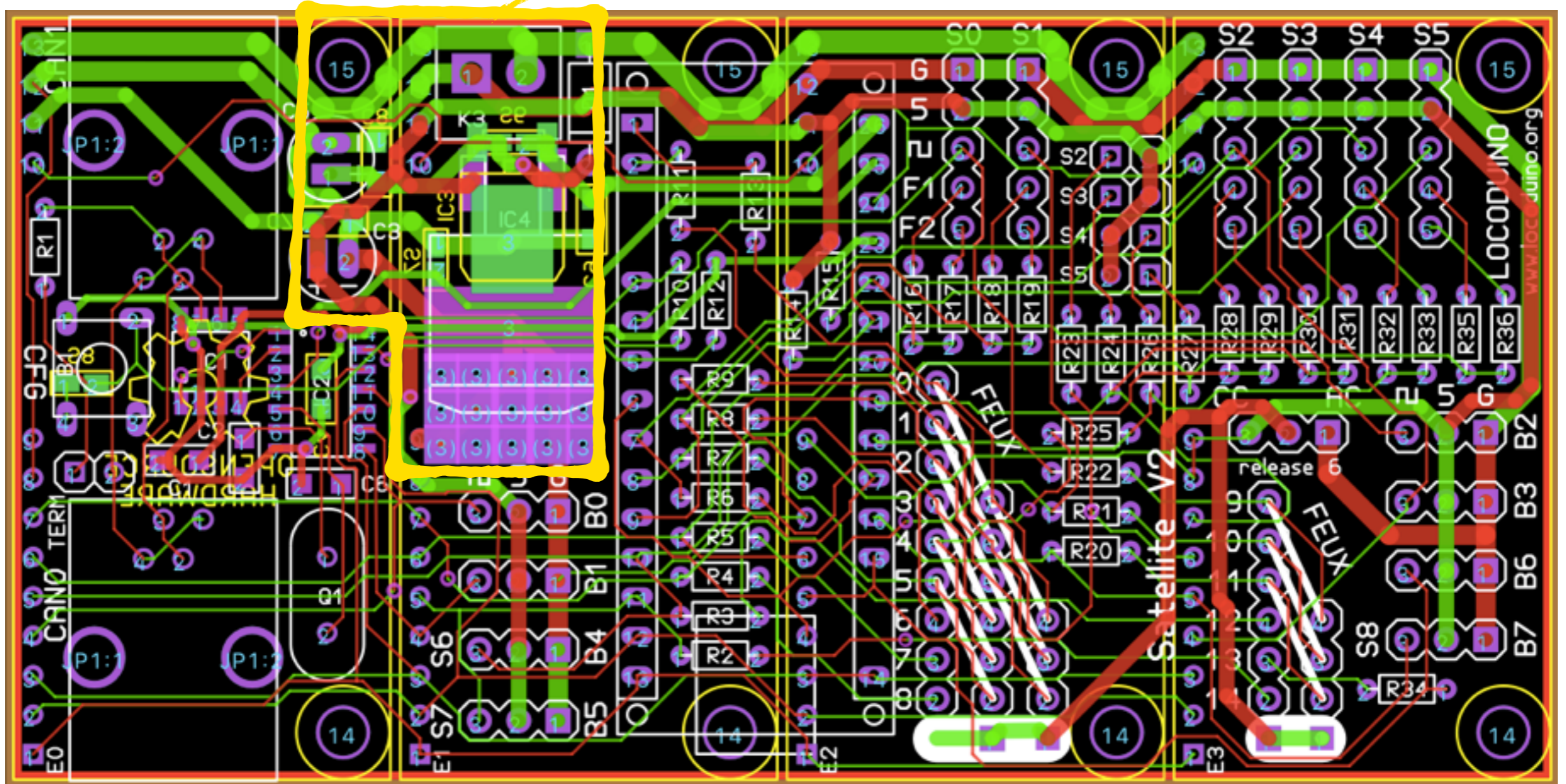


# Implantation de carte Satellite V2



# Implantation de carte Satellite V2

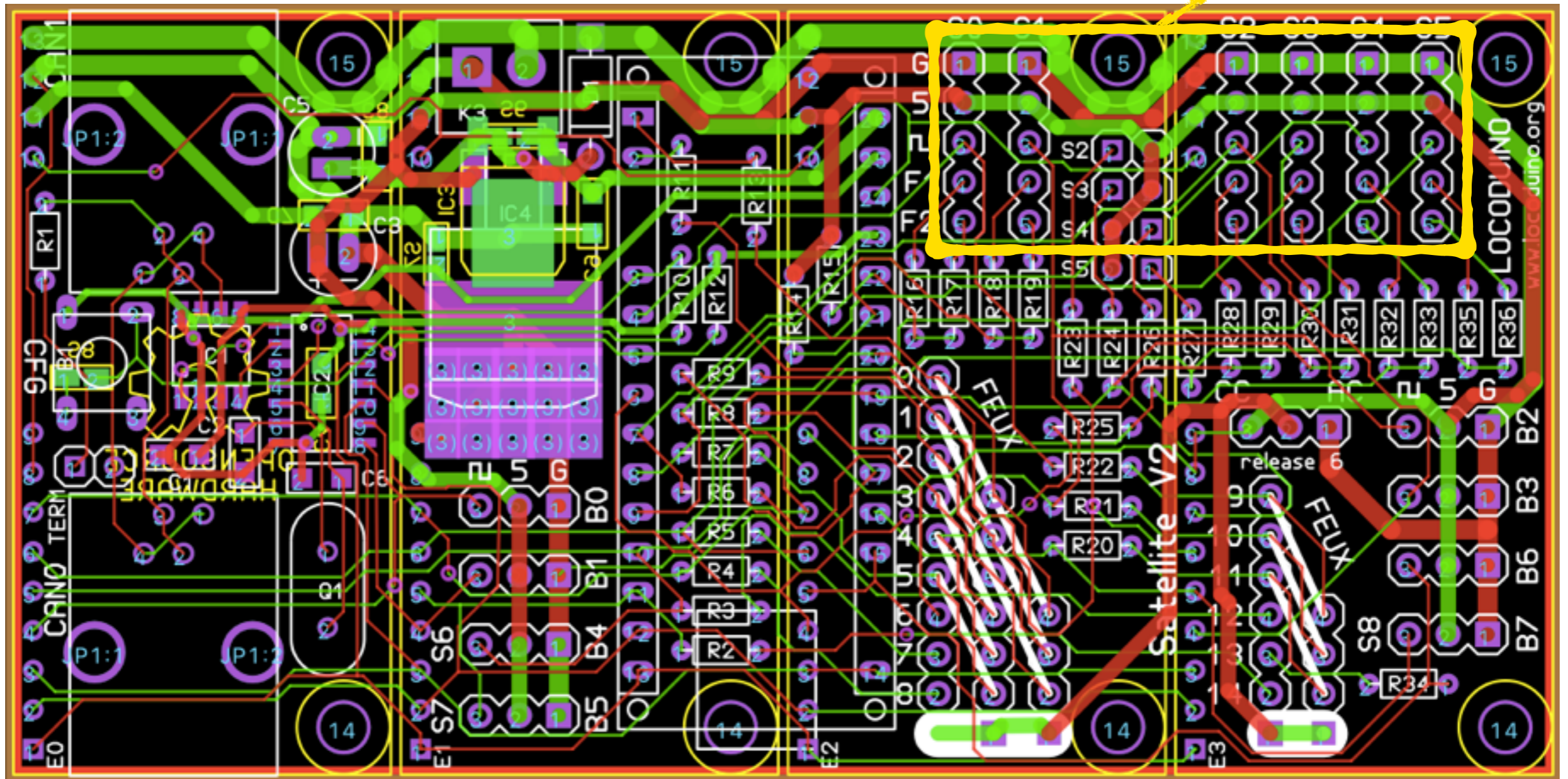
Alimentation



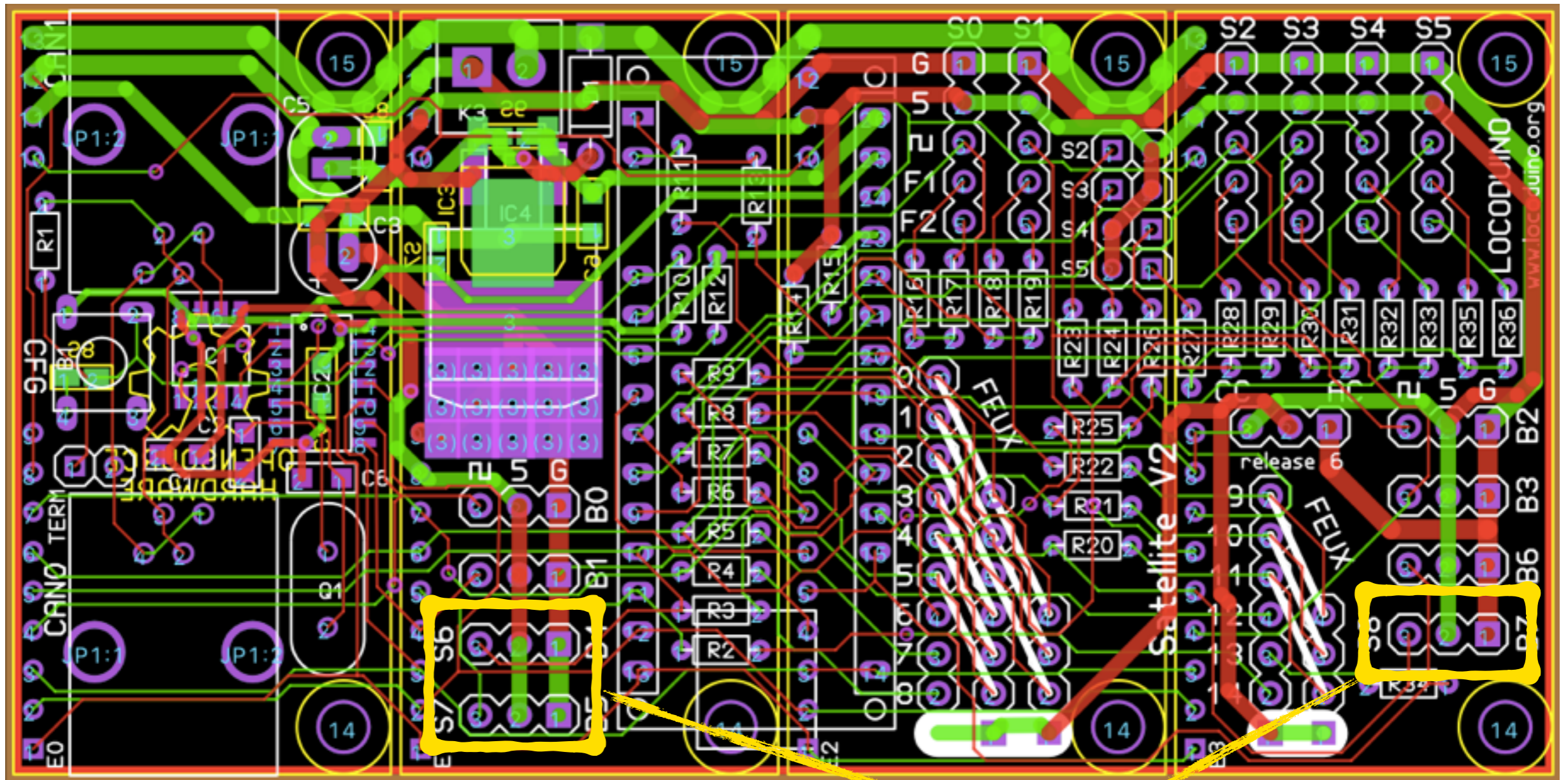


# Implantation de carte Satellite V2

Connecteurs servos avec FDC

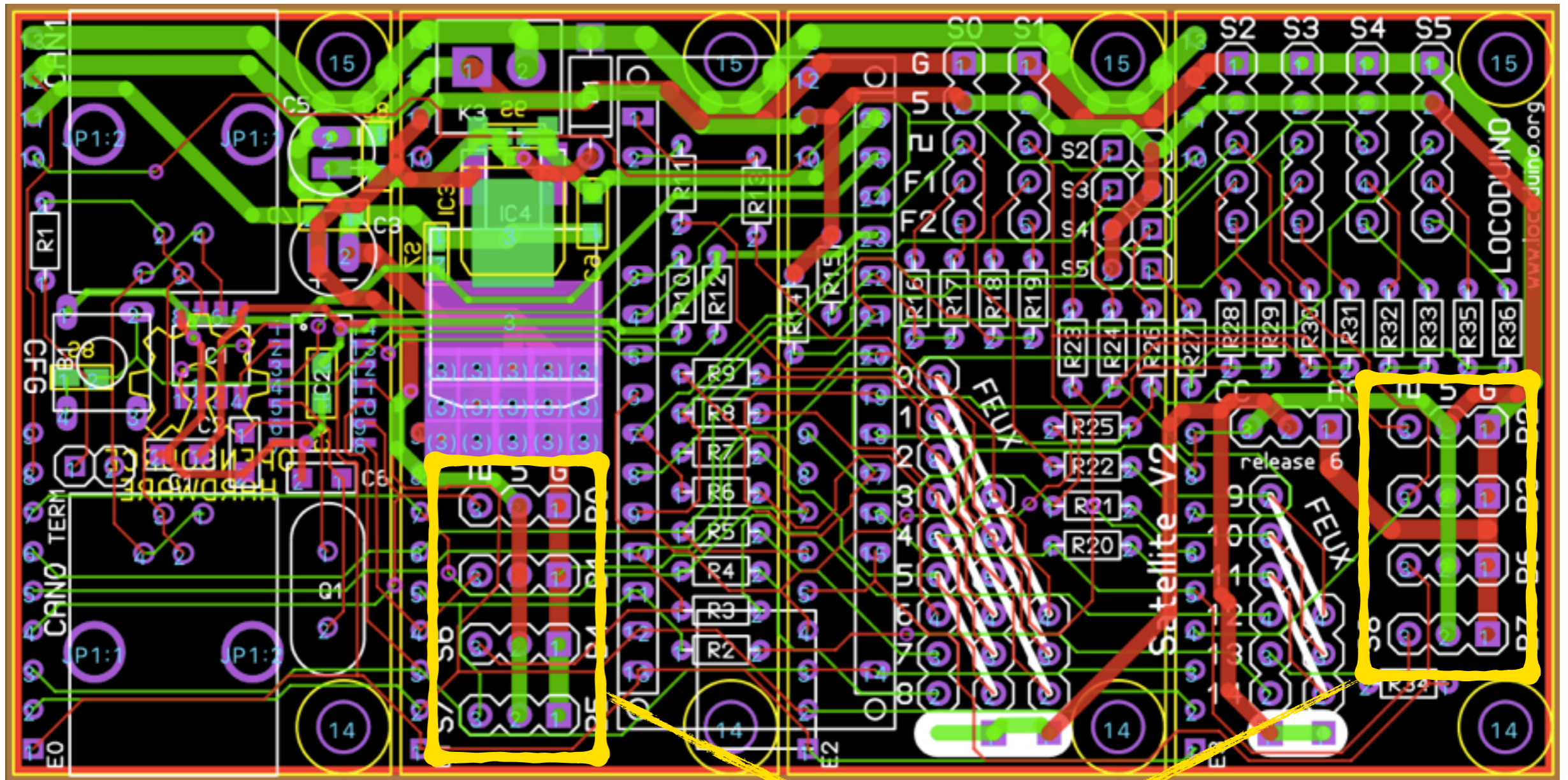


# Implantation de carte Satellite V2



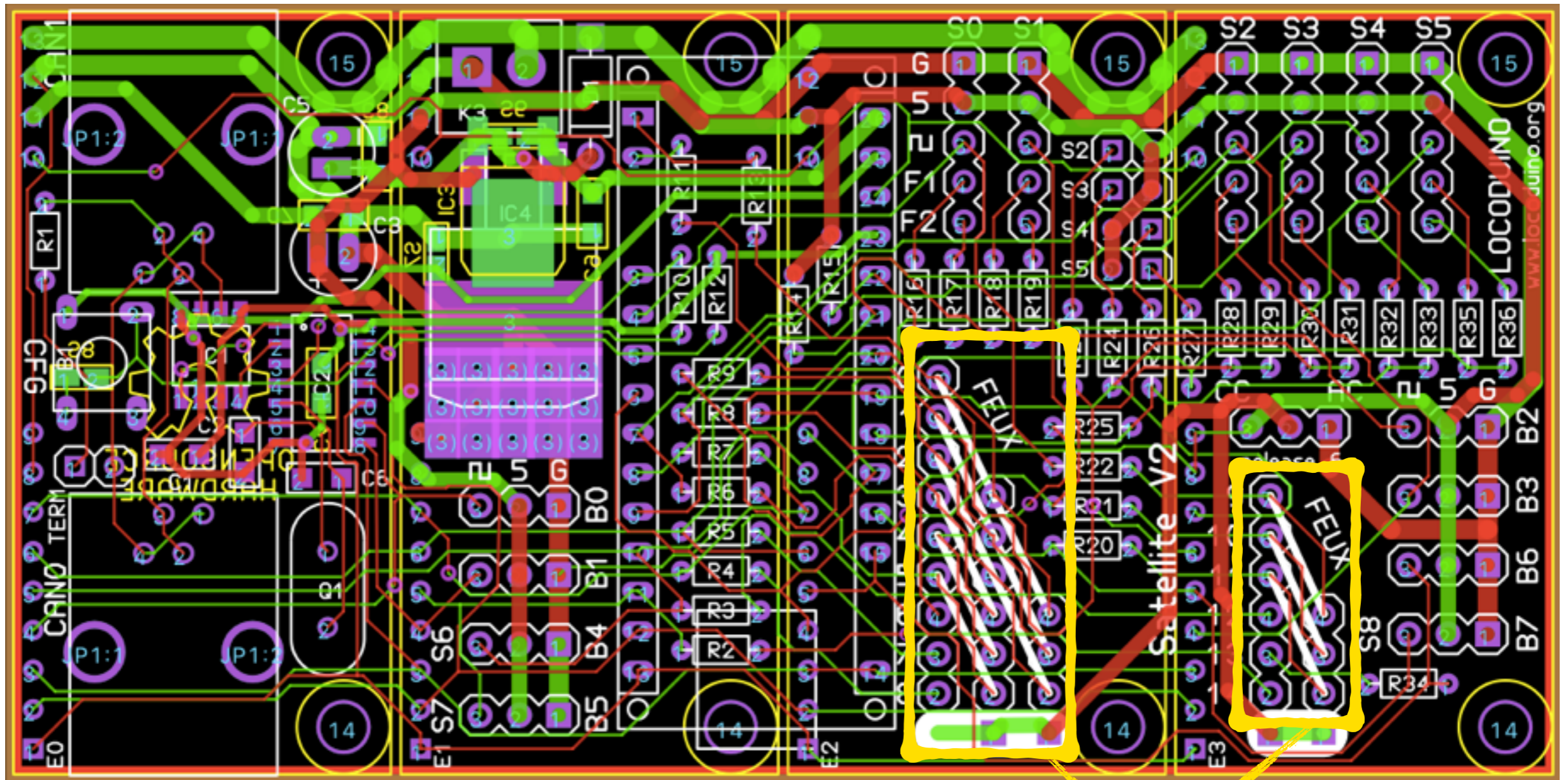
Connecteurs servos sans FDC

# Implantation de carte Satellite V2



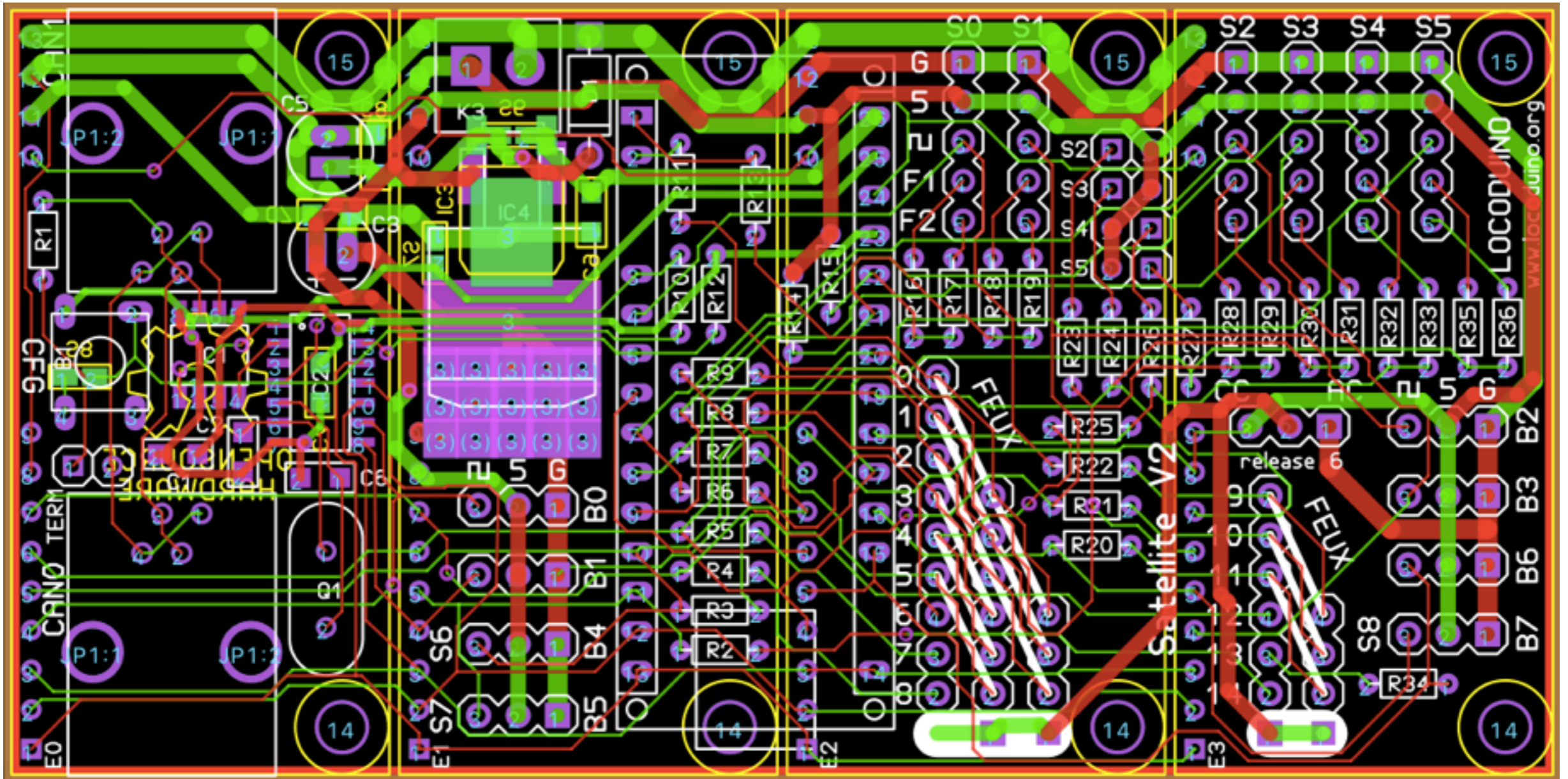
Connecteurs détecteurs

# Implantation de carte Satellite V2



Connecteurs feux

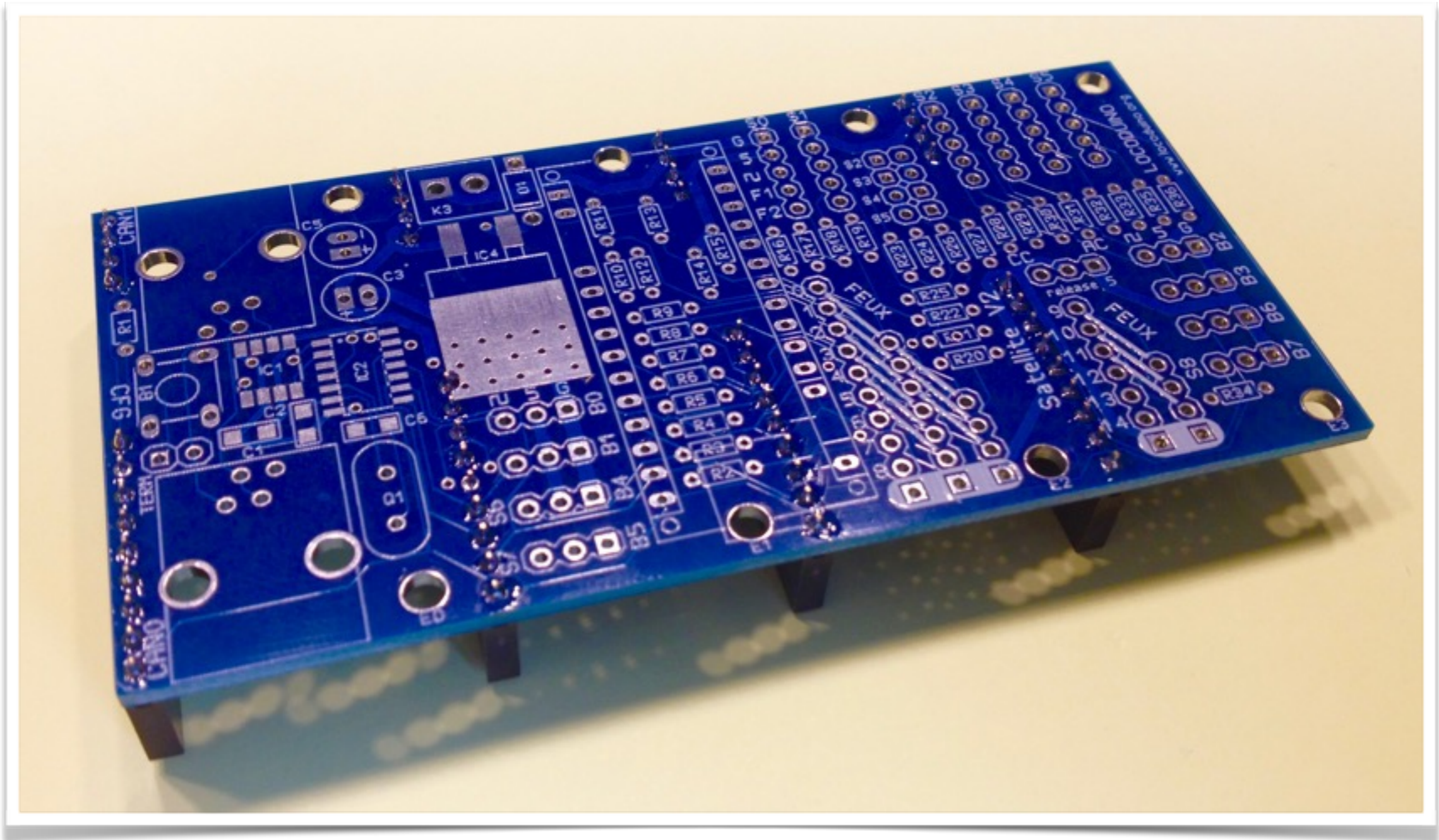
# Implantation de carte Satellite V2



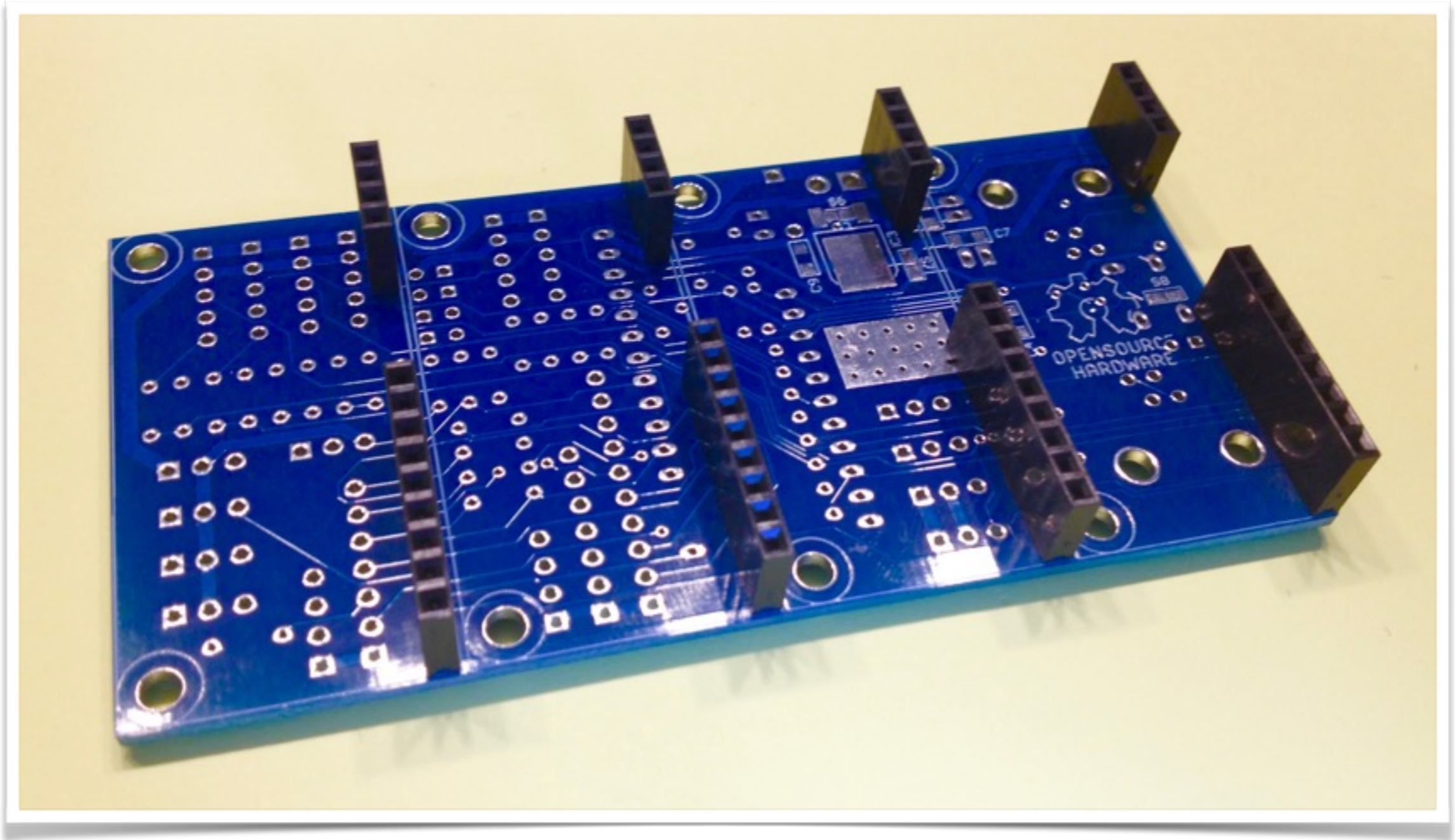
# Affectation des broches

Broche	Communication	Servo	Fin de course	détection	carte d'extension	LED
RX0	série					L1
TX1	série					L0
D2	Interruption CAN					
D3		S0				L9
D4	Chip select CAN					
D5		S1		B2	E2	L10
D6		S2		B3	E3	L6
D7		S3				L5
D8		S4				L3
D9		S6		B4	E0	L14
D10		S7		B5	E1	L13
D11	SPI MOSI					
D12	SPI MISO					
D13	SPI SCK					
A0		S5		B6	E2	L12
A1		S8		B7	E3	L11
A2			F5			L2
A3			F4			L4
A4	I2C SDA		F3			L8
A5	I2C SCL		F2			L7
A6			F1			
A7			F0			
IO0				B0	E0	
IO1				B1	E1	

# Montage des cartes d'extension

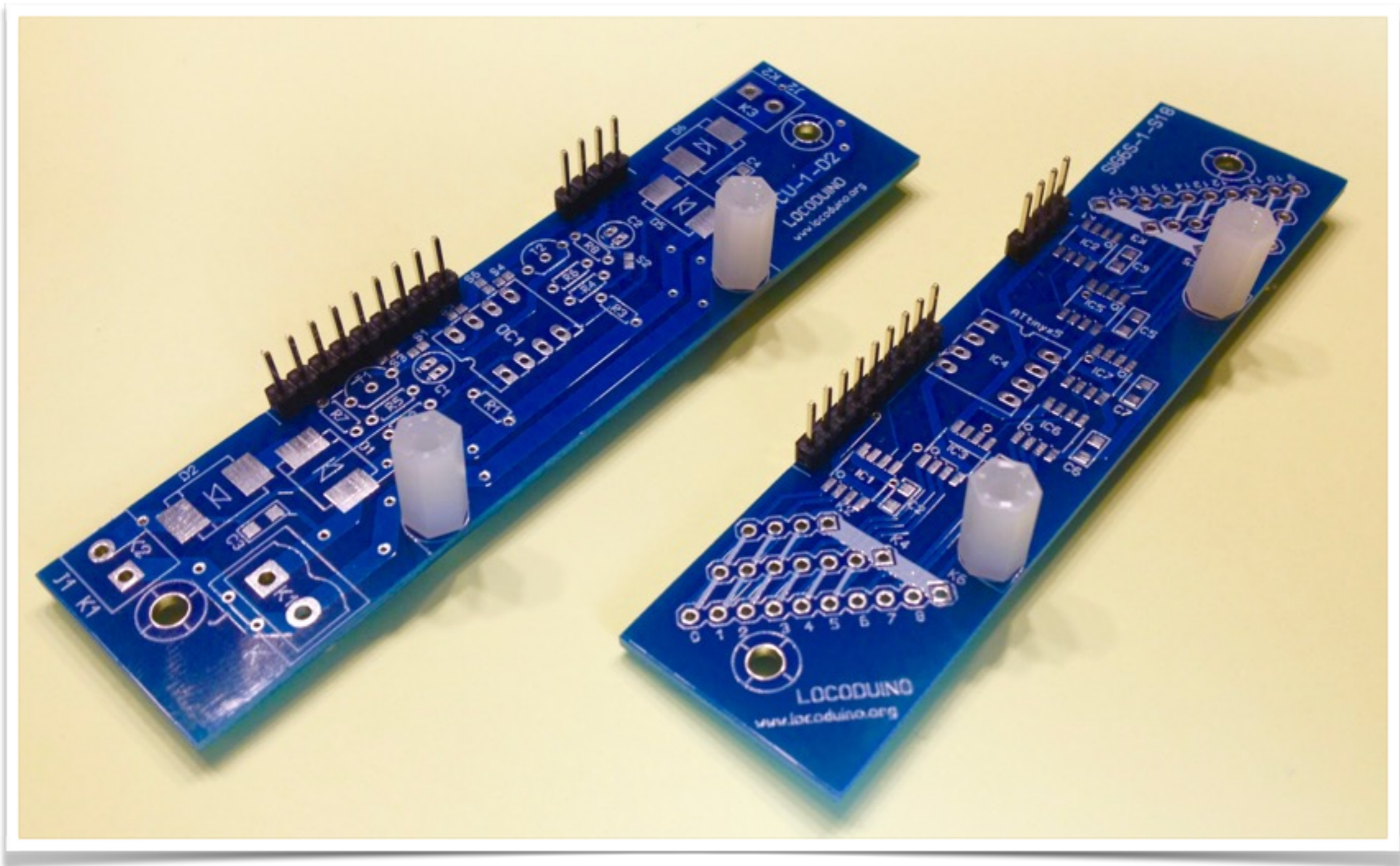


# Montage des cartes d'extension



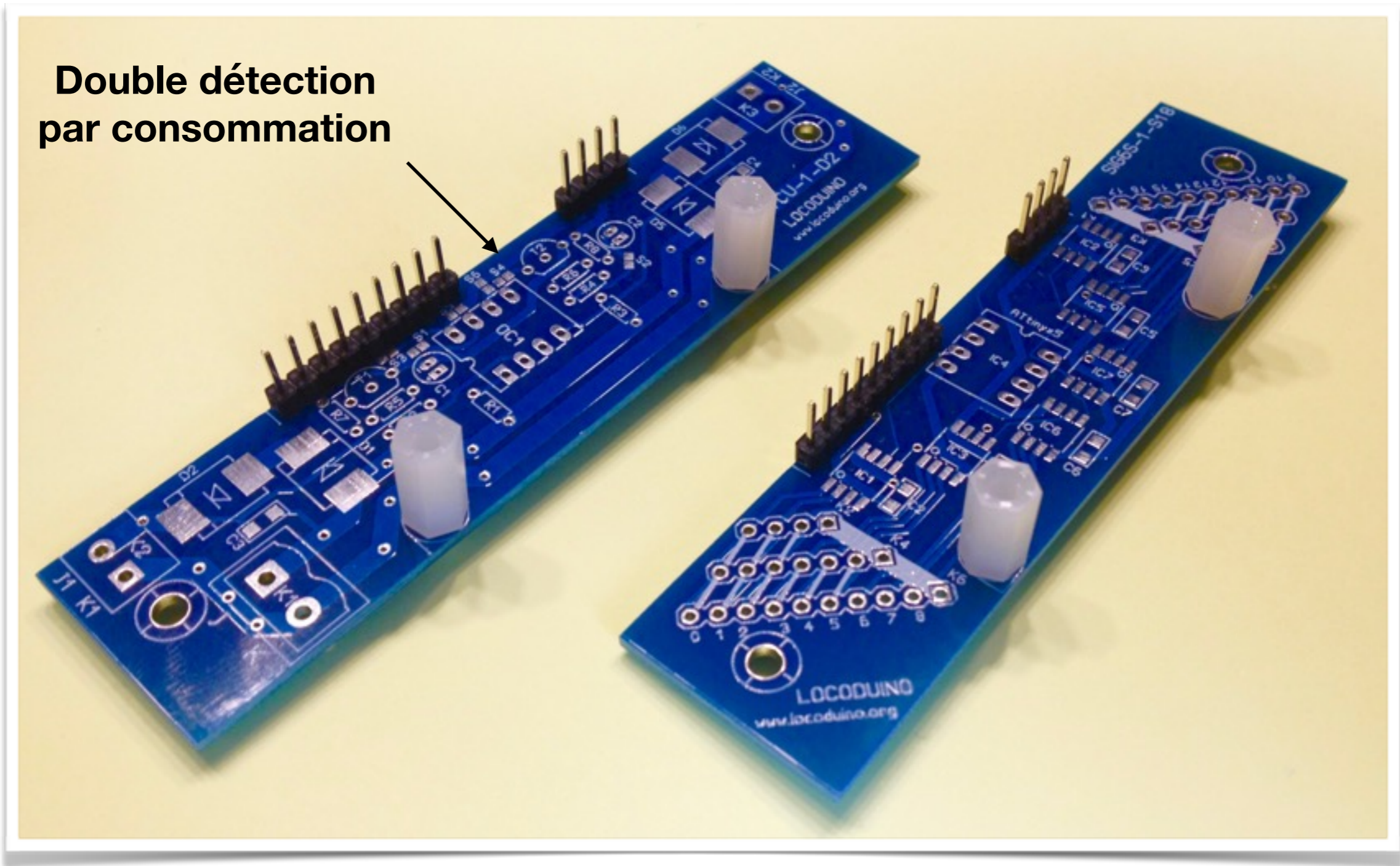


# Montage des cartes d'extension

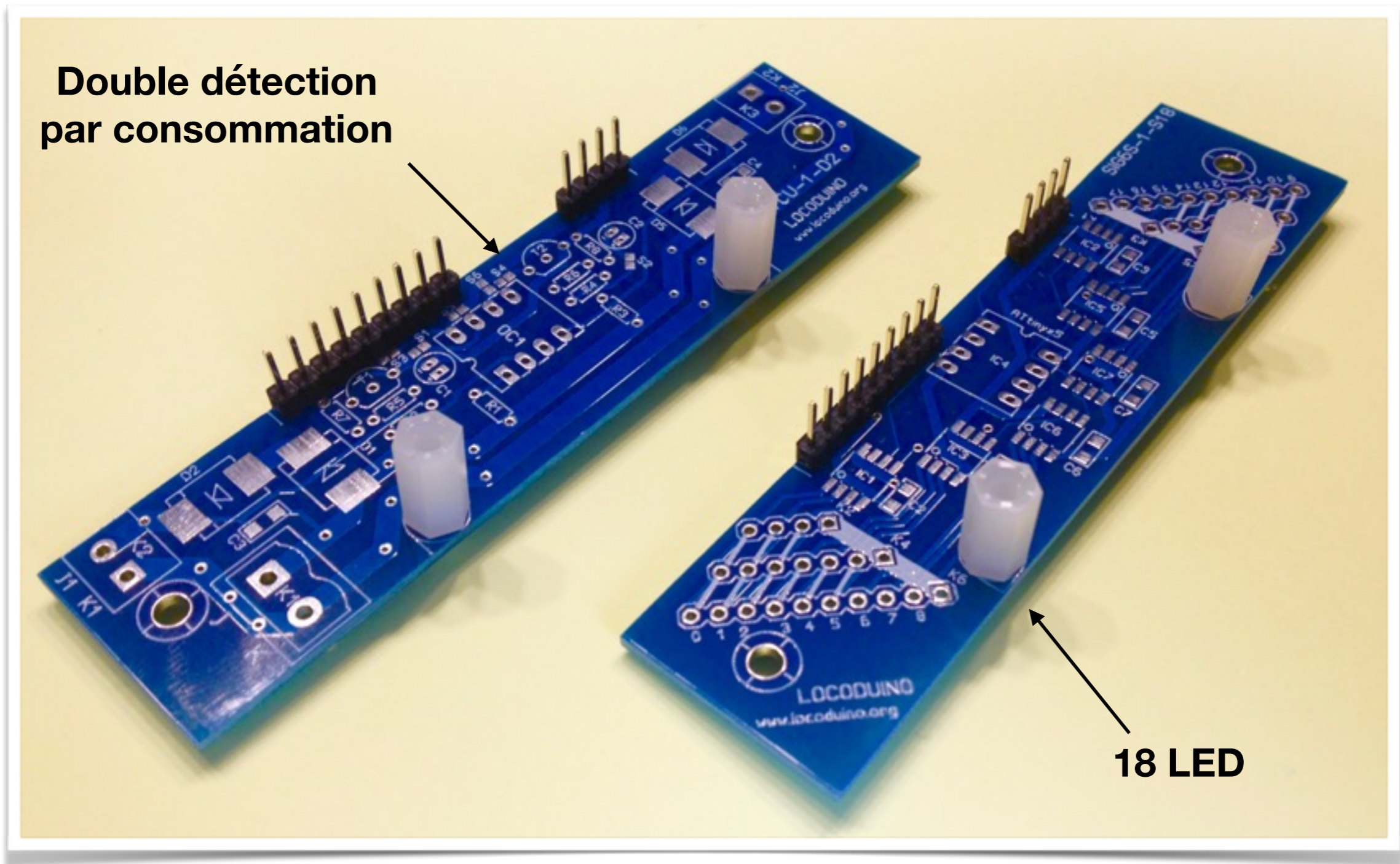


# Montage des cartes d'extension

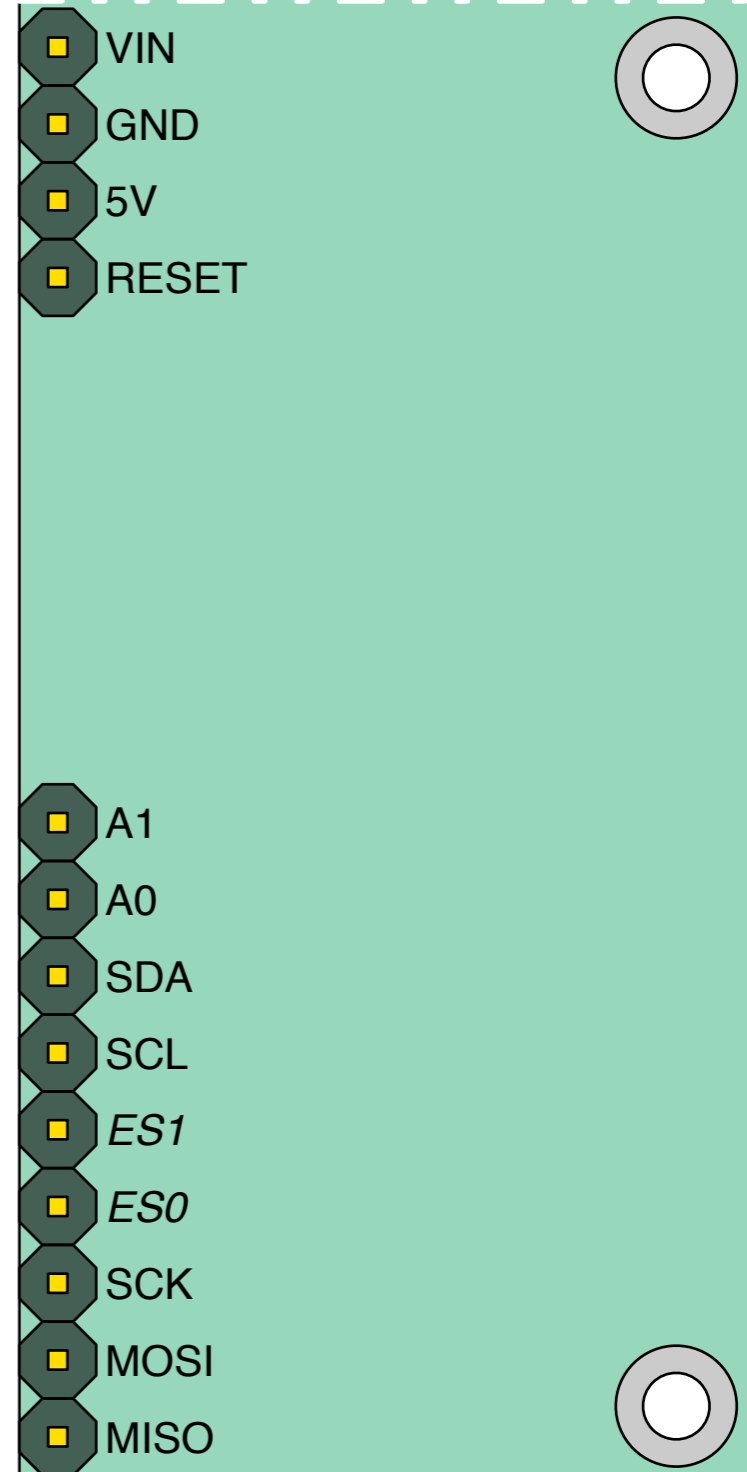
Double détection  
par consommation



# Montage des cartes d'extension

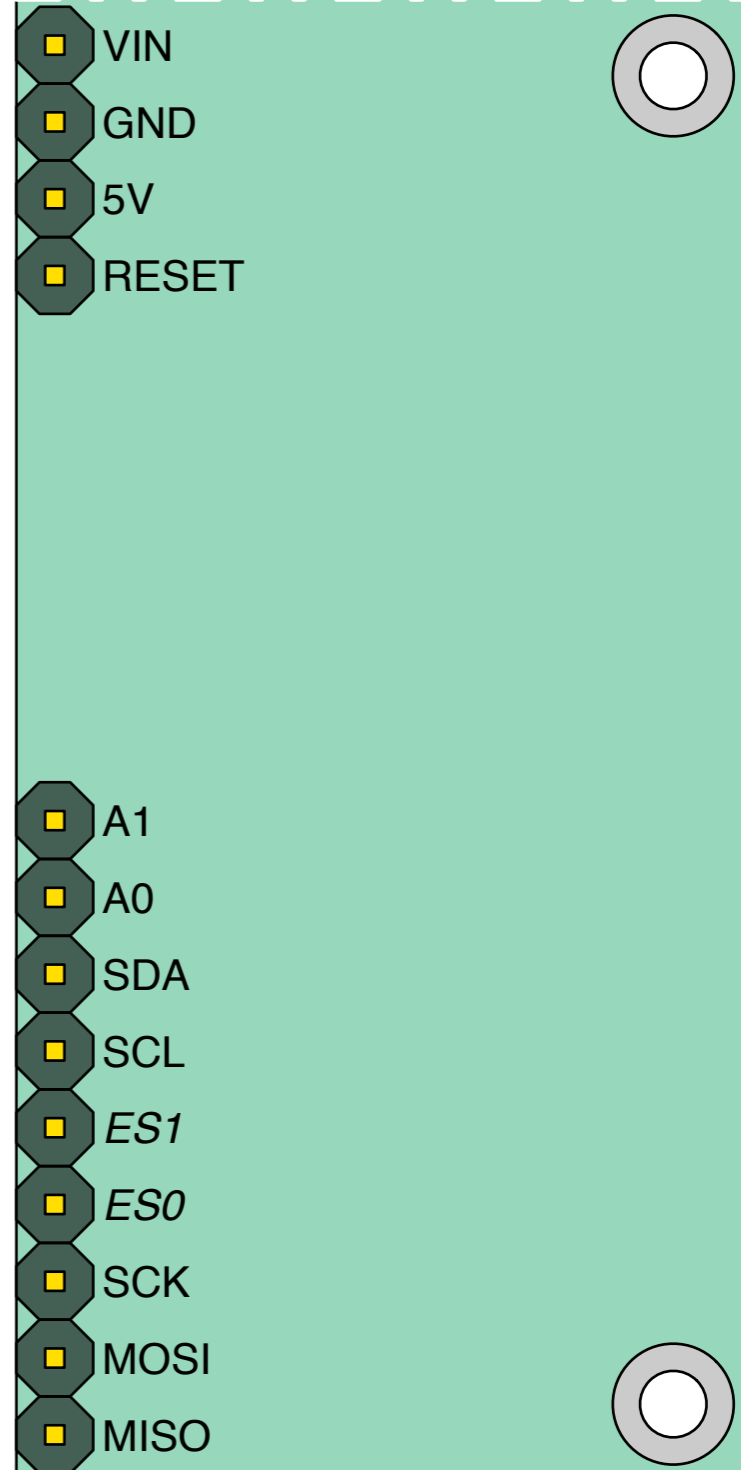


# Connecteur d'extension



# Connecteur d'extension

Alimentation de la carte Satellite

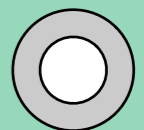
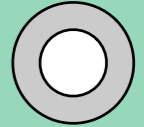


# Connecteur d'extension

Alimentation de la carte Satellite  
Masse

- VIN
- GND
- 5V
- RESET

- A1
- A0
- SDA
- SCL
- ES1
- ES0
- SCK
- MOSI
- MISO

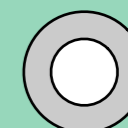
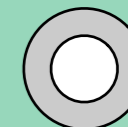


# Connecteur d'extension

Alimentation de la carte Satellite  
Masse  
5V du régulateur de la carte Satellite

VIN  
GND  
5V  
RESET

A1  
A0  
SDA  
SCL  
*ES1*  
*ES0*  
SCK  
MOSI  
MISO



# Connecteur d'extension

Alimentation de la carte Satellite  
Masse  
5V du régulateur de la carte Satellite  
RESET de l'Arduino

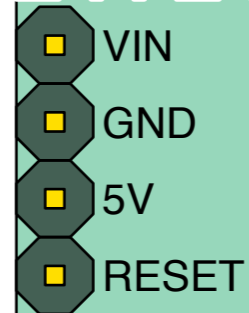
VIN  
GND  
5V  
RESET

A1  
A0  
SDA  
SCL  
*ES1*  
*ES0*  
SCK  
MOSI  
MISO

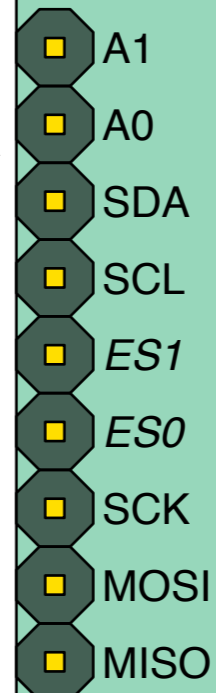


# Connecteur d'extension

Alimentation de la carte Satellite  
Masse  
5V du régulateur de la carte Satellite  
RESET de l'Arduino

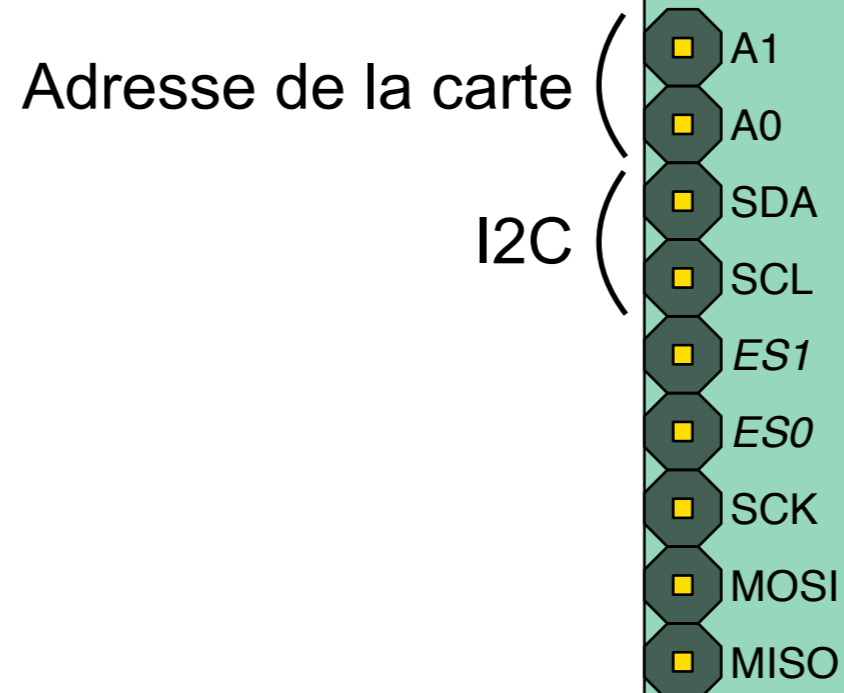
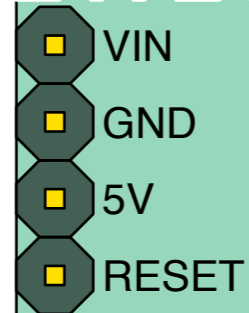


Adresse de la carte (



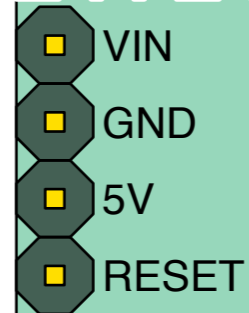
# Connecteur d'extension

Alimentation de la carte Satellite  
Masse  
5V du régulateur de la carte Satellite  
RESET de l'Arduino



# Connecteur d'extension

Alimentation de la carte Satellite  
Masse  
5V du régulateur de la carte Satellite  
RESET de l'Arduino



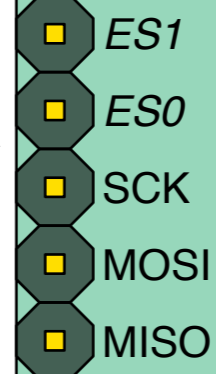
Adresse de la carte (



I2C (

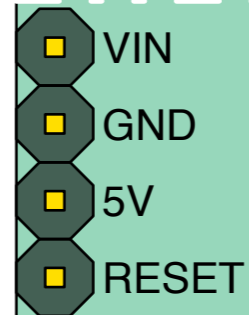


Broches de l'Arduino (



# Connecteur d'extension

Alimentation de la carte Satellite  
Masse  
5V du régulateur de la carte Satellite  
RESET de l'Arduino



Adresse de la carte



I2C



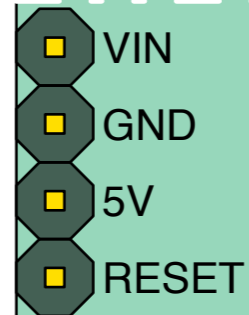
Broches de l'Arduino



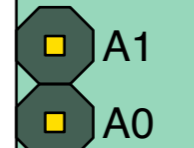
Emplacement	ES1	ES0
E0	D9	IO0
E1	D10	IO1
E2	A0	D5
E3	A1	D6

# Connecteur d'extension

Alimentation de la carte Satellite  
Masse  
5V du régulateur de la carte Satellite  
RESET de l'Arduino



Adresse de la carte



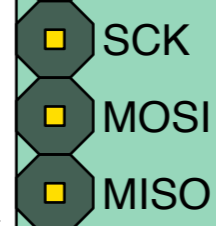
I2C



Broches de l'Arduino

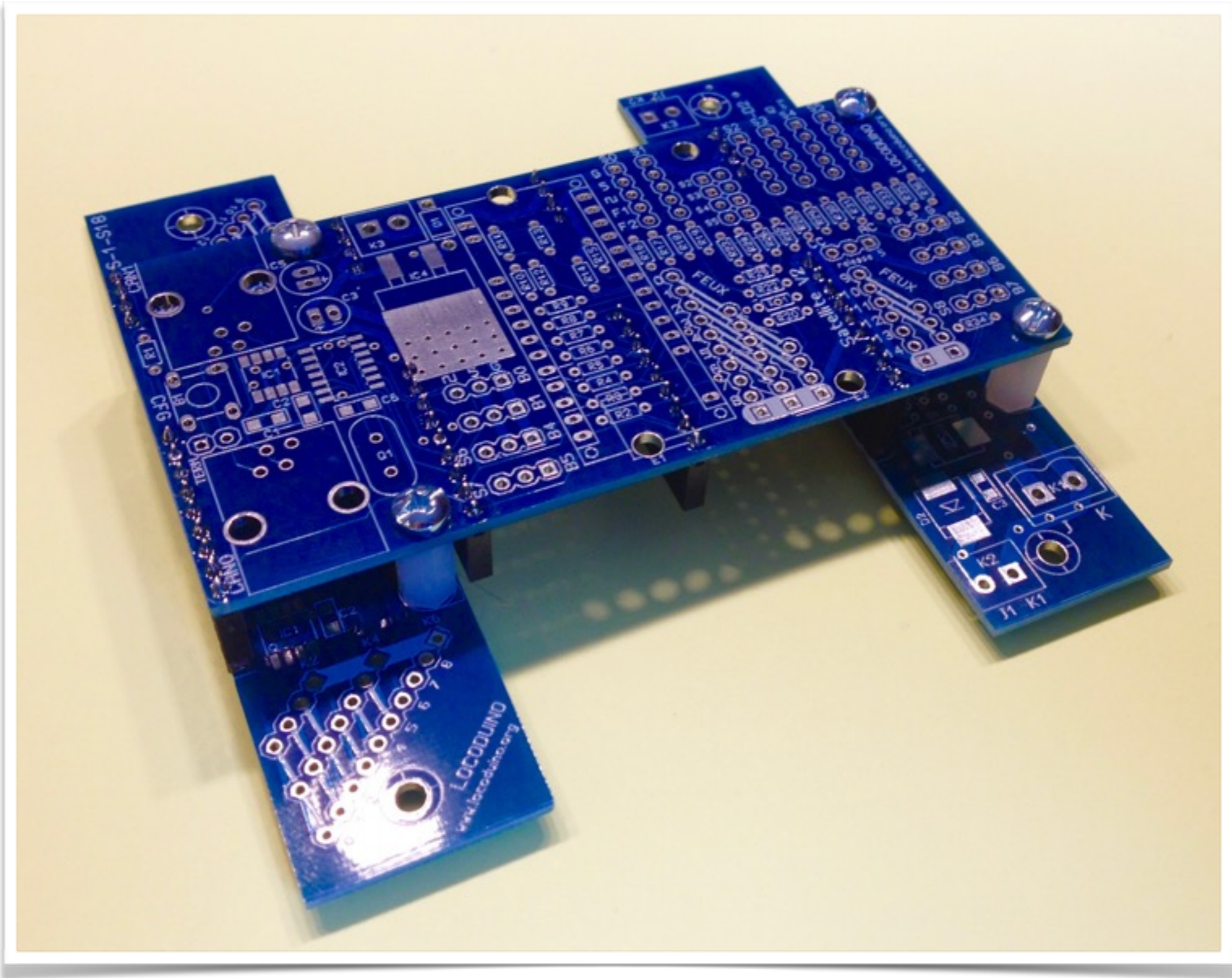


SPI



Emplacement	ES1	ES0
E0	D9	IO0
E1	D10	IO1
E2	A0	D5
E3	A1	D6

# Montage des cartes d'extension



# À suivre...

- Développement du logiciel de la carte Satellite V2
- Développement du(es) configurateur(s)
- Développement de cartes d'extension
- Passerelle Wifi/CAN

# À suivre...

- Développement du logiciel de la carte Satellite V2
- Développement du(es) configurateur(s)
- Développement de cartes d'extension
- Passerelle Wifi/CAN

**Merci de votre attention !**