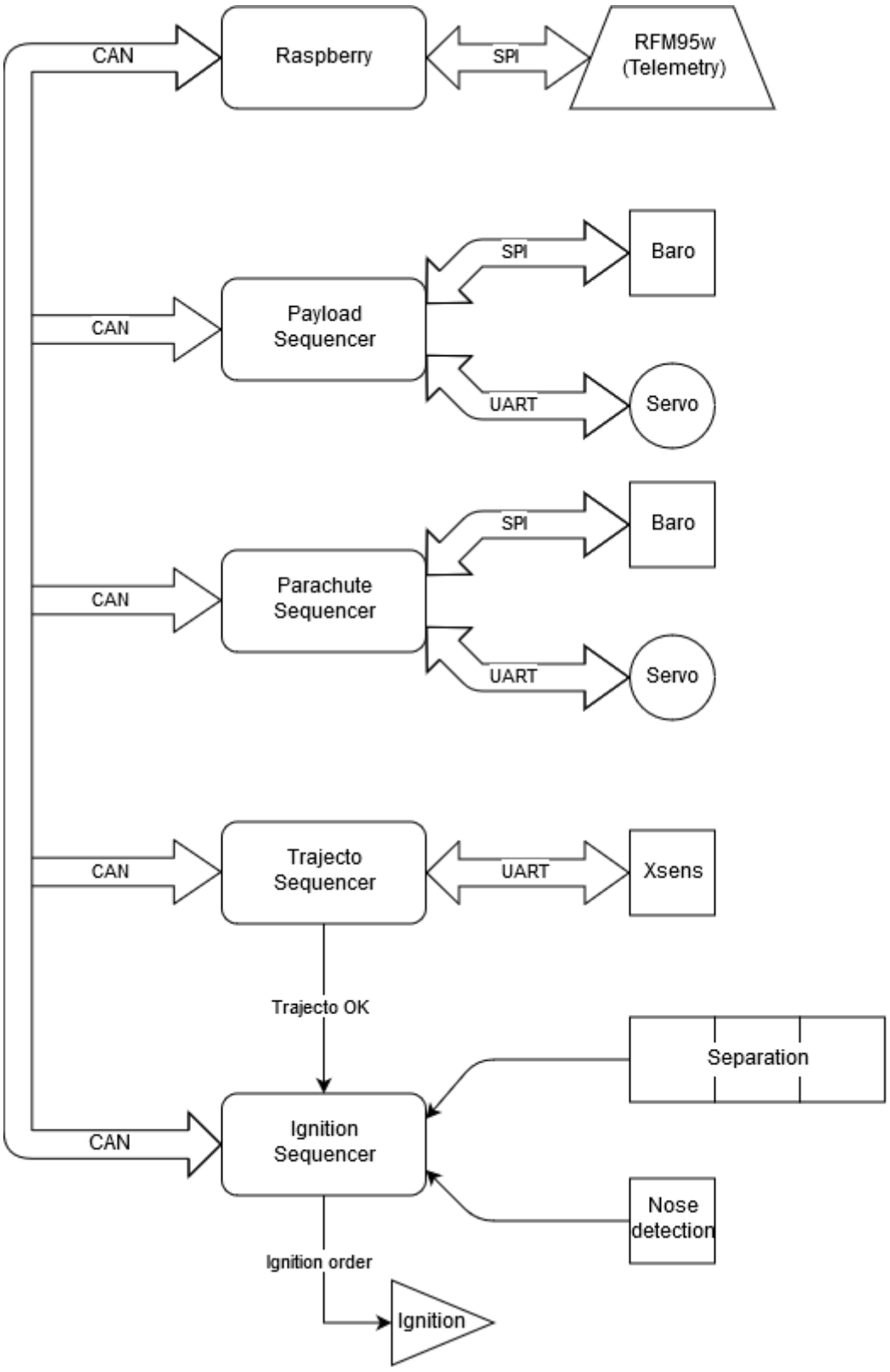
Organisation Informatique Case Electronique



Raspberry Pi

La Raspberry Pi écoute le bus CAN et enregistre toutes les données qu'elle reçoit. Elle en envoie ensuite une partie par télémétrie. Pour plus d'information, lire le readme.txt du dépôt Git sources_raspi.

Séquenceur Récupération (ou Séquenceur Parachute, ou « recup »)

Le Séquenceur récupération doit éjecter la coiffe de la fusée dans une fenêtre temporelle située à l'apogée. La date et la durée de la fenêtre temporelle sont déterminées à l'avance selon de cahier des charges (date théorique de l'apogée ± 10%). Le Séquenceur ajuste la date exacte de l'éjection en utilisant les mesures de pressions du baromètre pour détecter l'apogée. En réalité, on essaye d'éjecter la coiffe 1s avant l'apogée pour que le parachute ait le temps de s'ouvrir (mais sans modifier la fenêtre, c'est interdit).

Comme le code du Le Séquenceur récupération est quasiment identique à celui du Séquenceur Charge Utile, Il n'y a qu'un seul projet MPLAB pour les deux, « Ejection_Sequencer.X », avec deux configurations différentes.

Séquenceur Charge Utile (ou Séquenceur Cansat, ou « chut »)

Le Séquenceur récupération doit éjecter la coiffe de la fusée dans une fenêtre temporelle située avant l'éjection du parachute. La date et la durée de la fenêtre temporelle sont fixés avant le lancement, de façon à ne pas chevaucher la fenêtre temporelle d'éjection du parachute. Le Séquenceur ajuste la date exacte de l'éjection en utilisant les mesures de pressions du baromètre pour connaitre sa position temporelle par rapport à l'apogée.

Comme le code du Le Séquenceur récupération est quasiment identique à celui du Séquenceur Charge Utile, Il n'y a qu'un seul projet MPLAB pour les deux, « Ejection_Sequencer.X », avec deux configurations différentes.

Séquenceur Trajectographie (ou « traj »)

Le Séquenceur Trajectographie calcule l'orientation de la fusée à partir des données de la centrale inertielle (Xsens mti-1). Le séquenceur calibre les gyroscopes pendant que la fusée

est au sol. Pendant le vol, il compare l'attitude de la fusée avec 2 cônes : le cône nominal et le cône de sécurité :

- Si l'attitude est dans le cône nominal, le Séquenceur trajectographie indique au Séquenceur Allumage que la trajectoire est nominale.
- Si l'attitude sort du cône nominal, il indique au Séquenceur Allumage d'inhiber la mise à feu jusqu'à ce que l'attitude retourne dans le cône nominal.
- Si l'attitude sort du cône de sécurité (qui contient le cône nominal), la mise à feu est définitivement inhibée.

La transmission de cette autorisation de mise à feu entre les séquenceurs Trajectographie et Allumage se fait grâce à une simple piste reliant un pin de chaque microcontrôleur.

En configuration biétage, le rôle du séquenceur Trajectographie sera aussi de détecter la poussée du second moteur pour le signaler au Séquenceur Récupération afin d'ajuster la fenêtre d'éjection du parachute.

Séquenceur Allumage (ou « pyro »)

Le rôle du séquenceur Allumage est de donner l'ordre de mise à feu lorsque les 4 conditions suivantes sont réunies :

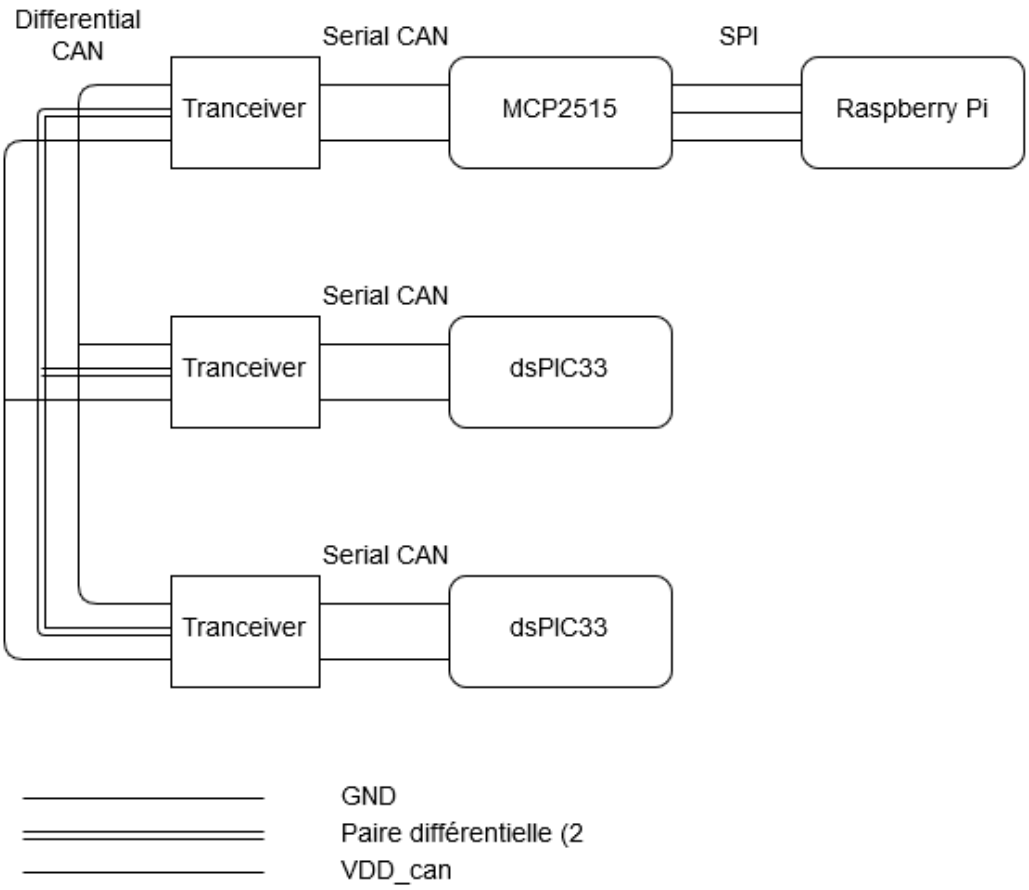
- L'ordre ne peut être donnée que pendant une fenêtre temporelle fixée avant le vol.
- Le Séquenceur Trajectographie doit indiquer que l'attitude est dans le cône nominal.
- La fusée doit être séparée depuis au moins N secondes (N à déterminer). Pour détecter la séparation, le séquenceur lit les états de trois capteurs (2 capteurs fin de course, 1 Capteur inductif). Si au moins 2 des 3 capteurs indiquent que la fusée est séparée, alors la condition est remplie.
- La coiffe doit encore être sur la fusée. Pour détecter la présence de la coiffe, le séquenceur lit l'état d'un capteur fin de course.

Si ces conditions sont remplies en même temps, le séquenceur donne l'ordre de mise à feu (ce qui fait aussi sonner le buzzer).

En configuration monoétage, le séquenceur Allumage à été reconverti pour faire sonner le buzzer afin d'indiquer l'état de la fusée.

Bus CAN

Le bus CAN permet de faire communiquer entre eux de façon sécurisée tous les appareils de la case électronique.



Couche physique

Le CAN se divise en deux parties, la partie série et la partie différentielle, séparées par des transceiver. Le transceiver est un composant dont le rôle est de faire le pont entre la partie série et la partie différentielle du CAN, tout en gardant les deux parties isolées électriquement l'une de l'autre.

- La partie série qui relie le microcontrôleur à un transceiver et composée de 2 pistes. Elle fonctionne comme l'UART : une piste transmet les données du microcontrôleur au transceiver, l'autre du transceiver au microcontrôleur. Comme la Raspberry ne peut pas contrôler elle-même le CAN, on utilise un composant intermédiaire, le MCP2515.
- La partie différentielle relie entre eux tous les transceiver, et est composée de 4 pistes : la masse, le 5V (VDD), et la paire différentielle. La valeur d'un bit sur la partie différentielle du CAN est déterminée par la différence de potentiel entre les deux pistes de la paire différentielle (d'où son nom).

Messages

Un message CAN est composé d'un identifiant codé sur 11 bits et de 0 à 8 bytes de données. Lorsqu'un message est transmis sur le CAN, tous les appareils le reçoivent, y compris l'expéditeur. Les microcontrôleurs peuvent être configurer pour filtrer les messages en fonction de leur identifiant. Si plusieurs appareils essaient de transmettre en même temps, le message avec l'identifiant le plus faible est prioritaire.