

CHAPITRE 3 : Conception du Parallélisme des Chargeurs

Introduction

Dans ce troisième chapitre, nous allons illustrer notre étude concernant la réalisation du parallélisme des chargeurs à savoir les choix et les explications des différentes parties et l'organigramme de gestion du parallélisme.

1. Choix de l'Architecture

On distingue deux Architectures en parallèles disponible:

- Architecture Parallèle / redondant : (batterie commune)
- Architecture Parallèle / redondant : (batterie séparer)

1.1. Architecture Parallèle / Redondant : (batterie commune)

Cette architecture dispose d'une batterie commune. Tous les chargeurs participent à la charge de la batterie en respectant la contrainte du pourcentage suivant la puissance, et en même temps ils alimentent la charge en agissant sur un seul paramètre, la tension.

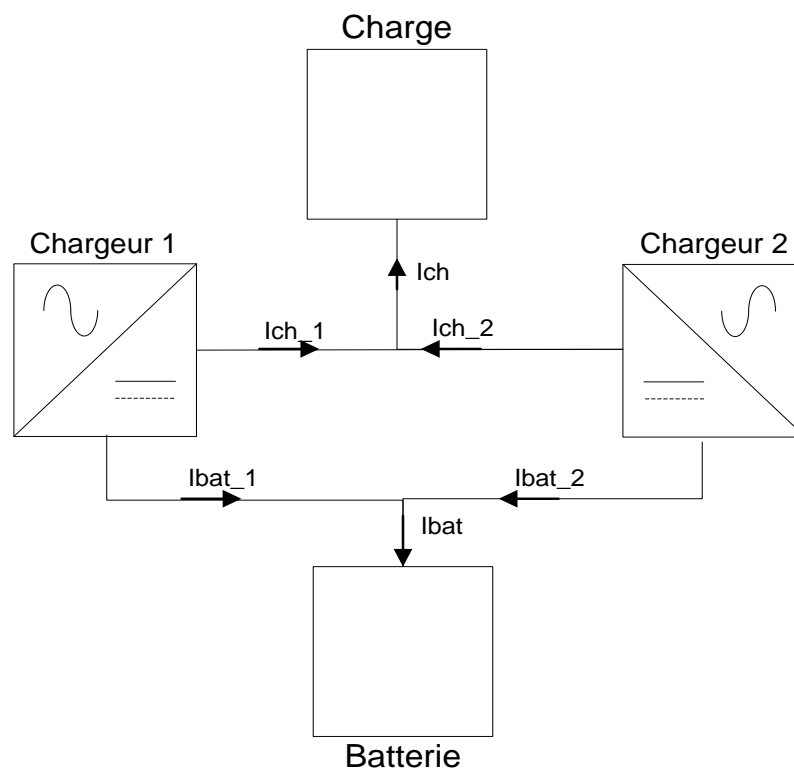


Figure 10 : Architecture Parallèle / Redondant : (batterie commune).

1.2. Architecture Parallèle / Redondant : (batterie séparer)

Quant à l'architecture parallèle/ Redondant avec batterie séparer, chaque chargeur a sa batterie ce qui rend la gestion de la communication pour assurer la distribution du courant plus difficile car on a des profils de charge de batterie différents. Un autre problème s'impose lors de la décharge des

batteries puisqu'on va avoir des courbes de décharge différentes dû à un déséquilibre entre les niveaux de charge des batteries ce qui réduit la durée de vie des batteries.

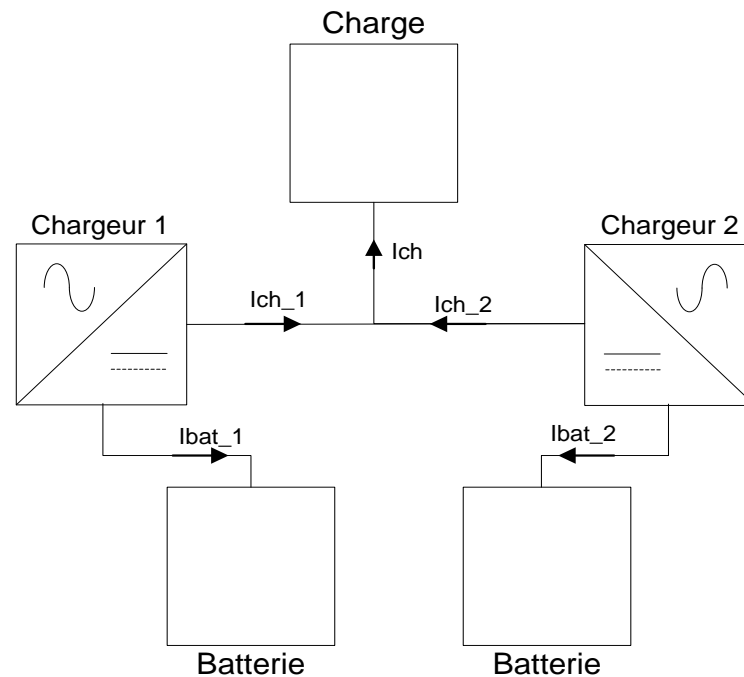


Figure 11 : Architecture Parallèle / redondant : (batterie séparer).

Les deux explications qui précèdent, nous amené à choisir l'Architecture Parallèle / Redondant : (batterie commune) parce qu'elle permet l'augmentation de la durée de vie des batteries et facilite la gestion de communication. Elle est donc la solution la plus convenable.

2. Fonction principale de la communication :

Comme le montre la figure 12, plusieurs chargeurs (monter on parallèle) alimentent la même charge, et chaque chargeur participe à la charge de la batterie suivant un pourcentage déterminé selon sa puissance. Il faut donc gérer la communication entre les chargeurs pour que chaque chargeur reçoit les informations de tous les autres chargeurs afin de connaître le courant de charge qu'il doit délivrer (de même manière pour le courant de la batterie).

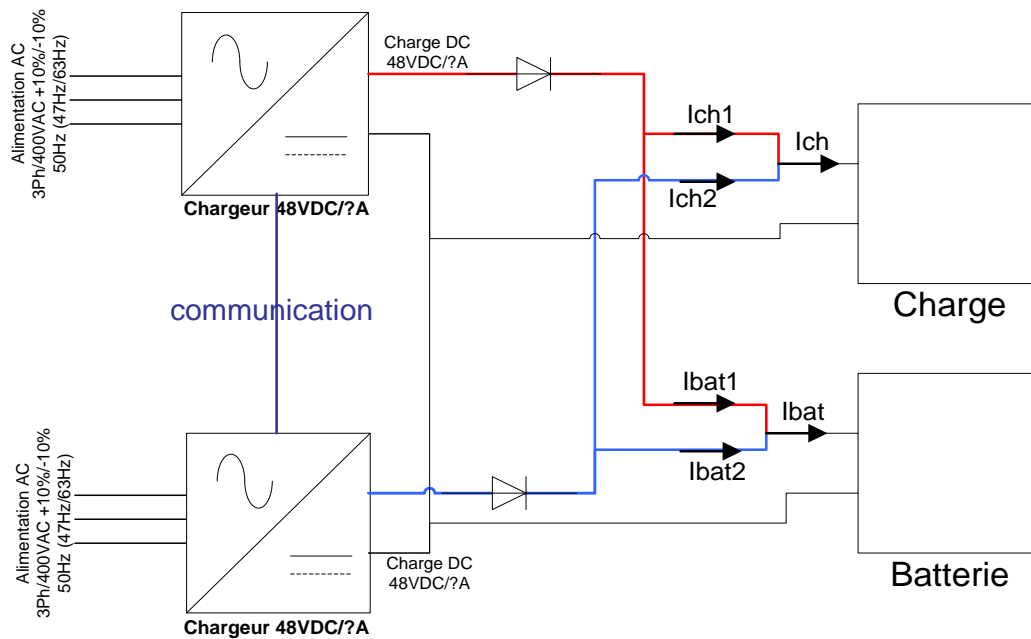


Figure 12 : Schéma de deux chargeurs en parallèle.

On peut varier le courant fourni par chaque chargeur par des petites variations dans sa tension de sortie en variant l'angle d'attaque du thyristor dans le redresseur.

✓ Exemple :

Le premier chargeur à un courant maximal de 100A et le courant maximal du deuxième chargeur est de 50A.

Après l'échange des données entre les chargeurs, supposant que la charge demande un courant de 99A, le premier chargeur doit délivrer un courant de charge égale à 66A;

$$Ich1 = \frac{100}{100 + 50} \times 99 = 66A$$

Avec :

I_{ch1} : Le courant de charge de premier chargeur.

Et le deuxième chargeur fournit un courant de charge égale à 33A ;

$$Ich2 = \frac{50}{100 + 50} \times 99 = 33A$$

Avec :

I_{ch2} : Le courant de charge de deuxième chargeur.

3. Choix de bus de communication

Le choix du bus de communication est affecté par un ensemble de contraintes à savoir :

- La robustesse contre les bruits électromagnétiques car les chargeurs travaillent dans un milieu industriel.
- aucune trame ne peut être perdue, car il peut engendrer le dysfonctionnement du système.
- Fonctionnement multi-maîtres. (Chaque chargeur peut communiquer avec un autre sans intervention des autres chargeurs).

- le nombre des chargeurs dans le bus doit être indifférent.
- Le transfert des données nécessaires pour chaque itération ne doit pas dépasser 10 ms.
- la distance doit être dans l'ordre d'une dizaine de mètres.

3.1. SPI (Serial Peripheral Interface)

La liaison SPI est un bus pour la transmission synchrone de données série entre un maître et un ou plusieurs esclaves (en théorie, il n'y a pas de limitations dans le nombre d'esclaves). La transmission a lieu en Full Duplex, SPI peut être configuré pour fonctionner à un débit supérieure à 10 Mb / s. [1']

Le schéma suivant illustre le bus SPI maître avec trois esclaves indépendants :

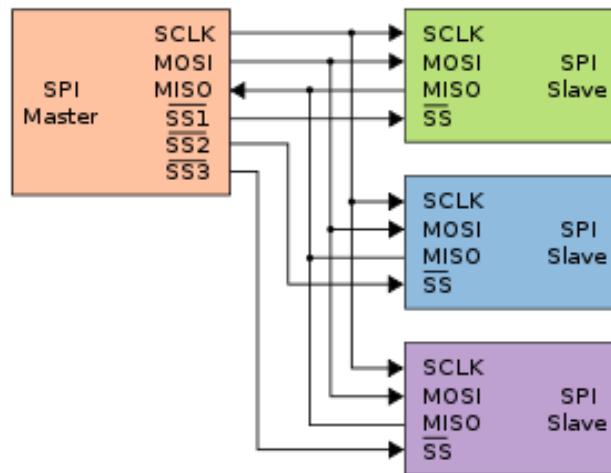


Figure 13 : Bus SPI maître et trois esclaves indépendants.

Les avantages :

- ✓ Communication Full duplex.
- ✓ Débit assez important par rapport aux autres communications série.
- ✓ Flexibilité du nombre de bits à transmettre ainsi que du protocole en lui-même.
- ✓ Simplicité de l'interface matérielle.
- ✓ Aucun arbitre nécessaire car aucune collision possible.
- ✓ Les esclaves utilisent l'horloge du maître et n'ont donc pas besoin d'oscillateur propre. [1']

Les inconvénients :

- ✓ Monopolise plus de broches d'un boîtier.
- ✓ Aucun adressage possible, il faut une ligne de sélection par esclave.
- ✓ Le protocole n'a pas d'acquittement. Le maître peut parler dans le vide sans le savoir.
- ✓ Ne s'utilise que sur de courtes distances (sur le même boîtier).
- ✓ structure maître-esclave. [1']

3.2. Bus I²C (Inter-Integrated Circuit)

Le bus I²C a été défini pour interconnecter des composants placés sur une même carte. Il n'a pas de robustesse vis à vis des perturbations, puisque le retour se fait par la masse. Son emploi est très risqué de carte à carte, à cause des problèmes de mode commun. [2']

Les avantages :

- ✓ Utilisation de 2 lignes seulement pour transmettre les informations.
- ✓ Les données peuvent circuler dans les deux sens sur le bus.
- ✓ Le bus est multi-maître.
- ✓ Chaque équipement relié au bus, dispose d'une adresse codée sur 7 bits, soit une possibilité de connecter 128 équipements.
- ✓ Les composants programmables récents comprennent des fonctions permettant de gérer le protocole I2C. [3]

Les inconvénients :

- ✓ Utilisé pour la communication interne.
- ✓ La moitié des données envoyées sont "utiles". Le reste, ce sont des adresses d'esclaves, des conditions et signaux divers de contrôle de la communication.
- ✓ Le principe de dialogue série est complexe et difficile à s'approprier.
- ✓ Le bus limite le débit et ne permet pas des applications dans lesquelles la vitesse de transmission est importante. (100Kbits maximum, 400Kbits sur des équipements récents). [3]

3.3. L'Interface RS485

L'interface RS485 est une spécification électrique pour des systèmes multipoints qui utilisent des lignes différentielles (la donnée est constituée de 2 tensions d'état contraire sur 2 fils séparés).

Une liaison RS485 peut avoir jusqu'à 32 modules. Chaque module doit avoir une impédance d'entrée de 12kOhms. Si on utilise des récepteurs haute impédance, on peut en connecter jusqu'à 256 sur une même ligne RS485. [3']

Longueur et Débit :

La longueur peut aller jusqu'à 1200 mètres et peut transférer des données jusqu'à 10Mbps mais pas les deux à la fois. Sur 1200 mètres, on peut transférer 90kbps. Sur 100 mètres, on peut arriver à 1Mbps. Pour atteindre 10Mbps, la longueur ne doit pas dépasser 15 mètres. Si on souhaite créer une ligne plus longue, il faut utiliser des répéteurs qui régénèrent le signal et débute une nouvelle ligne. [3']

Les avantages :

- ✓ L'immunité électromagnétique est assurée.
- ✓ Câblage réduit.
- ✓ Ordre des nœuds indifférent.
- ✓ Longue distance.
- ✓ Vitesse important sur des distances de dixième de mètre.

Les inconvénients :

- ✓ structure maître-esclave.

3.4. Bus CAN (Controller Area Network)

Le bus CAN (Controller Area Network) est un bus système série très répandu dans beaucoup d'industries, notamment l'automobile. CAN est un bus de données série bidirectionnel half-duplex. [1']

Longueur et Débit :

La figure ci-après montre l'effet de la longueur sur le débit, plus la longueur de bus augmente plus le débit diminue.

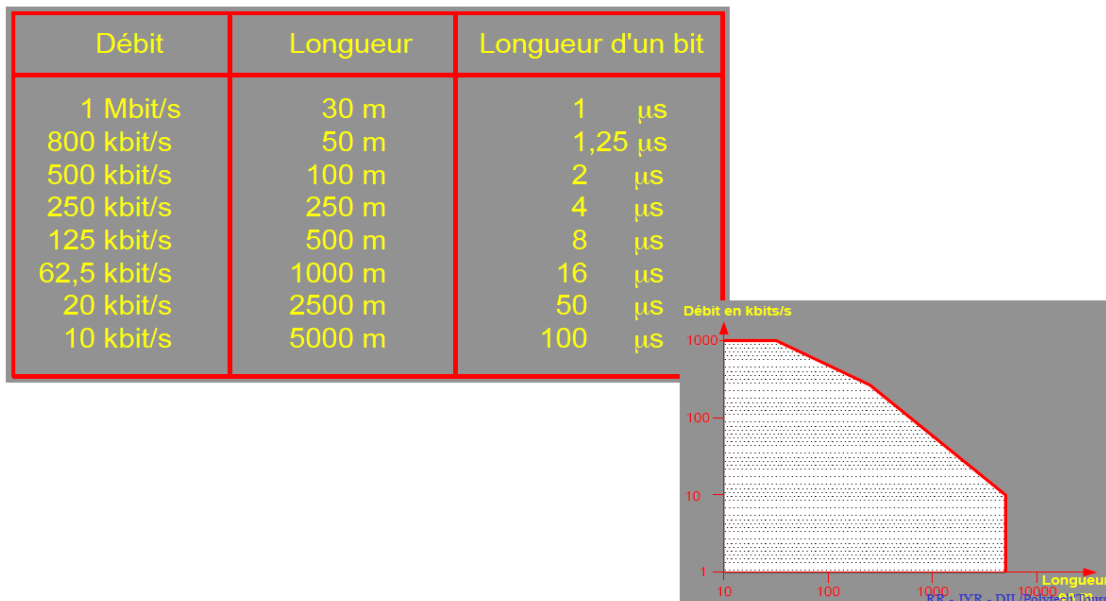


Figure 14 : Longueur de bus et Débit.

Les avantages :

- ✓ L'immunité électromagnétique est assurée.
- ✓ Configuration simple.
- ✓ Longue distance.
- ✓ Câblage réduit.
- ✓ Ordre des nœuds indifférent.
- ✓ Fonctionnement multi-maître. [4]

Les inconvénients :

- ✓ L'ensemble des nœuds est affecté en cas de défaut.
- ✓ Une défaillance est plus délicate à diagnostiquer.
- ✓ Outils de diagnostic et de maintenance spécialisés. [4]

➤ **Bus choisi :**

Parmi les quatre bus précités, et en analysant les avantages et les inconvénients de chacun d'eux, le meilleur choix est posé sur le bus CAN parce qu'il répond mieux aux critères prédéfinis plus haut.

4. Données de communication

Après avoir analysé attentivement les organigrammes existants du fonctionnement du chargeur et pour optimiser les données transmises, nous avons choisi les données illustrées dans le tableau 3.

Donnée	Description
Courant maximum du chargeur	Représente la puissance du chargeur puisque la tension est fixe et presque égale à 48 V. ce courant est utilisé pour calculer le pourcentage de courant de charge qu'il faut assurer.
Courant fournit à la charge	Utilisé pour calculer le courant qu'il doit fournir à la charge.

Tableau 3 : Les données de communication.

5. Organigramme de gestion du parallélisme

Nous avons proposé plusieurs solutions et plusieurs formes de l'organigramme qui gère la communication entre les chargeurs et après une analyse profonde de chaque proposition toute en respectant les recommandations de l'encadrant et du chef de projet, on s'est met d'accord sur l'organigramme final dont son diagramme d'état est représenté par la figure suivante :

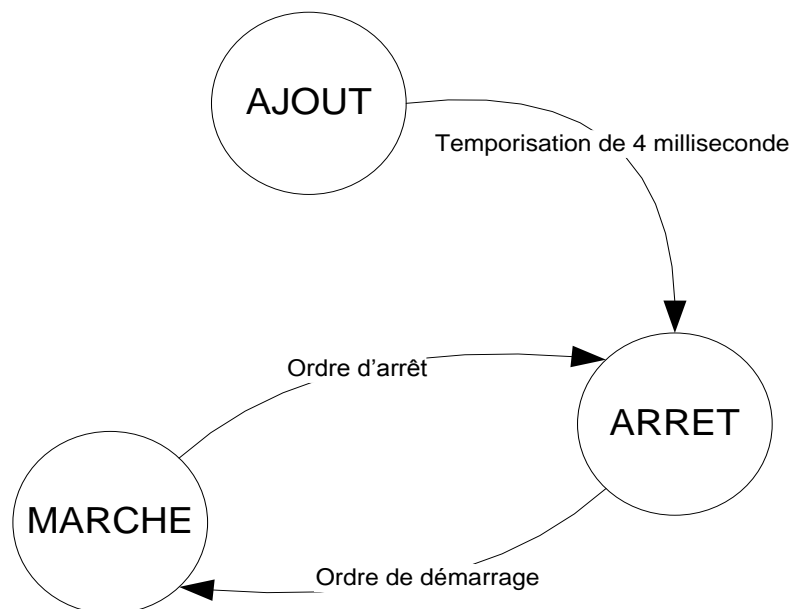


Figure 15 : L'organigramme d'état.

Dans notre organigramme on distingue trois principaux états:

- L'état «AJOUT » : elle assure l'ajout du chargeur au réseau.
- L'état «ARRET » : la partie commande est en marche et la partie puissance du chargeur est arrêtée. Au niveau de cet état l'utilisateur peut ajouter d'autres chargeurs au réseau.
- L'état «MARCHE » : les deux parties (partie commande et partie puissance) du chargeur sont en marche, le chargeur dans cet état fait la régulation sur son pourcentage de courant demandé par la charge.

5.1. L'état AJOUT

Au début le chargeur envoie un message de demande d'information sur le réseau, et après avoir reçu le message de réponse de chaque chargeur du réseau, il choisit un identifiant. Ensuite, il envoie le message d'ajouter chargeur au réseau (contient son identifiant et son courant maximum) et à la fin il fait la transition à l'état ARRET. Comme il est représenté dans la figure suivante :

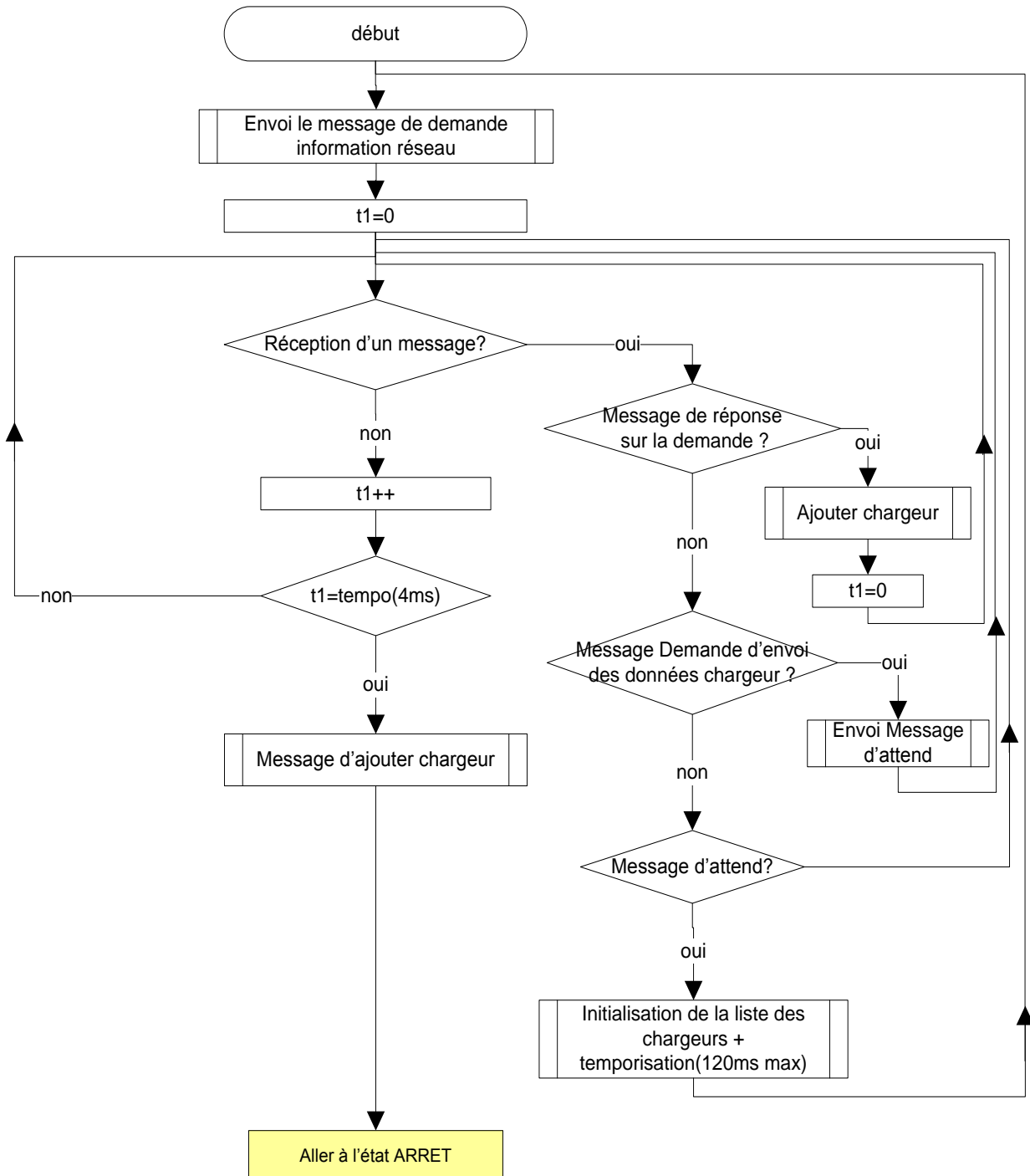


Figure 16 : L'organigramme d'ajout.

5.2. L'état ARRET

Dans cet état, l'utilisateur peut donner l'ordre de démarrage en utilisant des messages appelés "Message de Marche-Arrêt", cet ordre est applicable depuis tous les chargeurs qui appartient au réseau, en même temps le chargeur attend la réception d'un message : si c'est un message de demande d'information sur le réseau il renvoie le message de réponse et si c'est un message d'ajout chargeur, il ajoute le nouveau chargeur à sa liste. Comme montre la figure ci-dessous :

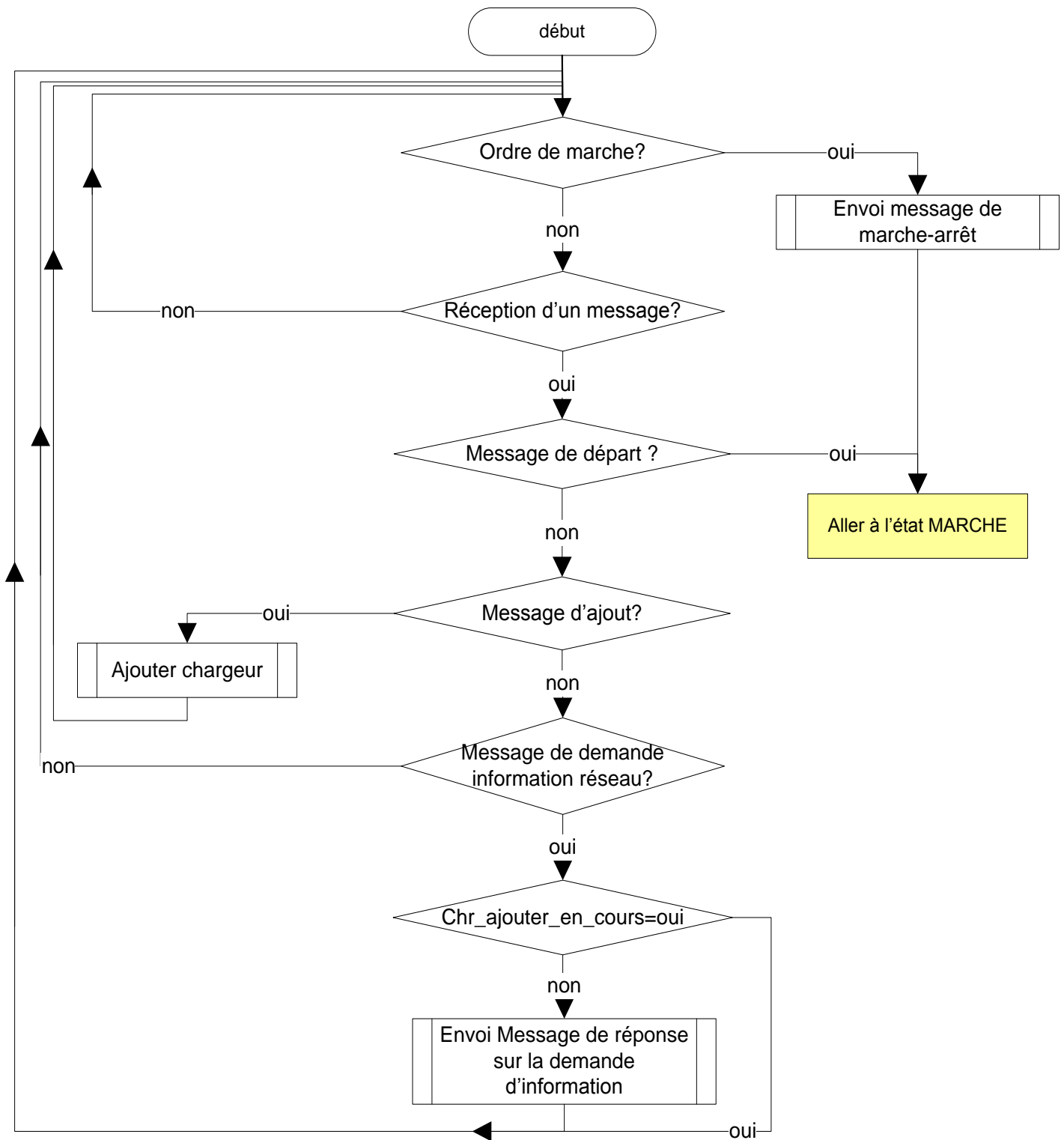


Figure 17 : L'organigramme d'arrêt.

5.3. L'état MARCHE

Dans la troisième phase le chargeur envoie chaque 5 milliseconde le message de données (son courant délivré à la charge) et en même temps attend la réception des messages des autres chargeurs pour calculer son nouveau courant de charge. Cette démarche est illustrée par la figure suivante :

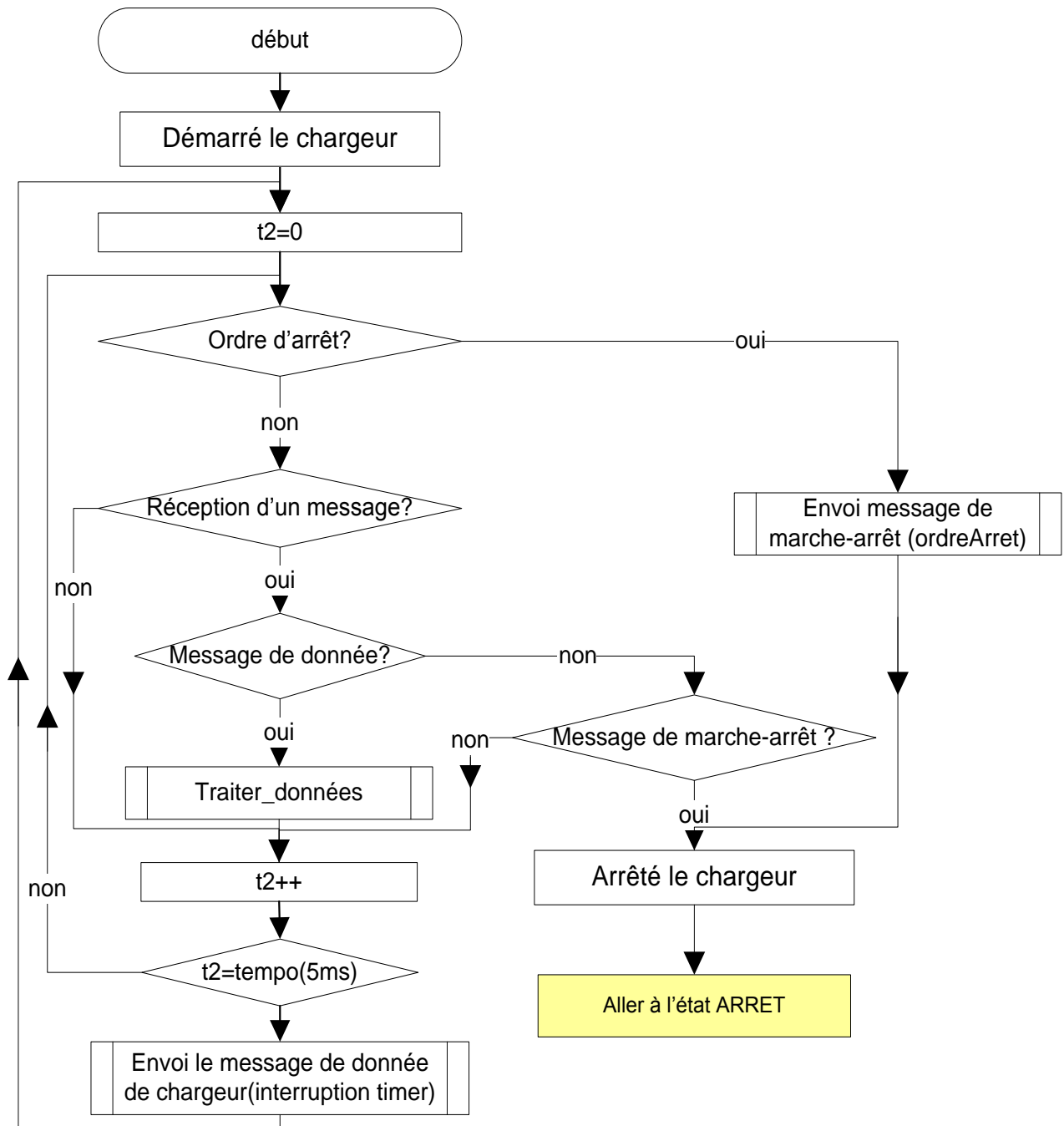


Figure 18 : L'organigramme de marche.

5.4. Les fonctions utilisées

Dans cette partie on va illustrer l'organigramme associé à chaque fonction.

5.4.1. Les variables utilisés

Le tableau suivant donne les explications des variables qu'on va utiliser dans les fonctions.

Les variables	Description
Tab_chr[]	Tableau contient les identifiants des chargeurs du réseau.
Nbr_chr	Nombre des chargeurs dans le réseau.
tab_courant_charge []	Tableau des courants des charges.
Tab_courant_max[]	Tableau des courants maximums.
Tab_elim[]	Tableau compteur pour éliminer un chargeur qui est déconnecté du réseau.
courant_max_tot	La somme des courants maximaux des chargeurs.
courant_charge_tot	Le courant total demandé par la charge.
Tab_envoi[]	Tableau contient le message à émettre.
Tab_recep[]	Tableau contient le message reçu.
Chr_ajouter_en_cours	Variable indique s'il y a un chargeur à l'état d'ajout.
Id	A la fin de la boucle il contient le plus grand identifiant des chargeurs.
nombre_chr_reste	Nombre des chargeurs accessibles dans le réseau – nombre chargeur ajouté.
MES_ATTEND, MES_AJOUTER, MES_REPONSE, MES_DONNEE, MES_DEMANDE_INFO, MES_MARCHE_ARRET.	Des constantes indiquant le type du message.

Tableau 4 : Table des variables.

5.4.2. Les fonctions de l'état AJOUT

- La fonction : Envoi Message de réponse sur la demande d'information.

L'organigramme de la fonction d'envoi de message de "demande d'information du réseau" est représenté dans la figure 19.

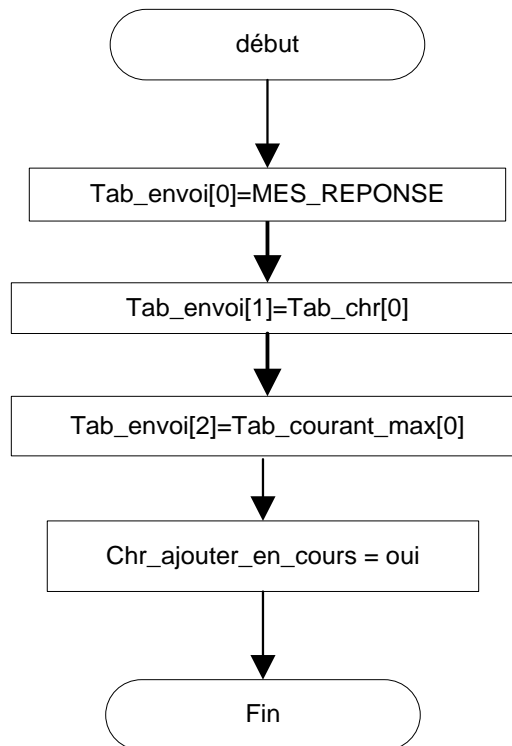


Figure 19 : L'organigramme d'Envoi Message de réponse sur la demande d'information.

➤ La fonction : Ajouter chargeur.

La figure suivante montre comment ajouter un chargeur à la liste.

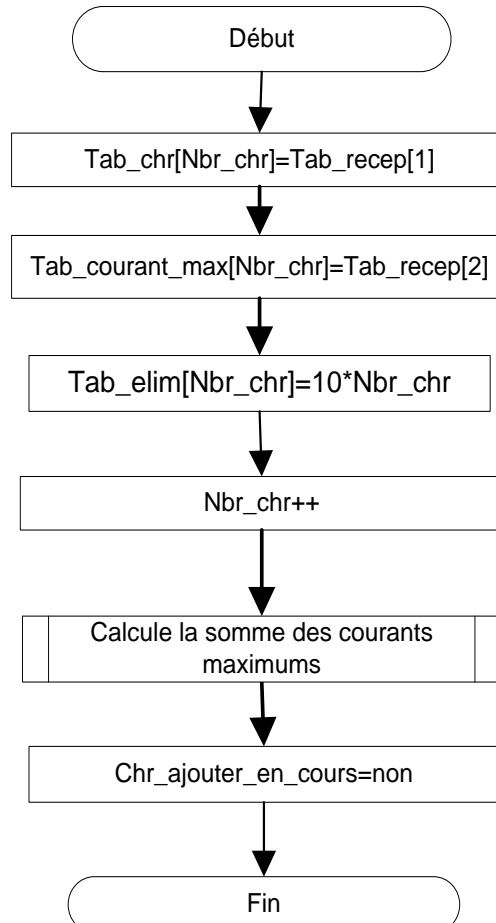


Figure 20 : L'organigramme d'Ajouter chargeur.

➤ La fonction : Envoi Message d'attente

Lorsque deux chargeurs s'ajoutent au même temps, il faut retarder l'un par rapport à l'autre. La figure ci-dessous montre le message qui permet de faire ce retard.

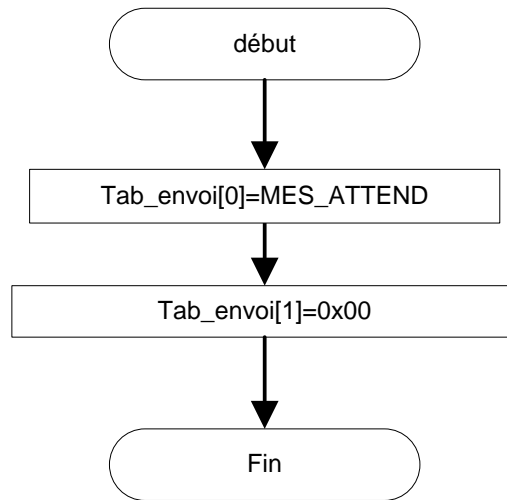


Figure 21 : L'organigramme d'Envoi Message d'attente.

➤ La fonction : Initialisation de la liste des chargeurs et temporisation

La figure suivante représente la fonction qui permet de faire l'initialisation de la liste des chargeurs et la temporisation.

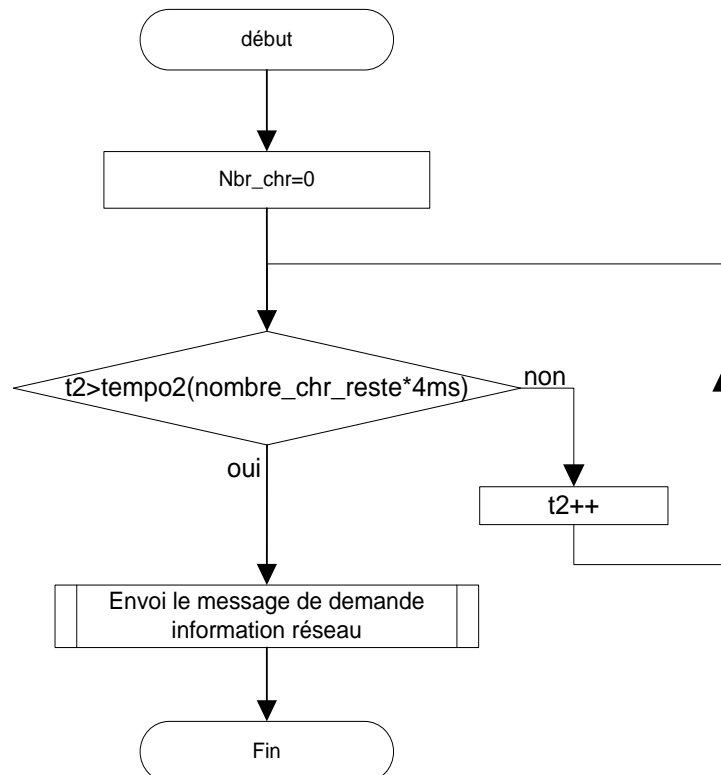


Figure 22 : L'organigramme d'Initialisation de la liste des chargeurs et temporisation.

➤ La fonction : Message d'ajout chargeur.

L'organigramme qui fait le choix de l'identifiant du chargeur et envoie le message "Ajouter chargeur" est représenté dans la figure ci-après.

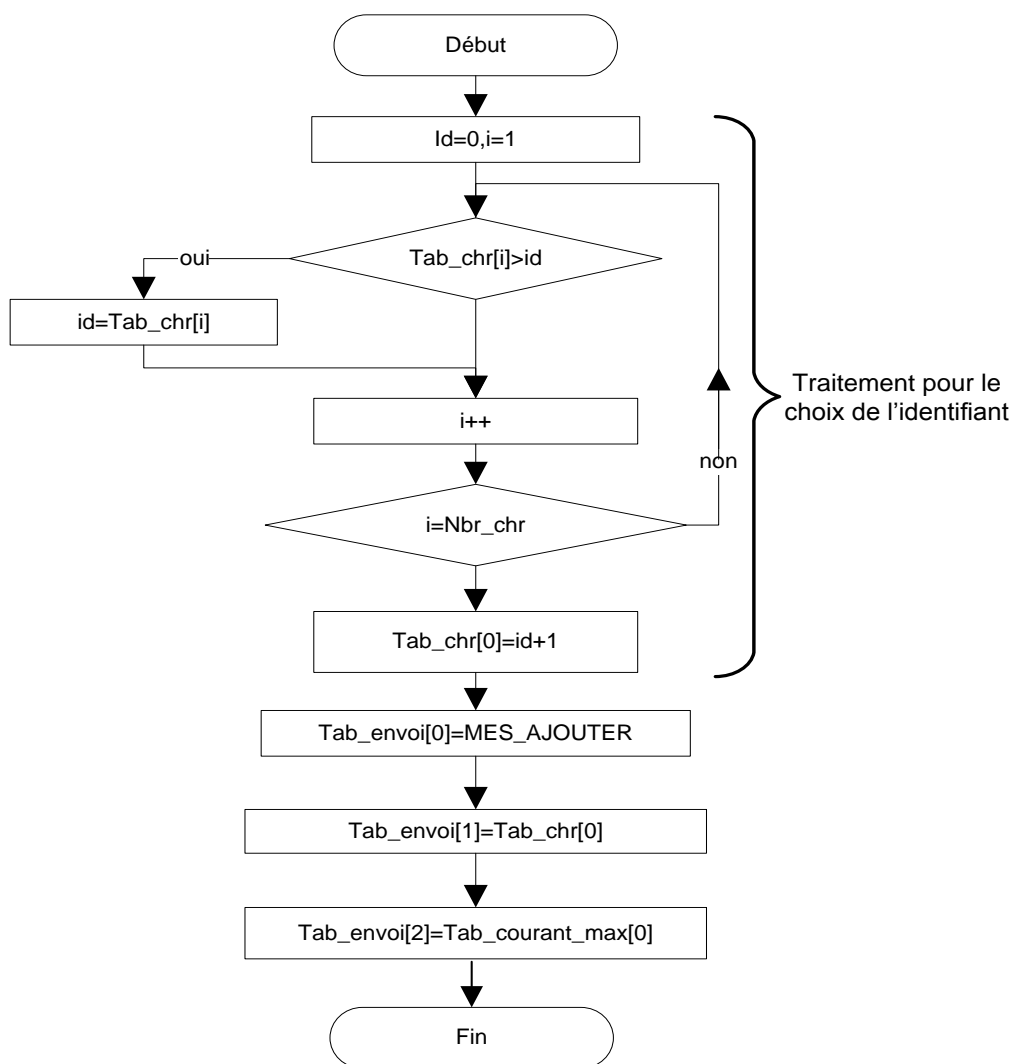


Figure 23 : L'organigramme d'Envoi message d'ajout chargeur.

5.4.3. Les fonctions de l'état ARRET

- La fonction : Envoi message de marche-arrêt.

La figure ci-dessous montre l'étape d'envoi d'un message de Marche-Arrêt.

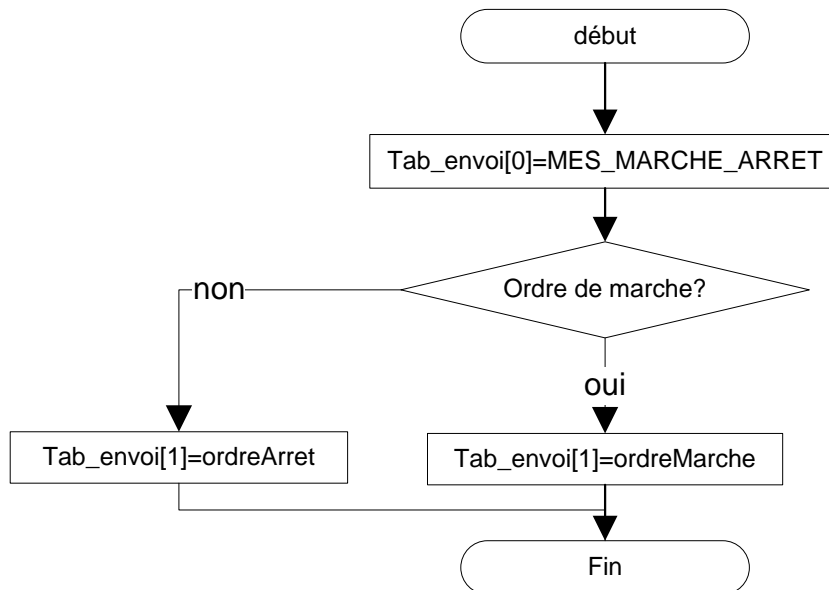


Figure 24 : L'organigramme d'Envoi message de marche-arrêt.

- La fonction : Envoi message de réponse sur la demande d'information.

Pour répondre au message de demande d'information, on envoie les informations du chargeur comme le montre la figure suivante :

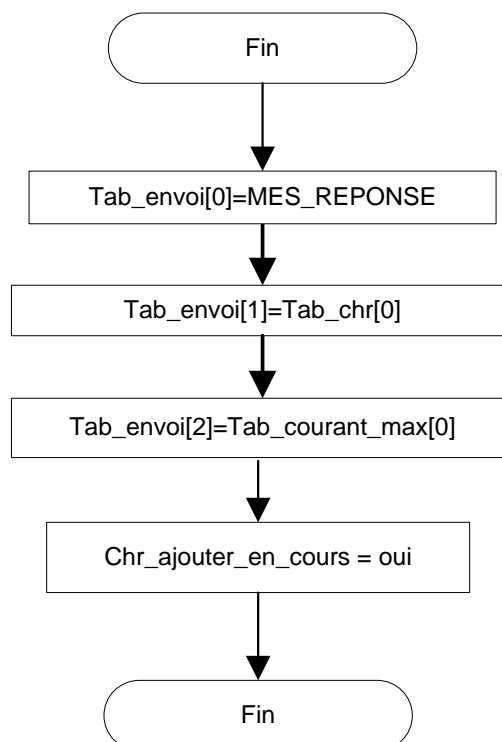


Figure 25 : L'organigramme d'Envoi message de réponse sur la demande d'information.

5.4.4. Les fonctions de l'état MARCHE

- La fonction : Envoi le message de donnée du chargeur.

La figure suivante montre le message d'envoi de données chargeur :

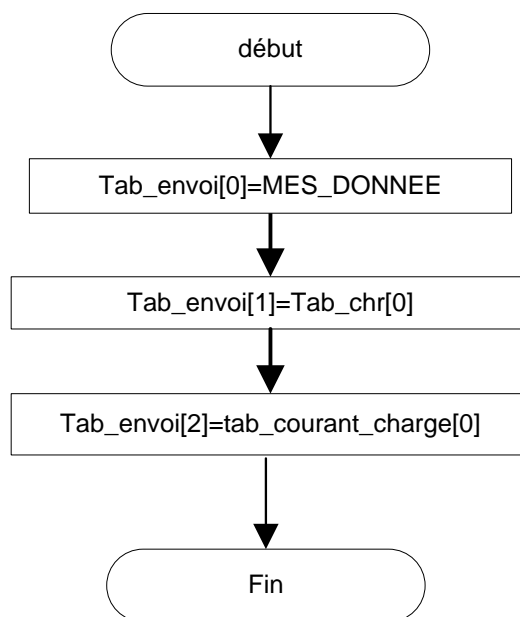


Figure 26 : L'organigramme d'Envoi le message de donnée du chargeur.

- La fonction : Traiter données.

La figure 27 montre l'organigramme de traitement des données :

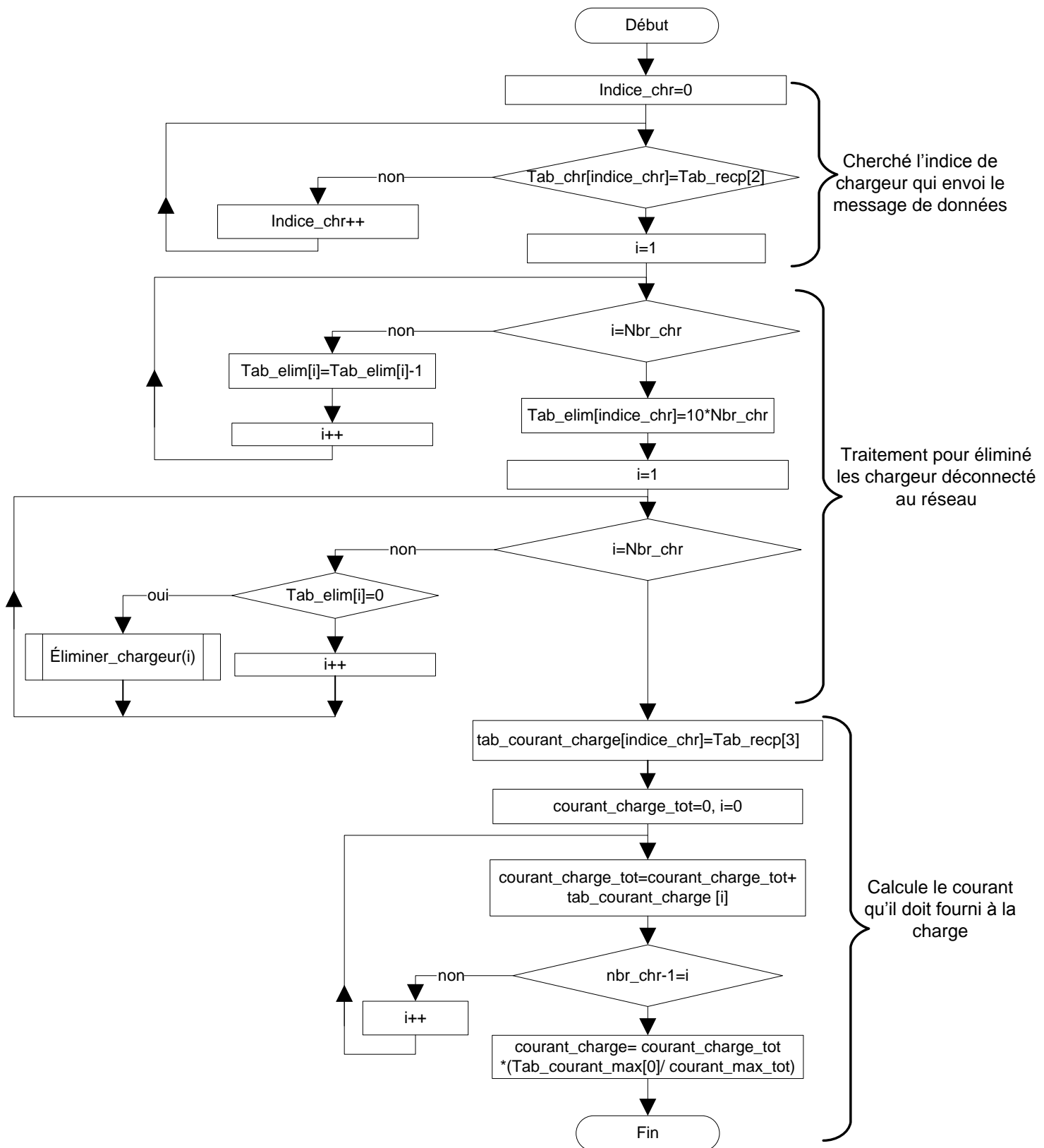


Figure 27 : L'organigramme de traitements données.

➤ La fonction : Élimination chargeur.

La façon d'éliminer un chargeur est représentée dans la figure 28.

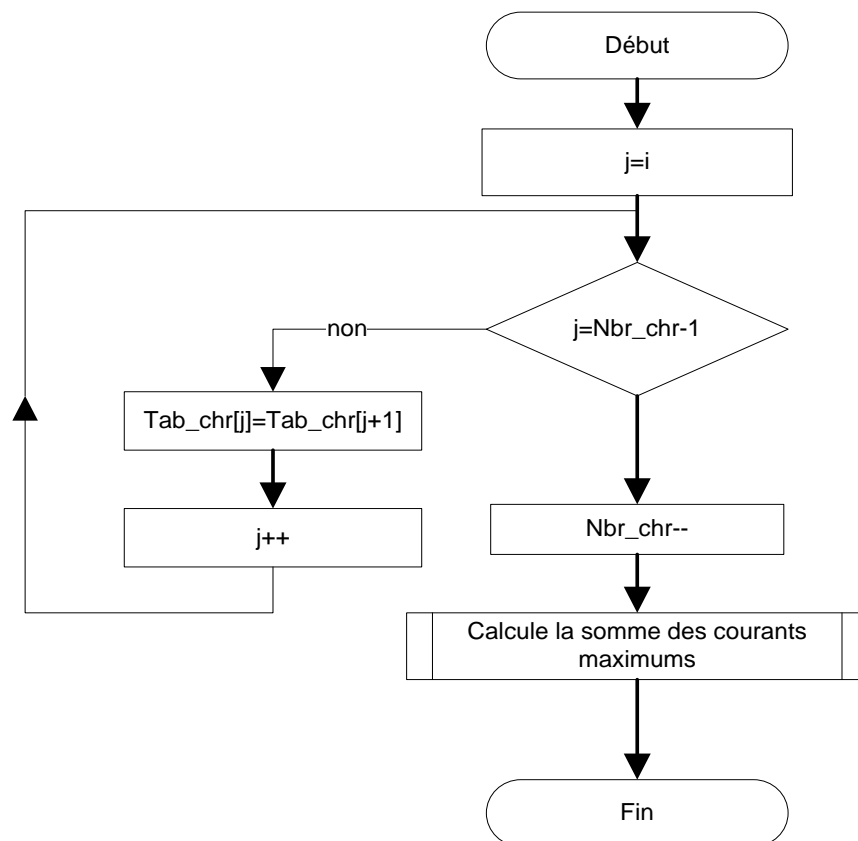


Figure 28 : L'organigramme d'Élimination chargeur.

La même chose pour les tableaux suivants: $Tab_courant_max[]$, $Tab_courant_charge[]$, $Tab_elim[]$.

- La fonction : Calcul de la somme des courants maximaux

La figure ci-dessous montre l'organigramme qui permet de calculer la somme des courants maximaux des chargeurs :

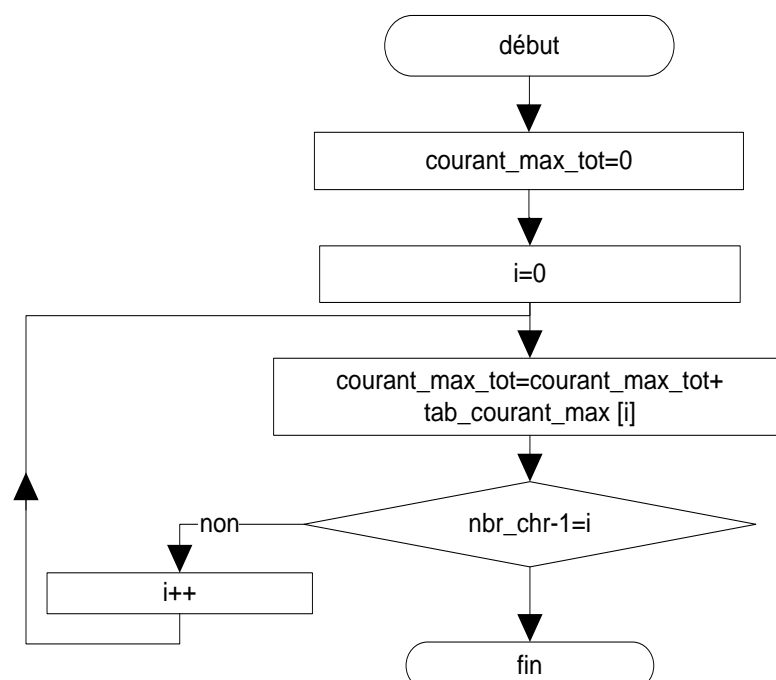


Figure 29 : L'organigramme de Calcul de la somme des courants maximaux.

5.5. L'étude des temporisations

Dans les conditions où l'exploitation de bus CAN est maximale, autrement dit avec 30 chargeurs, et une taille de données de 64 bits, nous avons choisi de travailler avec une vitesse de 800kbps, et nous pouvons déduire les résultats suivants:

- ✓ La longueur de la trame est de 111bit (sans les stuffing).
- ✓ Si on est dans le cas où tous les bits stuffing sont utilisés donc la longueur de la trame est de $111 + (101/4)$ soit 136 bits.
- ✓ Donc le temps pour l'envoi d'une trame est de $136 * 1.25$ soit 170 μ s.
- ✓ La durée maximale d'un message de surcharge ou d'un message d'erreur active est de $23 * 1.25$ soit 28.75 μ s.
- ✓ La durée maximale d'un message d'erreur active est de $23 * 1.25$ soit 28.75 μ s.

➤ Remarque :

Chaque chargeur doit être capable de recevoir tous les messages pour éviter les trames de surcharge.

Le bus doit avoir une bonne qualité de transmission afin de minimiser les messages d'erreur.

➤ Temporisation 1 : le temps pour recevoir un message de réponse.

- ✓ Message de réponse (170 μ s).
- ✓ Autre message sur le bus (170 μ s).
- ✓ Message de surcharge (28.75 μ s).
- ✓ Message d'erreur active (28.75 μ s).

L'estimation moyenne est de 4 ms.

➤ Temporisation 2 : le temps pour ajouter un chargeur (réception de tous les messages de réponse).

Le temps attribué à la réception de 30 messages de réponse est estimé à $30 * 4$ ms soit 120 ms

➤ Temporisation 3 : le temps pour recevoir tous les messages de données.

L'estimation du temps attribué à la réception de tous les messages de données est évalué à $170 \mu s * 30$ soit 5100 μ s.

6. Les trames de communication

Dans cette partie on a illustré les trames de communication qu'on a crée pendant la phase de conception de l'organigramme de communication.

Les valeurs des champs des messages sont choisies selon la priorité de message à accès au bus.

➤ Message de demande information sur le réseau

Chaque nouveau chargeur connecté au réseau envoie ce message pour que les autres chargeurs lui communiquent leurs informations (identifiant et le courant maximal).

Type de message: message de demande information sur le réseau(010)	ID(0x00)
--	----------

Figure 30 : Message de demande information sur le réseau.

➤ Message de réponse sur la demande d'information réseau

Chaque chargeur qui reçoit un message de demande information renvoie ce message.

Type de message: Message de réponse sur la demande d'information réseau(100)	Identifiant de chargeur émetteur(1 octet)	Courant maximum (4 octet)
--	---	---------------------------

Figure 31 : Message de réponse sur la demande d'information réseau.

➤ Message d'ajout d'un chargeur

Après la réception du message de réponse de tous les chargeurs, le chargeur envoie ce message pour s'ajouter au réseau.

Type de message: message d'ajout d'un chargeur(001)	Identifiant de chargeur émetteur(1 octet)	Courant maximum (4 octet)
---	---	---------------------------

Figure 32 : Message d'ajout d'un chargeur.

➤ Message de données

Ce message s'envoie en permanence par les chargeurs pour que chaque chargeur calcule le courant qu'il doit fournir à la charge.

Type de message: message de donnée(110)	Identifiant de chargeur émetteur(1 octet)	courant de charge délivré par le chargeur émetteur(4 octet)
---	---	---

Figure 33 : Message de données.

➤ Message d'attente

Ce message traite le problème de deux chargeurs qui s'ajoute en même temps. Dans ce cas l'un parmi eux envoie ce message pour retarder l'ajout de l'autre.

Type de message: message d'attend(011)	Nombre chargeur ajouté(1 octet)
--	---------------------------------

Figure 34 : Message d'attente.

➤ Message de Marche-Arrêt

On a construit ce message pour que tous les chargeurs démarrent ou s'arrêtent en même temps.

Type de message: Message de Marche-Arrêt(000)	Type de l'ordre: Marche ou Arrêt
---	----------------------------------

Figure 35 : Message de Marche-Arrêt.

7. Microcontrôleur

La carte de commande du chargeur comme le montre la figure ci-dessous dispose de deux microcontrôleurs STM32F107VC et STM32F407VG. Ce dernier assure le fonctionnement du chargeur et l'autre fait la gestion des communications du chargeur avec l'extérieur, donc nous allons travailler sur le microcontrôleur STM32F107VC.

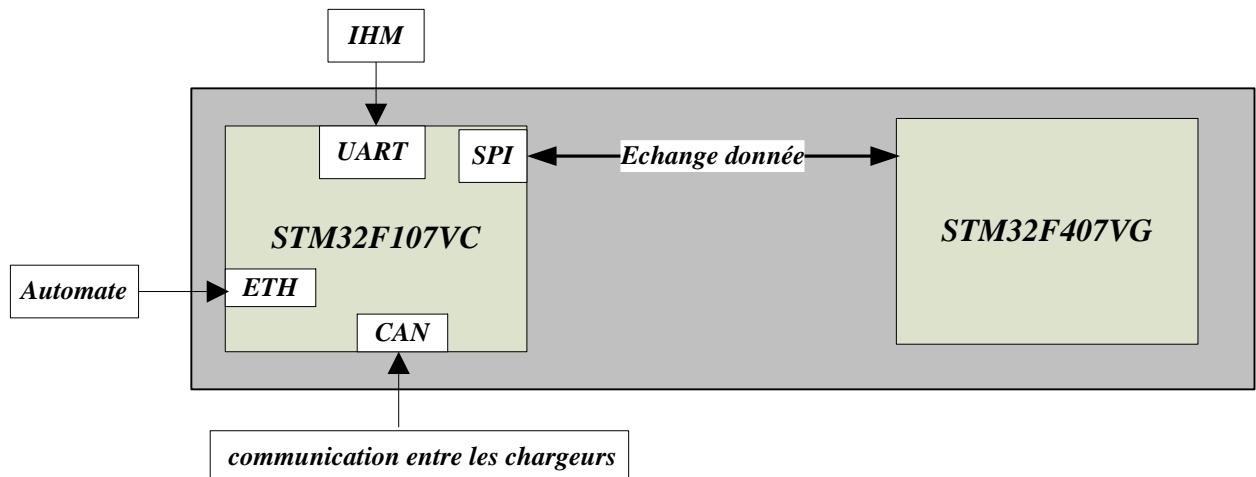


Figure 36 : La Carte de Commande du chargeur.

7.1. Série STM32F1

La famille STM32 de microcontrôleurs 32 bits Flash est basée sur l'exceptionnel noyau ARM Cortex™ M3 - un noyau spécialement développé pour les applications intégrées. La famille STM32 bénéficie des améliorations architecturales du Cortex-M3, notamment le jeu d'instructions Thumb-2 pour améliorer la performance avec une meilleure densité de codes, une réponse nettement plus rapide aux interruptions, le tout associé à une très faible consommation.

- ✓ Excellent comportement en temps réel.
- ✓ Exceptionnelle efficacité énergétique.
- ✓ Périphériques supérieurs et innovants.
- ✓ Intégration maximale.
- ✓ Compatibilité broche à broche, périphérique et logicielle au sein de la famille. [4']

7.2. Microcontrôleurs série STM32F107

Les processeurs STM32F107 de STMicroelectronics avec noyau Cortex M3 présentent des vitesses allant jusqu'à 72 MHz. La série STM32 se compose de microcontrôleurs 32 bits ARM avec des "Timers" 16 bits, des interfaces CAN, ADC, USB 2.0 pleine vitesse/OTG, Ethernet MAC et interface de communication (I2C, SPI, UART). Le microcontrôleur Flash 32 bits STM32 ARM Cortex-M3 fonctionne à basse tension et à faible consommation, tout en disposant de capacités en temps réel. L'architecture du microcontrôleur comprend une plate-forme STM32 facile à utiliser, qui offre trois boîtiers différents. Ces dispositifs intégrés fonctionnent dans les applications telles que les entraînements de moteur, les imprimantes et les scanners, les alarmes et les applications industrielles.

La figure suivante montre l'architecture interne de STM32F107:

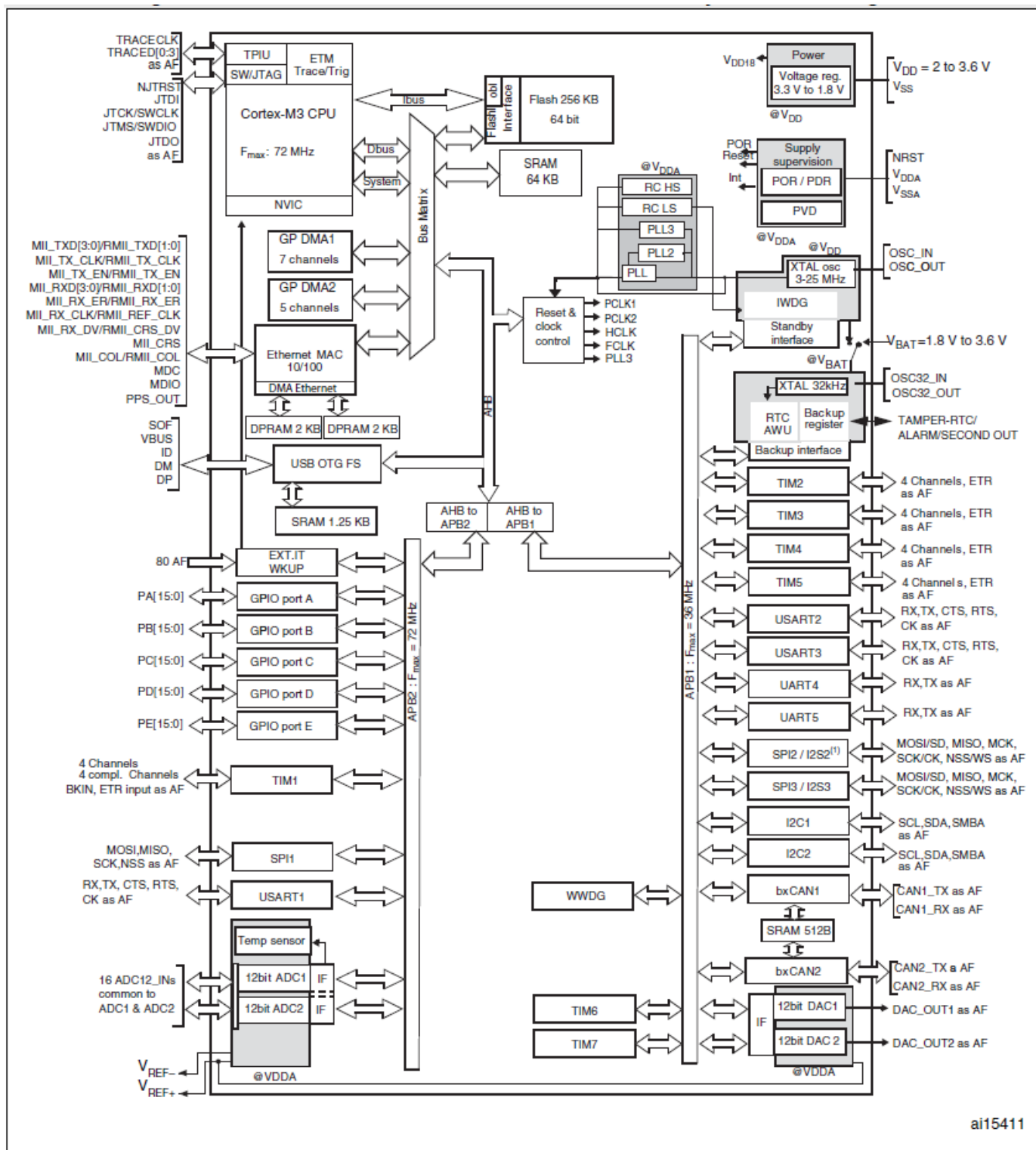


Figure 37 : Architecture interne de STM32F107. [1]

7.3. Le Cortex M3 :

Le cœur Cortex-M3 est basé sur l'architecture ARMv7-M et il est structuré de façon hiérarchique. Le noyau contient la CPU (appelé CM3Core) avec des composants périphériques supplémentaires tel que le contrôleur d'interruptions, les unités de protection de la mémoire et les circuits de débogage.

Le cœur Cortex-M3 dispose des propriétés suivantes:

- ✓ Processeur 32 bits avec des registres 32 bits et une interface mémoire.
- ✓ Une architecture de Harvard : il possède deux bus différents pour accéder à la mémoire contenant les instructions du programme et à la mémoire contenant les données. Ce qui augmente les performances du système. En effet les accès aux instructions et aux données du programme peuvent être exécutés en parallèle (c.à.d. en même temps).
- ✓ Un pipeline de trois étages (fetch, decode, execute) avec une prédiction spéculative pour les instructions de branchement.
- ✓ Jeu d'instruction Thumb-2 : Mélange d'instructions 16 et 32 bits pour une performance optimale avec les exigences minimales en ressource mémoire (code compact).
- ✓ Accès bit à certaines zones de la mémoire (bit-banding, bit-band regions), ce qui permet la modification d'un bit sans les cycles de lecture/écriture standards.
- ✓ Support hardware des opérations de multiplications et de divisions 32-bits.
- ✓ Contrôleur d'interruptions vectorielles imbriquées (Nested Vectored Interrupt Controller : NVIC) avec tableau de vecteurs et huit niveaux de priorité d'interruption. Les opérations de sauvegarde sur la pile (sauvegarde des registres avec les exceptions) sont réalisées de façon hardware.
- ✓ Unité de protection de la mémoire (Memory Protection Unit : MPU) pour la protection des zones mémoires (applications RTOS).
- ✓ Support de débogage. [2]

Conclusion

Nous avons abordé à travers ce chapitre, la conception du parallélisme des chargeurs qui nécessite une étude profonde afin de faire les choix convenables pour établir un protocole de communication solide. Ce protocole doit respecter toutes les contraintes imposées par le cahier des charges et assurer le fonctionnement correct des chargeurs montés en parallèle.