Polytech Clermont-Ferrand

Département Génie électrique 5éme année

Sécurisation d'une communication par bus CAN

Note d'application

Plantin Anthony

Tables des matières

Tables des figures

Introduction

Dans le cadre des projets de fin d'études de génie électrique à polytech Clermont-Ferrand, la société Eaton spécialisée dans les systèmes électriques et hydrauliques nous a donné la possibilité de travailler sur une problématique concernant les technologies tactiles et une communication CAN sécurisée. Cette note d'application abordera seulement la sécurisation d'une communication par bus CAN.

La société avec laquelle nous travaillons est une filiale d'Eaton, Cooper Security sas basé à Riom qui est spécialisé dans la sécurité des bâtiments. Le projet proposé est de repenser un tableau qui gère les organes de sécurité et qui envoie les informations à une carte mère qui peut être située loin du tableau, on comprend donc l'intérêt d'avoir une liaison sécurisée entre la carte mère et le tableau. Nous allons donc voir plusieurs stratégies pour sécuriser une liaison CAN.

Fonctionnement du bus CAN

Dans cette partie nous allons détailler fonctionnement du bus CAN et la composition des trames de données ainsi que les mécanismes de détection des erreurs.

Topologie du bus CAN

Le bus CAN (Contol area network) est un bus de donnée série bidirectionnelle half-duplex. On peut connecter jusqu'à 30 nœuds en mode « high speed » et 20 en « low speed ». Chaque nœud est connecté au bus par l'intermédiaire d'une paire torsadée. Les deux extrémités du bus doivent être rebouclées par des résistances de terminaison de 120 Ω.

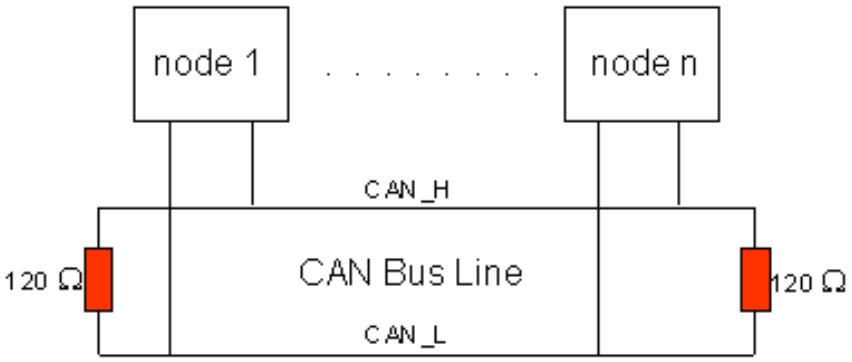

Figure 1 Topologie du bus CAN

Composition d'une trame de donnée

Une trame de donnée sert à envoyer des données aux autres nœuds. Une trame de donnée se compose de 7 différentes parties. Le début de trame ou SOF (Start Of Frame) matérialisé par 1 bit dominant. Le champ d'arbitrage (identificateur) composé de 12 ou 30 bits selon la norme du bus CAN (bus CAN 2.0a ou 2.0b). Le champ de commande (ou de contrôle) composé de 6 bits. Le champ de données composé de 0 à 64 bits (de 0 à 8 octets). Le champ de CRC composé de 16 bits. Le champ d'acquittement composé de 2 bits. La fin de trame ou EOF (End of Frame) matérialisée par 7 bits récessifs.

Figure 2 Trame CAN

L'encodage des bits est de type NRZ (Non-retour à zéro).

Sécurité des données

Types d'erreurs

Le CAN implémente cinq mécanismes de détection des erreurs, 2 au niveau bits (le "bit monitoring" et le "bit stuffing") et 3 au niveau messages (vérification du CRC, de la forme des trames et de l'acquittement). Ces cinq types d'erreurs différents qui peuvent être détectés par un nœud sont :

- Bit error : Un nœud envoyant un bit sur le bus regarde aussi en même temps les bits qu'il reçoit (Bit monitoring). Il considère comme une erreur de bit lorsque le bit envoyé est différent du bit reçu, à l'exception de l'envoi d'un bit récessif durant l'arbitrage (cas de la perte d'arbitrage) ou pendant l'ACK Slot (trame acquittée).
- Stuff error : Le nœud détecte une erreur de stuffing lorsqu'il reçoit 6 bits consécutifs de même valeur dans une partie d'un message qui devrait être codé avec la m´méthode du bit stuffing.
- CRC error : Une erreur de CRC est détectée lorsque le CRC calculé par un récepteur est différent de la valeur du CRC contenu dans la trame.
- Form error : Une "Form error" est détectée lorsqu'un bit qui devrait être à une certaine valeur est à une valeur différente (un délimiteur par exemple).
- ACK error : Le transmetteur détecte une erreur d'acquittement lorsqu'il ne reçoit pas de bit dominant pendant l'ACK Slot.

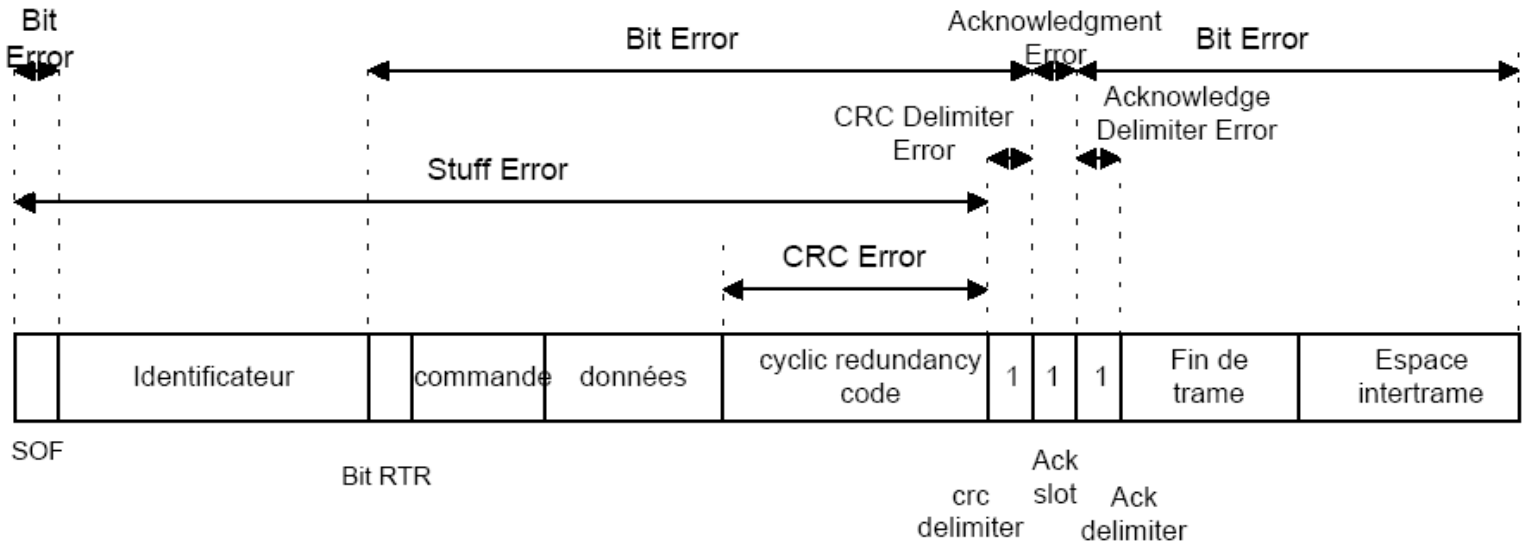

Figure 3 Types d'erreurs dans une trame

Trames d'erreurs

Une trame d'erreur est constituée de deux parties. La première est formée par la superposition des différents "Error flags" mis par les noeuds du bus. La seconde partie est un délimiteur. Un noeud qui détecte une erreur la signale en envoyant un Error flag. Celui-ci viole la règle du bit stuffing (6 bits dominants consécutifs) et par conséquent, tous les autres noeuds détectent aussi une erreur et commencent à envoyer un Error flag. La séquence de bits dominants qui existe alors sur le bus est le résultat de la superposition de plusieurs Error flags, et sa longueur varie entre 6 et 12 bits.

Il existe deux types d'Error flags :

- Active error flag : 6 bits dominants consécutifs.

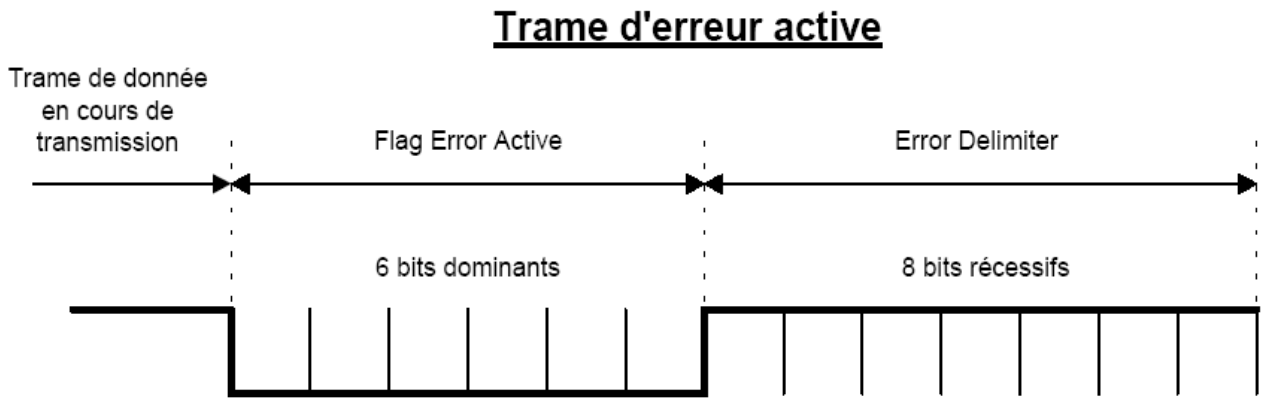

Figure 4 Trame d'erreur active

- Passive error flag : 6 bits récessifs consécutifs, jusqu'à ce qu'ils soient écrasés par des bits dominants.

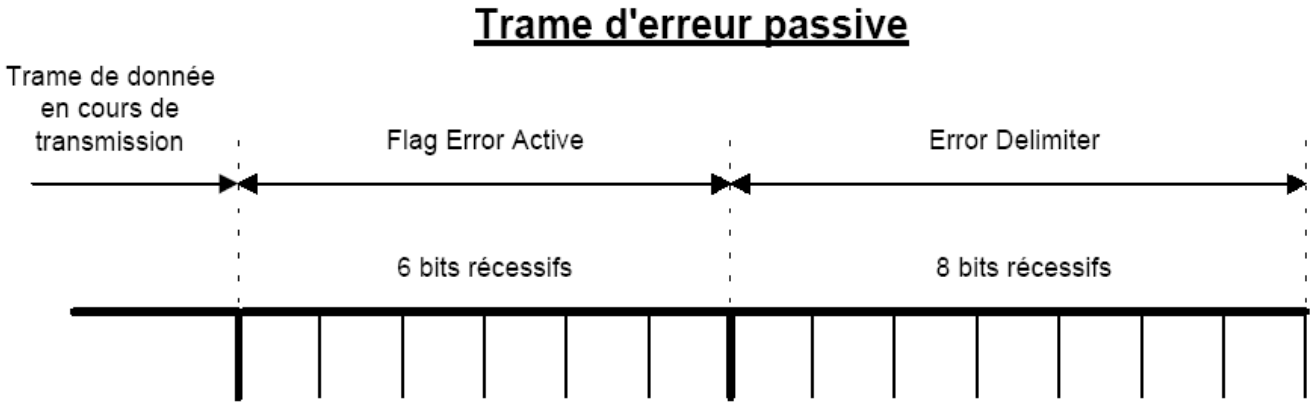

Figure 5 Trame d'erreur passive

L'Error delimiter est composé de 8 bits récessifs. En fait, après avoir transmis son Error flag, chaque nœuds envoie des bits récessifs et observe le bus jusqu'à ce qu'il détecte un bit récessif, après quoi il envoie encore 7 bits récessifs supplémentaires.

Gestion des erreurs

Le confinement des erreurs est un mécanisme permettant de faire la différence entre des erreurs temporaires ou permanentes. Les erreurs temporaires peuvent être causées par des glitchs par exemple, tandis que des erreurs permanentes sont en général dues à de mauvaises connexions ou à des composants défaillants. Ce système va permettre d'enlever un nœud défaillant du bus qui sinon aurait pu perturber les autres nœuds. Un nœud peut être dans trois états : error-active, error-passive ou bus-off.

1. Un nœud en mode d'erreur actif (error-active) peut prendre part normalement dans la communication sur le bus. Il transmettra un Active error flag s'il détecte une condition d'erreur. 2. Un nœud en mode d'erreur passif (error-passive) peut prendre part dans la communication, mais s'il détecte une condition d'erreur sur le bus, il transmettra un Passive error flag. Ce mode indique un nœud à problèmes. 3. Un nœud en mode bus-off n'est pas autorisé à avoir une quelconque influence sur le bus. Deux compteurs d'erreurs sont implémentés dans chaque nœud : celui des erreurs en transmission (Transmit error count) et celui des erreurs en réception (Receive error count). Les grandes règles de modifications des compteurs d'erreurs sont les suivantes. Elles sont détaillées dans le schéma ci-dessous :

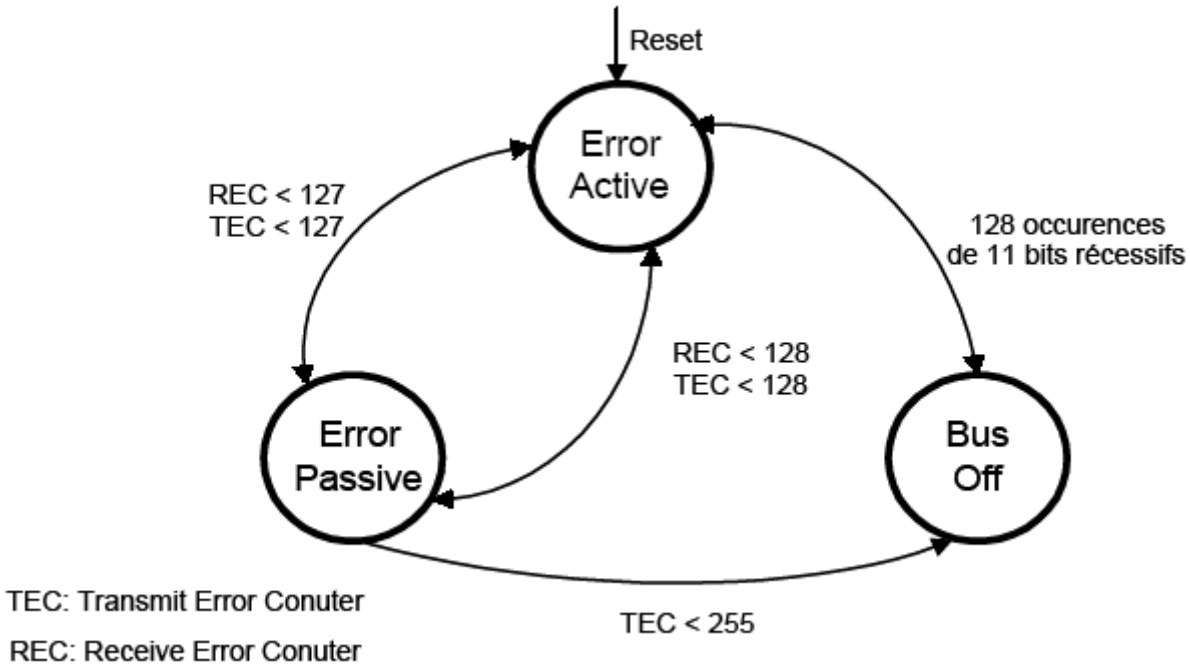

Figure 6 Machine d'état gérant les erreurs

– Lorsqu'un récepteur détecte une erreur, son Receive error count est augmenté de 1.
– Lorsqu'un transmetteur envoie un Error flag, son Transmit error count est augmenté de 8.
– Après une transmission réussie, le Transmit error count est diminué de 1.
– Après une réception réussie, le Receive error count est diminué de 1.

Un nœud est en mode error-active si ses deux compteurs d'erreurs sont inférieurs à 127. Il est en

error-passive si l'un des deux est compris entre 128 et 256. Si le Transmit error count est supérieur à 256, le nœud est en bus-off.

Redondance d'information

On se rend compte qu'avec tout le mécanisme de détection et de gestion d'erreurs, le bus CAN est un bus de communication robuste et sécuritaire. On se demande alors en quoi une redondance d'information peut ajouter une sécurité supplémentaire et dans quel cas cela peut être utile ? Pour se remettre dans le contexte de notre projet, une carte mère gérant des organes de sécurité incendie (ventouse de portes-coupes feux, alarme incendie, bouche de désenfumage, etc…) est reliée à un tableau à destination des pompiers. La connexion entre la carte mère et le tableau se fait par bus CAN. La carte mère peut être déportée à plusieurs kilomètres de câbles du tableau. Il faut donc trouver un moyen d'envoyer les informations à la carte mère même si il y a une coupure d'un bus Can et il faut un moyen de détecter la coupure afin de faire remonter le problème.

La solution développée consiste à utiliser deux bus CAN différents et envoyer les mêmes informations sur les deux bus CAN. Cette stratégie permet même en cas de coupure de l'un des bus CAN d'envoyer l'information à la carte mère et de pouvoir quand même piloter les organes de sécurité. On peut ainsi détecter la coupure.

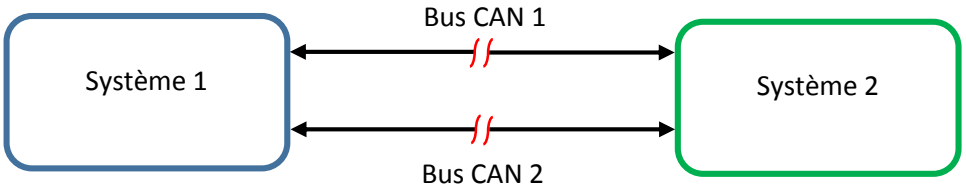

Figure 7 Schéma de principe du bus CAN redondant

Plusieurs stratégies sont possibles pour l'envoi des données, la première est l'envoi des mêmes paquets de données sur les deux bus CAN. Une autre stratégie plus robuste et plus efficace consiste a justement ne pas envoyer les mêmes données sur les bus CAN. On peut imaginer plusieurs stratégies. L'une des stratégies mises en place fut d'envoyer les paquets de manière croisée.

Octet de donnée	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6	Octet 7	Octet 8
CAN 1	0x01	0x02	0xAA	0xF2	0x23	0x00	0x80	0x5A
CAN 2	0x5A	0x80	0x00	0x23	0xF2	0xAA	0x02	0x01

Figure 8 Exemples de trame de données sur les bus CAN

Conclusion

Pour conclure, le bus CAN est déjà un bus de communication sécuritaire avec des mécanismes efficaces de gestion des erreurs. Ce bus de communication est robuste aux perturbations également grâce au codage des bits (NRZ). En clair le bus CAN est un bus robuste et assez sur.

Ce qui peut motivé à utiliser la redondance de l'information dépend du projet et du contexte. La redondance d'information permet d'augmenter la sécurité du produit. En effet lors d'une coupure de l'un des bus CAN on transmet quand même l'information et cela nous permet de détecter la coupure.