

Usine de tri

Projet de fin de session

Alex Vézina

Philippe Lemoine

John-William Lebel

Samuel Duclos

Ayé Claude Brou

Électronique Programmable et Robotique

5^e session, gr.005

Travail présenté à

M. Yves Roy

M. Simon Ayoub

M. Stéphane Deschênes

Département de Génie électrique

Cégep Limoilou

Le 13 novembre 2020

Table des matières

Architectures

Système

Poste de commande

Interface graphique

Centre de tri

Unité de gestion des transports

Station de pesage

Connexions électriques

Communication CAN

Planification

Architectures

Système

La figure suivante présente l'architecture système du projet. Le système se compose donc de 6 modules qui communiquent par CAN, Wi-Fi ou RS-232.

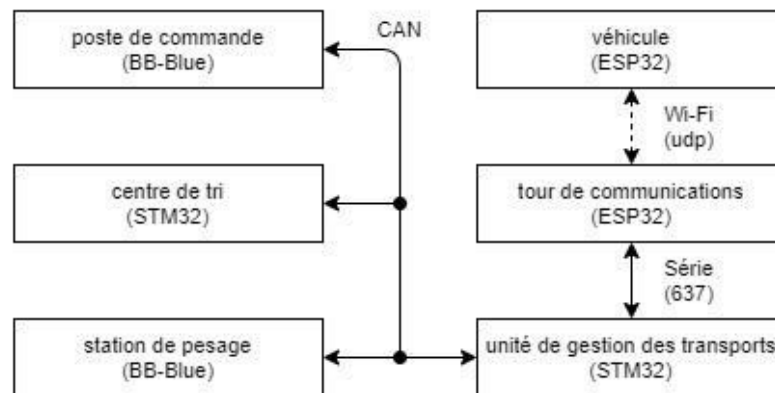


Figure 1 : Architecture système du projet

La figure suivante présente l'exécution des transitions d'état dans le temps à travers les différentes stations.

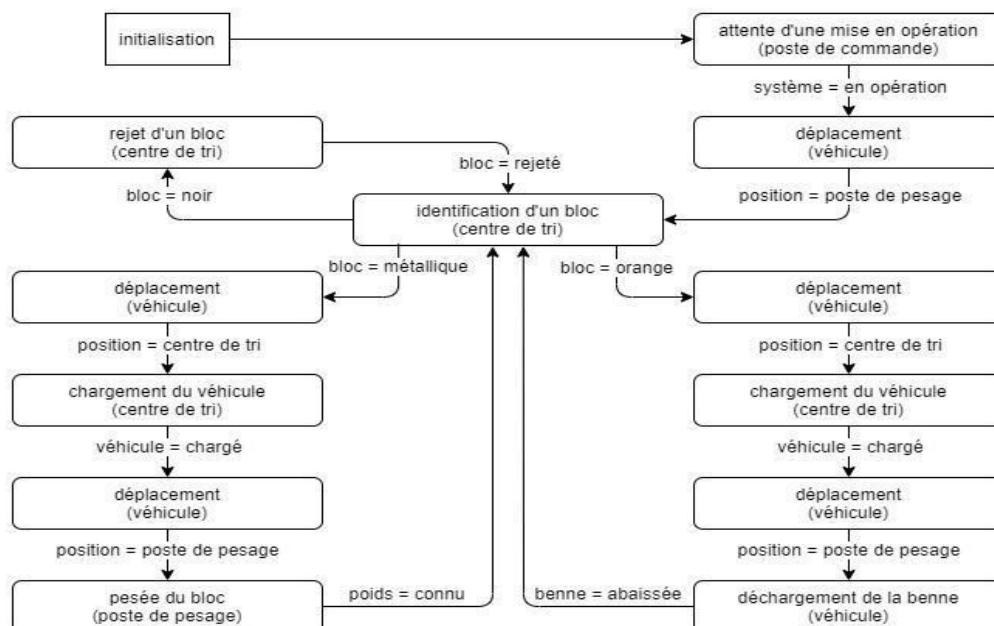


Figure 2 : Machine à états du projet

Poste de commande

Architecture matérielle

Un ordinateur commande l'ensemble des éléments du projet. L'ensemble du système est représenté par une interface graphique qui implémente des “widgets” dans *Visual Studio*. Ces “widgets” servent à interagir avec l'utilisateur pour aisément lui permettre de configurer les nœuds avec leurs différentes fonctionnalités. L'ordinateur connecté par USB communique par protocole série avec le *Beaglebone Blue*. Celui-ci ne sert que de passerelle pour relayer les communications vers et du reste du système, par protocole CAN.

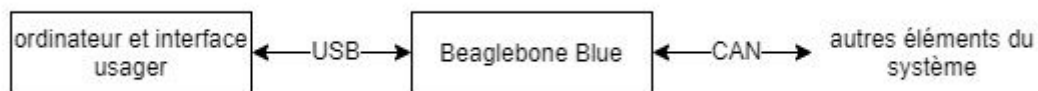


Figure 3 : Architecture matérielle du poste de commande

Architecture logicielle

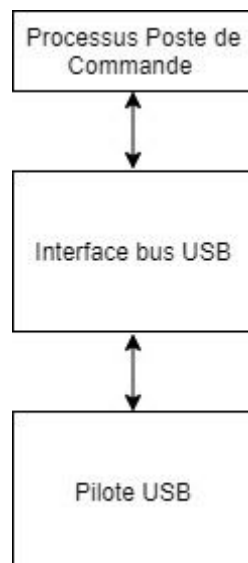


Figure 4 : Architecture logicielle du PC du poste de commande

Le *Beaglebone Blue* relaie les communications entre le PC et les autres éléments du système en redirigeant les entrées avec les sorties et vice-versa par des descripteurs de fichiers qui sont liés à un socket qui communique avec le serveur CAN en mode "broadcast manager". Les primitives disponibles facilitent la gestion en léguant la base de temps au noyau du système d'exploitation. Ce "design" simplifie grandement la programmation et la communication entre les éléments du projet. La figure suivante présente cette architecture logicielle.

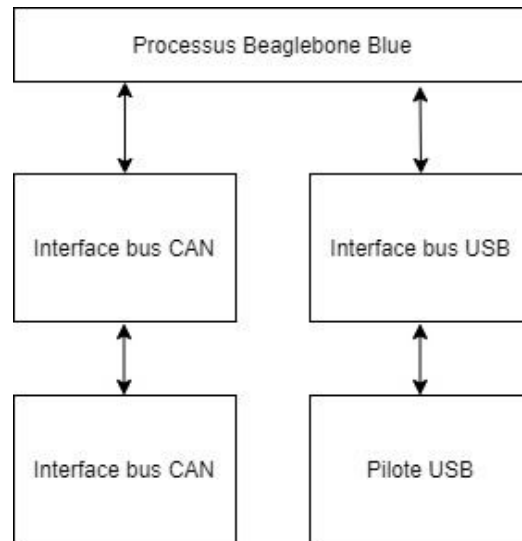


Figure 5: Architecture logicielle du *Beaglebone* du poste de commande

Machines à états

Les machines à états suivantes présentent le fonctionnement du *Beaglebone Blue*. La seconde figure celui du PC.

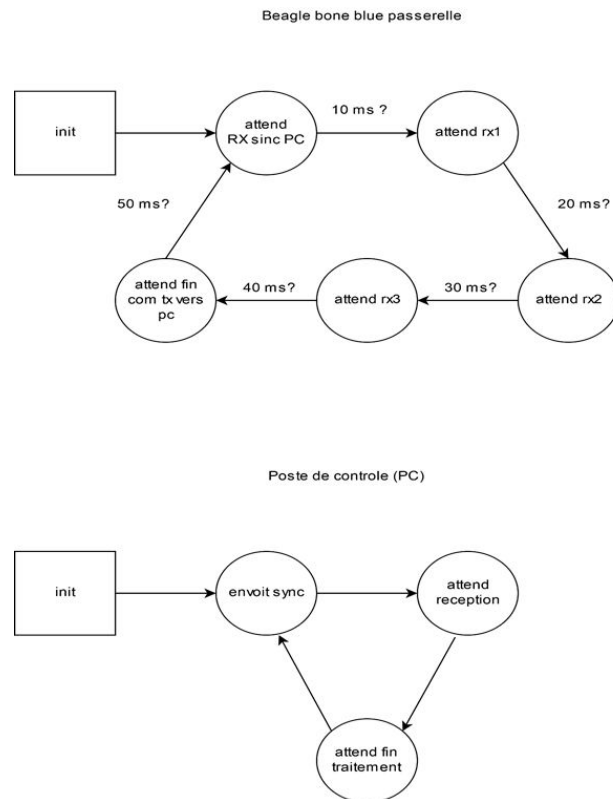


Figure 6 : Diagramme d'états du *Beaglebone Blue* passerelle

Interface graphique

L'interface graphique véhicule plusieurs informations concernant le déroulement du projet en temps quasi-réel. Il permet de contrôler l'usine, par exemple pour l'arrêter ou la démarrer. Parmi les informations affichées, nous pouvons savoir si le véhicule est chargé, quelle est la couleur du bloc, le poids du bloc et l'unité de la balance.

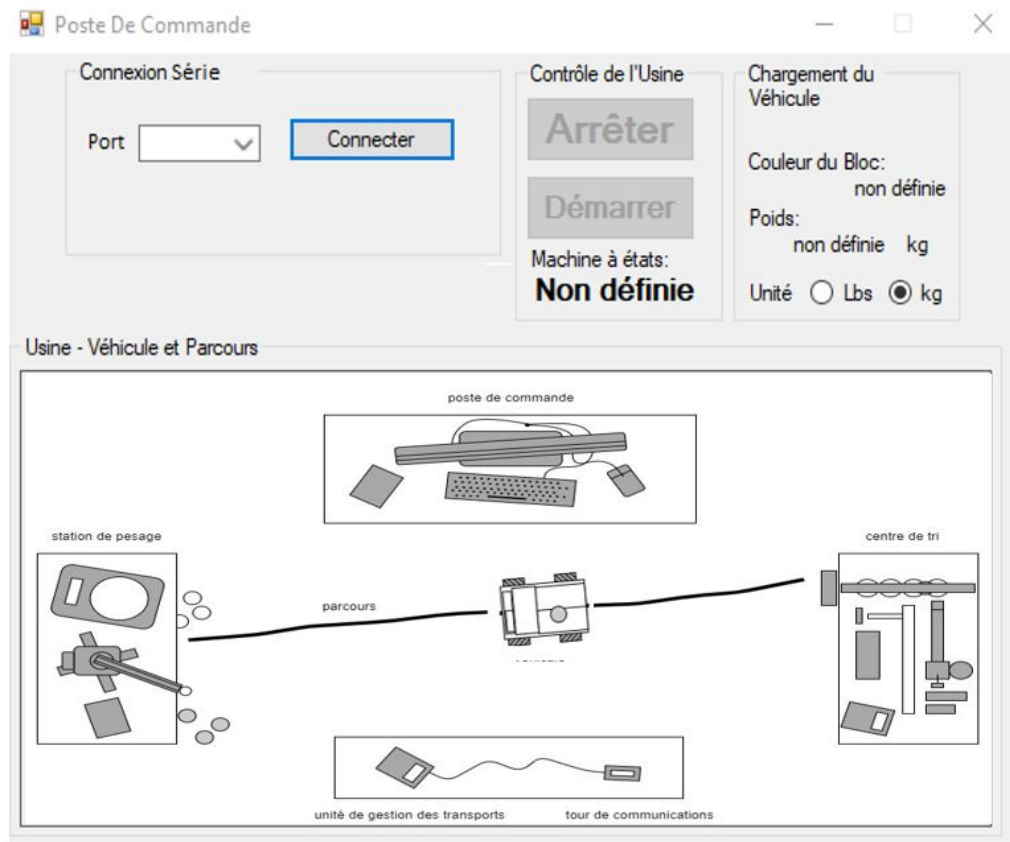


Figure 7 : Représentation de l'interface graphique

Centre de tri

Architecture matérielle

Le centre de tri constitue le cœur de l'usine et se compose de plusieurs capteurs, témoins lumineux et d'autres actuateurs mécaniques. La figure 8 présente l'architecture matérielle du centre de tri.

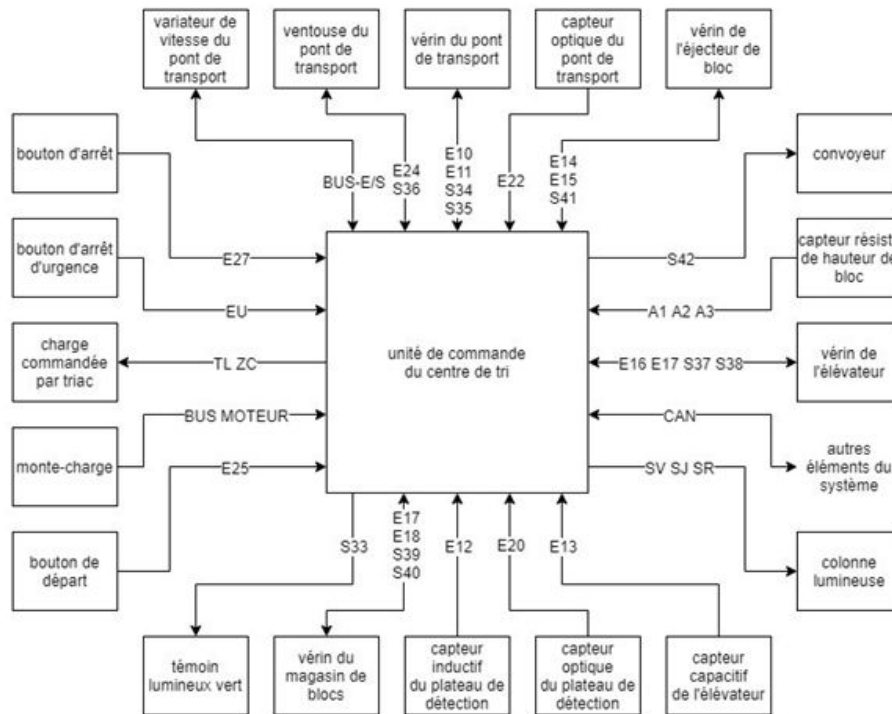


Figure 8 : Architecture matérielle du centre de tri

La figure 9 présente l'architecture matérielle de l'unité de commande qui est attachée au centre de tri.

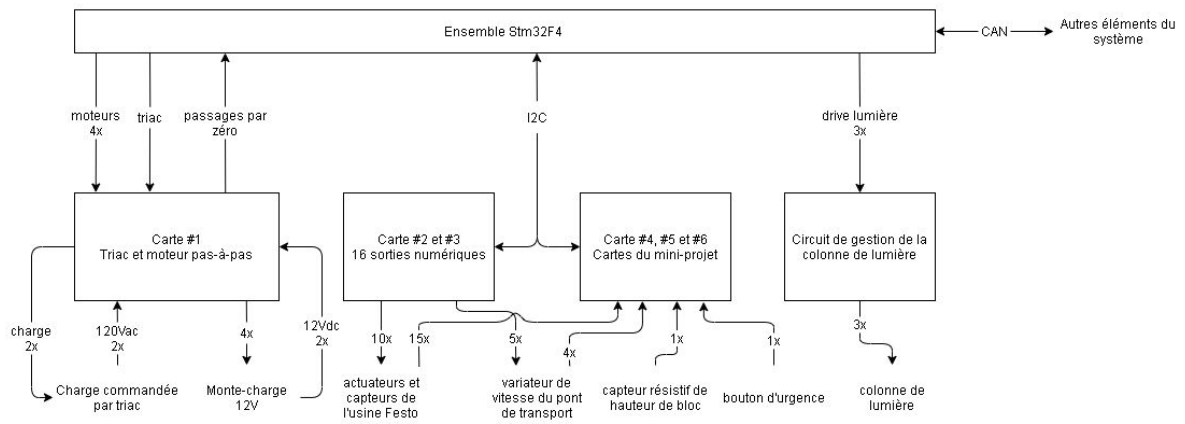
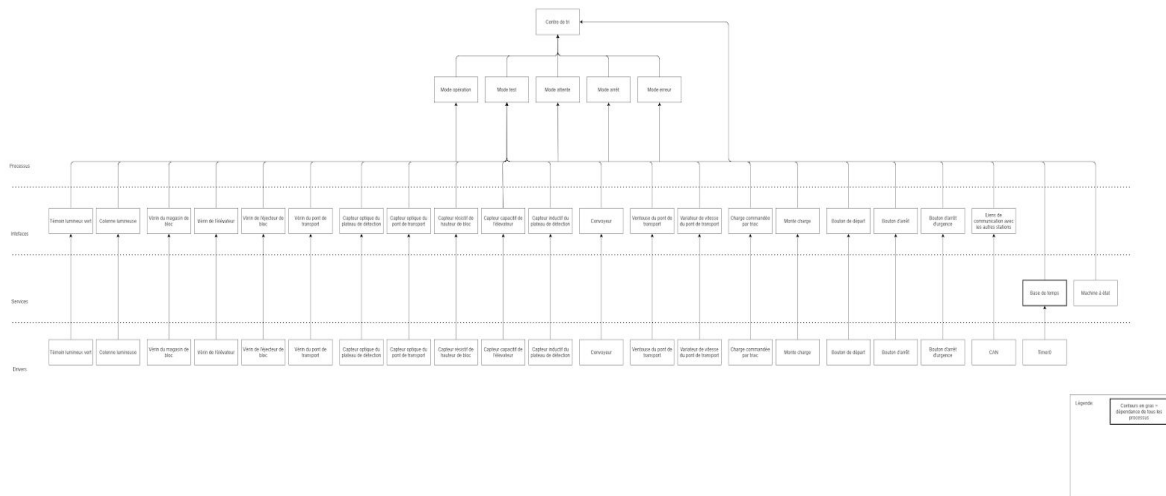


Figure 9 : Architecture matérielle de l'unité de commande du centre de tri

Architecture logicielle

Machines à états

Figure 10 : Architecture logicielle du centre de tri



La figure suivante présente la machine à états de la gestion des modes d'opération du centre de tri.

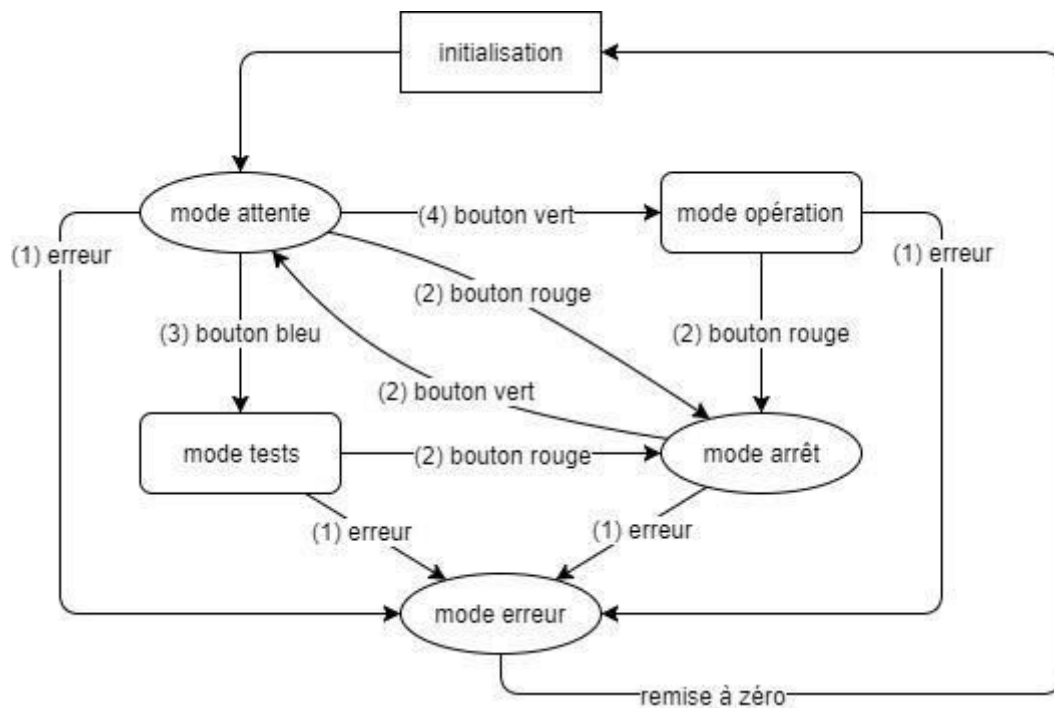
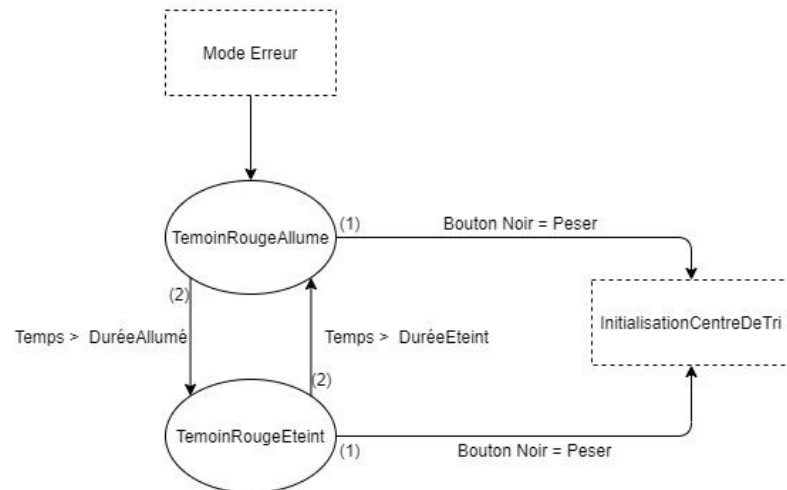


Figure 11 : Machine à états de la gestion des modes d'opération du centre de tri

Le programme du centre de tri aura aussi à gérer les erreurs. Cette gestion d'erreur est confinée dans le graphique de la figure 12 et est diffusée par CAN aux autres unités du



circuit.

Figure 12: Machine à états du mode erreur du centre de tri

D'abord le mode attente dont la machine à état est représenté à la figure 13, est le mode par défaut du centre de tri dont il ne sortira qu'à l'appuie d'un bouton ou à la détection d'une erreur.

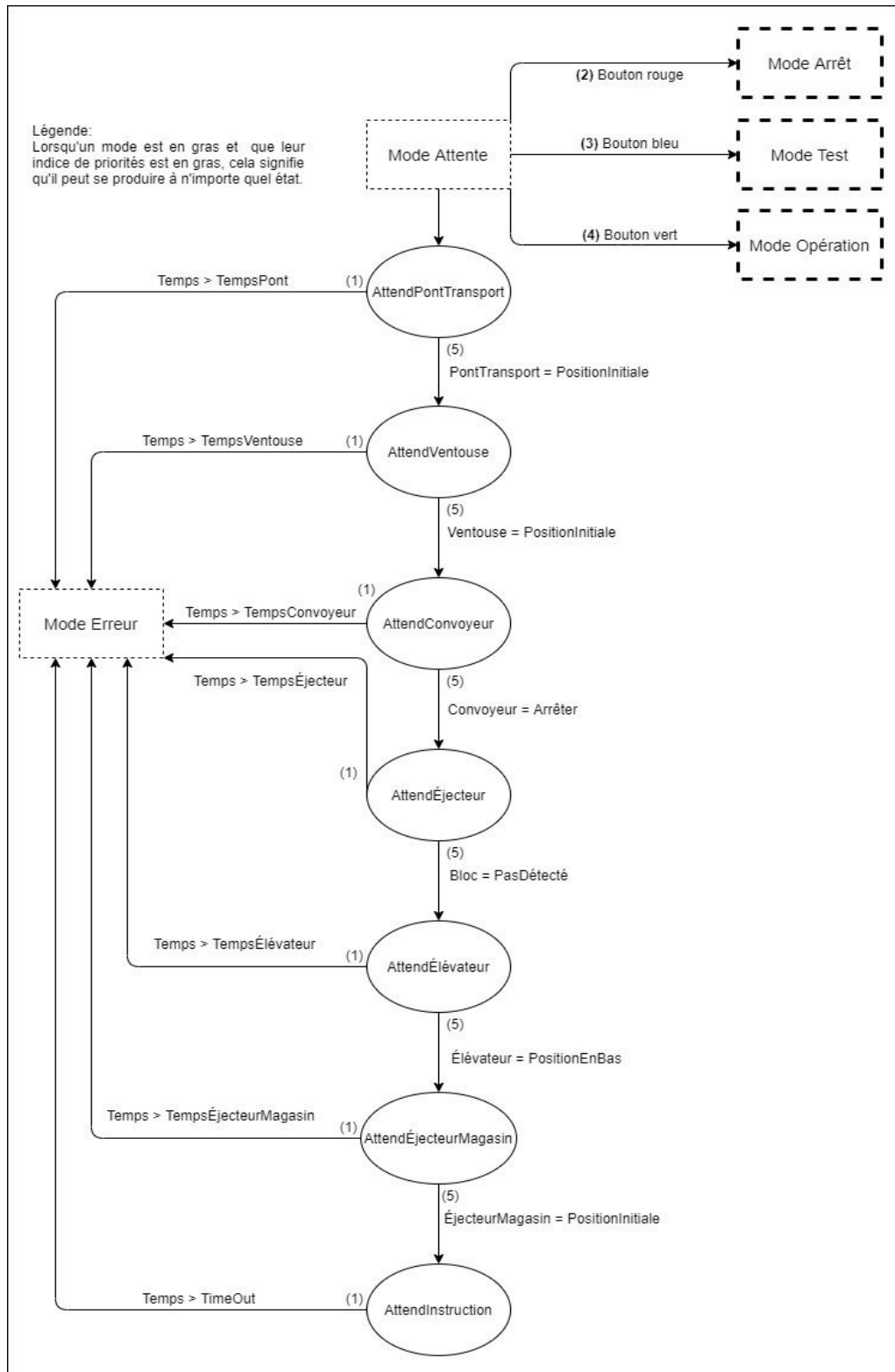


Figure 13 : Machine d'états du mode d'attente du centre de tri

Sinon, à l'appuie du bouton vert, le centre se mettra à opérer, c'est-à-dire faire ses actions principales et le diagramme qui y correspond se trouve à la figure 14.

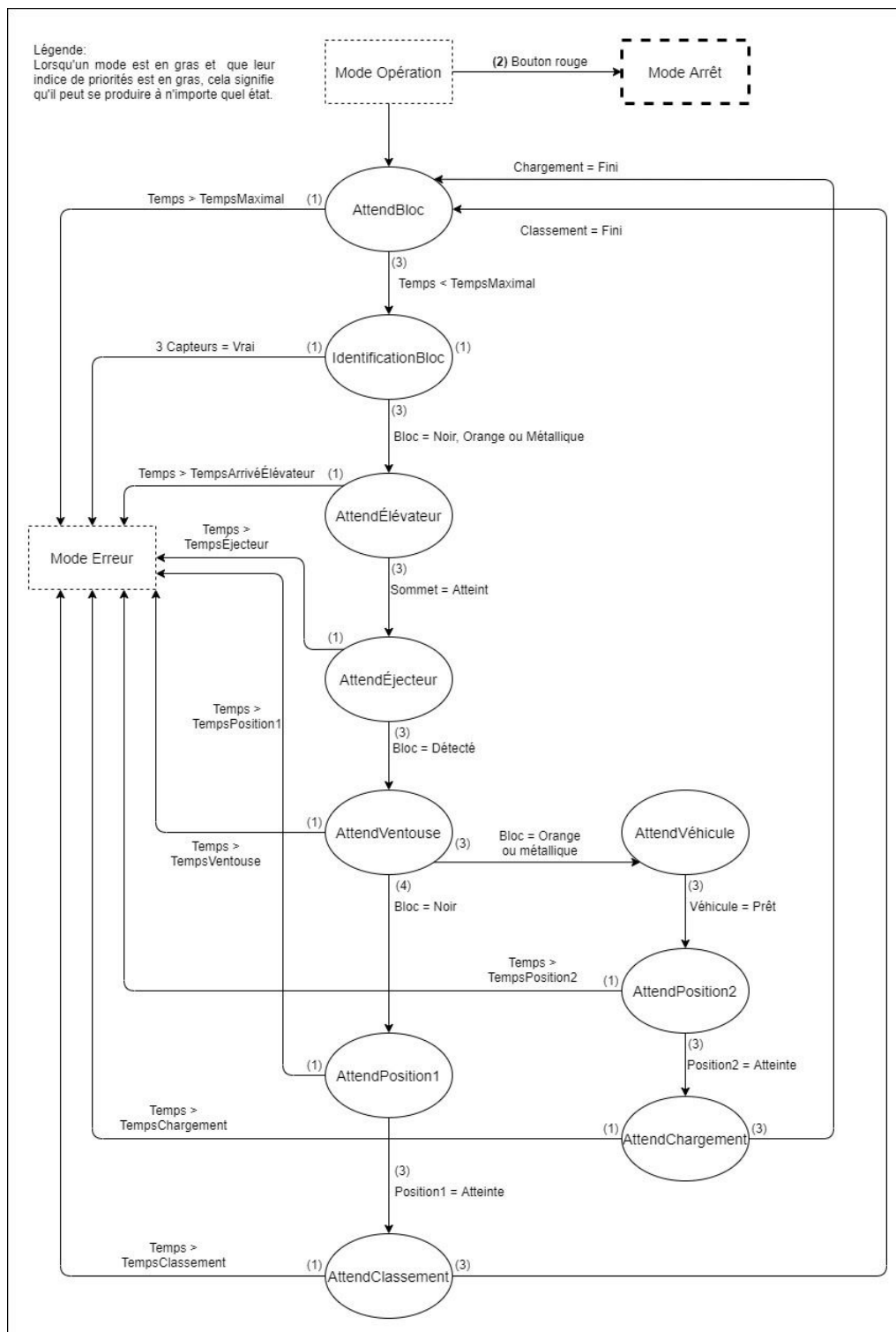


Figure 14 : Machine à états du mode d'opération du centre de tri

L'appuie du bouton rouge met le programme dans un mode arrêt où il ne fait pas grand chose. Nous nous référerons à la figure 15 pour voir de quoi il s'agit.

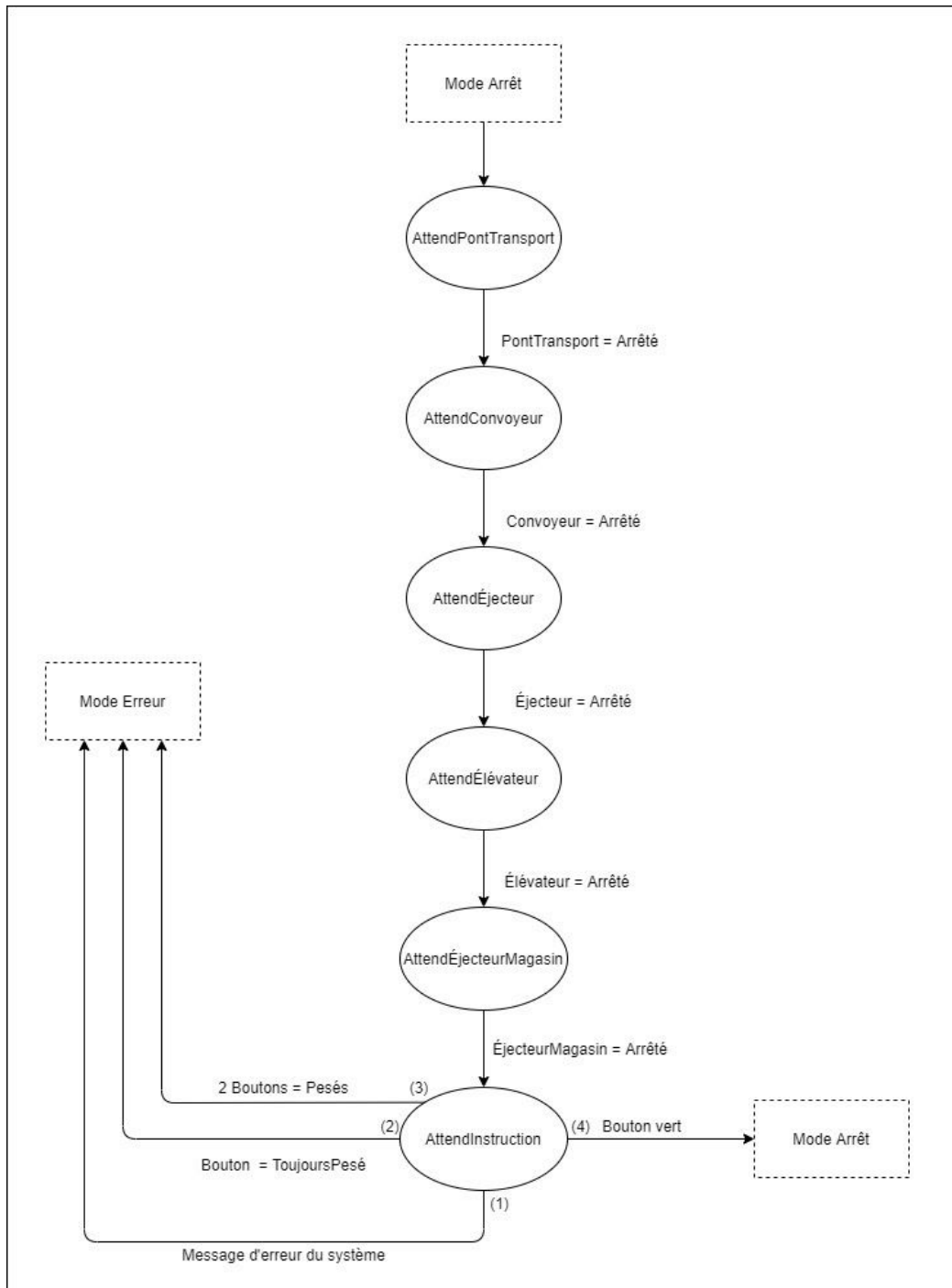


Figure 15 : Machine à états du mode d'arrêt du centre de tri

Le mode qui permettra de vérifier le fonctionnement de cette station. Il nous faudra appuyer sur le bouton vert pour passer d'un mode à l'autre et les tester chacun comme l'illustre la figure 16.

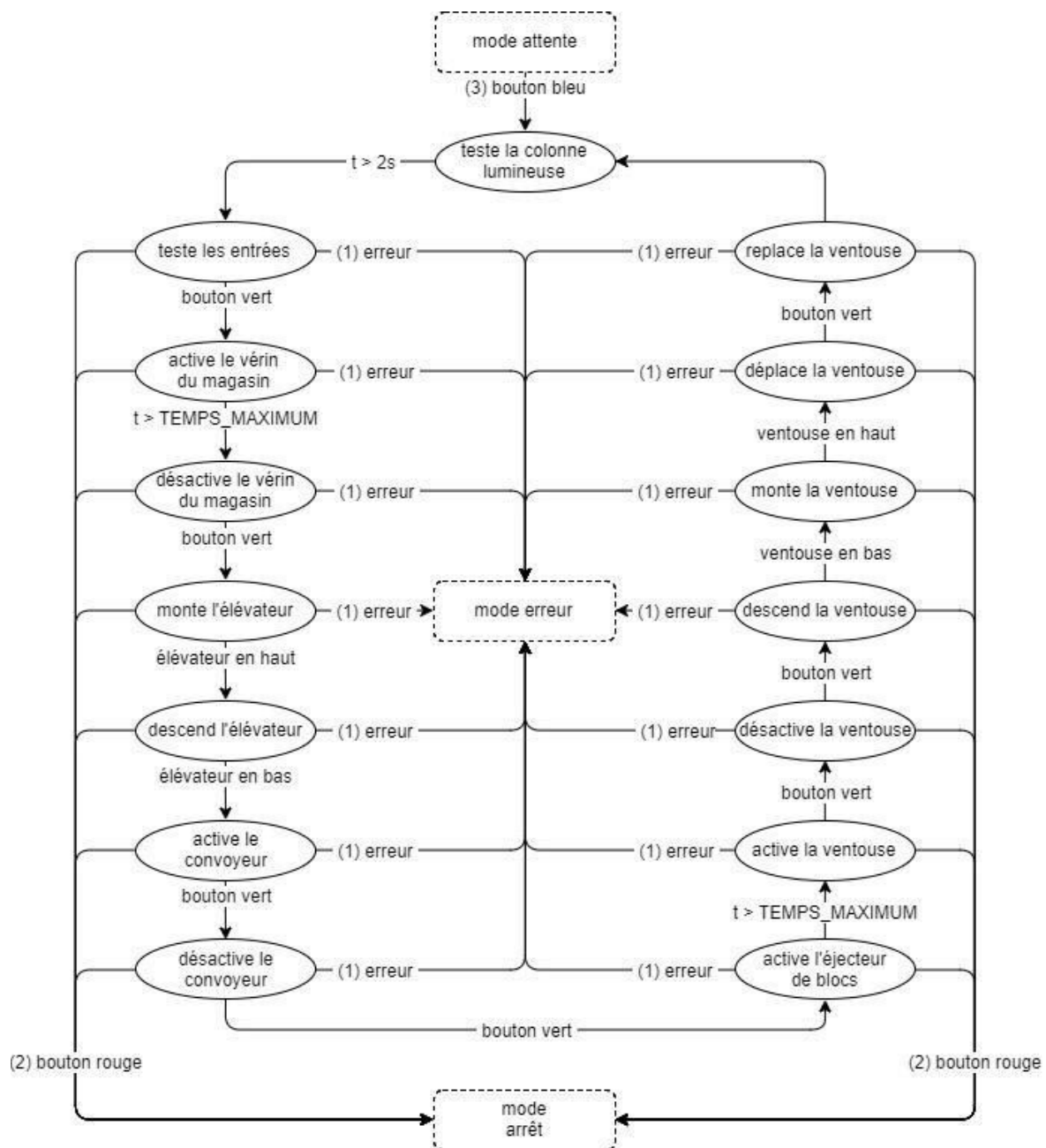


Figure 16: Machine à état du mode de test du centre de tri

Enfin, les leds sont très importantes dans le centre de tri car elles nous permettront de connaître l'état de l'exécution des modes. La machine à état de la figure 17 montre comment elles seront gérées.

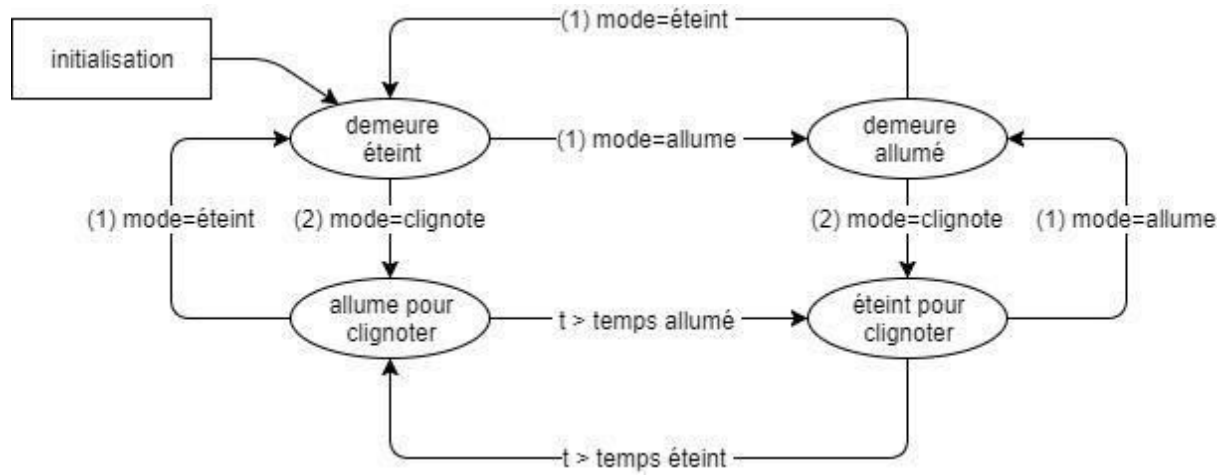


Figure 17: Machine à état de la gestion du fonctionnement des leds

Unité de gestion des transports et de la tour de communication

Architecture matérielle

L'unité de gestion des transports et de la tour de communication est constituée de deux modules principaux : de la STM32 en tant qu'unité de gestion des transports et de l'ESP32 en tant que tour de communications. La figure suivante représente les modules leurs liens.

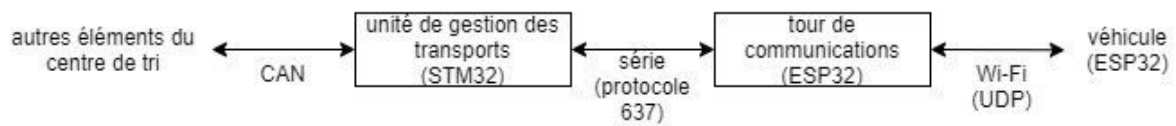


Figure 18: Architecture matérielle de l'unité de gestion des transports et de la tour de communications

Architecture logicielle

Le STM32 qui fait office de passerelle pour la communication CAN entre les différentes stations et fait le relais avec un protocole série RS-232 vers le ESP32 n'a pas de transducteur. Cette conception simplifie le matériel, mais un STM32 ne serait pas requis. L'architecture logicielle est montrée dans la figure suivante.

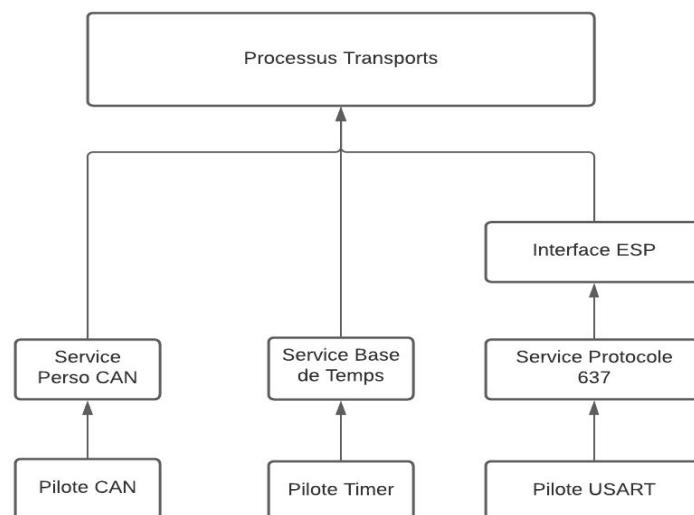


Figure 19: Architecture logicielle de l'unité de gestion des transports

Le ESP8266 passerelle communique par protocole série avec le STM32 et relaie les communications Wi-Fi en envoyant et recevant des datagrammes UDP par envoi massif de salves. L'architecture logicielle est présentée sur la figure suivante.

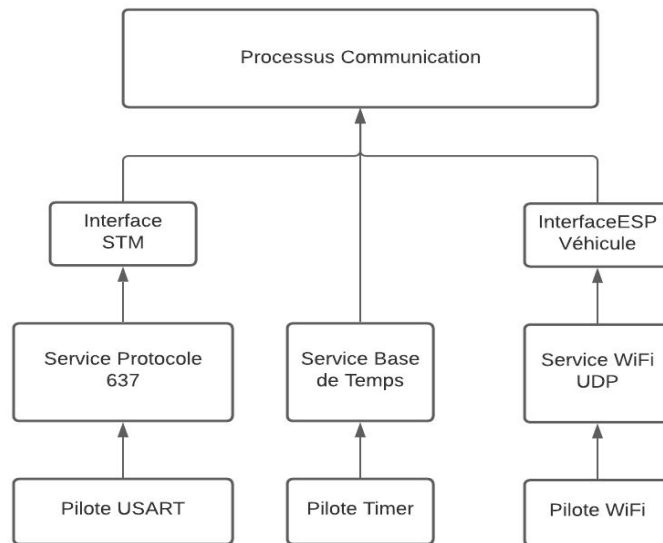


Figure 20: Architecture logicielle de la tour de communication

Machines à états

La machine d'états du STM32 est décrite par la figure 21.

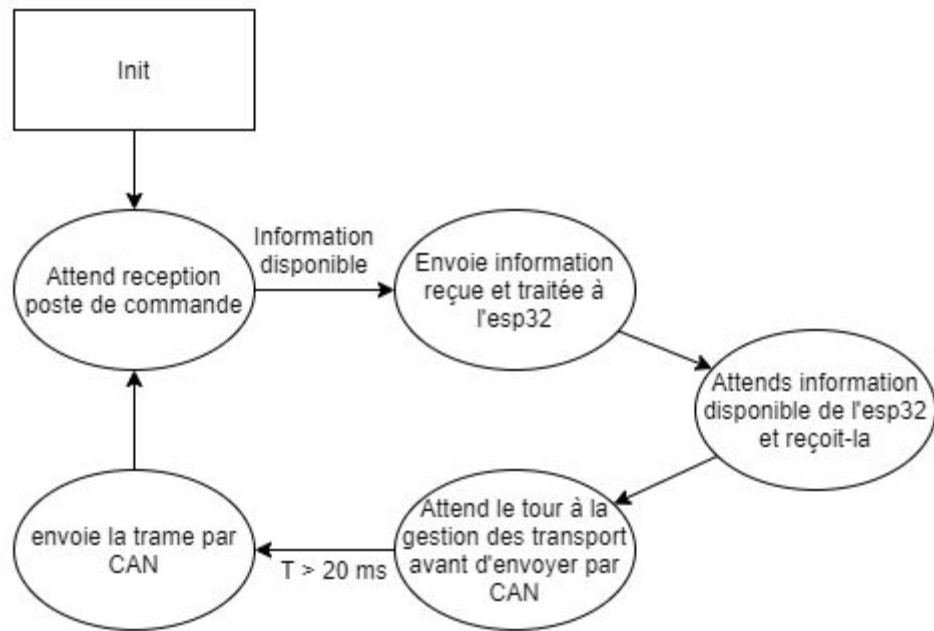


Figure 21: Machine à états de la STM32

La gestion des transports est assurée par un ESP32 qui effectue les actions qui correspondent à la machine à états de la figure 22.

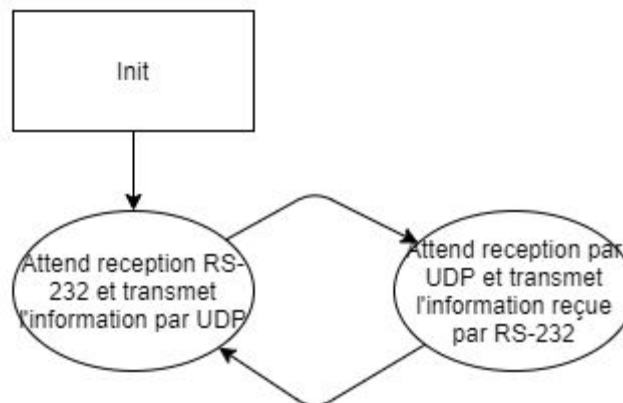


Figure 22 : Machine à états du ESP32

Véhicule

Architecture matérielle

Le véhicule transporte les blocs. Un vecteur de capteurs permet la détection d'une bande réfléchissante placée au sol définissant son parcours. L'alimentation est assurée par une batterie 3.7 V avec un débit de 110 mAh. Un pont en H gère les moteurs. Le microcontrôleur est un ESP32 HUZZAH. La figure 23 illustre l'architecture matérielle du véhicule.

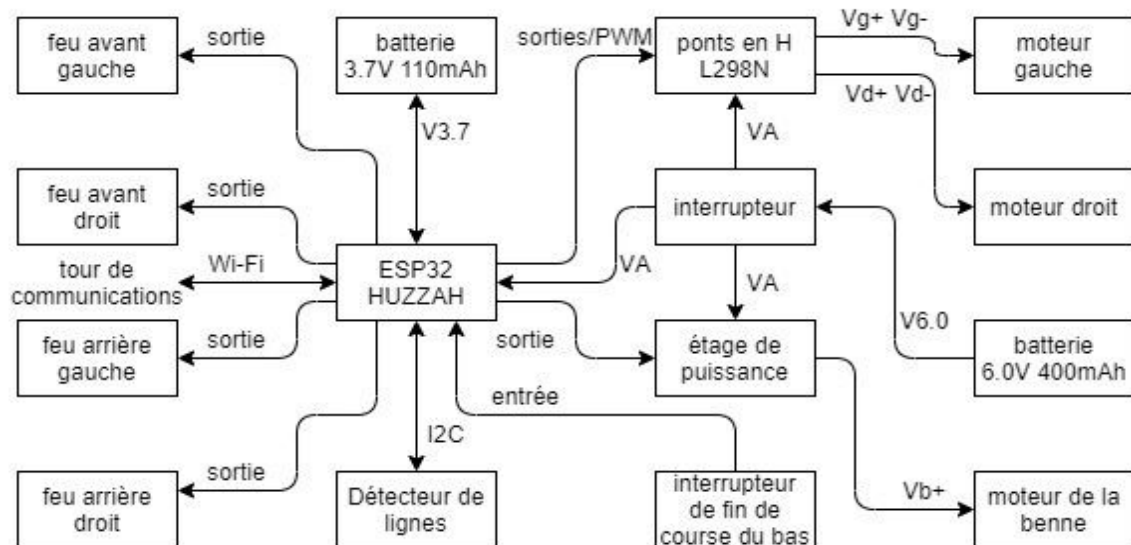


Figure 23 : Architecture matérielle du véhicule

Architecture logicielle

Le véhicule communique avec le reste du système par UDP via le Wi-Fi. Des bits du message permettent de connaître la fonction courante: erreur, déplacement, marche/arrêt et position. La figure 24 illustre l'architecture logicielle du véhicule.

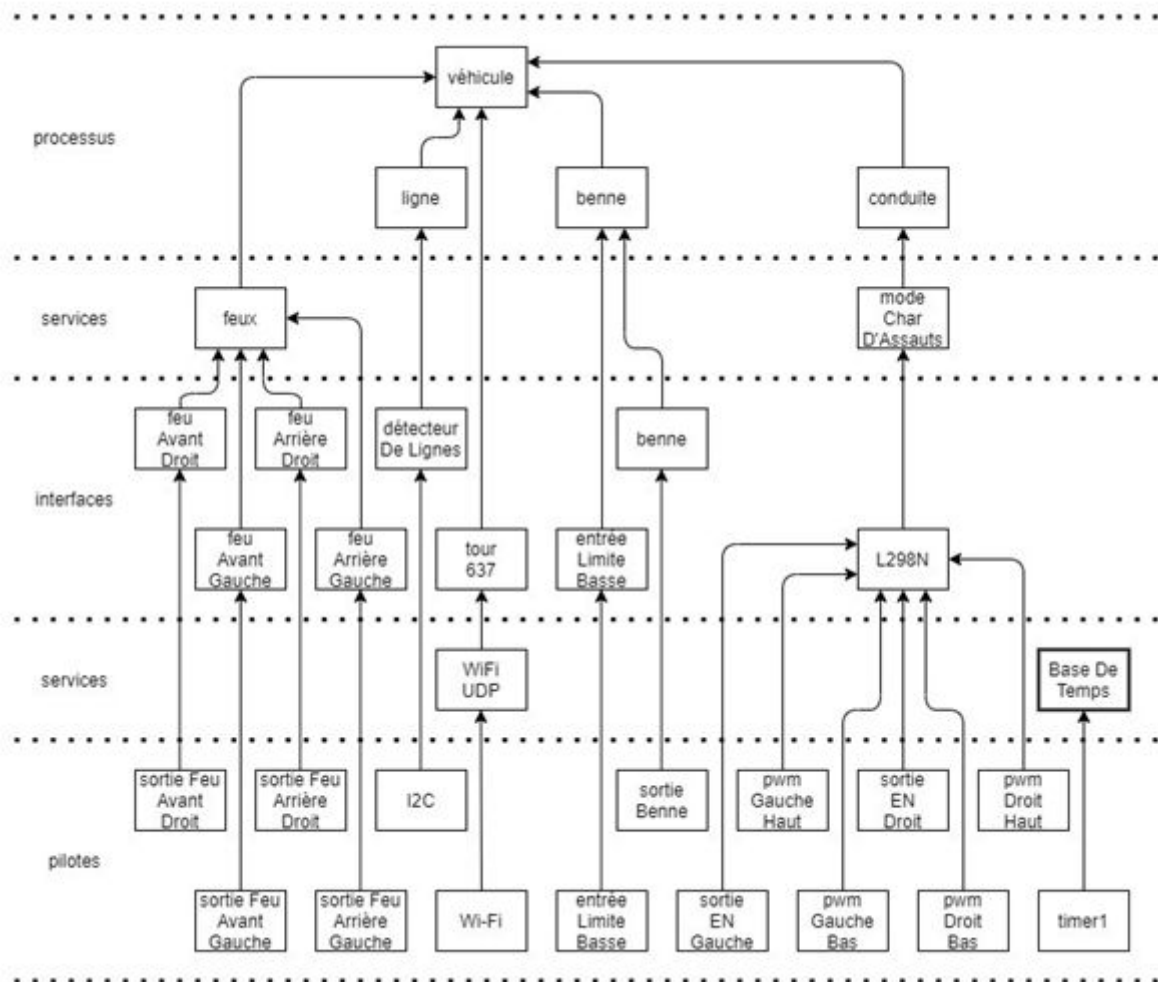
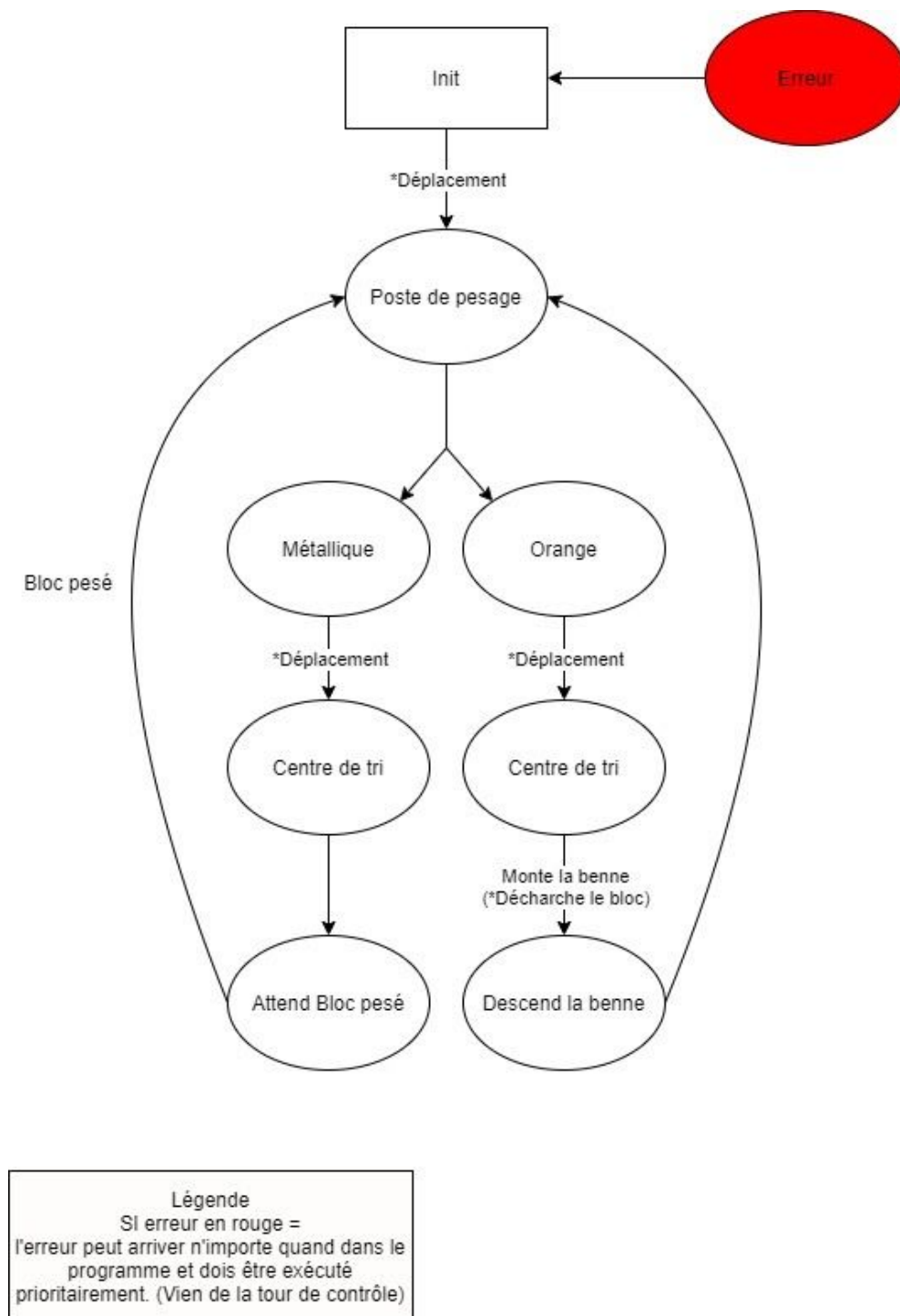


Figure 24 : Architecture logicielle du véhicule

Machines à états

La machine à état ci-dessous décrit le transport des blocs. En gros, le premier déplacement du camion sera d'aller au poste de pesage, par la suite, il va aller au centre de tri pour soit pesé le bloc (pour le bloc métallique) ou monter sa benne pour laisser le bloc par terre (bloc orange) et par la suite, retourner au poste de pesage.



"Provient***"

Figure 25 : Machine à états du fonctionnement du véhicule

À la figure 26, une machine à état montre comment le véhicule communique avec la tour de communication. Cette machine à état fonctionnera en parallèle avec la machine à état de la figure 25. La machine à état de la figure 25 montre que le véhicule attend constamment une

réception en provenance de la tour de communication. Après avoir reçu l'information, elle envoie directement de l'information, contenu dans le véhicule, à la tour de communication, pour ensuite la transférer comme information utilisable par les autres stations en protocole CAN.

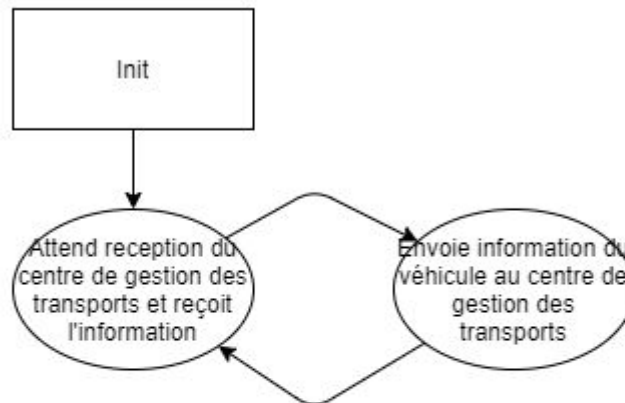


Figure 26: Seconde machine à états du véhicule pour la communication du ESP32 avec la tour de communication

Station de pesage

Architecture matérielle

La station de pesage est composée de plusieurs éléments tel que montre la figure 27.

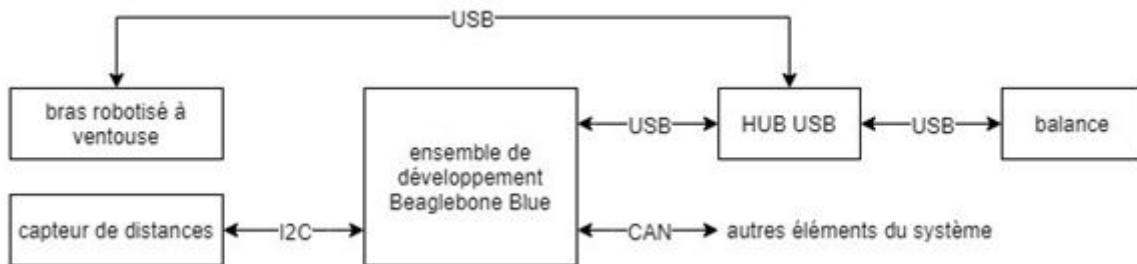


Figure 27 : Architecture matérielle de la station de pesage

Architecture logicielle

La figure suivante présente l'architecture logicielle de la station de pesage.

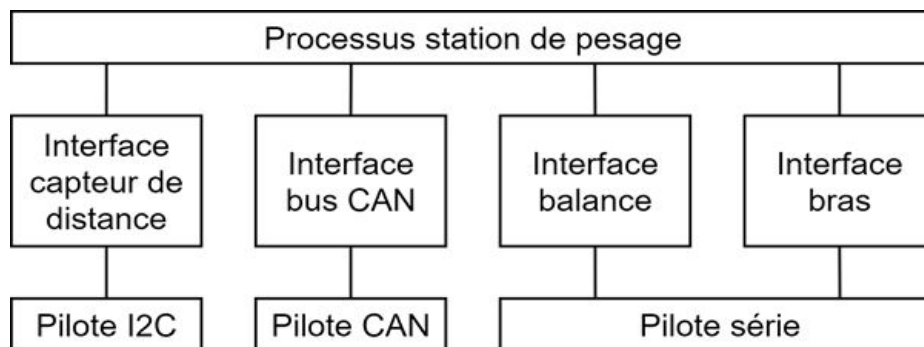


Figure 28 : Architecture logicielle de la station de pesage

Machines à états

La figure suivante présente le fonctionnement de la station de pesage. Premièrement, il attend la couleur, par la suite, attend le véhicule. Lorsque le véhicule est arrivé, il attend le bloc, par la suite, attend que le bloc soit pesé et après le bloc est entreposé.

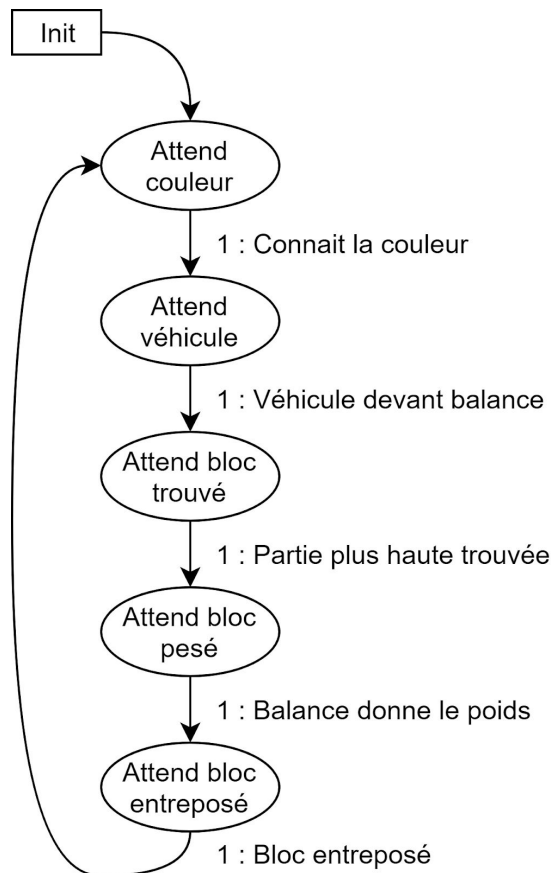


Figure 29: Diagramme d'états de la station de pesage

Connexions électriques

Ce tableau... + lien

NUMÉRO DU BORNIER	TYPE IN OUT (vu du Festo)	DESCRIPTION CARTE I2C In/Out
10	OUT	CYLINDRE DE LA VENTOUSE HAUT
11	OUT	CYLINDRE DE LA VENTOUSE BAS
12	OUT	DÉTECTEUR MAGNÉTIQUE PLATEAU DÉTECTION TYPE BLOC
13	OUT	DÉTECTEUR CAPACITIF PLATEAU DÉTECTION TYPE BLOC
14	OUT	EJECTEUR BLOC POSITION ENTRÉE
15	OUT	EJECTEUR BLOC POSITION SORTIE
16	OUT	ÉLÉVATEUR BAS
17	OUT	ÉLÉVATEUR HAUT

18	OUT	POUSSOIR DU MAGASIN DE BLOC POSITION ENTRÉE
19	OUT	POUSSOIR DU MAGASIN DE BLOC POSITION SORTIE
20	OUT	DÉTECTEUR OPTIQUE PLATEAU DÉTECTION TYPE BLOC
22	OUT	DÉTECTEUR OPTIQUE DE PRÉSENCE DE BLOC À LA CHUTE
25	OUT	BOUTON DÉPART
27	OUT	BOUTON ARRÊT
29	OUT	INDICATION DE LA PRESSION DE LA VENTOUSE
1	IN (5V)	DÉTECTEUR HAUTEUR BLOC (résistance 1k entre les bornes 1 et 3)
2	OUT	DÉTECTEUR HAUTEUR BLOC (centre de la résistance variable de 1K)
3	IN (GND)	DÉTECTEUR HAUTEUR BLOC (résistance 1k entre les bornes 1 et 3)
33	IN	LUMIÈRE VERTE PANNEAU AVANT

34	IN	CYLINDRE DE LA VENTOUSE BAS solénoïde 704
35	IN	CYLINDRE DE LA VENTOUSE HAUT solénoïde 707
36	IN	VACUUM ON solénoïde 710
37	IN	ÉLÉVATEUR POSITION BASSE solénoïde 713
38	IN	ÉLÉVATEUR POSITION HAUTE solénoïde 716
39	IN	POUSSOIR DU MAGASIN DE BLOC ENTRÉ solénoïde 719
40	IN	POUSSOIR DU MAGASIN DE BLOC SORTIE solénoïde 722
41	IN	EJECTEUR BLOC POSITION SORTIE solénoïde 725
42	IN	RELAIS OMRON 24V MOTEUR CONVOYEUR

Tableau 3: description de l'usage des entrées/sorties associées au variateur de vitesse de l'usine *Festo*

Colonne de trop (ou remplir).

NUMÉRO DU BORNIER	TYPE IN OUT (vu du variateur de vitesse de l'usine Festo)	DRIVE MOTEUR VIS SANS FIN CARTE MULTI I/O
47		DRIVE MOTEUR MODE
48		DRIVE MOTEUR STOP
56	IN	SÉLECTION POSITION 0
57	IN	SÉLECTION POSITION 1
58	IN	SÉLECTION POSITION 2
59		SÉLECTION POSITION 3
60	IN	END STAGE ENABLE
61		CONTROLLER RELEASE

62		LIMIT SWITCH 0
63		LIMIT SWITCH 1
64	IN	START POSITION PROCESS
65		MODE
68	OUT	READY FOR OPERATION
69	OUT	DEFAULT MOTION COMPLETE
71	OUT	DEFAULT ACK START
72	OUT	DEFAULT ERROR
73-74	GND	Alimentation 0 Volt
75-76	24 V	Alimentation 24 Volts

Schéma des connexions

Ci-dessous est représenté le schéma des connexions physiques du centre de tri avec la stm32 qui gère ce centre de tri.

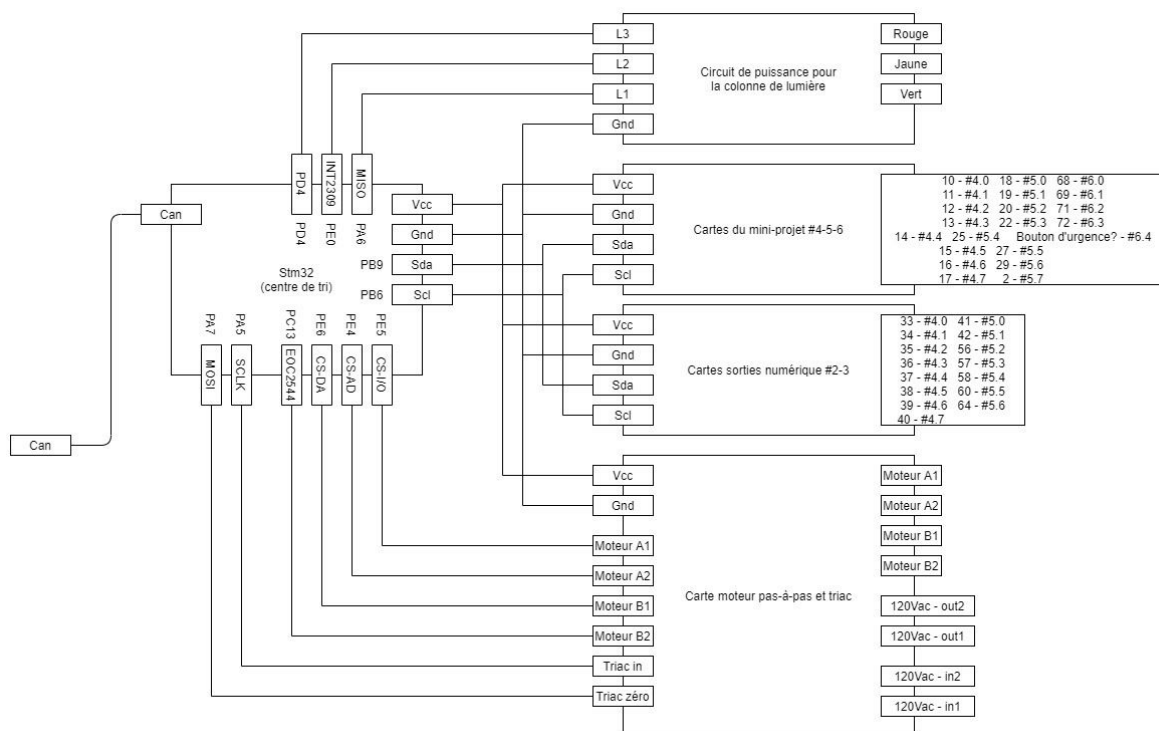


Figure 30: schéma des circuits de connexion du centre de tri

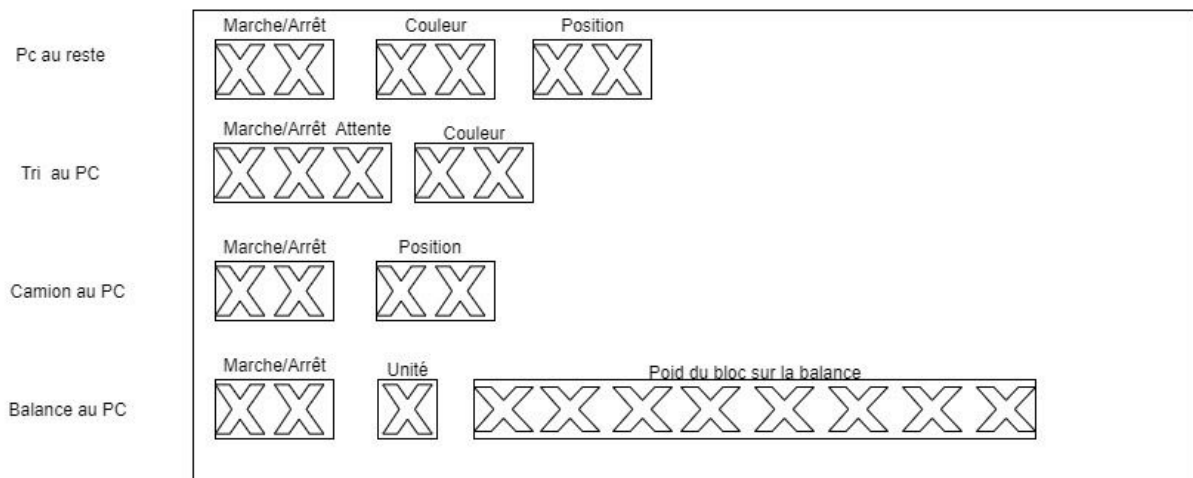
Protocole de communication CAN

Le protocole CAN met en application le multiplexage pour raccorder à un même bus un grand nombre de calculateurs qui communiquent à tour de rôle. Cette technique élimine le besoin de câbler des lignes dédiées pour chaque information à faire transiter (connexion point-à-point). Dès qu'un système (voiture, avion, réseau téléphonique, etc.) atteint un certain niveau de complexité, l'approche point-à-point devient impossible à cause de

l'immense quantité de câblage à installer et de son coût (en masse, matériaux, main d'œuvre, maintenance).

Toutes les stations (autres que le poste de commande), bien que toutes connectées entre-elles sur le bus CAN, n'écoutent que les messages qui proviennent du poste de commande. Le poste de commande écoute tous les messages qui proviennent des autres stations. Le poste de commande sert donc de point central dans la communication. Il enregistre toutes les informations contenues dans les transmissions CAN, et les distribuent pour toutes les autres stations.

Trame Can



Légende:

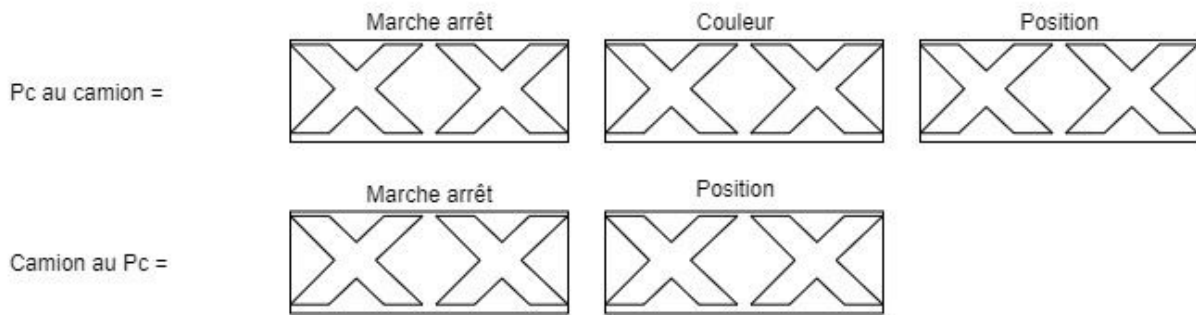
Marche= 01(0)
 Arrêt =00(0)
 Erreur = 10(0)
 Test = 11(0)
 Attente= 001
Couleur :
 Pas de bloc = 00
 orange = 01
 noir = 10
 métallique = 11
Position :
 Position 1 = 00
 Position 2 = 01
 Position 3 = 10
 Position 4 = 11
Unité :
 Grammes= 0
 Onces = 1
Poid de la balance
 Unité de poid (Unsigned char)

Protocole de communication par UDP

Le protocole User Datagram Protocol (UDP) est défini dans le but de fournir une communication par paquet unique entre deux processus dans un environnement réseau étendu. Ce protocole suppose l'utilisation du protocole IP comme support de base à la communication. Ce protocole définit une procédure permettant à une application d'envoyer un message court à une autre application, selon un mécanisme minimaliste. Ce protocole est transactionnel et ne garantit ni la délivrance du message, ni son éventuelle duplication. Les applications nécessitant une transmission fiabilisée et ordonnée d'un flux de données implémenteront de préférence le protocole TCP.

Le protocole de communication par UDP est simple car il communique seulement avec le camion. En effet, il a seulement besoin de savoir comme toutes les autres stations le (Marche arrêt), la couleur du bloc pour savoir quoi faire et la position pour montrer en direct sur l'interface graphique.

Trame UDP



Légende:

Marche= 01

Arrêt =00

Erreur = 10

Test = 11

Couleur :

Pas de bloc = 00

orange = 01

noir = 10

métallique = 11

Position :

Position 1 = 00

Position 2 = 01

Position 3 = 10

Position 4 = 11

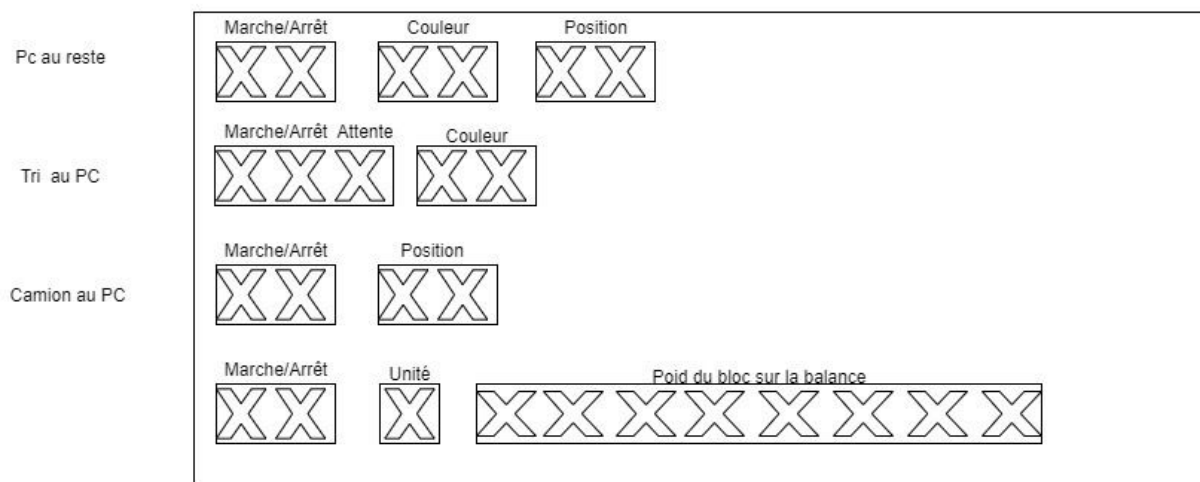
Protocole de communication RS-232

Le protocole RS-232 est un protocole série. asynchrone et duplex. Pour établir une communication via RS-232, il est nécessaire de définir le protocole utilisé. Le débit de la transmission, le codage utilisé, le découpage en trame, etc. La norme RS-232 laisse ces points libres, mais en pratique on utilise souvent des UART qui découpent le flux en trames d'un caractère constituées d'un bit de départ, de sept à huit bits de données, d'un bit de parité optionnel et d'un ou deux bits d'arrêt. Le bit de départ a un niveau logique de zéro alors que le bit d'arrêt est de niveau logique un. Le bit de données de poids faible est envoyé en premier, puis les autres s'ensuivent. Pour générer un signal électrique alternatif carré avec un rapport cyclique un à un sur le port série, il faut imprimer une suite consécutive de zéros et de uns alternés (caractère "U", ce qui donne dans le temps zéro (le bit de départ) 0xAA, le bit d'arrêt (un) et zéro bit de parité. Les niveaux électriques sont inversés. La spécification RS-232 prescrit des débits inférieurs à 20 000 bit/s. Cependant, les débits utilisés en pratique varient entre 75 bit/s et 115 200 bit/s. Un niveau logique de zéro est représenté par une tension de +3V à +25V et un niveau logique de "un" par une tension de

-3V à -25V (codage NRZ). D'habitude, des niveaux de +12V et -12V sont utilisés. La norme V.28 indique qu'un "1" est reconnu si la tension est inférieure à -3 V, et un 0 est reconnu si la tension est supérieure à +3V.

Communication USB. Le PC envoie toutes les commandes et informations aux autres composantes grâce au USB qui est par la suite converti en CAN... Dans le sens inverse, tous les composants parlent en CAN et par la suite, converti pour passer dans le USB pour que le PC aille tout l'information qu'il a de besoin.

Trame USB PC



Légende:

Marche= 01(0)
 Arrêt =00(0)
 Erreur = 10(0)
 Test = 11(0)
 Attente= 001
Couleur :
 Pas de bloc = 00
 orange = 01
 noir = 10
 métallique = 11
Position :
 Position 1 = 00
 Position 2 = 01
 Position 3 = 10
 Position 4 = 11
Unité :
 Grammes= 0
 Onces = 1
Poid de la balance
 Unité de poid (Unsigned char)

```
#!/bin/bash
```

```
#
```

```
balance.sh
```

```
cat -v < /dev/ttyUSB0 &
```

```
PID=$!
```

```
echo -ne "T\n\r" > /dev/ttyUSB0
```

```
sleep  
1
```

```
echo -ne "Z\n\r" > /dev/ttyUSB0
```

```
sleep  
1
```

```
echo -ne "P\n\r" > /dev/ttyUSB0
```

```
sleep  
5
```

```
echo -ne "T13.37\n\r" > /dev/ttyUSB0
```

```
sleep  
1
```

```
echo -ne "P\n\r" > /dev/ttyUSB0
```

```
sleep  
5
```

```
kill $PID > /dev/null
```

****com pour le bras / balance / capteur de distance****

Planification