

AGRÉGATION

SESSION 2005

CONCOURS INTERNE

Section : GÉNIE ÉLECTRIQUE

Option A : ÉLECTRONIQUE ET INFORMATIQUE INDUSTRIELLE

ÉTUDE D'UN SYSTÈME INDUSTRIEL

DURÉE : 8 HEURES, COEFFICIENT : 1

Aucun document n'est autorisé.

Calculatrice autorisée (conformément à la circulaire n° 99-186 du 16 novembre 1999)
Convertisseur en Euro autorisé

SYSTÈME DE GESTION DES ÉQUIPEMENTS D'UN AUTOBUS

Ce sujet comporte 4 dossiers distincts :

📁 Dossier de présentation	12 pages
📁 Texte du sujet avec le travail demandé	10 pages
📁 Documents réponses DR1 à DR4	3 pages
📁 Documents annexes	64 pages
📁 Schémas structurels format A3	7 pages

Ce sujet comporte trois parties indépendantes :

Partie **A** : Étude du système.

Partie **B** : Étude d'une unité d'entrées sorties.

Partie **C** : Étude de la communication.

Une lecture préalable et complète du sujet est indispensable.

Il sera tenu compte de la cohérence avec laquelle les candidats traiteront chaque partie, le jury préférant une réflexion d'ensemble de la partie abordée à un éparpillement des réponses.

Les questionnements portant sur les développements pédagogiques se verront affectés de 60 % des points de l'épreuve. Il est donc conseillé d'y consacrer 60 % du temps de l'épreuve.

Les candidats sont invités à numérotter chaque page de leur copie et à indiquer clairement le numéro de la question traitée.

Les candidats sont priés de rédiger les différentes parties du problème sur feuilles séparées et clairement repérées. Chaque question est identifiée par une police italique et repérées par un numéro.

Il leur est rappelé qu'ils doivent utiliser les notations propres au sujet, présenter clairement les calculs et dégager ou encadrer tous les résultats.

Tout résultat incorrectement exprimé ne sera pas pris en compte. En outre les correcteurs leur sauront gré d'écrire lisiblement et de soigner la qualité de leur copie.

Il sera tenu compte de la qualité de rédaction, en particulier pour les réponses aux questions ne nécessitant pas de calcul. Le correcteur attend des phrases complètes respectant la syntaxe de la langue française.

Pour la présentation des applications numériques, il est rappelé que lors du passage d'une forme littérale à son application numérique, il est recommandé aux candidats de procéder comme suit :

- après avoir rappelé la relation littérale, chaque grandeur est remplacée par sa valeur numérique en respectant la position qu'elle avait dans la relation puis le résultat numérique est donné sans calculs intermédiaires et sans omettre son unité.

Si le texte du sujet, de ses questions ou de ses annexes, vous conduit à formuler une ou plusieurs hypothèses, il vous est demandé de la (ou les) mentionner explicitement dans votre copie.

Sommaire de la présentation du système

1	PRÉSENTATION DU SYSTÈME	10
1.1	Présentation générale	10
1.1.1	Modules génériques.....	10
1.1.2	Le multiplexage au sein d'un véhicule	10
1.1.3	Description des unités d'entrée sortie.....	11
1.2	Système de gestion des équipements d'un autobus.....	11
1.2.1	Fonctionnalités	11
1.2.2	Architecture du système de gestion des équipements de l'autobus.....	12
2	SOUS SYSTÈME D'OUVERTURE DES PORTES	13
2.1	Présentation	13
2.2	Schématisation du système d'ouverture des portes	13
2.2.1	Architecture matérielle	13
2.2.2	Schéma fonctionnel.....	14
2.3	Acquisition des entrées (FP1)	16
2.4	Transmission des mesures via les réseaux CAN (FP2)	17
2.5	Contrôle de la porte (FP3).....	17
2.5.1	Echange de données	18
2.5.2	Description des différents états de la porte.....	18
2.6	Transmission des commandes (FP5)	19
2.7	Commande des actionneurs FP6.....	19
2.8	Gestion des tâches FP7	19

DOSSIER DE PRÉSENTATION

1 PRÉSENTATION DU SYSTÈME

1.1 PRÉSENTATION GÉNÉRALE

Dans les véhicules industriels (tramways, autobus, camions), les fonctionnalités offertes par les équipements électroniques sont de plus en plus nombreuses et variées. Le système doit être adaptable au cahier des charges du client.

Compte tenu des multiples fonctionnalités offertes, l'interconnexion des équipements électriques est de plus en plus complexe. Une architecture conventionnelle basée sur un câblage classique n'est plus adaptée, pour des raisons de complexité, fiabilité et coût de mise en œuvre. Les véhicules actuels utilisent des réseaux locaux pour interconnecter les équipements.

La solution de la société ACTIA est une solution générique d'interconnexion des équipements destinée à contrôler les fonctions électriques. Elle s'adapte sur différents véhicules industriels. Elle est personnalisée en fonction des souhaits des clients. Cette solution est composée d'un ensemble de modules calculateurs spécialisés pour des véhicules, communiquant entre eux par réseau CAN.

1.1.1 MODULES GÉNÉRIQUES

Il existe plusieurs types de modules génériques :

Des modules spécialisés pour le tableau de bord :

- Les « SLU » (Screen and Ligh Unit) : ce sont des modules d'entrées sorties permettant d'interfacer des diodes ou des ampoules, des commutateurs, interrupteurs ou boutons poussoirs.
- Les « SCU » (Screen Control Unit) et « MCU » (Management Control Unit) : elles permettent de gérer les SLU au niveau applicatif et assurent les fonctions de gestion du réseau CAN. Les SCU possèdent une interface graphique sous forme d'afficheur à cristaux liquides.



Des modules spécialisés pour le véhicule :

- Les unités d'entrée sortie « IOU » (Input Output Unit) : ce sont des modules permettant d'interfacer des capteurs – actionneurs sur le réseau CAN.
- Les unités de supervision CMU (Central Management Unit) : elles permettent de contrôler le fonctionnement du système. Elles pilotent les unités d'entrées sorties (IOU) à travers le réseau CAN et disposent également d'entrées ou de sorties propres.

1.1.2 LE MULTIPLEXAGE AU SEIN D'UN VÉHICULE

Le multiplexage correspond à l'interconnexion des équipements au sein d'un véhicule.

L'architecture générique comporte trois parties :

L'interconnexion des équipements associés au châssis (moteur, suspension, boîte de vitesse, freins).

L'interconnexion des équipements du corps du véhicule et son environnement (portes, éclairage externe et interne, signalisation, essuie glace, climatisation, agenouillement du bus).

Et enfin l'interconnexion des équipements du tableau de bord.

D'autres options sont également proposées; elles concernent généralement des équipements de confort (radio, éclairage, etc.).

Suivant les fonctionnalités, les spécifications pour le réseau CAN ne sont pas les mêmes, on distingue par exemple :

- Inter system bus CAN SAE J1939 au niveau du châssis,
- Intra system CAN network ISO 11898 au niveau du corps du véhicule,
- J1708 pour les équipements optionnels de confort.

La société ACTIA utilise pour des couches applicatives spécifiques : MULTIBUS CAN pour les équipements du corps du véhicule et MULTIBUS Dashboard pour le tableau de bord.

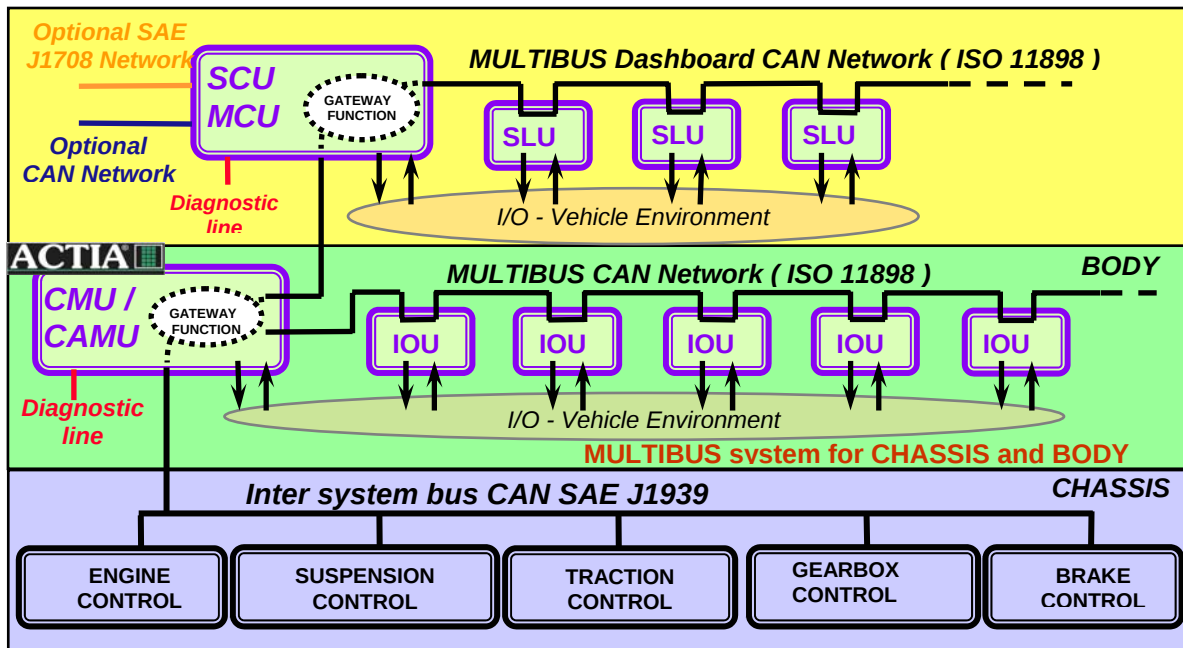
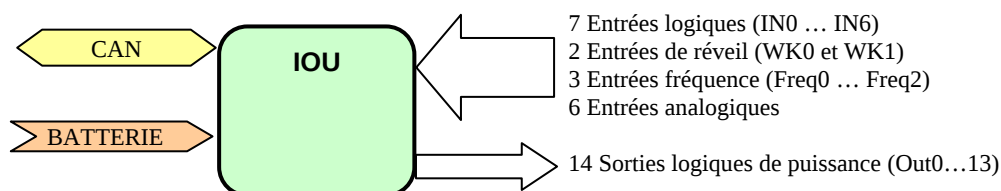


Figure 1. Architecture générique des réseaux au sein d'un véhicule :

1.1.3 DESCRIPTION DES UNITÉS D'ENTRÉE SORTIE



Les 14 sorties logiques de puissance se décomposent de la manière suivante :

- 2 sorties pouvant délivrer 9 A (Out 0, Out12)
- 4 sorties pouvant délivrer 7 A (Out 5, Out11, Out13, Out14)
- 3 sorties pouvant délivrer 3,2 A (Out1, Out2 et Out3)
- 6 sorties pouvant délivrer 2 A (Out 4, Out6, Out7, Out8, Out9 et Out10)

4 sorties pouvant former deux ponts en H commandés en PWM pouvant délivrer 5 A

Ces ponts sont équipés d'une mesure de courant en interne (CPU_SHUNT0 pour le pont constitué de Out11 et Out12 et CPU_SHUNT1 pour le pont constitué de Out13 et Out14)

A chacune de ces 14 sorties est associée une entrée logique de mesure de défaut (CPU_STAx)

Toutes les entrées peuvent être configurées pour être utilisées en entrées logiques.

Les modules sont alimentés par les connecteurs CN1 et CN5.

- Au connecteur CN1 : V_{AMS} ou V_{ACB} . (ACB : Après coupe-batterie = AMS : After Master Switch)
Alimentation principale de puissance du module
- Au connecteur CN5 : (Vbat Unswitched)
Alimentation secondaire du module. Permet la sortie du mode veille du module et l'utilisation de certaines sorties lorsque le contact est coupé.

1.2 SYSTÈME DE GESTION DES ÉQUIPEMENTS D'UN AUTOBUS

1.2.1 FONCTIONNALITÉS

Dans le cas de l'autobus à 3 portes de la ville de Lyon, les fonctionnalités que doit gérer le système sont :

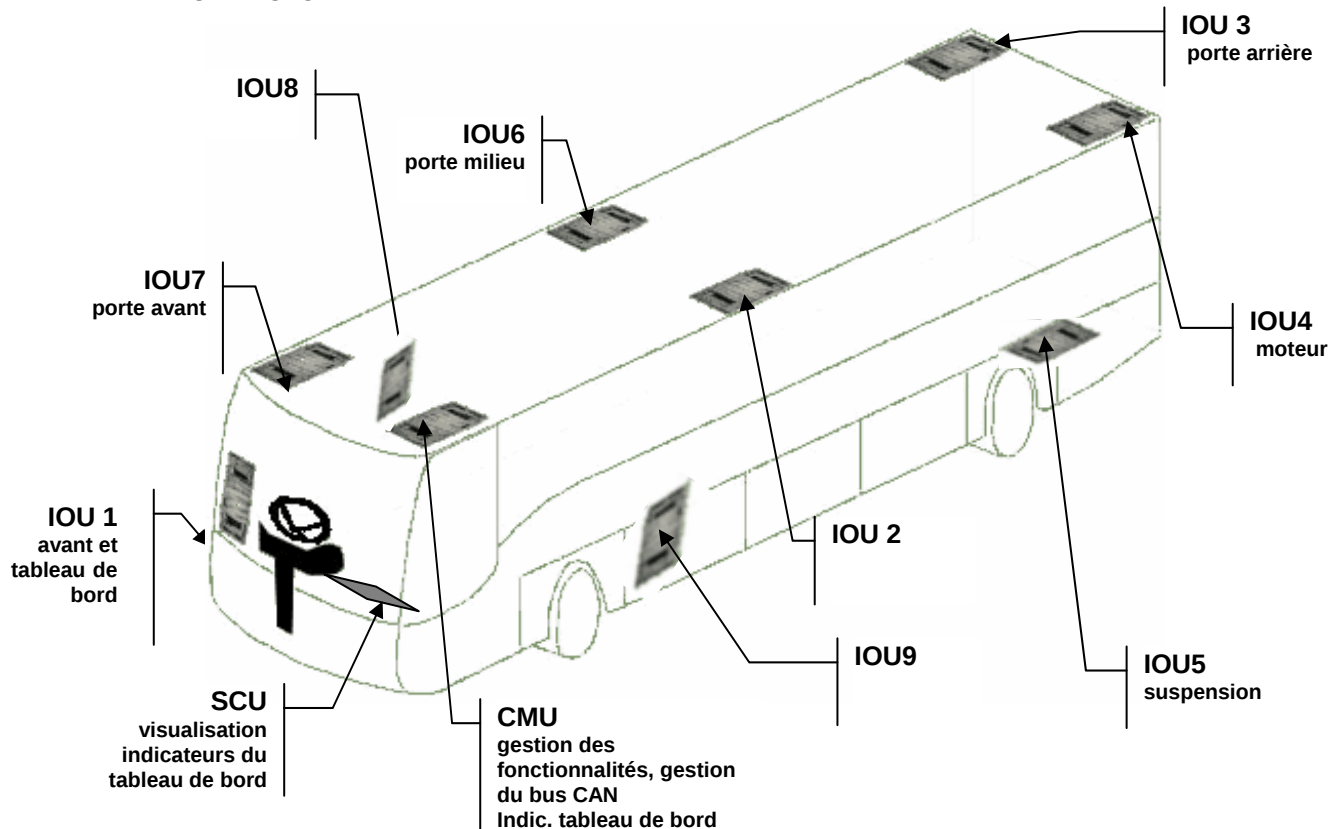
- La commande de l'ouverture et la fermeture des portes

- La gestion de la rampe d'accès pour handicapé
- La commande des essuie-vitres
- La gestion de l'affichage des panneaux (arrêt demandé, etc.)
- La commande du microphone et de la caméra
- La commande de l'éclairage externe (feux, phares, etc.) et interne du bus
- Le contrôle du moteur
- Le contrôle des freins
- La gestion de la suspension (abaissement, soulèvement, agenouillement du véhicule)
- La commande des équipements thermiques (dégivrage, chauffage, climatisation)
- La commande des voyants de défaut

Le fonctionnement dépend du cahier des charges fourni par le client de l'autobus.

1.2.2 ARCHITECTURE DU SYSTÈME DE GESTION DES ÉQUIPEMENTS DE L'AUTOBUS

• ARCHITECTURE MATÉRIELLE



Pour l'ensemble de l'autobus, neuf unités d'entrées sorties (IOU) sont associées aux capteurs actionneurs du moteur et du châssis. Ces unités d'entrées sorties sont pilotées par une unité de supervision (CMU) qui dialogue avec l'unité de contrôle de la visualisation (SCU) associé au tableau de bord.

C'est le CMU qui contrôle le fonctionnement et réalise « l'application client » c'est-à-dire les fonctionnalités du cahier des charges du client (modalités de commande de l'ouverture et la fermeture des portes, gestion de la rampe d'accès pour handicapé, des essuie vitres, de l'éclairage, le contrôle du moteur, des freins, etc.).

• ARCHITECTURE LOGICIELLE

Les modules d'entrées sorties (IOU) et les modules de visualisation (SLU) sont des modules esclaves. Ce sont des calculateurs n'ayant pas de logiciel lié à l'application client. Ils répondent exclusivement aux ordres du calculateur maître (CMU) transitant via le réseau CAN.

Les modules de gestion du fonctionnement et de la visualisation (CMU, SCU) sont les modules maîtres. Ce sont des calculateurs qui contiennent des structures logicielles liées aux fonctionnalités souhaitées par l'application client. Ils échangent des données applicatives entre eux via un réseau CAN et pilotent les modules esclaves via le réseau CAN.

L'architecture logicielle des unités de supervision (CMU) est composée de trois parties

- * boot
- * moniteur

* application client (gestion des équipements de l'autobus, gestion des portes, gestion de l'éclairage, de l'agenouillement, etc.).

Les fonctionnalités que doivent assurer le CMU sont organisées en tâches. Le séquençage de ces tâches est assuré par le moniteur qui s'appuie sur le noyau temps réel RTX Tiny. L'appel des tâches est effectué cycliquement avec un temps de cycle de 50 ms.

Dans la suite de l'étude, nous nous limiterons au sous système d'ouverture des portes.

2 SOUS SYSTÈME D'OUVERTURE DES PORTES

2.1 PRÉSENTATION

L'autobus étudié comprend 3 portes : une porte avant, une porte centrale (nommée : *Milieu1*) et une porte arrière.

Les portes de l'autobus sont constituées de deux vantaux. Ces deux vantaux sont couplés mécaniquement et pilotés par un seul moteur à courant continu.

Une porte peut être ouverte à partir de :

- L'appui sur un bouton poussoir situé sur le tableau de bord du bus
- L'appui sur un bouton poussoir situé au niveau de la porte à l'intérieur du bus
- L'appui sur un bouton poussoir situé à l'extérieur du bus (pour la porte avant seulement)
- Une porte peut être fermée à partir de l'appui sur un bouton poussoir situé sur le tableau de bord du bus

L'ouverture et la fermeture des portes doivent respecter des contraintes de sécurité. Par exemple :

- L'ouverture des portes peut être interdite tant que le véhicule n'est pas à l'arrêt ou encore si le véhicule est en cours d'abaissement, de soulèvement ou d'agenouillement
- Le conducteur doit autoriser l'ouverture des portes (bouton poussoir « habilitation porte » ou « autorisation porte avant de l'extérieur » situé sur le tableau de bord)

Le chauffeur du bus doit être prévenu de la demande d'arrêt d'un passager. Des boutons poussoirs « arrêt demandé » sont répartis dans le bus. Un affichage de la demande est effectué dans le bus et sur le tableau de bord.

Les principales grandeurs en sorties sont les alimentations des éléments suivants :

- Moteurs des portes (*Avant, Arrière et Milieu1*)
- Freins des portes (*Avant, Arrière et Milieu1*)
- Voyant arrêt demandé sur le tableau de bord
- Voyant toutes portes fermées sur le tableau de bord
- Voyant autorisation de l'ouverture de la porte avant de l'extérieur
- Voyant Portes ouverte (*Avant, Arrière, Milieu1*) sur le tableau de bord
- Voyant inhibition des sécurités des portes
- Voyant habilitation du bouton poussoir porte (*Avant, Arrière, Milieu1*)
- Éclairage arrêt demandé (*porte Milieu1, Avant*),
- Sonnerie arrêt demandé

2.2 SCHÉMATISATION DU SYSTÈME D'OUVERTURE DES PORTES

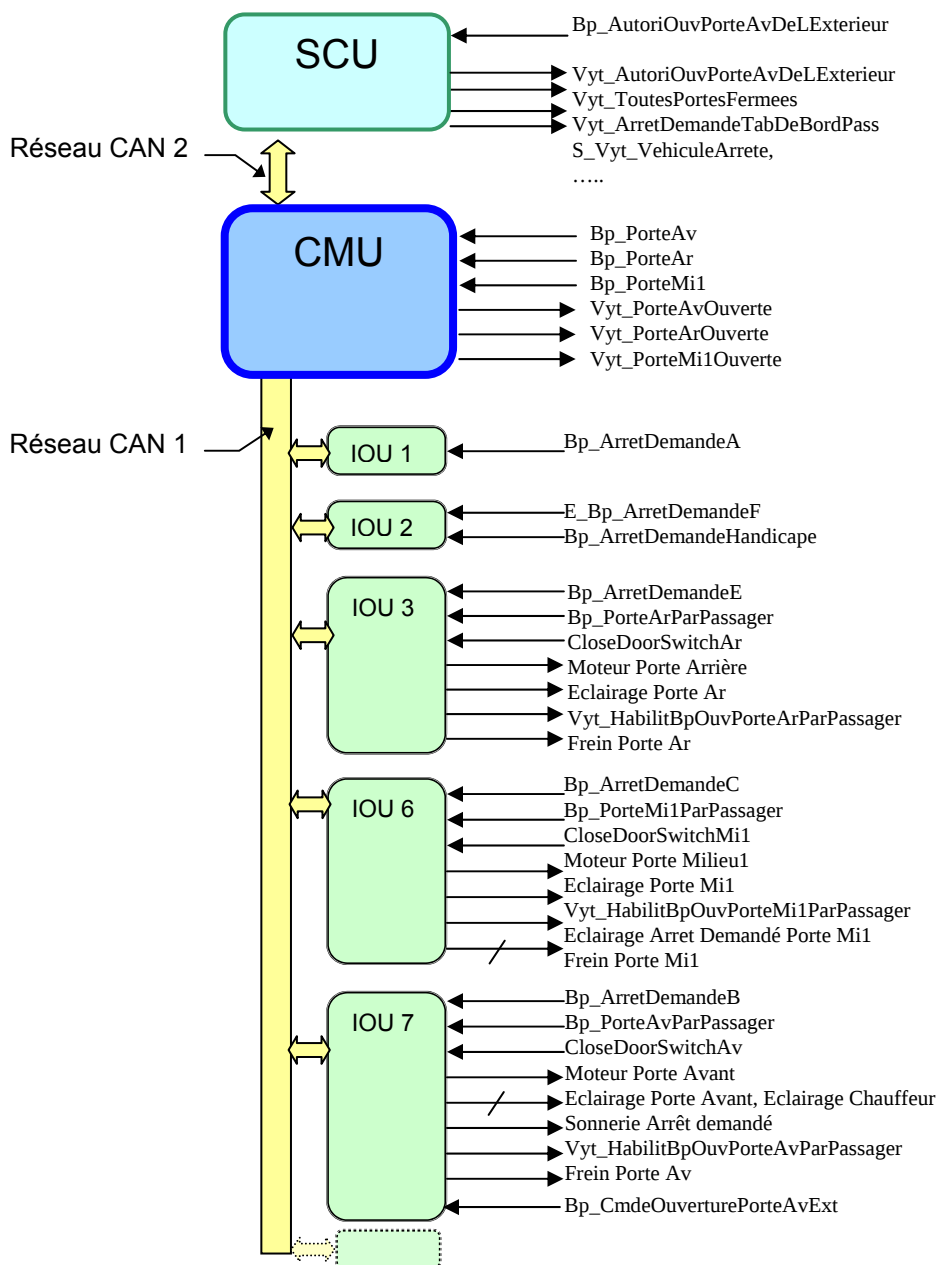
2.2.1 ARCHITECTURE MATÉRIELLE

Le système d'ouverture de portes de l'autobus est essentiellement basé sur six unités d'entrées sorties (IOU), une unité de supervision (CMU) et une unité de contrôle de la visualisation (SCU).

Les unités d'entrées sorties (IOU) sont reliées aux capteurs (capteur de position de la porte, bouton poussoir de demande d'ouverture, ...) et aux actionneurs (moteurs, voyants état des portes, etc.). L'IOU7 assure la commande du moteur de la porte avant, l'IOU6 la commande de la porte du milieu et l'IOU3 celle de la porte arrière. Ces unités d'entrées sorties (IOU) alimentent les moteurs des portes à partir de la tension fournie par la batterie via un pont en H. Les capteurs associés à la gestion de la porte sont reliés aux unités d'entrées sorties suivant leur localisation dans l'autobus. Les unités d'entrées sorties communiquent avec l'unité de supervision (CMU) via un réseau CAN (réseau CAN1).

C'est l'unité de supervision (CMU) qui contient le logiciel de gestion de l'ouverture des portes. Ce module est également en interaction avec :

- Une unité de contrôle de la visualisation (SCU) qui gère plus particulièrement les équipements raccordés au tableau de bord via un second réseau CAN (CAN2).
- D'autres unités d'entrées sorties (IOU) gérant d'autres équipements (vitesse véhicule, état de la rampe d'accès pour handicapé etc.) via le réseau CAN1. Ceux-ci n'ont pas été représentés.



Remarque :

Les unités d'entrées sorties sont également utilisées pour des capteurs et actionneurs concernant d'autres fonctionnalités que la gestion des portes.

2.2.2 SCHÉMA FONCTIONNEL

Sauf indications contraires, la description se limite au fonctionnement normal du système, la phase d'initialisation étant terminée.

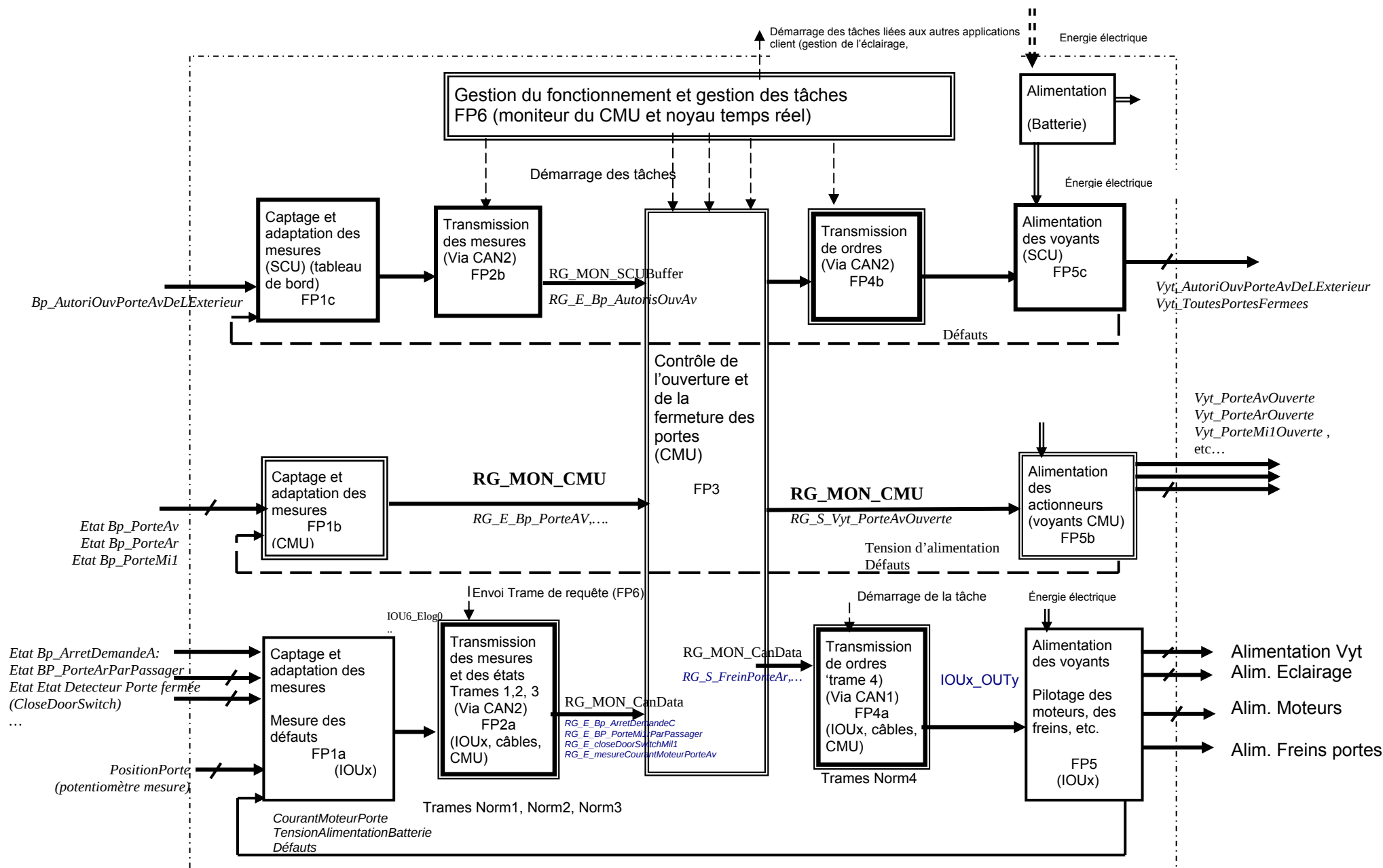


Figure 2. Schéma fonctionnel du système de commande de l'ouverture de portes pour le fonctionnement normal (initialisation terminée)

Toutes les entrées sorties fonctionnelles ne sont pas représentées

2.3 ACQUISITION DES ENTRÉES (FP1)

• VUE D'ENSEMBLE DES FONCTIONS FP1A, FP1B, FP1C

Ces fonctions sont assurées par un ensemble de structures (matérielles et logicielles) situées dans les unités d'entrées sorties IOU (FP1a), dans le CMU (FP1b) et le SCU (FP1c).

Elles permettent d'assurer l'acquisition des données externes comme :

- État des boutons poussoir ouverture de porte par passager (*Avant, Milieu1, Arrière*)
- État des boutons poussoir arrêt demandé (*A, B, C, D, E, Handicapé*)
- États des détecteurs porte fermée (*Avant, Milieu1, Arrière*)
- États des boutons poussoir extérieur d'ouverture de la porte avant
- État du bouton poussoir autorisation ouverture porte avant de l'extérieur (sur le tableau de bord)
- La position de la porte

Elles permettent également d'assurer l'acquisition de données internes au système comme :

- Défaut alimentation des moteurs (*Avant, Milieu1, Arrière*)
- Défaut des freins des moteurs de portes (*Avant, Milieu1, Arrière*)
- Défaut de la commande de la direction des moteurs de portes (*Avant, Milieu1, Arrière*)
- Le courant dans les moteurs (*Avant, Milieu1, Arrière*)
- La tension de la batterie (IOU3, IOU6, IOU7) et du CMU
- La température des IOU et du CMU

Suivant la nature des entrées (logiques, analogiques, fréquentielles) divers traitements (amplification, filtrage, compensation des offsets, comparaison à un seuil) sont effectués. Pour une entrée analogique, la sortie corrigée est reliée à l'entrée EAna0 par une relation du type :

$$Scorr = (K2 \cdot E_Ana0 / K1 + K2 \cdot Dt1 + Dt2 + B) \cdot A$$

K1, K2, Dt1 et Dt2 sont des imperfections de la chaîne analogique. A et B sont là pour les corriger, en fonction de l'estimation de K1, K2, Dt1 et Dt2 obtenue lors du calibrage.

La grandeur d'entrée est numérisée et filtrée par un filtre numérique. Les paramètres de traitement sont stockés une Eprom associée à chaque unité d'entrées sorties (de même pour le CMU et le SCU).

Le calculateur des modules (IOU, CMU, SCU) est muni d'un moniteur qui s'appuie sur un noyau temps réel. Les tâches d'acquisition des entrées s'effectuent avec un temps de cycle de 50 ms.

Les sorties sont de variables logicielles.

La fonction FP1b qui assure l'acquisition des entrées du CMU, communique avec le logiciel de gestion de portes (FP3) via la structure de donnée RG_MON_CMU qui contient l'état des entrées (par exemple RG_E_Bp_PorteAV)

• MESURE DE LA POSITION DE LA PORTE

La position de la porte est mesurée par un potentiomètre (0 – 2,2 kΩ) couplé à la partie mécanique.

Un calibrage du capteur est réalisé de sorte que :

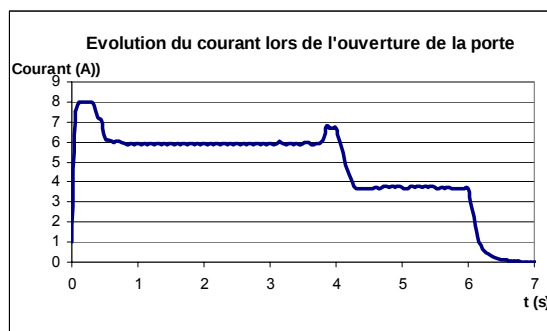
- Porte position fermée → la sortie corrigée RG_E_PositionPorte est égale à 0
- Porte position complètement ouverte → la sortie corrigée RG_E_PositionPorte est égale à 100

La position de la porte est également estimée à partir de l'évolution du courant dans le moteur (très utile pour le fonctionnement en mode dégradé).

• MESURE DU COURANT DANS LE MOTEUR

Le courant dans le moteur est mesuré dans l'unité d'entrées sorties (IOU).

Figure 3



Avant la livraison de l'autobus, une procédure de test et de calibrage est faite pour chaque porte à l'ouverture et à la fermeture. En phase d'exploitation, la valeur du courant donnera donc une indication sur l'état de la porte.

Le courant dans le moteur dépend de la vitesse de fermeture ou d'ouverture souhaitée, de la tension de la batterie, de la température, du dévers, du nombre de cycles d'ouverture fermeture dans la demi-heure précédente, mais aussi du vieillissement de la porte.

Donc en phase d'exploitation, l'évolution de la position de la porte ainsi que celle du courant dans le moteur sont mesurées à chaque ouverture et fermeture de porte.

Ces mesures serviront à faire évoluer le gabarit du courant pour les cycles suivants de fonctionnement. Les gabarits et le pilote de la porte sont auto adaptatifs.

Si le courant dans le moteur est supérieur à la valeur définie dans le gabarit, cela peut signifier un obstacle à la fermeture ou à l'ouverture de la porte. La mesure du courant dans le moteur participe donc à la détection d'obstacle. Ce point est important en cas de fonctionnement en mode dégradé (défaut capteur position par exemple).

En conclusion, la mesure du courant servira :

- À déterminer la position de la porte
- À participer à la détection d'un obstacle
- À faire évoluer le gabarit pour l'ouverture et la fermeture en fonction du vieillissement.

Ces fonctions sont réalisées par l'unité de supervision (CMU) dans la fonction FP3 (gestion des portes).

2.4 TRANSMISSION DES MESURES VIA LES RÉSEaux CAN (FP2)

• TRANSMISSION DES MESURES DES IOU VERS LE CMU (RÉSEAU CAN1) FP2A

Rôle de la fonction

Le rôle de cette fonction est de gérer l'émission des données des unités d'entrées sorties (IOU) et leur réception par l'unité de supervision (CMU) via le réseau CAN1. Elle est donc chargée de la création et l'émission des trames par un IOU d'une part, de la réception des trames et l'extraction des données par le CMU d'autre part.

Cette fonction est réalisée à l'aide :

- D'un moniteur embarqué dans le calculateur de l'IOU, d'un contrôleur CAN (pour chaque IOU)
- Des connecteurs et du câblage réseau CAN
- D'un contrôleur CAN et du moniteur du CMU.

Chaque IOU dispose de trois trames pour envoyer ses mesures vers le CMU (*Norm1, Norm2, Norm3*).

Description du réseau

Le réseau a une topologie de type bus (réseau non bouclé avec une seule branche). Le réseau répond aux spécifications CAN 2.0 part B. Les identifiants sont en mode étendu. Le débit est de 250 kbits/s.

La société ACTIA utilise une couche applicative spécifique pour le réseau CAN (MULTIBUS).

L'unité d'entrées sorties (IOU) se comporte en esclave et répond à une trame de requête en provenance de l'unité de supervision (CMU). L'émission de ces trames de requête est gérée par le moniteur de l'unité de supervision (tâche A du moniteur du CMU, cf. Fonction FP6).

Les IOU sont repérées par une adresse logique qui est utilisée pour définir les identifiants des trames CAN qui les concernent, ainsi que l'ordre d'adressage.

Toutes les 50 ms, des trames de requête sont envoyées par le CMU.

Le contrôleur CAN du CMU stocke les données des trames reçues dans ses registres.

Régulièrement le CMU vient lire les données de toutes les trames du réseau CAN2 et les stockent dans la zone de mémoire partagée de la RAM.

La lecture des registres du contrôleur CAN avec la mise à jour de la zone de mémoire partagée du CMU est découpée en 4 tâches.

• TRANSMISSION DES MESURES DU SCU VERS LE CMU (RÉSEAU CAN2) FP2B

Le réseau a une topologie en bus et répond aux spécifications CAN 2.0 part B. Les identifiants sont en mode étendu. Le débit est de 250 kbits/s. Les deux réseaux CAN1 et CAN2 sont indépendants.

L'unité de contrôle de la visualisation (SCU) fonctionne en maître secondaire et non pas en esclave comme les unités d'entrées sorties (IOU).

Les données de toutes les trames du réseau CAN2 sont stockées dans la zone de mémoire partagée de la RAM de l'unité de supervision (CMU).

2.5 CONTRÔLE DE LA PORTE (FP3)

C'est le calculateur de l'unité de supervision (CMU) qui contrôle le fonctionnement de la porte et réalise les fonctionnalités de l'application imposée par le cahier des charges. Il n'y a pas de logiciel lié à l'applicatif du contrôle des portes dans les unités d'entrées sorties (IOU).

Les sorties de cette fonction sont les paramètres de commandes

- Des moteurs des portes avant, arrière et milieu (arrêt/marche, sens de rotation, rapport cyclique de la tension d'alimentation -- neuf valeurs possibles)

- Des freins des portes avant, arrière et milieu (deux états possibles)
- Des voyants reliés aux IOU, aux SCU, aux CMU
- De l'éclairage des panneaux.

Les données de toutes les trames du réseau CAN1, du réseau CAN2, ainsi que les données échangées avec le module d'entrées sorties intégré au CMU, sont stockées dans une zone de mémoire partagée de la RAM de l'unité de supervision (CMU).

Trois structures de données sont utilisées :

- *RG_MON_SCUBuffer* pour les échanges avec le SCU (réseau CAN2)
- *RG_MON_CanData* pour les échanges avec les unités d'entrées sorties (IOU) (réseau CAN1)
- *RG_MON_CMU* pour les entrées sorties directement reliées au CMU.

Cette fonction s'appuie sur trois tâches principales :

- La tâche *MAJ_Entree* a pour rôle de lire les données contenues dans ces structures, de mettre à jour les variables de l'application avec les données en provenance des SCU, des IOU et des entrées directement reliés au CMU
- La tâche *Exe_fonctionsporte* qui détermine les commandes à appliquer aux moteurs des portes et aux voyants
- La tâche *Maj_Sortie* qui place les commandes destinées aux IOU dans la table *RG_MON_candata*, dans *RG_MON_scubuffer* pour les commandes du SCU et dans *RG_MON_CMU* pour les sorties directement reliées au CMU.

Le rôle de la tâche *Exe_FonctionsPorte* est de :

- Déterminer l'état de la tension d'alimentation de chaque IOU qui pilote le moteur d'une porte (15 états possibles)
- Gérer les voyants de défaut, gérer l'inhibition de la sécurité des portes, l'autorisation d'ouverture des portes par les passagers
- Gérer le fonctionnement de la porte. C'est-à-dire :
 - Déterminer l'état du fonctionnement de chaque porte, détecter la présence d'obstacle ou de dysfonctionnement
 - Décider de l'évolution et calculer les commandes à appliquer aux sorties (sens de rotation du moteur et commande PWM)
- Gérer les voyants de signalisation de l'état des portes
- Déterminer les commandes à appliquer aux panneaux arrêt demandé et la sonnerie correspondant à une demande d'arrêt d'un passager.

2.5.1 ECHANGE DE DONNÉES

L'accès aux données d'entrées sorties des IOU se fait par deux index élaborés à partir du numéro global de la trame et du numéro de l'octet de la trame choisie. Pour obtenir le numéro global d'une trame, il faut connaître l'unité d'entrée sortie (IOU) à laquelle elle appartient.

Pour les données en provenance du SCU, le logiciel de gestion de portes (FP3) utilise la table *RG_MON_SCU_BUFFER*.

Pour les données échangées avec le module d'entrées sorties intégré au CMU, le logiciel de gestion de portes (FP3) utilise la structure de donnée *RG_MON_CMU*.

```
typedef struct // extrait du fichier RGmon.h
{
    T_BOOLEEN ELog0;
    T_BOOLEEN EAnaOL;
    T_BOOLEEN PresACB;
    T_BOOLEEN ERev0;
    ..
}T_RG_MON_CMU;
```

2.5.2 DESCRIPTION DES DIFFÉRENTS ÉTATS DE LA PORTE

Le fonctionnement de l'application de gestion des portes (Tâche *Exe_FonctionsPorte*) est essentiellement basé sur des machines à états finis.

Pour la position de la porte, les états sont : porte fermée, porte non fermée et porte ouverte.

Le nombre d'états considérés pour la porte dépend du mode de fonctionnement : nominal ou dégradé. Le mode dégradé est utilisé en cas de défaut des capteurs, de la communication entre l'unité de supervision (CMU) et les unités d'entrées sorties (IOU).

États pour la porte en fonctionnement « mode nominal » :

- Inactive
- En cours de fermeture
- En cours de réouverture suite à un obstacle détecté à la fermeture

- Inactive suite à un défaut lors de la réouverture
- En cours de fermeture après un arrêt lors de la fermeture (suite à un obstacle)
- En cours d'ouverture
- En cours de fermeture suite à un obstacle détecté à l'ouverture
- Inactive suite à un défaut lors de l'ouverture (attente avant nouvel essai d'ouverture)
- En cours de fermeture après un arrêt lors de l'ouverture (suite à un obstacle)
- En cours de réouverture après un arrêt lors de l'ouverture (suite à un obstacle)
- Frein actif
- Frein inactif.

États pour la porte en fonctionnement « mode dégradé » :

- Inactive
- En cours de fermeture
- En cours d'ouverture.

2.6 TRANSMISSION DES COMMANDES (FP5)

Le rôle de cette fonction FP5A est de gérer la transmission par l'unité de supervision (CMU) des actionneurs reliés aux unités d'entrée sortie (IOU) via le réseau CAN1, de leur réception par les IOU. Cette fonction s'appuie sur le même réseau et les mêmes composants que pour la fonction FP2a. Chaque unité d'entrée sortie (IOU) dispose d'une trame (*Norm4*) pour recevoir les commandes qui lui sont destinées. Les données à envoyer dans la trame sont stockées dans une zone de mémoire partagée de la RAM de l'unité de supervision (CMU). La mise à jour du contrôleur CAN du CMU avec les commandes à envoyer est réalisée par des tâches nommées *C1*, *C20*, *C3*, *C4*. Ces tâches sont pilotées par le moniteur du CMU (cf. fonction FP7).

Le rôle de la fonction FP5B est de gérer la transmission par l'unité de supervision (CMU) des commandes à destination de l'unité de contrôle de la visualisation (SCU) via le réseau CAN2.

2.7 COMMANDE DES ACTIONNEURS FP6

Le rôle de cette fonction est de générer la tension d'alimentation des voyants, les alimentations des moteurs, des freins, de la sonnerie et de l'éclairage des panneaux en fonction des commandes extraites des trames CAN *Norm4* envoyées par le CMU.

L'énergie nécessaire est fournie par la batterie via les IOU.

Le calculateur d'un IOU est muni d'un moniteur qui s'appuie sur un noyau temps réel. La tâche de lecture des commandes provenant du CMU via le réseau CAN et de mise à jour des sorties s'effectue avec un temps de cycle de 50 ms.

Les principales grandeurs en sorties du système d'ouverture des portes sont :

- Les tensions : *Umoteurx*, différences de potentiel aux bornes du moteur, de type PWM, d'amplitude 24 V, de fréquence 50 khz, de rapport cyclique fourni par le CMU (*RG_S_PWM_Freq_cmdemoteurportex*)
- Les tensions *Ufreinx* : (frein des portes avant, arrière et milieu) prennent deux états
- Les tensions *Uvoyantxx* alimentant :
 - Les voyants habilitation bouton poussoir porte
 - Les voyants portes ouverte sur le tableau de bord
 - Le voyant autorisation de l'ouverture de la porte avant de l'extérieur
 - Le voyant arrêt demandé sur le tableau de bord
 - Le voyant toutes portes fermées sur le tableau de bord
 - Le voyant inhibition des sécurités des portes
- La tension *USonnerie Arrêt demandé*
- Les tensions *Uéclairage* (éclairage arrêt demandé milieu et avant).

2.8 GESTION DES TÂCHES FP7

C'est l'unité de supervision (CMU) qui pilote le fonctionnement et réalise les fonctionnalités de l'application imposée par le cahier des charges.

L'architecture logicielle des unités de supervision (CMU) est composée de trois parties :

- Boot
- Moniteur
- Application (dépend du cahier des charges du client et des équipements de l'autobus).

Les fonctionnalités que doit assurer l'unité de supervision (CMU) sont organisées en tâches. Le séquençage de ces tâches est assuré par le moniteur qui s'appuie sur le noyau temps réel *RTX Tiny*. L'appel des tâches est effectué cycliquement avec un temps de cycle de 50 ms.

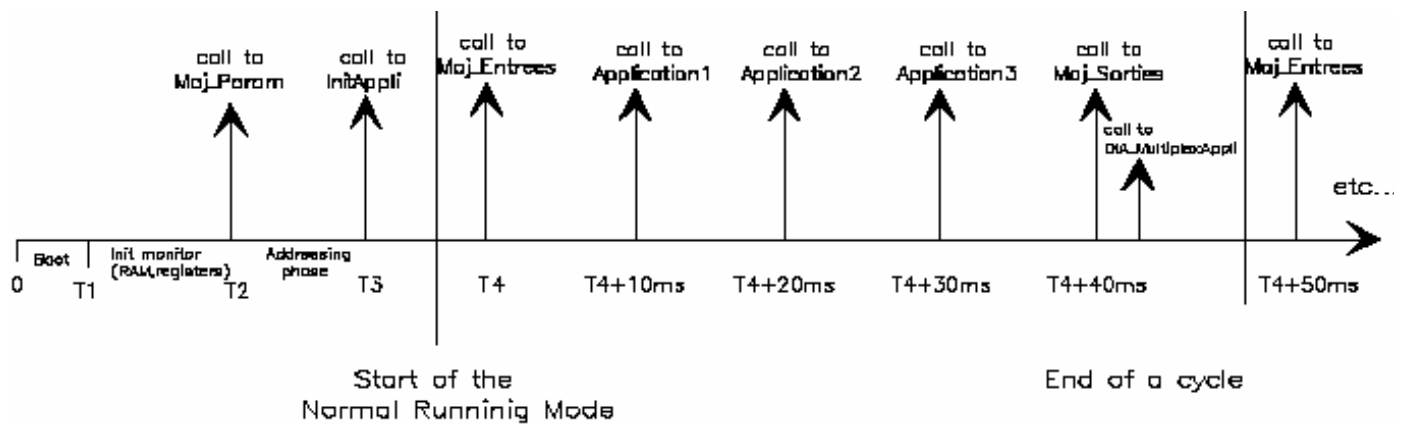


Figure 4: Séquencement des tâches, au démarrage puis en fonctionnement normal.

On distingue deux phases de fonctionnement successives : le démarrage puis le mode normal.

Après la phase de boot et l'initialisation du moniteur, une initialisation du réseau (réveil et adressage) est effectuée, puis on a une initialisation de l'application. Les phases de réveil et d'adressage sont indépendantes de l'application.

En mode « normal », le séquencement des appels à l'applicatif est le suivant :

MAJ_ENTREES à T (actualisation des entrées : lecture de la zone de mémoire partagée)

APPLICATION1 à T + 10 ms

APPLICATION2 à T + 20 ms

APPLICATION3 à T + 30 ms

MAJ_SORTIES à T + 40 ms (actualisation des sorties : écriture de la zone de mémoire partagée)

DIA_MultiplexAppli

La période d'appel des tâches est de 50 ms.

Le cœur de l'applicatif client est organisée en trois sections : *application1* (début), puis *application2* et *application3* (fin). L'applicatif de contrôle des portes (FP3) est lancé dans la tâche « *application3* ».

DIA_MultiplexAppli : a pour fonction principale le diagnostic. Elle permet d'analyser une réponse de la trame de diagnostic et de définir en conséquence la réponse.

En fonctionnement « normal », outre le cadencement des tâches, le moniteur du CMU doit également assurer les tâches spécifiques suivantes :

- La lecture des entrées logiques et analogiques du CMU
- L'émission des trames de requête et des trames de gestion du réseau CAN
- La détection des défauts des sorties du CMU
- L'émission et la réception des trames de données CAN (gestion par blocs) avec la mise à jour de la zone d'échange des données avec l'application
- Le traitement et la mise à jour des sorties logiques et fréquentielles du CMU.

Sommaire du sujet

SUJET	22
A ANALYSE DU FONCTIONNEMENT.....	22
A.1 Étude d'une demande d'Arrêt du bus.....	22
A.2 Etude de l'ouverture de la porte AVANT	23
A.3 Influence de la tension de la batterie.....	24
A.4 Estimation de la position de la porte AVANT	24
A.5 Élaboration des documents de présentation du système	24
B ÉTUDE DE LA STRUCTURE D'UNE UNITÉ D'ENTRÉES-SORTIES	25
B.1 Étude des entrées Logiques.....	25
B.2 Étude des sorties des IOU.....	25
B.3 Étude partielle de la chaîne de mesures analogiques	26
B.4 Exploitation pédagogique concernant les unités d'entrées sorties	26
B.4.1 Présentation de la chaîne d'acquisition analogique.....	27
B.4.2 Définition de la séquence	27
B.4.3 Élaboration détaillée d'une partie de la séquence	28
C ÉTUDE DE LA COMMUNICATION ENTRE LES UNITÉS D'ENTRÉES SORTIES (IOU) ET L'UNITÉ DE SUPERVISION (CMU)	28
C.1 Étude de la liaison CAN.....	28
C.2 Exploitation pédagogique concernant la communication	29
C.2.1 Contexte de l'exploitation pédagogique	29
C.2.2 Documents de présentation de la communication par réseau CAN.....	30
C.2.3 Définition d'une séquence	30
C.2.4 Élaboration détaillée d'une partie de la séquence	30

SUJET

Sauf indications contraires, l'étude est limitée au fonctionnement « normal » du système : les éléments du système (IOU, CMU, etc.) étant « réveillés » et les phases d'adressage, de test et d'initialisation terminées.

A ANALYSE DU FONCTIONNEMENT

A.1 ÉTUDE D'UNE DEMANDE D'ARRÊT DU BUS

On considère le cas d'utilisation suivant : un passager appuie sur le bouton poussoir *arrêt_demandé_C*, les deux panneaux arrêt demandé (avant et milieu du bus) doivent s'allumer, ainsi que le voyant sur le tableau de bord du conducteur et la sonnerie arrêt demandé.

On souhaite expliciter le rôle des différents objets et leur interaction pour ce cas d'utilisation.

On considère que l'unité de supervision (CMU) est décomposée en 3 éléments principaux : un module d'entrée sortie, le moniteur qui gère le fonctionnement du CMU (cf. la description de la fonction gestion des tâches) et le logiciel de gestion de l'ouverture et de la fermeture des portes du bus. On distingue également les cinq unités d'entrées sorties (IOU) et l'unité de contrôle de la visualisation (SCU).

Pour les questions suivantes, on se référera essentiellement aux informations données dans le dossier de présentation et les annexes A.2, A.4.2 et B.3.

Q1. Identification des éléments du système concernés par la gestion d'une demande d'arrêt, et des interactions associées.

- Repérer le numéro de l'IOU et le type d'entrée utilisée pour la détection de l'appui sur le bouton poussoir Arrêt demandé C.**
- Préciser à quel élément du système (CMU, SCU, IOU), cette donnée sera envoyée. Préciser le type de communication utilisée (liaison directe, liaison série, réseau CAN).**
- Dans le cas d'une communication par réseau CAN, préciser la trame concernée (Norm1, Norm2, Norm3, Norm4).**
- Repérer le module qui commande le panneau « Arrêt Demandé avant », le panneau « Arrêt Demandé milieu », le voyant « Arrêt Demandé » du tableau de bord et la sonnerie.**

Q2. Représentation de la chronologie des échanges

Pour le cas d'utilisation décrit ci-dessus (gestion d'une demande d'arrêt), on souhaite inventorier les échanges entre les différents éléments du système en respectant leur chronologie. Ces interactions sont repérées par une flèche sur le document réponse N°1.

- Indiquer sur le document réponse DR 1, la signification du message ou les opérations principales associées à chaque flèche, en respectant la chronologie.**

Q3. Analyse du délai de réaction du système

L'ordonnancement des tâches réalisées par l'unité de supervision (CMU), présenté dans l'annexe A.4.2 peut se commenter comme suit :

- La tâche C1 correspond à l'émission des trames (Norm4) contenant les commandes pour les IOU du bloc 1 (IOU1, IOU2, IOU3)
- La tâche C20 correspond à l'émission des trames (Norm4) contenant les commandes pour les IOU 4, 5 et 6 d'une part et d'autre part à la lecture des données reçues sur le contrôleur CAN du CMU en provenance des IOU 1, 2 et 3 et leur stockage dans la zone de mémoire partagée
- MAj-Entrees correspond à la mise à jour des variables d'entrées de l'application (lecture de la mémoire partagée)
- MAj_Sorties correspond à la mise à jour de la mémoire partagée avec les sorties de l'application (écriture de la mémoire partagée).

L'application de gestion des portes est incluse dans le module « application3 ».

On ne tiendra pas compte dans cette question, du temps de transfert des informations sur le réseau CAN ou dans les structures électroniques. Le délai entre la réception d'une trame de requête et l'émission de la réponse par l'IOU est négligeable (très inférieure à 10 ms). Pour un IOU, le délai maximum entre le changement sur une entrée

de l'IOU et sa prise en compte dans la mise à jour dans la zone de donnée du contrôleur CAN est de 20 ms. De même le délai maximum entre la réception d'une trame de commande et le changement sur ses sorties est de 20 ms.

On supposera que toutes les tâches exécutées par l'unité de supervision (CMU) sont terminées dans le temps imparti.

- Quel est le délai maximal entre l'appui sur le bouton poussoir « Arrêt DemandeC » et la réception de cette information dans la mémoire partagée du CMU (RG_MON_CanData)?**
- Donner la valeur du délai total entre l'appui sur le bouton poussoir et l'allumage du panneau lumineux « arrêt demandé » situé à l'avant du bus.**
- Compléter le document réponse DR 2 en faisant apparaître la chronologie des opérations intermédiaires.**

A.2 ETUDE DE L'OUVERTURE DE LA PORTE AVANT

On s'intéresse maintenant au cas d'utilisation de l'ouverture de la porte.

Les moteurs qui entraînent les portes sont pilotés en tension (commande de type PWM). Il n'y a pas d'asservissement de courant.

Q4. Evolution de la position de la porte en fonction du temps

La position de la porte est mesurée par un angle d'ouverture qui peut aller de 0° (porte fermée) à 100° (position maximale). La porte est déclarée ouverte à partir d'une position qui dépend du type de porte.

La tension appliquée au moteur dépend de l'état de fonctionnement (ouverture ou fermeture), de la position de la porte, de la tension d'alimentation du module IOU, du mode de fonctionnement (normal ou urgence).

Lors de l'ouverture ou la fermeture d'une porte, on a une phase de démarrage, puis une phase de fonctionnement qui dure environ 4 s et enfin une phase de ralentissement.

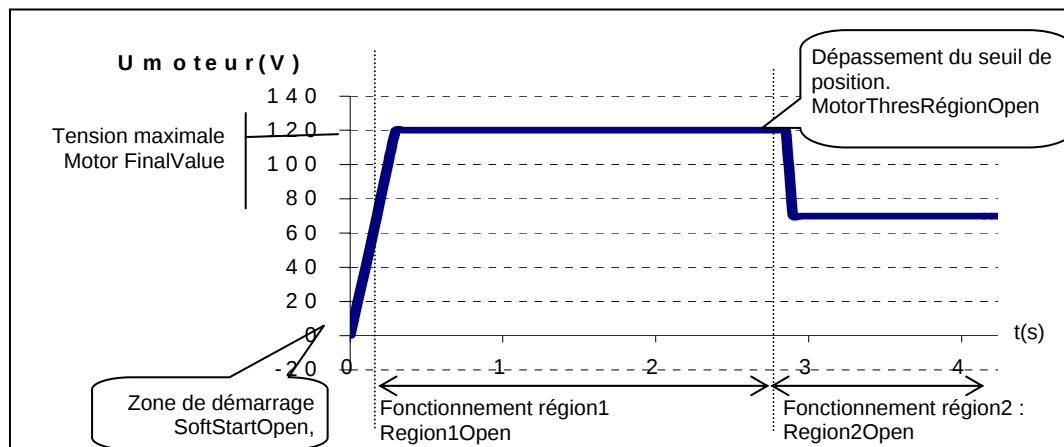


Figure 1 Évolution de la tension appliquée au moteur pendant l'ouverture

Pendant la phase de démarrage du moteur, la tension aux bornes du moteur est augmentée par pas, à chaque cycle de fonctionnement (toutes les 50 ms). Le passage de la « région1 » à la « région2 » apparaît lorsque la position de la porte dépasse un seuil d'ouverture. Ce seuil dépend du type de porte.

- Justifier la présence de la phase de démarrage du moteur.**
- À partir des données extraites du fichier d'initialisation init_Appli (cf. annexe A.4.1.3), montrer que la durée de cette phase est de 0,3 s si la tension de la batterie est de 23,5 V.**
- Sur le document réponse DR 3, tracer en fonction du temps l'allure de la vitesse d'ouverture de la porte AVANT et l'évolution de l'angle d'ouverture de la porte AVANT. Préciser l'échelle sur les angles d'ouverture.**

Q5. Étude du rapport cyclique de la tension PWM

La tension générée par l'IOU qui pilote le moteur est de type PWM. (Cf. schéma électronique folio 6 des IOU fournis en annexe).

La tension de la batterie dans le cas idéal est de 24 V (CT_THEORIC_VALUE_BATTERY=24V).

- Donner l'expression du rapport cyclique en fonction de la tension souhaitée aux bornes du moteur.**

- b) *Compte tenu des valeurs de rapport cyclique disponibles pour la génération des tensions PWM et en supposant que la tension de la batterie soit égale à 24 V, tracer sur le document réponse DR 3, l'évolution du rapport cyclique en fonction du temps. On repèrera bien les zones de fonctionnement : SoftStartOpen, Region1Open, Region2Open. Les axes seront correctement gradués à la fois sur l'axe des abscisses et sur l'axe des ordonnées.*
- c) *Dans l'application logicielle de gestion des portes, le rapport cyclique est toujours calculé en supposant que la tension de la batterie est égale à la valeur théorique de 24 V. Quelle sera dans ces conditions l'influence de la tension réelle de la batterie sur l'évolution de la position de la porte ?*

Q6. Étude du courant dans le moteur

- a) *Justifier la forme du courant dans le moteur fournie figure 3 dans le document de présentation.*

Le tableau *TableCourantPorteAv* (cf annexe A.4.1.3. *Init_Appli*) précise les valeurs de courant pour les différentes zones de fonctionnement à l'ouverture et à la fermeture, en fonction de la tension mesurée par la batterie.

D'autre part, le cahier des charges de l'application indique : « En cas de rencontre d'un obstacle correspondant à une force supérieure à 150 N sur une des portes, une inversion du fonctionnement doit intervenir ».

- b) *Expliquer comment l'utilisation du tableau *TableCourantPorteAv* permet de détecter un obstacle, à partir de la mesure du courant dans le moteur.*

A.3 INFLUENCE DE LA TENSION DE LA BATTERIE

Les fonctions *Porte_Cmde_TensionPourPorte* et *Porte_Cmde_CalculParametreParRapportVBat* représentent des tâches incluses dans l'application de gestion des portes (application3 pour le moniteur).

Q7. Étude de la fonction *Porte_Cmde_TensionPourPorte* (cf. annexe A.4.3.2)

- a) *Expliciter ce que représente la variable *RG_TensionPorteAv*. Quel est l'intérêt de cette fonction ?*
- b) *Proposer en la justifiant, une valeur pour *CT_PeriodeMoyenneTensionPorte* et donner sa déclaration.*

Q8. Étude de la fonction *Porte_Cmde_CalculParametreParRapportVBat* (cf. annexe A.4.3.3)

- a) *Expliciter le rôle de cette fonction et tracer la caractéristique donnant la valeur de la variable *RL_systCptPositionAv* en fonction de la tension de la batterie *Ubat*.*
- b) *Tracer la caractéristique donnant la valeur de *RL_systCptCourantAv* en fonction de la tension de la batterie *Ubat*.*

A.4 ESTIMATION DE LA POSITION DE LA PORTE AVANT

Q9. Estimation de la position de la porte

Une estimation de la position de la porte doit être faite par l'application de gestion des portes. Pour ce faire, la variable *Temporal_Pos_Counter_Porte* représente l'estimation de la position de la porte.

La table *TableCompteurPositionPorteAv* (cf. fichier *init_Appli* dans l'annexe A.4.1.3) permet de définir un pas d'incrément de la porte à chaque cycle de mesure. L'évolution de la position de la porte dépend de la tension de la batterie.

Six zones de fonctionnement sont repérées, trois pour l'ouverture et trois pour la fermeture. Les trois zones de fonctionnement pour l'ouverture sont la phase de démarrage (*SoftStartOpen*), la zone de la région1 (*Region1Open*), la zone de la région 2 (*Region2Open*). (cf. Figure 1 au paragraphe A.2 du questionnaire).

Chaque porte est décrite par une structure *struct T_Porte* dont un extrait est donné dans l'annexe A 4.1.1.

*Proposer l'algorithme de la fonction *Porte_Cmde_DoorEstimationPosition* qui estime la position de la porte pour le fonctionnement dans la région 1.*

A.5 ÉLABORATION DES DOCUMENTS DE PRÉSENTATION DU SYSTÈME

Il est envisagé d'exploiter ce système afin d'illustrer les concepts mis en jeu dans **les unités d'entrées sorties** d'une part, dans **la transmission des données** d'autre part. Dans cette partie, il est demandé au candidat de didactiser le système et de prévoir les documents de présentation associés. Le public sera choisi dans des sections d'électronique ou de STS IRIS. Ces documents pourraient être utilisés lors de différentes séquences d'enseignement.

Q10. Choisir la mise en situation générale, (c'est-à-dire restreindre et délimiter le système étudié pour illustrer un ensemble de concepts, définir le système dans son contexte)

Q11. Élaborer les principaux schémas de présentation associés (schémas fonctionnels, schéma d'architecture matérielle, diagrammes UML, algorithmes, etc.)

Q12. Expliciter les caractéristiques principales retenues (caractéristiques fonctionnelles, matérielles ou comportementales, contraintes du cahier des charges)

Attentes du jury :

Pour les documents de présentation réclamés, on attend une production optimale de 2 à 4 pages format A4.

Critères d'évaluation du candidat :

- L'exactitude technique des documents présentés et le respect du système proposé
- La pertinence pédagogique de la mise en situation choisie et des caractéristiques principales retenues
- La cohérence et la pertinence pédagogique des représentations utilisées au niveau choisi
- La qualité des descriptions et des explications proposées (exactitude technique, précision et concision, expression écrite, adéquation par rapport au contexte).

B ÉTUDE DE LA STRUCTURE D'UNE UNITÉ D'ENTRÉES-SORTIES

B.1 ÉTUDE DES ENTRÉES LOGIQUES

On se propose de valider les choix de composants de la chaîne de filtrage des perturbations électromagnétiques de l'acquisition des entrées logiques. Cette étude se rapporte au schéma structurel des IOU et à leur documentation (annexe B.1.2).

Q13. Validation de la structure vis-à-vis des perturbations

- a) Représenter cette chaîne d'acquisition, en se limitant aux éléments placés entre le connecteur d'entrée CN104 et le port du microcontrôleur (CPU_IN_x).**
- b) Caractériser les deux types de perturbation intervenant dans la définition des immunités électriques (annexe B.1.2 EMC/ECM immunity).**
- c) Calculer la tension à l'entrée du microcontrôleur pour chacun des signaux d'impulsion normalisés pulse2, pulse3a et pulse3b (norme ISO7637-3) appliqué sur le connecteur CN104. Le candidat justifiera les approximations éventuellement utilisées.**
- d) Commenter les résultats obtenus du point de vue de l'immunité aux perturbations.**
- e) Préciser quel post-traitement, interne au microcontrôleur de l'IOU, permet finalement d'obtenir une immunité vis à vis des impulsions normalisées.**

B.2 ÉTUDE DES SORTIES DES IOU

On considère que l'unité d'entrées sorties (IOU) vient de recevoir une trame correspondant à l'allumage du panneau « Arrêt demandé » situé près de la porte du milieu (sortie 5 de l'IOU6). On se propose dans un premier temps d'étudier cette sortie. Dans toute cette étude, on considérera que le signal SMART_PROTECT2 est relié à la masse (fonctionnement « normal »).

Q14. Étude du dispositif de détection des défauts sur la sortie OUT5

- a) Identifier la structure et tracer un schéma simplifié du montage du conditionnement de la sortie OUT5 d'après le folio 5 (chaîne de détection de défaut de la sortie OUT5).**
- b) Expliquer le fonctionnement de la détection d'un circuit ouvert et d'un circuit fermé. Donner pour chacun de ces défauts la valeur du seuil de détection pour une charge résistive (respectivement R_{chmax} et R_{chmin}).**

Q15. Étude du pont en H

On se propose maintenant d'étudier le pont en H destiné à l'alimentation du moteur de gestion de la porte du milieu. Ce pont, désigné par Pont3, est composé des sorties Out13 et Out14.

La commande `RG_IOU3_BR3D1_COMMAND` est utilisée pour commander le demi-pont formé par les interrupteurs U608 et Q603. La fréquence de hachage est fixée en interne dans l'IOU à 50 kHz.

- a) *Expliquer pourquoi les deux interrupteurs du demi-pont ne sont pas réalisés avec des composants de même technologie.*

Lors de la fermeture de la porte, les interrupteurs commandés sont le U608 et le Q603.

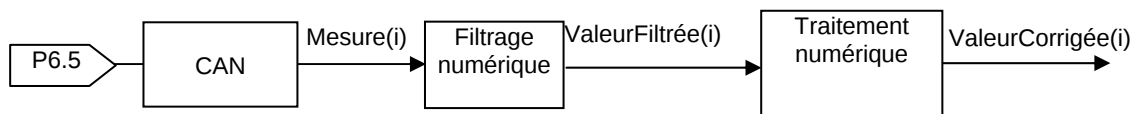
- b) *Représenter la structure du pont en précisant pour chacun des états de Q601 (fermé/ouvert) le cheminement du courant moteur.*

Q16. Étude de la mesure du courant

- a) *Tracer l'allure du chronogramme du courant dans la résistance R636.*
 b) *Déterminer la fonction de transfert liant la tension désignée par CPU_SHUNT1 au courant dans la résistance R636. En déduire le rôle de ce dispositif.*
 c) *Donner la caractéristique statique de la conversion analogique numérique $N = f(I)$. Préciser la valeur de la résolution en courant et valider la plage de mesure du courant.*

B.3 ÉTUDE PARTIELLE DE LA CHAÎNE DE MESURES ANALOGIQUES

Une partie de la chaîne de mesure analogique peut se représenter par le schéma suivant :



L'acquisition par le microcontrôleur des entrées est réalisée avec une période 10 ms.

La grandeur ainsi échantillonnée Mesure(i) traverse un filtre numérique dont la relation est :

$$\text{ValeurFiltrée}(i) = ((K-1) \text{ValeurFiltrée}(i-1) + \text{Mesure}(i)) / K$$

Par ailleurs, un traitement numérique est réalisé dans le but de corriger des défauts de la chaîne analogique et de permettre une mise à l'échelle des mesures.

Q17. Étude du filtre numérique

On supposera qu'initialement la valeur filtrée est égale à la mesure ($\text{ValeurFiltrée}(0) = \text{Mesure}(0)$). Le coefficient K est fixé à 4 dans cette partie.

- a) *Le signal en entrée du convertisseur étant un échelon défini sur le document réponse DR 4, calculer et représenter sur celui-ci les premiers échantillons de la sortie du filtre numérique.*
 b) *Déterminer la transformée en Z du filtre. Montrer qu'il s'agit d'un filtre passe-bas du premier ordre et donner son gain statique.*
 c) *Proposer un filtre analogique équivalent.*
 d) *En le justifiant, proposer un choix concernant le type de données à utiliser pour la programmation du filtre numérique.*

Q18. Étude du traitement numérique

Le premier étage du traitement numérique est chargé de la correction des défauts. Il ne sera pas traité dans cette étude, on considèrera donc que la ValeurCorrigée = ValeurFiltrée. On s'intéresse donc maintenant au troisième étage, chargé de la mise à l'échelle des mesures.

Ce module de traitement numérique applique la transformation suivante à la valeur corrigée :

$$\text{Valeur convertie} = ((C1/C2) * \text{Valeur corrigée}) + D$$

Elle fait apparaître trois coefficients (C1, C2 et D) codés sur 8 bits signés.

- a) *Calculer les coefficients C1, C2 et D programmés dans l'EEPROM de l'IOU afin que la mesure de la tension de batterie V_{ams} fournisse une valeur convertie avec une résolution de 200 mV/bit.*

B.4 EXPLOITATION PÉDAGOGIQUE CONCERNANT LES UNITÉS D'ENTRÉES SORTIES

Le public visé est une classe de terminale STI option électronique.

Les compétences visées feront principalement partie des compétences D et E (cf. annexe D).

Les savoirs associés feront principalement partie des domaines suivants :

- Traitement des signaux analogiques
- Transmission de l'information : conversion de données

B.4.1 PRÉSENTATION DE LA CHAÎNE D'ACQUISITION ANALOGIQUE

Q19. Rédiger le document permettant de présenter à des élèves de terminale STI la chaîne d'acquisition des signaux analogiques. Ce document destiné à être utilisé lors de différentes séquences devra comporter les schémas, descriptions et explications nécessaires (par exemple : décomposition d'une fonction, schématisation d'une architecture matérielle, caractérisation des entrées sorties, relations entrées sorties, algorithme, modélisation d'une fonctionnalité, d'une structure ou d'un comportement).

Conseil aux candidats

Le candidat proposera une étude qui s'articulera autour des 5 blocs suivants :

- Adaptation/filtrage des entrées
- Multiplexage
- Amplification
- Conversion A/N
- Traitement numérique

Cette étude devra mettre en évidence les caractéristiques des signaux traités et détaillera le rôle des fonctions de la chaîne, la nature des entrées/sorties, les relations liant les entrées sorties, l'influence des paramètres stockés en mémoire EEPROM, la présentation du dispositif numérique de correction de la chaîne analogique et du dispositif de mise à l'échelle.

Les fonctions intervenant dans la communication CAN avec le CMU ne devront pas faire partie de cette étude. La chaîne de mesure sera délimitée entre le connecteur CN104 et la structure de données partagées dans la RAM de l'unité d'entrées sorties (IOU) noté *RG_MON_IOU* qu'il faudra définir.

Consigne :

Pour mener à bien cette étude et réaliser les documents demandés, vous serez amené à compléter l'étude technique des unités d'entrées sorties. La démarche de cette étude complémentaire et ses résultats principaux (schémas, calculs, explications) devront être inclus dans votre copie. Toutefois, il est impératif de les dissocier de manière explicite des documents élèves (par exemple en les étiquetant « documents professeur »).

Critères d'évaluation du candidat : voir question Q12

B.4.2 DÉFINITION DE LA SÉQUENCE

Il s'agit de proposer en une page maximum, la description d'une séquence d'enseignement pour laquelle le système étudié permet une mise en situation pertinente. La séquence et surtout la phase qui sera détaillée par le candidat, devront permettre de viser un ou plusieurs des objectifs suivants :

- Être capable de définir les fonctions intervenant dans une chaîne d'acquisition numérique et de caractériser les entrées-sorties
- Valider que la fonction secondaire de FP1a concernant l'acquisition de données analogiques par un IOU est assurée par la structure utilisée
- Être capable d'évaluer que la fonction secondaire FP1a est assurée par la structure utilisée et de justifier le choix des paramètres principaux de la chaîne d'acquisition par rapport au cahier des charges
- Être capable de déterminer les valeurs stockées en EEPROM, les paramètres principaux de la chaîne d'acquisition étant donnés.

Q20. Donner les caractéristiques principales de la séquence, c'est-à-dire :

- a) Situer la séquence dans le cycle de formation et sa durée et définir la compétence principale visée**
- b) Préciser l'objectif de la séquence, les savoirs nouveaux et les pré-requis.**

Q21. Donner l'organisation de la séquence, c'est-à-dire :

- a) Proposer un déroulement en définissant les étapes permettant d'atteindre l'objectif**
- b) Donner l'objectif de chaque étape (phases d'enseignement) et détailler les savoirs nouveaux associés**
- c) Préciser les modalités des interventions du professeur et les activités des élèves**
- d) Indiquer les éléments sur lesquels portera l'évaluation.**

B.4.3 ÉLABORATION DÉTAILLÉE D'UNE PARTIE DE LA SÉQUENCE

Q22. Définition de la partie de séquence choisie

- a) *Repérer sa place au sein de la séquence et donner ses caractéristiques principales (organisation, objectif)*
- b) *Définir le concept principal visé et/ou les centres d'intérêts*

Q23. Élaborer un document qui guide l'élève dans son travail

Ce document de 3 à 4 pages devra respecter les contraintes suivantes :

- Il s'agit d'un questionnaire destiné aux élèves pour des travaux dirigés associant ou non des parties expérimentales
- Les réponses attendues devront être fournies.
- Les parties de cours ou les interventions de l'enseignant ne seront pas développées, elles seront simplement citées. Les documents ressources et les schémas nécessaires pour les explications sont ceux élaborés dans les questions précédentes.

Critères d'évaluation du candidat : voir question Q12 et adéquation des productions aux objectifs visés.

C ÉTUDE DE LA COMMUNICATION ENTRE LES UNITÉS D'ENTRÉES SORTIES (IOU) ET L'UNITÉ DE SUPERVISION (CMU)

Le réseau CAN a une topologie de type bus avec des résistances de terminaison.

C.1 ÉTUDE DE LA LIAISON CAN

Q24. Étude de la liaison physique

Le schéma associé au contrôleur CAN est présenté sur le schéma structurel folio 7.

- a) *Compte tenu de la valeur des composants passifs associés au contrôleur CAN 82C251, indiquer le mode de fonctionnement choisi. Quel est l'intérêt de ce choix ?*
- b) *Donner le schéma électrique équivalent du réseau CAN, en supposant qu'un des IOU est émetteur. Déterminer le nombre maximal de nœuds que l'on pourrait connecter au réseau en négligeant la résistance du câble et l'influence du transformateur.*

Q25. Transmission des trames de données échangées entre le CMU et les IOU

Le nombre maximal d'unité d'entrées sorties (IOU) avec lequel l'unité de supervision (CMU) peut échanger est fixé à 12.

- a) *Déterminer la durée minimale nécessaire pour la transmission de l'ensemble des trames de données échangées entre les IOU et le CMU*
- b) *Donner les valeurs des identifiants des trames de données (Norm1 à Norm4) pour l'IOU1*
- c) *Expliciter le mécanisme d'arbitrage utilisé sur un bus CAN. Définir un ordre de priorité entre les trames Norm1, Norm2, Norm3 et les trames Norm4 suivant les unités IOU*
- d) *Montrer que le choix du débit et l'organisation de la lecture des trames CAN reçues permet de respecter les contraintes temporelles de l'application.*

Q26. Lecture et mise à jour de la table RG_MON_CanData

On considère les extraits donnés ci-dessous :

Extrait de RGI0U.h

```
#define RG_I0U6_ELOG4
```

```
(T_BOOLEEN)MC_TestBit(RG_MON_CanData[CT_Trm1_I0U6][4],CT_Bit4)
```

Extrait de Maj_Entree.c

```
RG_E_Bp_ArretDemandeC = RG_I0U6_ELOG4
```

- a) **Expliciter le rôle de ces deux lignes de code, en vous appuyant sur les extraits du fichier de définition des constantes et des macros const.h donnée en annexe A.4.1.4**

On trouve dans le fichier Maj_Sortie.c les lignes suivantes (cf. annexe A.4.2.3)

```
RG_I0U7_OUT5_COMMAND (RG_S_EclairageArretDemandePorteAv)
```

```
RG_I0U7_BR2_PARAMETER (RG_S_Pwm_Freq_CmdeMoteurPorteAv)
```

```
RG_I0U7_BR1_STOP
```

```
RG_I0U7_BR2D1_COMMAND
```

- b) **Expliciter l'action de ces lignes. Préciser comment ces commandes seront transférées et exploitées par l'unité d'entrées sorties (I0U).**

C.2 EXPLOITATION PÉDAGOGIQUE CONCERNANT LA COMMUNICATION

C.2.1 CONTEXTE DE L'EXPLOITATION PÉDAGOGIQUE

Le candidat a le choix entre deux options :

- Soit une exploitation dans une section de TS SE : « systèmes électroniques »
- Soit une exploitation dans une section TS IRIS : « informatique et réseaux pour l'industrie et les services techniques ».

OPTION 1 : EXPLOITATION EN STS SE

Les compétences visées feront principalement partie de :

- A1 Expliciter un schéma fonctionnel
- M1 Installer ou configurer un nouvel équipement ou produit
- M2 Valider le bon fonctionnement de l'équipement ou du produit.

Les savoirs associés feront principalement partie des domaines suivants :

- Transmission et transport de l'information
- Connexions entre constituants électroniques ou connexions inter-systèmes.

La séquence et surtout la phase qui sera détaillée par le candidat, devront permettre de viser un ou plusieurs des objectifs suivants :

- Associer les caractéristiques d'un équipement aux spécifications de la couche physique d'un réseau
- Valider le choix d'une solution technologique pour la couche physique et le média de transmission d'un réseau par rapport au cahier des charges de l'application.

OPTION 2 : EXPLOITATION EN STS IRIS

Les principales compétences visées peuvent faire partie des extraits de référentiels fournis en annexe D :

- C3.8 Caractériser les contraintes principales d'un système de transmission de l'information
- C3.9 Décrire les tâches d'une application de communication
- C3.10 Valider la configuration d'un réseau local industriel ou d'un bus de terrain

Les savoirs associés peuvent être faire partie des domaines suivants :

- Notions fondamentales sur les réseaux et mode de transmission
- Principe de la transmission de niveau 1
- Règles d'échange de données. Modèles des échanges.

La séquence et surtout la phase qui seront explicités par le candidat, devront permettre de cibler un ou plusieurs des concepts suivants ;

- Caractérisation des méthodes d'accès au média et des mécanismes d'arbitrage
- Modélisation des échanges dans un mécanisme de communication.

C.2.2 DOCUMENTS DE PRÉSENTATION DE LA COMMUNICATION PAR RÉSEAU CAN

Q27. Choisir la mise en situation, (c'est-à-dire restreindre et délimiter le système étudié, définir le contexte ou la « situation problème » retenue)

Q28. Expliciter les caractéristiques principales retenues (caractéristiques fonctionnelles, matérielles ou comportementales, contraintes du cahier des charges)

Q29. Rédiger en 2 ou 3 pages les documents détaillés permettant de présenter à des élèves la communication par réseau CAN dans le système étudié

Critères d'évaluation du candidat : voir question Q12 et adéquation des productions aux objectifs visés.

Il est impératif de dissocier de manière explicite, les études complémentaires (« documents professeur ») du « document élève » demandé. Il n'est pas demandé d'élaborer des documents de cours mais un document ressource présentant la communication par réseau CAN au sein du système étudié.

C.2.3 DÉFINITION D'UNE SÉQUENCE

Il s'agit de proposer en une page maximum, la description d'une séquence d'enseignement pour laquelle le système étudié permet une mise en situation pertinente.

Q30. Donner les caractéristiques principales de la séquence, c'est-à-dire :

- a) *Situer la séquence dans le cycle de formation et sa durée. Définir la compétence principale visée*
- b) *Préciser l'objectif de la séquence, les savoirs nouveaux et les pré-requis.*

Q31. Donner l'organisation de la séquence, c'est-à-dire :

- a) *Proposer un déroulement en définissant les étapes permettant d'atteindre l'objectif*
- b) *Donner l'objectif de chaque étape (phases d'enseignement) et détailler les savoirs nouveaux associés*
- c) *Préciser les modalités des interventions du professeur et les activités des élèves*
- d) *Indiquer les éléments sur lesquels portera l'évaluation.*

C.2.4 ÉLABORATION DÉTAILLÉE D'UNE PARTIE DE LA SÉQUENCE

Q32. Définition de la partie de séquence choisie

- a) *Repérer sa place au sein de la séquence et donner ses caractéristiques principales (organisation, objectif)*
- b) *Définir le concept principal visé et/ou les centres d'intérêts*

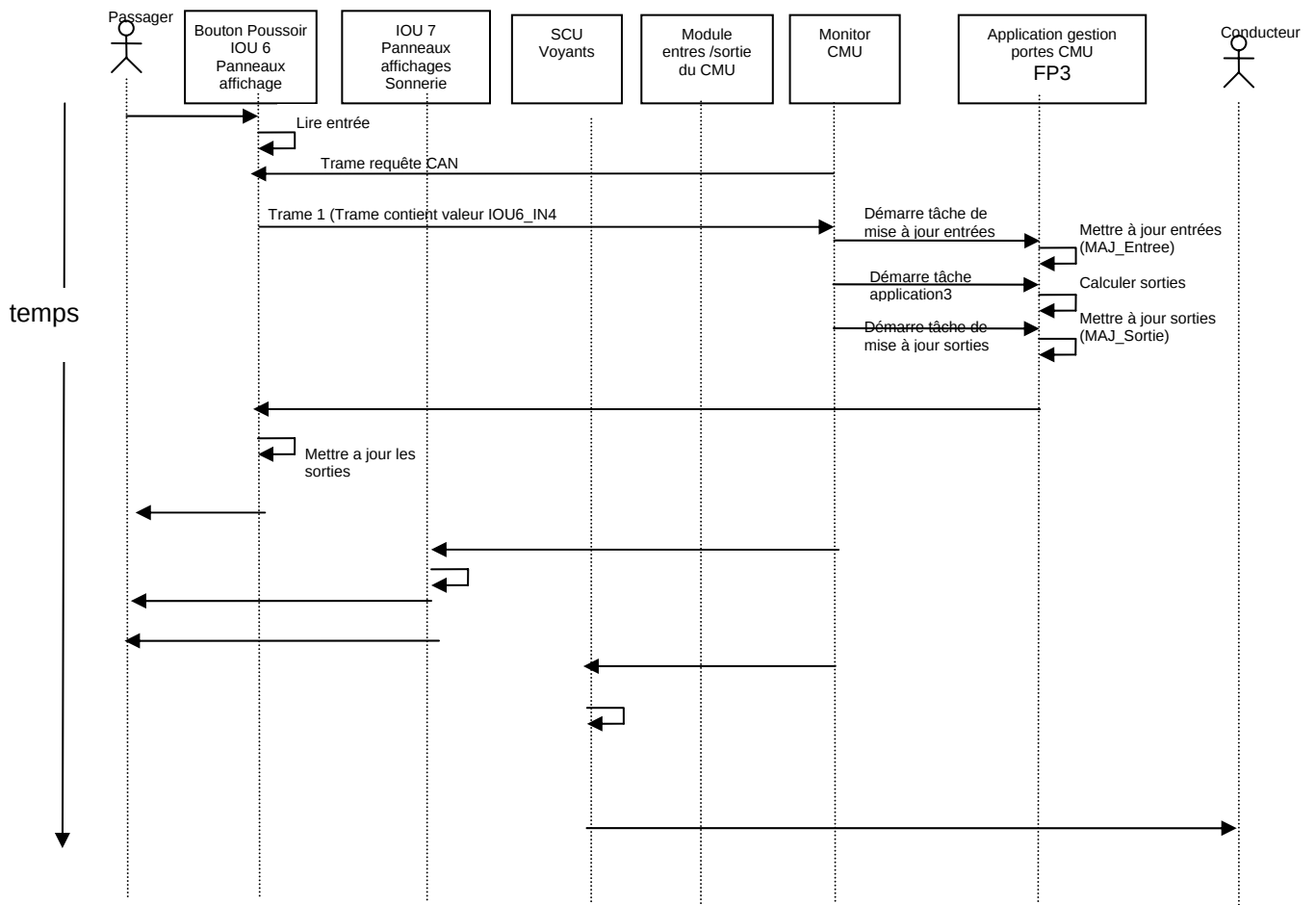
Q33. Élaborer un document qui guide l'élève dans son travail

Ce document de 3 à 4 pages devra respecter les contraintes suivantes :

- Il s'agit d'un questionnaire destiné aux élèves pour des travaux dirigés associant ou non des parties expérimentales
- Les réponses attendues devront être fournies.
- Les parties de cours ou les interventions de l'enseignant ne seront pas développées, elles seront simplement citées. Les documents ressources et les schémas nécessaires pour les explications sont ceux élaborés dans les questions précédentes.

Critères d'évaluation du candidat : voir question Q12 et adéquation des productions aux objectifs visés.

Document réponse DR 1

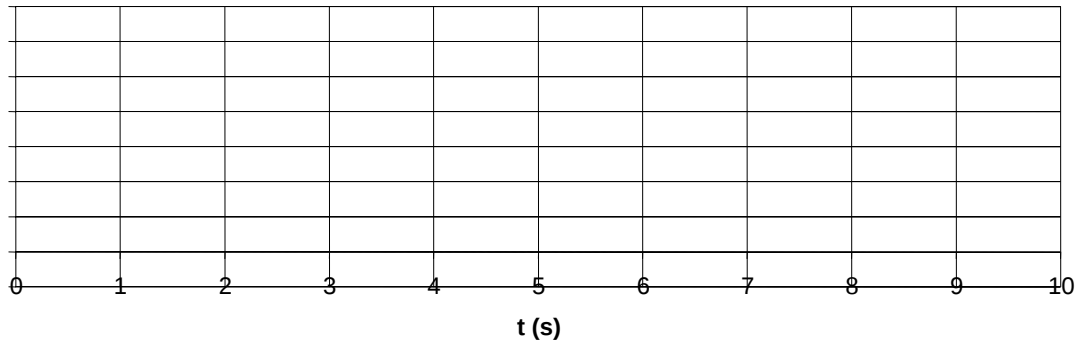


Document réponse DR 2

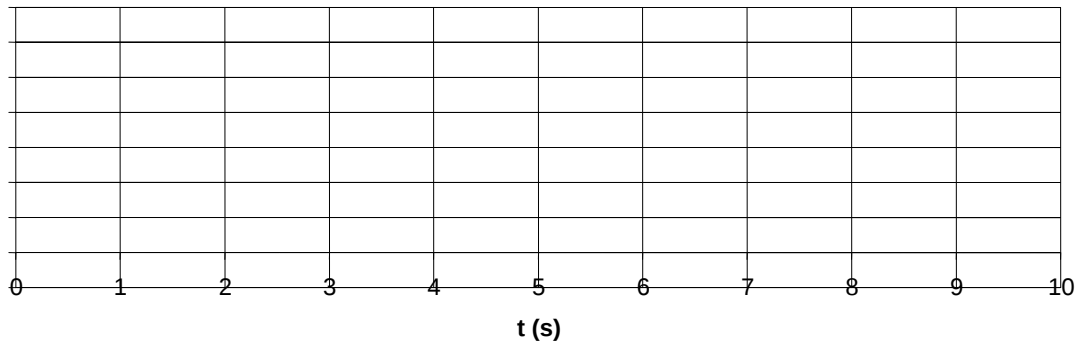


Document réponse DR 3

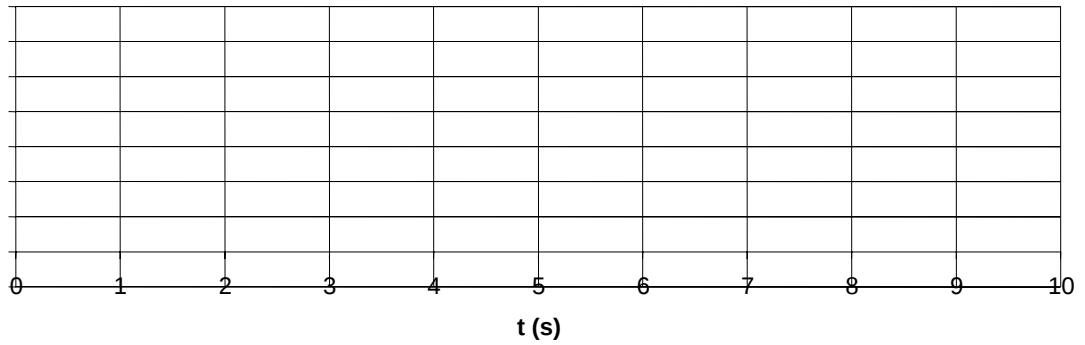
Évolution de la vitesse d'ouverture de la porte



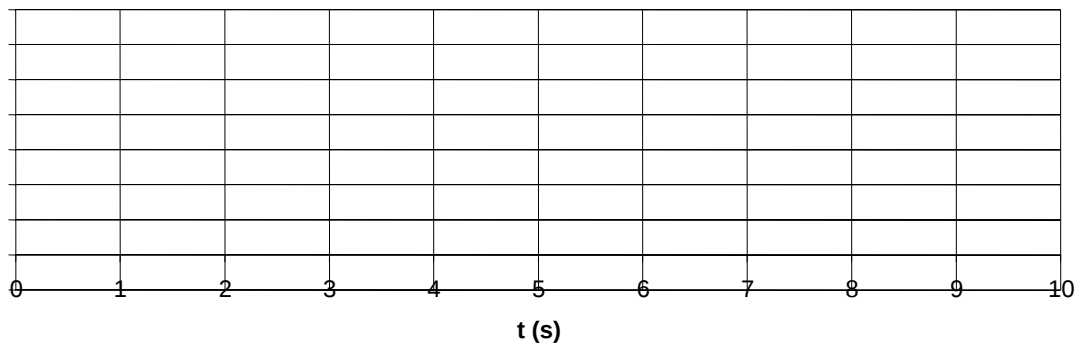
Évolution de l'angle d'ouverture de la porte



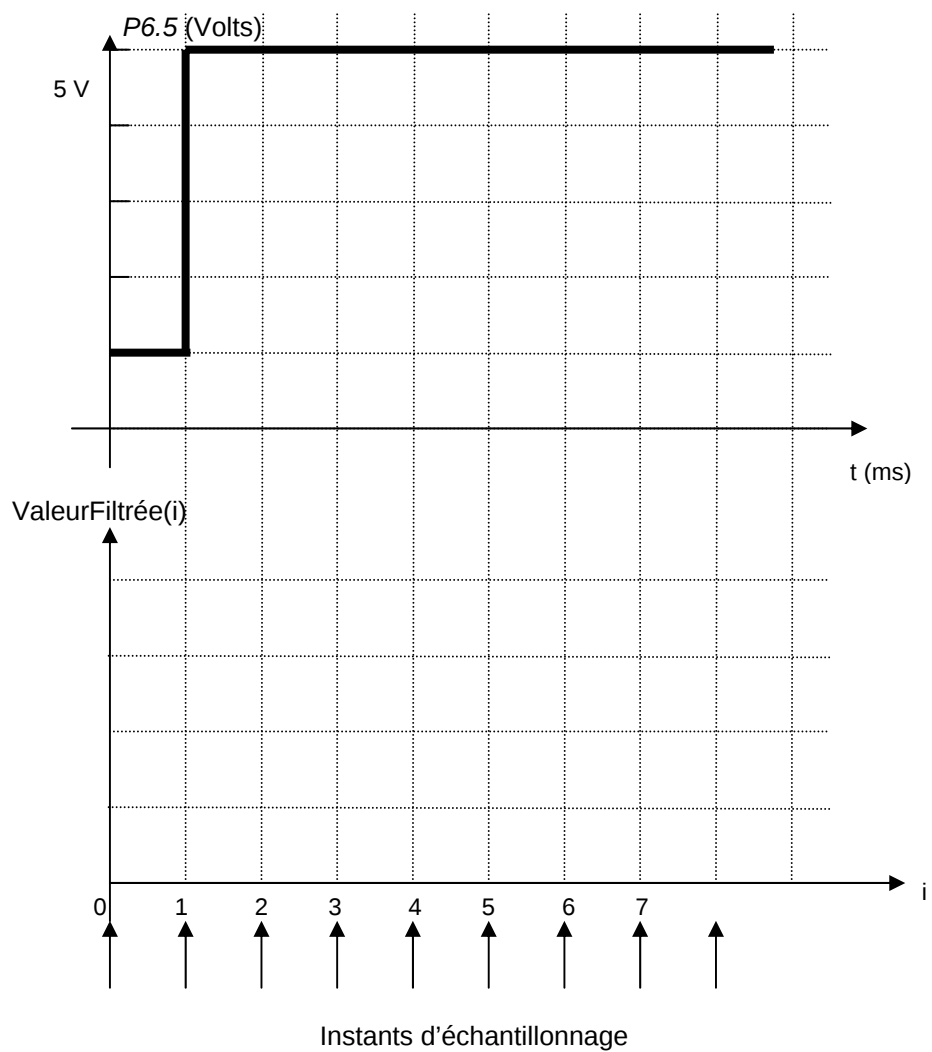
Évolution du rapport cyclique de la tension aux bornes du moteur



Évolution du courant dans le moteur



Document réponse DR 4



SOMMAIRE DES DOCUMENTS ANNEXES

A	ANNEXE A : STRUCTURES MATÉRIELLES ET LOGICIELLES.....	36
A.1	GLOSSAIRE	36
A.2	DETAILS DES ENTRÉES SORTIES DE LA GESTION DE PORTES	36
A.3	SCHÉMAS STRUCTURELS DES UNITÉS D'ENTRÉES SORTIES (IOU)	38
A.4	EXTRAITS DE STRUCTURES LOGICIELLES.....	38
A.4.1	Extraits de fichiers de configuration.....	38
1)	Extrait de la structure struct T_Porte.....	38
2)	Extrait du Fichier : RG_MON.H.....	39
3)	Extrait du Fichier INIT_APPLI	40
4)	Extrait du fichier Const.h	42
5)	Extraits du fichier RGIUO.h	43
A.4.2	Tâches applicatives en Mode Normal.....	45
1)	Descriptif	45
2)	Extrait de Maj_Entree.c	46
3)	Extrait de Maj_Sortie.c	46
A.4.3	Application gestion des portes (Porte_Exe_FonctionsPorte)	46
1)	Extrait de Porte_Cmde_Portes	47
2)	Extrait de Porte_Cmde_TensionPourPorte	48
3)	Extrait de Porte_Cmde_CalculParametreParRapportVBat	48
A.4.4	Pilote de porte	49
	Vue générale du pilote	49
B	ANNEXE B : EXTRAITS DE NOTICES TECHNIQUES DU SYSTÈME	50
B.1	DOCUMENTATION DES MATÉRIELS.....	50
B.1.1	Moteur	50
B.1.2	Unités d'entrées sorties (IOU).....	50
	Caractéristiques matérielles d'une IOU	50
B.1.3	Unité de supervision (CMU).....	53
B.2	COMMUNICATION ENTRE LE CMU ET LES AUTRES ÉQUIPEMENTS	53
B.2.1	Gestion du réseau.....	53
B.2.2	Structure du câblage	54
B.2.3	Phase de réveil	54
B.2.4	Phase d'adressage	54
B.3	SPÉCIFICITÉS DU RÉSEAU CAN (MULTIBUS ACTIA)	54
B.3.1	Trames échangées entre les IOU et le CMU en fonctionnement normal	55
B.3.2	Définition des identifiants des trames pour le réseau CAN1 entre le CMU et les IOU.....	57
C	ANNEXE C : DOCUMENTATION DE COMPOSANTS.....	60
C.1	COMPOSANTS ÉLECTRONIQUES	60
C.1.1	Microcontrôleur MB90F540 /545.....	60
C.1.2	Documentation constructeur VNB20N07	63
C.1.3	Documentation constructeur BTS426L1	65
C.1.4	Documentation Multiplexeur 74 HC 4051	67
C.1.5	Documentation Contrôleur CAN	68

C.2	GÉNÉRALITÉS SUR LE RÉSEAU DE TERRAIN CAN	82
	Controller Area Network – How CAN Works	84
D	ANNEXE D: EXTRAITS DE RÉFÉRENTIELS ET PROGRAMMES.....	86
D.1	EXTRAITS DU PROGRAMME DU BACCALAURÉAT GÉNIE ÉLECTRONIQUE :	86
	D.1.1 Extraits des Compétences (compétences D et E)	86
	D.1.2 Extraits des Savoirs	86
D.2	EXTRAITS DU RÉFÉRENTIEL DE BTS IRIS	88
	D.2.1 Extraits des Compétences	88
	D.2.2 Extraits de savoirs.....	90
D.3	EXTRAITS DU RÉFÉRENTIEL DE BTS SE.....	93
	D.3.1 Extraits des champs des savoirs associés aux compétences	97
	D.3.2 Relations savoirs compétences	99

A ANNEXE A : STRUCTURES MATÉRIELLES ET LOGICIELLES

A.1 GLOSSAIRE

CAN : Control Area Network

Ligne d'adressage : Ligne reliant 2 calculateurs et servant à la propagation du réveil et à la phase d'adressage.

Entrée réveil : Entrée logique dédiée au réveil. Son activation provoque le réveil du boîtier.

Esclave : Calculateur n'ayant pas de logiciel application client et répondant exclusivement aux ordres du calculateur maître reçus par le CAN.

Maître : Calculateur contenant du logiciel application client.

ELOGx : Logical Input number x.

EREVx: Wake-Up input number x

EANAx : Analog Input number x

EFREQx : Frequency Input number x

OUTx : Output number x

BRx : Bridge number x (Bridge 1 = Out9 and 10, Bridge 2 = Out11 and 12, Bridge3 = Out13 and 14)

BRxDy : Bridge x Direction y (to change direction of the bridge)

ITEMP : Internal Temperature

VAMS : Voltage After Master Switch +ACB : Après Coupe Batterie

SSUPPLY : Sensor SUPPLY

IOU : Input Output Unit

CMU : Central Management Unit

SCU : Screen Control Unit

SLU : Switch & Light Unit

ECU : Electronic Control Unit

CNX : Connector number X.

HS : High-Side

LS : Low-Side

PWM : Pulse Width Modulation.

LAD : Logical Address (for an IOU)

ONUM: Order NUMber (for an IOU)

Variables

Les variables utilisées dans la spécification sont de trois types: Les entrées, les sorties et les variables internes

Entrées

- E entrée simple
- EDef entrée de défaut
- CMU Message CMU (unité de contrôle et de management)

Sorties

- S sortie
- SVyt sortie Voyant
- Saff sortie affichage pictogramme (pour les unités du tableau de bord)
- SPWM sortie pour les ponts

Variables internes

- Sys variable intermédiaire globale
- P paramètres
- Cmd variable intermédiaire
- Def défaut
- Etat Etat du système
- Temp temporisation
- CT constante initialisée à la déclaration dans le dictionnaire
- Int variable intermédiaire locale

A.2 DETAILS DES ENTRÉES SORTIES DE LA GESTION DE PORTES

Entrées	Module
RG_E_Bp_DeblocageSecuritesDePortes = RG_MON_CMU.ELog5;	CMU
RG_E_Bp_PorteMi1 = RG_MON_CMU.ELog6	
RG_E_Bp_PorteAv = RG_MON_CMU.ELog7;	
RG_E_Bp_PorteAr = RG_MON_CMU.ELog9;	
RG_E_Bp_DeblocagePortes = RG_MON_CMU.EAna1L;	

RG_E_Bp_ArretDemandeA	= RG_I0U1_ELOG6;	IOU 1
RG_E_Bp_ArretDemandeHandicape	= RG_I0U2_EANA1_LOGICAL;	IOU 2
RG_E_Bp_ArretDemandeF	= RG_I0U2_EANA3_LOGICAL;	
RG_E_Bp_ArretDemandeE	= RG_I0U3_ELOG1;	IOU 3
RG_E_CloseDoorSwitchAr	= RG_I0U3_ELOG0;	
RG_E_BP_PorteArParPassager	= RG_I0U3_ELOG2;	
RG_E_A_MesureCourantMoteurPorteAr	= RG_I0U3_BR2CURRENT_MEASURE;	
RG_E_A_MesTensionBatterieIOU12	= RG_I0U3_VAMS_MEASURE;	
RG_Def_S_AlimMoteurPorteAr	= RG_I0U3_OUT13_DEFAULT;	
RG_Def_S_FreinPorteAr	= RG_I0U3_OUT14_DEFAULT;	
RG_Def_S_CmdeDirectMoteurPorteAr	= RG_I0U3_BR2_DEFAULT;	
RG_E_BP_PorteMi1ParPassager	= RG_I0U6_ELOG2;	IOU 6
RG_E_Bp_ArretDemandeC	= RG_I0U6_ELOG4;	
RG_E_CloseDoorSwitchMi1	= RG_I0U6_ELOG6;	
RG_E_A_MesureCourantMoteurPorteMi1	= RG_I0U6_BR2CURRENT_MEASURE;	
RG_E_A_MesTensionBatterieIOU8	= RG_I0U6_VAMS_MEASURE;	
RG_Def_S_FreinPorteMi1	= RG_I0U6_OUT6_DEFAULT;	
RG_Def_S_AlimMoteurPorteMi1	= RG_I0U6_OUT13_DEFAULT;	
RG_Def_S_CmdeDirectMoteurPorteMi1	= RG_I0U6_BR2_DEFAULT;	
RG_E_BP_PorteAvParPassager	= RG_I0U7_ELOG2;	IOU7
RG_E_Bp_ArretDemandeB	= RG_I0U7_ELOG4;	
RG_E_CloseDoorSwitchAv	= RG_I0U7_ELOG6;	
RG_E_A_MesureCourantMoteurPorteAv	= RG_I0U7_BR2CURRENT_MEASURE;	
RG_E_A_MesTensionBatterieIOU7	= RG_I0U7_VAMS_MEASURE;	
RG_Def_S_FreinPorteAv	= RG_I0U7_OUT6_DEFAULT;	
RG_Def_S_AlimMoteurPorteAv	= RG_I0U7_OUT13_DEFAULT;	
RG_Def_S_CmdeDirectMoteurPorteAv	= RG_I0U7_BR2_DEFAULT;	
RG_E_Bp_CmdeOuverturePorteAvExt	= RG_I0U7_EREV0;	
RG_E_Bp_AutoriOuvPorteAvDeLExterieur	= MC_TestBit(RG_MON_SCUBuffer[CT_FromSCU_2][6],CT_Bit1);	SCU CAN2

Sorties		Module
RG_MON_CMU.Out8	= RG_S_Vyt_PorteAvOuverte	CMU
RG_MON_CMU.Out9	= RG_S_Vyt_PorteArOuverte;	
RG_MON_CMU.Out10	= RG_S_Vyt_PorteMi10ouverte	
RG_I0U3_OUT0_COMMAND(RG_S_Vyt_HabilitBpOuvPorteArParPassager); RG_I0U3_OUT5_COMMAND(RG_S_EclairagePorteArriere); RG_I0U3_OUT13_COMMAND(RG_S_FreinPorteAr); RG_I0U3_BR2_PARAMETER(RG_S_Pwm_Freq_CmdeMoteurPorteAr) RG_I0U3_BR2D1_COMMAND RG_I0U3_BR2D2_COMMAND		IOU 3
RG_I0U6_OUT1_COMMAND(RG_S_EclairagePorteMilieu1); RG_I0U6_OUT4_COMMAND(RG_S_FreinPorteMi1); RG_I0U6_OUT5_COMMAND(RG_S_EclairageArretDemandePorteMi); RG_I0U6_OUT10_COMMAND(RG_S_Vyt_HabilitBpOuvPorteMi1ParPassager)		IOU 6
RG_I0U6_BR2_PARAMETER (RG_S_Pwm_Freq_CmdeMoteurPorteMi1) ;		
RG_I0U7_OUT1_COMMAND(RG_S_EclairagePorteAvant); RG_I0U7_OUT2_COMMAND(RG_S_EclairageChauffeur); RG_I0U7_OUT3_COMMAND(RG_S_SonnerieArretDemande); RG_I0U7_OUT5_COMMAND(RG_S_EclairageArretDemandePorteAv) RG_I0U7_OUT7_COMMAND(RG_S_Vyt_HabilitBpOuvPorteAvParPassager); RG_I0U7_OUT10_COMMAND(RG_S_FreinPorteAv);		IOU7
MC_LoadBit(RG_MON_SCUBuffer[CT_ToSCU_1][6],CT_Bit1, RG_S_Vyt_AutoriOuvPorteAvDeLExterieur)		SCU CAN2
MC_LoadBit(RG_MON_SCUBuffer[CT_ToSCU_3][2],CT_Bit7, RG_S_Signal0TEPorteArOuv);		
MC_LoadBit(RG_MON_SCUBuffer[CT_ToSCU_3][6],CT_Bit3, RG_S_Vyt_ArretDemandeTabDeBordPass);		
MC_LoadBit(RG_MON_SCUBuffer[CT_ToSCU_3][6],CT_Bit5, RG_S_Vyt_VehiculeArrete);		
MC_LoadBit(RG_MON_SCUBuffer[CT_ToSCU_3][6], CT_Bit6,RG_S_Vyt_ToutesPortesFermees);		
MC_LoadBit(RG_MON_SCUBuffer[CT_ToSCU_9][1],CT_Bit7, RG_S_Aff_L_VytInhibitSecuritesPortes);		

A.3 SCHÉMAS STRUCTURELS DES UNITÉS D'ENTRÉES SORTIES (IOU)

Sept schémas structurels sont fournis au format A3

- Schéma structurel 1 General electric diagram: Vue d'ensemble de l'unité d'entrée sortie
- Schéma structurel 2 Power Supply
- Schéma structurel 3 Inputs
- Schéma structurel 4 CPU CORE
- Schéma structurel 5 Outputs 1
- Schéma structurel 6 Outputs 2
- Schéma structurel 7 Serial Links

Le chiffre des centaines dans le nom d'un composant est le numéro de folio : C601 est un condensateur du folio 6.
Tous les composants dont la valeur est notée NM ne sont pas montés sur la carte considérée.

A.4 EXTRAITS DE STRUCTURES LOGICIELLES

A.4.1 EXTRAITS DE FICHIERS DE CONFIGURATION

1) Extrait de la structure struct T_Porte

```
typedef struct T_Porte
{
    // ***** Déclaration des variables d'interface avec l'application *****
    T_BOOLLEEN I_Open_Door;
    T_BOOLLEEN I_Close_Door;
    T_BOOLLEEN I_Stop_Door;
    T_BYTE I_Door_Position; //Sortie 3 états : inactive,
                            // en cours d'ouverture, en cours de fermeture.
    T_BYTE I_Door_Status; //Sortie 3 états : porte ouverte, porte
                          // non fermée, porte fermée
    T_BYTE I_Obstacle_Status; //Sortie 3 états : pas d'obstacle, obstacle à
                              // ouverture, obstacle à fermeture
    T_BOOLLEEN I_Close_Door_Switch; // Entree
    T_BOOLLEEN I_Emergency_Switch; // Entree : Actif/Inactif
    T_BOOLLEEN I_Obstacle_Detection_Inhibit; // Entrée : désactive la détection d'obstacle
    T_BYTE I_Current_Measurement; // Mesure du courant : 0,1A/Bit
    T_BOOLLEEN I_Motor_Fault; // Sortie
    T_BYTE I_Motor_Direction; // Sortie 3 états : pilote en ouverture,
                              // en fermeture, arrêté
    T_BYTE I_Motor_Duty_Cycle; // Sortie rapport cyclique
    T_BOOLLEEN I_Obstacle_Sensor;
    T_BOOLLEEN I_Init_Driver; // Sortie utilisé lors d'un reset
    T_BOOLLEEN I_N_Tentatives_Ok; // Sortie
    T_BOOLLEEN I_Frein_De_Porte; // Sortie

    // ***** Déclaration des Paramètres *****
    T_WORD P_DoorSteadyModeStepRegion1Opening;
    T_WORD P_DoorSteadyModeStepRegion2Opening;
    T_WORD P_DoorSoftStartModeStepOpening;
    T_WORD P_Door_Threshold_Open;
    T_BYTE P_Motor_Supply_Step;
    T_BYTE P_Maximum_Opening_Time;
    T_WORD P_Temporal_Pos_Counter_Max_Porte;
    // ***** Déclaration des variables internes au driver *****/
    T_WORD Temporal_Pos_Counter_Porte;
    T_BYTE Command_Motor;
    T_BYTE Motor_Supply;
    T_BYTE Motor_Final_Value;
    T_WORD Motor_Supply_Region1_Open;
    T_WORD Motor_Supply_Region2_Open;
    T_BYTE Motor_Status;
    T_BYTE Motor_Phase;
    T_BOOLLEEN Obstacle_Detection;
    T_BYTE Opening_Timeout_Counter;
    T_BOOLLEEN Timeout_Opening;
    T_BYTE Automate_Door;
    T_BYTE Automate_Motor;
    T_BYTE Automate_Detection;
    T_BYTE Automate_Normal_Mode;
    T_BYTE Automate_Emergency_Mode;
```

```

T_BYTE      Automate_Driver;
T_WORD      Duty_Cycle;
T_BYTE      Direction ;

```

```

} T_Porte , *T_PtrPorte ;

```

2) Extrait du Fichier : RG_MON.H

```

//-----
// MULTIBUS NETWORK CONFIGURATION
// REMARK: These next 6 variables have to be initialized by application
// for a good start-up of the monitor.
//-----
// Net configuration.
// The four LSB bits indicate the number of IOU on branch 0 or the total
// number of IOU if the net is looped.
// The four MSB bits indicate the number of IOU on branch 1
// Must be initialized by application.
extern T_BYTE RG_CAN_ConfigReseau;

// The next 5 variables have to be initialized by application. The CMU
// compares, for each IOU it addresses, the value given by the IOU (cf CAN
// frame 2 of the addressing mode) and these variables.
// If its wrong the CMU put to sleep the IOU or reprogramming its E2PROM
// (for a RG_AdrCalc different)
// Customer Identification the same for all IOU
extern T_BYTE RG_CodeClient;

// Calculator Type : 1 = IOU
// The index of the table is the order of addressing
extern T_BYTE RG_TypeCalculeur[CT_NbMaxIOU+1];

// Hardware Configuration of each IOU
// The index of the table is the order of addressing
extern T_BYTE RG_ConfigHard[CT_NbMaxIOU+1];

// Value of the checksum of each IOU E2prom Zone 1
// The index of the table is the order of addressing
extern T_BYTE RG_Checksum_Zone1[CT_NbMaxIOU+1];

// Logic address of each IOU
// The index of the table is the order of addressing
extern T_BYTE RG_AdrCalc[CT_NbMaxIOU+1];
//-----
// END OF MULTIBUS NETWORK CONFIGURATION
//-----

// CAN INTERFACE DECLARATION
// In normal running mode you can access directly the four CAN frames of each
// IOU. (cf Multibus CAN interface description frame)
// To index this table use the constant define in const.h
// Example : Byte 1 of the frame3 of the IOU number 6 (bridge 2 current):
// ( 6 is its logical address fixed by application in RG_AdrCalc)
// Bridge2_Current = RG_MON_CanData[CT_Trm3_IOU6][1]
// In order to drive IOU you should use the macro defined in rgiou.h. It's
// more simple than use RG_MON_CanData directly.
// if you use rgiou.h the example is written :
// Bridge2_Current = RG_IOU6_BR2CURRENT_MEASURE
//-----
extern T_BYTE RG_MON_CanData[CT_CAN_TailleBloc*4][8];
//-----
// END OF CAN INTERFACE DECLARATION
//-----

// SCU DRIVER DECLARATION
//-----
// table of the exchange datas between CMU and SCU
// 160 bytes are exchange every 50ms
// the 80 first bytes are for reception from SCU
// the 80 last bytes of this table is for emission to SCU
// the CMU sends 80 bytes to SCU in 10 frames of 8 bytes by CAN bus(2 frames
// every 10ms) (CT_SCU_PacketNumberMax = 5)
// the SCU does the same thing in reverse
// to index this table, use the constants declared in const.h:
// CT_FromSCU_1, , CT_FromSCU_10, CT_ToSCU_1, , CT_ToSCU_10
extern T_BYTE RG_MON_SCUBuffer[CT_SCU_PacketNumberMax*4][8];

// every 50 ms the CMU scan the frames received from SCU
// if there is a missed frame it increased the byte corresponding
// in the next table (10 frames max : from 0 to 9).
// if the frame is received the byte is set to zero.

```

```

// for example if RG_MON_SCUMissedFrame[CT_FromSCU_1] = 10
// it means that you don't have the frame CT_FromSCU_1 since : 10 * 50ms = 500ms.
extern T_BYTE RG_MON_SCUMissedFrame[CT_SCU_PacketNumberMax*2];

// application has to initialize the two next variables with the number of
// frame that must be send(Tx) and received(Rx) from SCU.
// if the two are set to zero it's like there is no SCU for the CMU
extern T_BYTE RG_MON_SCUTxFrameMax;
extern T_BYTE RG_MON_SCURxFrameMax;

```

3) Extrait du Fichier INIT_APPLI

Initialisations des fonctions de gestion des portes

```

//Seuil de début de zone de ralentissement
// en ouverture
#define CT_MOTOR_THRESH_REGION_OPEN_AV 5000 //50°
#define CT_MOTOR_THRESH_REGION_OPEN_MI1 5000 //50°
#define CT_MOTOR_THRESH_REGION_OPEN_AR 5000 //50°
//...en fermeture
#define CT_MOTOR_THRESH_REGION_CLOSE 1500
//Seuil à partir de laquelle la porte peut être déclaré ouverte.
#define CT_DOOR_THRESHOLD_OPEN_AV 6800 //68°
#define CT_DOOR_THRESHOLD_OPEN_MI1 7500 //75°
#define CT_DOOR_THRESHOLD_OPEN_AR 7500 //75°
#define CT_TemporalPosCounterMaxPorte 10000 //100°
#define CT_THEORIC_VALUE_BATTERY 24 //24V
//Constante intervenant dans la formule d'alimentation.
#define CT_SUPPLY_STEP 20 // 0.2V/bit * 100 precision
//Seuil d'incrémentatation de la tension lors du démarrage moteur en ouverture ou fermeture.
#define CT_MOTOR_SUPPLY_STEP 20 // Inc. de 20 (2V)toute les 50ms
//Les valeurs des tensions pour les régions de fonctionnement 1 et 2 sont initialisées à l'aide de
tableaux.
TableNORMAL_VOLTAGE_OPEN_1 [5] = {95, // < 18V
80, // 18V <= Vbat < 22V
120, // 22V <= Vbat < 26V //V1.03 : 70% --> 90%
120, // 26V <= Vbat < 30V //V1.03 : 70% --> 90%
80}; // 32V <= Vbat
const T_BYTE TableNORMAL_VOLTAGE_OPEN_2 [5] = {95, // < 18
80, // 18V <= X < 22V
70, // 22V <= X < 26V //V1.03 : 50% --> 60%
70, // 26V <= X < 30V //V1.03 : 50% --> 60%
60}; // 32V <= X
//Seuil de tension pendant la zone début de fonctionnement porte en fermeture 0V au seuil.
const T_BYTE TableNORMAL_VOLTAGE_CLOSE_1 [5] = {70, // < 18
70, // 18V <= X < 22V
70, // 22V <= X < 26V
70, // 26V <= X < 30V
70}; // 32V <= X
//Seuil de tension nominal pendant fonctionnement porte en fermeture.
const T_BYTE TableNORMAL_VOLTAGE_CLOSE_2 [5] = {70, // < 18
70, // 18V <= X < 22V
70, // 22V <= X < 26V
70, // 26V <= X < 30V
70}; // 32V <= X
//Time-Out temps d'ouverture ou de fermeture de porte avant de passer en mode Emergency.
const T_BYTE TableMAXIMUM_OPENING_TIME [5] = {250, // 12.5s < 18
200, // 10s 18V <= X < 22V
160, // 8s 22V <= X < 26V //V1.03 6s-->8s
160, // 8s 26V <= X < 30V //V1.03 5s-->8s
160 // 8s 32V <= X //V1.03 5s-->8s
};
const T_BYTE TableMAXIMUM_CLOSING_TIME [5] = {240, // 12s < 18
200, // 10s 18V <= X < 22V
200, // 10s 22V <= X < 26V //V1.03 6s-->10s
200, // 10s 26V <= X < 30V //V1.03 5s-->10s
200 // 10s 32V <= X //V1.03 5s-->10s
};

// Valeurs de courant pour les différentes zones de fonctionnement à l'ouverture
// et à la fermeture, en fonction de la tension mesurée par la batterie
// SoftStat Open / Région 1 Open / Région 1 Open / Région 2 Open/ Région 2 Open /
// SoftStat Close / Région 1 Close / Région 1 Close / Région 2 Close/ Région 2 Close
const T_BYTE TableCourantPorteAv [5][10] = {65,65,65,65,38, 34,34,43,43,43, // < 18
65,65,65,65,36, 50,50,44,44,44, // 18V <= X < 22V
85,65,65,70,42, 55,55,48,45,45, // 22V <= X < 26V
85,65,65,70,46, 60,60,50,45,45, // 26V <= X < 30V
65,65,65,65,48, 69,69,50,50,50 // 32V <= X
};

```

```
//Valeur des incréments du compteur (de position) suivant la position de la porte.
//-----
//SoftStat Open / Région 1 Open / Région 2 Open / SoftStat Close / Région 1 Close / Région 2 Close
const T_WORD TableCompteurPositionPorteAv [16][6]= {
    { 0, 80, 70, 0, 53, 53, // < 18
      5, 80, 75, 5, 73, 73, // 18V <= X < 19V
      8, 90, 90, 8, 75, 75, // 19V <= X < 20V
      10, 95, 93, 10, 83, 83, // 20V <= X < 21V
      12, 100, 95, 12, 94, 94, // 21V <= X < 22V
      15, 142, 79, 15, 103, 103, // 22V <= X < 23V
      18, 155, 90, 18, 112, 112, // 23V <= X < 24V
      20, 155, 90, 20, 112, 112, // 24V <= X < 25V
      22, 164, 100, 22, 123, 123, // 25V <= X < 26V
      25, 174, 111, 25, 134, 134, // 26V <= X < 27V
      28, 183, 122, 28, 145, 145, // 27V <= X < 28V
      30, 193, 130, 30, 157, 157, // 28V <= X < 29V
      32, 206, 139, 32, 165, 165, // 29V <= X < 30V
      35, 170, 122, 35, 173, 173, // 30V <= X < 31V
      38, 180, 128, 38, 181, 181, // 31V <= X < 32V
      40, 190, 135, 40, 188, 188 // 32V <= X
    };
};
```

Fonction MAJ-param

```
void Maj_Param(void)
{
    //-----
    // VARIABLES QU'IL EST NECESSAIRE D'INITIALISER POUR UN BON FONCTIONNEMENT DU MONITEUR
    // Network configuration (number of unit on each CAN bus)
    // 0) Configuration : 0x00 = not used , 0x01 = standard MULTIBUS ,
    //                    0x02 = VDO Dashboard , 0x03 = J1939
    // 1) Number of IOU on each branch : bit3-bit0 = on branch 0 (AD_CAN0)
    //                                   bit7-bit4 = on branch 1 (AD_CAN1)
    // 2) Network type : 0x00 = open , 0x01 = closed
    // 3) Number of frames for dialog with 2nd master : bit3-bit0 = From 2nd master
    //                                                  bit7-bit4 = To 2nd master
    // 4) Number of SLU
    //-----
    const T_BYTE far CT_NetworkConfig[3][5] =
    {
        { // CAN_1 :
          0x01, // configuration
          0x09, // number of IOU on each branch
          0x00, // network type (open)
          0x00, // number of frames for dialog with 2nd master
          0x00, // number of SLU
        },
        { // CAN_2 :
          0x01, // configuration
          0x00, // number of IOU on each branch
          0x00, // network type
          0x00, // number of frames for dialog with 2nd master
          0x00, // number of SLU
        },
        { // CAN_3 :
          etc.
        }
    };

    // -----
    // Adresse des IOU : RG_AdrCalc
    // Tableau indiquant l'adresse des IOU dans l'ordre où le CMU les adresse.
    RG_AdrCalc [1] = 1 ;
    RG_AdrCalc [2] = 2 ;
    RG_AdrCalc [3] = 3 ;
    RG_AdrCalc [4] = 4 ;
    RG_AdrCalc [5] = 5 ;
    RG_AdrCalc [6] = 6 ;
    RG_AdrCalc [7] = 7 ;
    RG_AdrCalc [8] = 8 ;
    RG_AdrCalc [9] = 9 ;
    // -----
    // Code client : RG_CodeClient
    // Code commun à tous les IOU du réseau
    // Lors de la phase d'adressage si un IOU retourne un code client différent
    // de celui indiqué alors il est endormi.
    RG_CodeClient = 0x80;
    //-----
    // Config de chaque IOU: variable de vérification de la conformité
}
```

```

// de chaque calculateur adressé par le CMU à l'init.
// Si un calculateur renvoie pendant la phase d'adressage une valeur non
// conforme à ce qui a été initialisé, le CMU l'endort.
// L'indice de chaque tableau correspond au numéro d'ordre sur le réseau
// et pas à l'adresse du calculateur.
// C'est le même indice que pour RG_AdrCalc.
// RG_TypeCalculateur : Caractérise le type de calculateur rencontre.
// RG_ConfigHard : Caractérise une version Hard du calculateur
// RG_Checksum_Zone1 : indique la valeur du checksum de la zone1 de l'E2PROM
// du calculateur.
//-----
// Paramétrage pour 9 IOU
RG_TypeCalculateur[1] = CT_IOU;
RG_ConfigHard [1] = 0x81;
RG_Checksum_Zone1 [1] = 0x81;

```

4) Extrait du fichier Const . h

Définition de constantes et de macros et de types

L'accès aux données d'une trame CAN se fait par deux index qui sont d'abord le numéro global de la trame puis le numéro de l'octet de la trame choisie. Pour obtenir le numéro global d'une trame, il faut connaître l'unité d'entrée sortie (IOU) à laquelle elle appartient. Celle-ci est définie dans le tableau

Table	IOU adresse logique	Numéro de trame
RG_MON_CanData[0][?]	1	NORM1
RG_MON_CanData[1][?]	1	NORM2
RG_MON_CanData [2][?]	1	NORM3
RG_MON_CanData [3][?]	1	NORM4
RG_MON_CanData [4][?]	2	NORM1
RG_MON_CanData [5][?]	2	NORM2
...	...	...
RG_MON_CanData [46][?]	12	NORM3
RG_MON_CanData [47][?]	12	NORM4

```

//*****
// DEFINITION
#define CT_Bit0 0x0001
#define CT_Bit1 0x0002
#define CT_Bit2 0x0004
#define CT_Bit3 0x0008
#define CT_Bit4 0x0010
#define CT_Bit5 0x0020
#define CT_Bit6 0x0040
#define CT_Bit7 0x0080
#define CT_Bit8 0x0100
#define CT_Bit9 0x0200
#define CT_Bit10 0x0400
#define CT_Bit11 0x0800
#define CT_Bit12 0x1000
#define CT_Bit13 0x2000
#define CT_Bit14 0x4000
#define CT_Bit15 0x8000
...
#define CT_Trm1_IOU1 0
#define CT_Trm2_IOU1 1
#define CT_Trm3_IOU1 2
#define CT_Trm4_IOU1 3
#define CT_Trm1_IOU2 4
#define CT_Trm2_IOU2 5
#define CT_Trm3_IOU2 6
#define CT_Trm4_IOU2 7
#define CT_Trm1_IOU3 8
#define CT_Trm2_IOU3 9
#define CT_Trm3_IOU3 10
#define CT_Trm4_IOU3 11
#define CT_FromSCU_1 1
etc.

//*****
// MACRO
#define NOT !
#define AND &&
#define OR ||
#define XOR ^
#define INV ~ // binary operator

#define MC_SetBit(mot,bit) (mot) |= (bit)

```

```

#define MC_ClrBit(mot,bit)    (mot) &= ~(bit)
#define MC_TestBit(mot,bit)  ((mot) & (bit) ? 1 : 0)
#define MC_LoadBit(mot,bit,val)  ((val) ?
    MC_SetBit(mot,bit) : MC_ClrBit(mot,bit))
#define MC_SetBitIfOk(mot,bit,val)
    if (val) MC_SetBit(mot,bit)

//*****
// Type def : data types
// Datas type definition used by
// the monitor
typedef unsigned char T_BOOLEAN ;
typedef unsigned int  T_BITFIELD;
typedef unsigned char T_BYTE;
typedef unsigned int  T_WORD;
typedef signed int    T_SWORD;
typedef unsigned long T_DWORD;
typedef T_WORD        T_EpaTime;
typedef T_BOOLEAN     T_Status;
typedef T_Status      *T_PtrStatus;
typedef T_DWORD       T_DtAcqFre;
typedef T_WORD        T_ValeurAna;
typedef T_ValeurAna   *T_PtrValeurAna;
// typedef for conversion of two bytes
// in a word
typedef union
{
    T_WORD Mot;
    struct
    {
        T_BYTE LSB;
        T_BYTE MSB;
    } Poids;
} T_IOU_LMSBToWord;

```

5) Extraits du fichier RGIOU.h

Dans le nom des macros, le x de la dénomination « IOUx » correspond à l'adresse logique de chaque IOU (entre 1 et 9).

Quand une macro est précédé d'un type, la macro retourne une valeur :

Exemple :

```

#define RG_IOU1_ITEMP_MEASURE (T_BYTE)(RG_MON_CanData[CT_Trm3_IOU1][3])
x = RG_IOU1_ITEMP_MEASURE ;    ↔    x = RG_MON_CanData[CT_Trm3_IOU1][3] ;

```

Quand il y a un nom après l'identificateur, la macro est une commande avec une variable d'entrée.

Exemple :

```

#define RG_IOU1_SSUPPLY_COMMAND(Val) RG_MON_CanData[CT_Trm4_IOU1][0] = (T_BYTE)Val
RG_IOU1_SSUPPLY_COMMAND(8) ;    ↔    RG_MON_CanData[CT_Trm4_IOU1][0] = 8 ;

```

Extrait du fichier RGIOU.h

```

/*****
/*                                IOU1                                */
/*****
// FRAME NUMBER 1 OF NORMAL RUNNING
#define RG_IOU1_EFREQ0_DEFAULT (T_BOOLEAN)MC_TestBit(RG_MON_CanData[CT_Trm1_IOU1][0],CT_Bit7)
#define RG_IOU1_EFREQ1_DEFAULT (T_BOOLEAN)MC_TestBit(RG_MON_CanData[CT_Trm1_IOU1][0],CT_Bit6)
#define RG_IOU1_EFREQ2_DEFAULT (T_BOOLEAN)MC_TestBit(RG_MON_CanData[CT_Trm1_IOU1][0],CT_Bit5)
#define RG_IOU1_EFREQ0_MEASURE (T_WORD)((T_WORD)RG_MON_CanData[CT_Trm1_IOU1][0]&0x1F)*256 \
    +(T_WORD)RG_MON_CanData[CT_Trm1_IOU1][1])
#define RG_IOU1_EFREQ0_LOGICAL (T_BOOLEAN) (RG_MON_CanData[CT_Trm1_IOU1][1] & 0x01)
#define RG_IOU1_EFREQ1_MEASURE (T_BYTE) (RG_MON_CanData[CT_Trm1_IOU1][2])
#define RG_IOU1_EFREQ1_LOGICAL (T_BOOLEAN) (RG_MON_CanData[CT_Trm1_IOU1][2] & 0x01)
#define RG_IOU1_EFREQ2_MEASURE (T_BYTE) (RG_MON_CanData[CT_Trm1_IOU1][3])
#define RG_IOU1_EFREQ2_LOGICAL (T_BOOLEAN) (RG_MON_CanData[CT_Trm1_IOU1][3] & 0x01)
#define RG_IOU1_EREV0 (T_BOOLEAN)MC_TestBit(RG_MON_CanData[CT_Trm1_IOU1][4],CT_Bit7)
#define RG_IOU1_ELOG6 (T_BOOLEAN)MC_TestBit(RG_MON_CanData[CT_Trm1_IOU1][4],CT_Bit6)
#define RG_IOU1_ELOG5 (T_BOOLEAN)MC_TestBit(RG_MON_CanData[CT_Trm1_IOU1][4],CT_Bit5)
etc.
#define RG_IOU1_ELOG1 (T_BOOLEAN)MC_TestBit(RG_MON_CanData[CT_Trm1_IOU1][4],CT_Bit1)
#define RG_IOU1_ELOG0 (T_BOOLEAN) (RG_MON_CanData[CT_Trm1_IOU1][4] & 0x01)
#define RG_IOU1_BR3_DEFAULT (T_BOOLEAN)MC_TestBit(RG_MON_CanData[CT_Trm1_IOU1][5],CT_Bit3)
#define RG_IOU1_BR2_DEFAULT (T_BOOLEAN)MC_TestBit(RG_MON_CanData[CT_Trm1_IOU1][5],CT_Bit2)
#define RG_IOU1_BR1_DEFAULT (T_BOOLEAN)MC_TestBit(RG_MON_CanData[CT_Trm1_IOU1][5],CT_Bit1)
#define RG_IOU1_EREV1 (T_BOOLEAN) (RG_MON_CanData[CT_Trm1_IOU1][5] & 0x01)
#define RG_IOU1_EANA0_MEASURE (T_BYTE) (RG_MON_CanData[CT_Trm1_IOU1][6])
#define RG_IOU1_EANA0_LOGICAL (T_BOOLEAN) (RG_MON_CanData[CT_Trm1_IOU1][6] & 0x01)
#define RG_IOU1_EANA1_MEASURE (T_BYTE) (RG_MON_CanData[CT_Trm1_IOU1][7])
#define RG_IOU1_EANA1_LOGICAL (T_BOOLEAN) (RG_MON_CanData[CT_Trm1_IOU1][7] & 0x01)

```

```

// FRAME NUMBER 2 OF NORMAL RUNNING
#define RG_I0U1_EANA5_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][0],CT_Bit7)
#define RG_I0U1_EANA4_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][0],CT_Bit6)
etc.
#define RG_I0U1_EANA0_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][0],CT_Bit2)
#define RG_I0U1_EANA2_MEASURE (T_WORD)((((T_WORD)RG_MON_CanData[CT_Trm2_I0U1][0] & 0x03)*256 \
+(T_WORD)RG_MON_CanData[CT_Trm2_I0U1][1])
#define RG_I0U1_EANA2_LOGICAL (T_BOOLLEEN)(RG_MON_CanData[CT_Trm2_I0U1][1] & 0x01)
#define RG_I0U1_E2PROM_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][2],CT_Bit7)
#define RG_I0U1_SSUPPLY_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][2],CT_Bit5)
#define RG_I0U1_ITEMP_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][2],CT_Bit4)
#define RG_I0U1_EANA4_MEASURE (T_WORD)((((T_WORD)RG_MON_CanData[CT_Trm2_I0U1][2] & 0x0C)>>2)\
*256 + (T_WORD)RG_MON_CanData[CT_Trm2_I0U1][4])
#define RG_I0U1_EANA4_LOGICAL (T_BOOLLEEN)(RG_MON_CanData[CT_Trm2_I0U1][4] & 0x01)
#define RG_I0U1_EANA3_MEASURE (T_WORD)((((T_WORD)RG_MON_CanData[CT_Trm2_I0U1][2] & 0x03)*256 \
+(T_WORD)RG_MON_CanData[CT_Trm2_I0U1][3])
#define RG_I0U1_EANA3_LOGICAL (T_BOOLLEEN)(RG_MON_CanData[CT_Trm2_I0U1][3] & 0x01)
#define RG_I0U1_EANA5_MEASURE (T_BYTE)(RG_MON_CanData[CT_Trm2_I0U1][5])
#define RG_I0U1_EANA5_LOGICAL (T_BOOLLEEN)(RG_MON_CanData[CT_Trm2_I0U1][5] & 0x01)
#define RG_I0U1_OUT7_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][6],CT_Bit7)
#define RG_I0U1_OUT6_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][6],CT_Bit6)
etc.
#define RG_I0U1_OUT0_DEFAULT (T_BOOLLEEN)(RG_MON_CanData[CT_Trm2_I0U1][6] & 0x01)
#define RG_I0U1_TOGGLEBIT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][7],CT_Bit7)
#define RG_I0U1_OUT14_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][7],CT_Bit6)
#define RG_I0U1_OUT13_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][7],CT_Bit5)
etc.
#define RG_I0U1_OUT9_DEFAULT (T_BOOLLEEN)MC_TestBit(RG_MON_CanData[CT_Trm2_I0U1][7],CT_Bit1)
#define RG_I0U1_OUT8_DEFAULT (T_BOOLLEEN)(RG_MON_CanData[CT_Trm2_I0U1][7] & 0x01)

// FRAME NUMBER 3 OF NORMAL RUNNING
#define RG_I0U1_BR1CURRENT_MEASURE (T_BYTE)(RG_MON_CanData[CT_Trm3_I0U1][0])
#define RG_I0U1_BR2CURRENT_MEASURE (T_BYTE)(RG_MON_CanData[CT_Trm3_I0U1][1])
#define RG_I0U1_BR3CURRENT_MEASURE (T_BYTE)(RG_MON_CanData[CT_Trm3_I0U1][2])
#define RG_I0U1_ITEMP_MEASURE (T_BYTE)(RG_MON_CanData[CT_Trm3_I0U1][3])
#define RG_I0U1_VAMS_MEASURE (T_BYTE)(RG_MON_CanData[CT_Trm3_I0U1][4])

// FRAME NUMBER 4 OF NORMAL RUNNING
#define RG_I0U1_SSUPPLY_COMMAND(Val) RG_MON_CanData[CT_Trm4_I0U1][0] = (T_BYTE)Val
#define RG_I0U1_OUT8_COMMAND(Val) \
MC_LoadBit(RG_MON_CanData[CT_Trm4_I0U1][1],CT_Bit7,(T_BYTE)Val)
#define RG_I0U1_OUT7_COMMAND(Val) \
MC_LoadBit(RG_MON_CanData[CT_Trm4_I0U1][1],CT_Bit6,(T_BYTE)Val)
etc.
#define RG_I0U1_OUT0_COMMAND(Val) \
MC_LoadBit(RG_MON_CanData[CT_Trm4_I0U1][1],CT_Bit0,(T_BYTE)Val)
#define RG_I0U1_BR2D1_COMMAND \
RG_MON_CanData[CT_Trm4_I0U1][2] = (RG_MON_CanData[CT_Trm4_I0U1][2] & 0x3F) | 0x40
#define RG_I0U1_BR2D2_COMMAND RG_MON_CanData[CT_Trm4_I0U1][2] |= 0xC0;
#define RG_I0U1_BR2_STOP MC_ClrBit(RG_MON_CanData[CT_Trm4_I0U1][2],CT_Bit6)
#define RG_I0U1_BR1D1_COMMAND \
RG_MON_CanData[CT_Trm4_I0U1][2] = (RG_MON_CanData[CT_Trm4_I0U1][2] & 0xCF) | 0x10
#define RG_I0U1_BR1D2_COMMAND RG_MON_CanData[CT_Trm4_I0U1][2] |= 0x30;
#define RG_I0U1_BR1_STOP MC_ClrBit(RG_MON_CanData[CT_Trm4_I0U1][2],CT_Bit4)
etc.
#define RG_I0U1_ASLEEP_COMMAND MC_SetBit(RG_MON_CanData[CT_Trm4_I0U1][2],CT_Bit0)
#define RG_I0U1_BR3D1_COMMAND RG_MON_CanData[CT_Trm4_I0U1][3] = \
(RG_MON_CanData[CT_Trm4_I0U1][3] & 0xFC) | 0x01
#define RG_I0U1_BR3D2_COMMAND RG_MON_CanData[CT_Trm4_I0U1][3] |= 0x03;
#define RG_I0U1_BR3_STOP MC_ClrBit(RG_MON_CanData[CT_Trm4_I0U1][3],CT_Bit0)
#define RG_I0U1_OUT5_COMMAND(Val) RG_MON_CanData[CT_Trm4_I0U1][4] = \
(RG_MON_CanData[CT_Trm4_I0U1][4] & 0xF0) | ((T_BYTE)Val & 0x0F)
#define RG_I0U1_OUT10_COMMAND(Val) RG_MON_CanData[CT_Trm4_I0U1][5] = \
(RG_MON_CanData[CT_Trm4_I0U1][5] & 0x0F) | ((T_BYTE)Val << 4)
#define RG_I0U1_OUT9_COMMAND(Val) RG_MON_CanData[CT_Trm4_I0U1][5] = \
(RG_MON_CanData[CT_Trm4_I0U1][5] & 0xF0) | ((T_BYTE)Val & 0x0F)
#define RG_I0U1_OUT12_COMMAND(Val) RG_MON_CanData[CT_Trm4_I0U1][6] = \
(RG_MON_CanData[CT_Trm4_I0U1][6] & 0x0F)\ | ((T_BYTE)Val << 4)
#define RG_I0U1_OUT11_COMMAND(Val) RG_MON_CanData[CT_Trm4_I0U1][6] = \
(RG_MON_CanData[CT_Trm4_I0U1][6] & 0xF0) | ((T_BYTE)Val & 0x0F)
#define RG_I0U1_BR2_PARAMETER(Val) RG_MON_CanData[CT_Trm4_I0U1][6] = \
(RG_MON_CanData[CT_Trm4_I0U1][6] & 0xF0)\ | ((T_BYTE)Val & 0x0F)
//Byte =Duty cycle (unit 10%) from 1 to 9 or frequency (Unit 50HZ) from 1 to 10

#define RG_I0U1_OUT14_COMMAND(Val) RG_MON_CanData[CT_Trm4_I0U1][7] = \
(RG_MON_CanData[CT_Trm4_I0U1][7] & 0x0F)\ | ((T_BYTE)Val << 4)
#define RG_I0U1_OUT13_COMMAND(Val) RG_MON_CanData[CT_Trm4_I0U1][7] = \
(RG_MON_CanData[CT_Trm4_I0U1][7] & 0xF0)\ | ((T_BYTE)Val & 0x0F)
#define RG_I0U1_BR3_PARAMETER(Val) RG_MON_CanData[CT_Trm4_I0U1][7] = \

```

```
(RG_MON_CanData[CT_Trm4_I0U1][7] & 0xF0) | ((T_BYTE)Val & 0x0F)
//Byte=Duty cycle (unit 10%) from 1 to 9 or frequency (Unit 50HZ) from 1 to 10
```

```
#define RG_I0U1_OFF_COMMAND      RG_MON_CanData[CT_Trm4_I0U1][0] = 0x00;\
                                RG_MON_CanData[CT_Trm4_I0U1][1] = 0x00;\
                                RG_MON_CanData[CT_Trm4_I0U1][2] = 0x00;\

                                etc.

                                RG_MON_CanData[CT_Trm4_I0U1][7] = 0x00
```

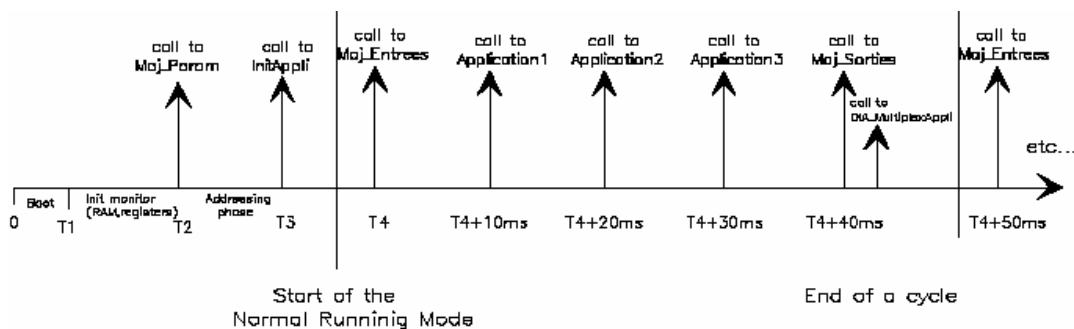
etc.

A.4.2 TÂCHES APPLICATIVES EN MODE NORMAL

1) Descriptif

Le séquençement des appels à l'applicatif en fonctionnement normal est le suivant :

MAJ_ENTREES	à T	(actualisation des entrées : lecture de la zone de mémoire partagée)
APPLICATION1	à T + 10 ms	
APPLICATION2	à T + 20 ms	
APPLICATION3	à T + 30 ms	
MAJ_SORTIES	à T + 40 ms	(actualisation des sorties : écriture de la zone de mémoire partagée)
DIA_MultiplexAppl		



Toute application doit être structurée de la manière suivante :

- traitement des entrées à câblage paramétrable (réaffectation des entrées suivant un paramètre optionnel)
- gestion des variables d'états/traitement des conditions de réveil (exemple : appel du driver permettant de déterminer les origines de réveil qui sont remontées à l'application)
- gestion des temporisations
- gestion des alimentations
- programmes (pères et fils associés)
- traitement de l'endormissement
- traitement des sorties à câblage paramétrable (réaffectation des sorties suivant un paramètre optionnel)

Application1 :

- Modification et neutralisation d'entrée spécifique, Gestion des origines du réveil.
- Calcul des principales variables d'états (exemple : Etat du moteur, Etat du mode Diagnostic, Etat Arrêt urgence,...)
- Désactivation des éléments du système pouvant être actifs dans le mode Arrêt Urgence
- Gestion des sorties dans le cas du mode arrêt urgence

Application2 :

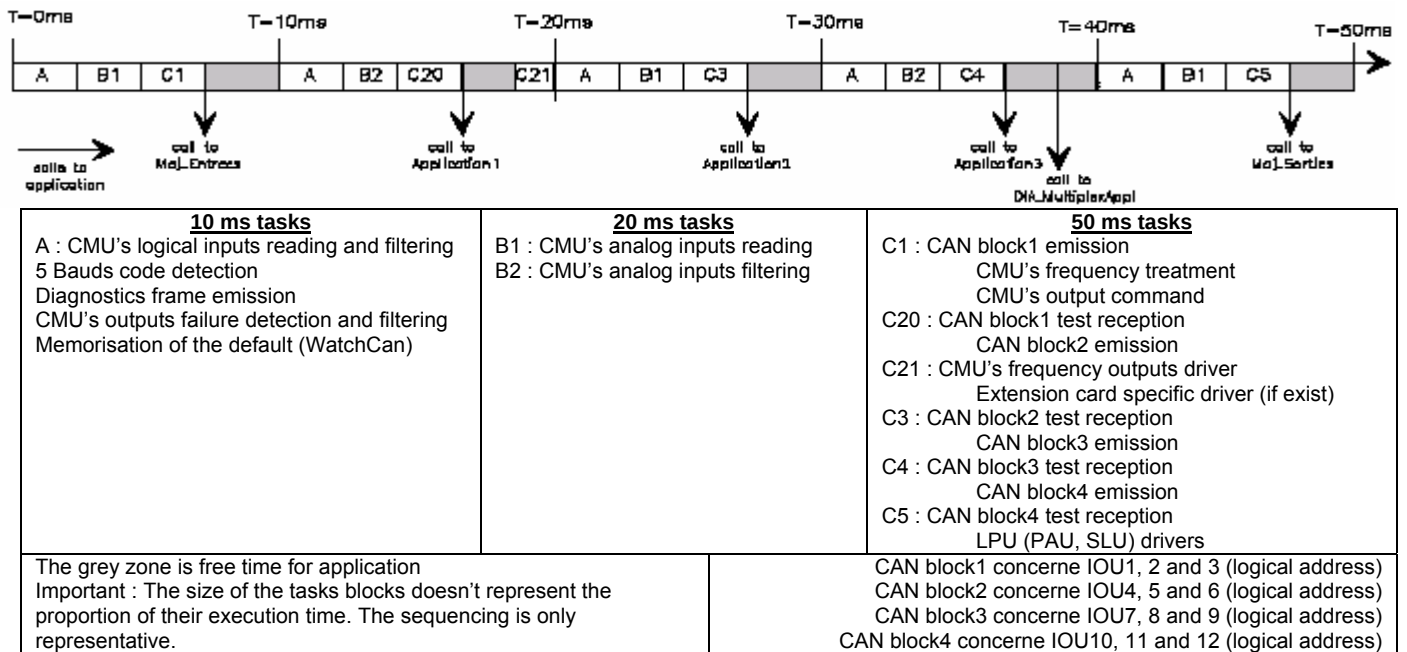
```
//Gestion des fonctions applicatives clients (première partie).
MCF_Eclair_Exe_FonctionsEclairage;
MCF_Moteur_Exe_FonctionsMoteur;
MCF_Frein_Exe_FonctionsFrein;
```

Application3 :

```
//Gestion des fonctions applicatives clients (deuxième partie).
MCF_Porte_Exe_FonctionsPorte;
MCF_Hand_Exe_FonctionsHandicape;
//Gestion Fonctions Autre1 (toujours en avant dernière position).
MCF_Autre_Exe_FonctionsAutre1;
MCF_Autre_Exe_FonctionsAutre2;
//Gestion endormissement (toujours en dernière position).
```

Mise à jour des sorties :

Module qui doit correspondre aux commandes des sorties des calculateurs en fonction de l'état des variables applicatives. Le point d'entrée de ce module est la fonction MAJ_SORTIES. Dans le cas du fonctionnement normal c'est le dernier point d'entrée appelé par le moniteur.



Séquencement des tâches pour un cycle du moniteur en mode de fonctionnement normal

2) Extrait de Maj_Entree.c

```
RG_E_Bp_ArretDemandeC = RG_IOU6_ELOG4 ;
```

3) Extrait de Maj_Sortie.c

```
//-----
// IOU7
//-----
RG_IOU7_OUT5_COMMAND(RG_S_EclairageArretDemandePorteAv);
RG_IOU7_OUT7_COMMAND(RG_S_Vyt_HabilitBpOuvPorteAvParPassager);
RG_IOU7_OUT9_COMMAND(RG_S_Plus15_5);
RG_IOU7_OUT10_COMMAND(RG_S_FreinPorteAv);
// Bridge 1
RG_IOU7_BR1_STOP;
// Bridge 2
if (!RG_S_BornePlusMoteurPorteAv AND !RG_S_BorneMoinsMoteurPorteAv)
{
    RG_IOU7_BR2_STOP;
}
else
{
    RG_IOU7_BR2_PARAMETER (RG_S_Pwm_Freq_CmdeMoteurPorteAv) ;
    if (RG_S_BornePlusMoteurPorteAv AND !RG_S_BorneMoinsMoteurPorteAv)
    {
        RG_IOU7_BR2D1_COMMAND; // direction1
    }
    else
    {
        RG_IOU7_BR2D2_COMMAND; // direction2
    }
}
...
// bit 3 : SCU S6
MC_LoadBit(RG_MON_SCUBuffer[CT_ToSCU_3][6],CT_Bit3,RG_S_Vyt_ArretDemandeTabDeBordPass);
...

```

A.4.3 APPLICATION GESTION DES PORTES (PORTE_EXE_FONCTIONSPORTE)

L'application de gestion de portes (Porte_Exe_FonctionsPorte) effectue les tâches suivantes :

- Calcul de la tension moyenne d'alimentation (batterie) Porte_Cmde_MoyenneTensionPourPorte()
- Gestion des états de la porte à l'aide du driver de porte. Porte_Cmde_Portes()

Les différents états de la porte sont gérés par un « driver de porte ». Ce pilote dispose des fonctionnalités suivantes.

- Porte_Cmde_DoorPositionDriver(S_Porte);
- Porte_Cmde_ObstructionDetectionDriver(S_Porte);
- Porte_Cmde_TimeoutDetectionDriver(S_Porte);
- Porte_Cmde_ModesDriver(S_Porte);
- Porte_Cmde_CalculTransitionDriverMode(S_Porte);
- Porte_Cmde_MotorDriver(S_Porte);

Chaque porte est décrite à l'aide d'une structure T_porte.

1) Extrait de Porte_Cmde_Portes

Cet extrait est limité à la gestion de la porte Avant

Organisation générale

```
{
// I . Fonctions préparatoires et annexe précédant le driver porte.
// Gestion des voyants de défauts moteurs des portes, de l'inhibition sécurité de porte,
// autorisation ouverture porte.;
// Gestion habilitation des portes, Gestion ouverture/fermeture porte avant / milieu 1 et
// arrière.
//-----
// II . Mise à jour ou initialisation des Interfaces d'entrée du driver porte -
//Porte Av
RG_Syst_PorteAv.I_Current_Measurement = RG_E_A_MesureCourantMoteurPorteAv ;
RG_Syst_PorteAv.I_Close_Door_Switch = RG_E_CloseDoorSwitchAv;
RG_Syst_PorteAv.I_Emergency_Switch = RG_E_Int_ExclusAlimFreinPorteAv;
RG_Syst_PorteAv.I_Motor_Fault = RG_Def_S_CmdeDirectMoteurPorteAv;
//Calcul tension moyenne d'alimentation des portes.
Porte_Cmde_CalculParametreParRapportVBat();
//-----
// III . Appel drivers Portes :
MCF_Porte_Exe_FonctionsDriverPorte;
//-----
// IV . Mise à jour des Interfaces de sortie du driver
//Cablage moteurs porte Avant.
RG_S_Pwm_Freq_CmdeMoteurPorteAv = RG_Syst_PorteAv.I_Motor_Duty_Cycle;
//Cablage Direction du moteur de portes.
switch(RG_Syst_PorteAv.I_Motor_Direction)
{
case CT_MOTOR_DIRECTION_OPEN:
RG_S_BornePlusMoteurPorteAv = CT_OFF;
RG_S_BorneMoinsMoteurPorteAv = CT_ON;
break;
case CT_MOTOR_DIRECTION_CLOSE:
RG_S_BornePlusMoteurPorteAv = CT_ON;
RG_S_BorneMoinsMoteurPorteAv = CT_OFF;
break;
case CT_MOTOR_DIRECTION_STOP:
default:
RG_S_BornePlusMoteurPorteAv = CT_OFF;
RG_S_BorneMoinsMoteurPorteAv = CT_OFF;
break;
}
//Cablage position porte : ouverte ou fermée.
RG_Syst_PositionPorteAv =RG_Syst_PorteAv.I_Door_Position;
//Cablage détection bord sensible. //Porte Av.
switch(RG_Syst_PorteAv.I_Obstacle_Status)
{
case CT_OBSTACLE_STATUS_FAULT_CLOSING:
RG_Syst_DetectDefautFermPorteAv =CT_ON;
RG_Syst_DetectDefautOuvPorteAv =CT_OFF;
break;
case CT_OBSTACLE_STATUS_FAULT_OPENING:
RG_Syst_DetectDefautFermPorteAv =CT_OFF;
RG_Syst_DetectDefautOuvPorteAv =CT_ON;
break;
case CT_OBSTACLE_STATUS_INACTIVE:
default:
RG_Syst_DetectDefautFermPorteAv =CT_OFF;
RG_Syst_DetectDefautOuvPorteAv =CT_OFF;
break;
}
//cablage des états et des commande porte : inactive / en ouverture / en fermeture.
//Porte Avant.
RG_Syst_CommandePorteAv=RG_Syst_PorteAv.I_Door_Status;
RG_S_FreinPorteAv=RG_Syst_PorteAv.I_Frein_De_Porte;
//-----
// V . Fonctions exploitant les résultats du pilote de porte
```

```
//Gestion voyants et signaux portes.
Porte_Cmde_VytEtSignauxPorte();
```

2) Extrait de Porte_Cmde_TensionPourPorte

```
void Porte_Cmde_TensionPourPorte(void)
{
    T_BYTE      RL_I;
    T_WORD      RL_Somme_TensionPorte;
    //Gestion du pointeur de tableau.
    //-----
    if ( RG_Int_PointeurMoyTensionPorte >= CT_PeriodeMoyenneTensionPorte )
    {
        RG_Int_PointeurMoyTensionPorte = 0;
    }
    RG_TableauTensionPorteAv[RG_Int_PointeurMoyTensionPorte] = RG_E_A_MesTensionBatterieIOU7 * 2;
    RL_Somme_TensionPorte = 0;
    for ( RL_I = 0 ; RL_I < CT_PeriodeMoyenneTensionPorte ; RL_I++ )
    {
        RL_Somme_TensionPorte = RL_Somme_TensionPorte + RG_TableauTensionPorteAv[RL_I];
    }
    RG_TensionPorteAv = RL_Somme_TensionPorte / CT_PeriodeMoyenneTensionPorte;

    //Gestion du pointeur de tableau.
    RG_Int_PointeurMoyTensionPorte = RG_Int_PointeurMoyTensionPorte + 1;
}
}
```

3) Extrait de Porte_Cmde_CalculParametreParRapportVBat

Cet extrait est limité à la gestion de la porte Avant

```
void Porte_Cmde_CalculParametreParRapportVBat (void)
{
    T_BYTE      RL_CptPositionAv;
    T_BYTE      RL_CptCourantAv;
    T_WORD      RG_Int_SeuilPositionTemporaire;
    //Porte Av.
    RL_CptPositionAv=0;
    if ( RG_TensionPorteAv >= 180 )
    {
        if ( RG_TensionPorteAv < 320 )
        {
            RL_CptPositionAv = (T_BYTE)((RG_TensionPorteAv-180)/(10))+1;  //1*10
        }
        else
        {
            RL_CptPositionAv = 15; //Ecrétage à 32V.
        }
    }
    if ( RL_CptPositionAv > 15 )
    {
        RL_CptPositionAv = 15; //Ecrétage à 32V.
    }
    //-----
    //Mise à jour suivant équations de tension.
    //-----
    //Calcul table compteur de position.
    //Porte avant.
    RG_Syst_PorteAv.P_DoorSoftStartModeStepClosing =
        RG_P_CompteurPositionAv[RL_CptPositionAv][3];
    RG_Syst_PorteAv.P_DoorSteadyModeStepRegion1Closing =
        RG_P_CompteurPositionAv[RL_CptPositionAv][4];
    RG_Syst_PorteAv.P_DoorSteadyModeStepRegion2Closing =
        RG_P_CompteurPositionAv[RL_CptPositionAv][5];

    //Calcul table Courant.
    RG_Syst_PorteAv.P_NorCurrValueOpen_1 = RG_P_NorCurrValuePorteAv[RL_CptCourantAv][0];
    RG_Syst_PorteAv.P_NorCurrValueOpen_2 = RG_P_NorCurrValuePorteAv[RL_CptCourantAv][1];
    RG_Syst_PorteAv.P_NorCurrValueOpen_3 = RG_P_NorCurrValuePorteAv[RL_CptCourantAv][2];
    RG_Syst_PorteAv.P_NorCurrValueOpen_4 = RG_P_NorCurrValuePorteAv[RL_CptCourantAv][3];
    RG_Syst_PorteAv.P_NorCurrValueOpen_5 = RG_P_NorCurrValuePorteAv[RL_CptCourantAv][4];
    RG_Syst_PorteAv.P_NorCurrValueClose_1 = RG_P_NorCurrValuePorteAv[RL_CptCourantAv][5];
    RG_Syst_PorteAv.P_NorCurrValueClose_2 = RG_P_NorCurrValuePorteAv[RL_CptCourantAv][6];
    RG_Syst_PorteAv.P_NorCurrValueClose_3 = RG_P_NorCurrValuePorteAv[RL_CptCourantAv][7];
    RG_Syst_PorteAv.P_NorCurrValueClose_4 = RG_P_NorCurrValuePorteAv[RL_CptCourantAv][8];
    RG_Syst_PorteAv.P_NorCurrValueClose_5 = RG_P_NorCurrValuePorteAv[RL_CptCourantAv][9];

    //Calcul table PWM. //An les zones sont les mêmes que pour le courant.
    RG_Syst_PorteAv.P_Normal_Voltage_Open_1 = TableNORMAL_VOLTAGE_OPEN_1[RL_CptCourantAv];
    RG_Syst_PorteAv.P_Normal_Voltage_Open_2 =
}
```

```

TableNORMAL_VOLTAGE_OPEN_2[RL_CptCourantAv];
RG_Syst_PorteAv.P_Normal_Voltage_Close_1 = TableNORMAL_VOLTAGE_CLOSE_1[RL_CptCourantAv];
RG_Syst_PorteAv.P_Normal_Voltage_Close_2 = TableNORMAL_VOLTAGE_CLOSE_2[RL_CptCourantAv];

//Calcul table PWM. //An les zones sont les mêmes que pour le courant.
RG_Syst_PorteAv.P_Maximum_Opening_Time = TableMAXIMUM_OPENING_TIME[RL_CptCourantAv];
RG_Syst_PorteAv.P_Maximum_Closing_Time = TableMAXIMUM_OPENING_TIME[RL_CptCourantAv];
RG_Syst_PorteMi1.P_Maximum_Opening_Time = TableMAXIMUM_OPENING_TIME[RL_CptCourantMi1];
RG_Syst_PorteMi1.P_Maximum_Closing_Time = TableMAXIMUM_OPENING_TIME[RL_CptCourantMi1];
RG_Syst_PorteAr.P_Maximum_Opening_Time = TableMAXIMUM_OPENING_TIME[RL_CptCourantAr];
RG_Syst_PorteAr.P_Maximum_Closing_Time = TableMAXIMUM_OPENING_TIME[RL_CptCourantAr];
}

```

A.4.4 PILOTE DE PORTE

Le pilote de porte est basé sur des machines à états finis.

Vue générale du pilote

```

// CD NAME : Porte_Cmde_ManagerDriver
// *****
// CD ROLE : Gere l'appel des différents sous-drivers
void Porte_Cmde_ManagerDriver(T_PtrPorte S_Porte)
{
    Porte_Cmde_DoorPositionDriver(S_Porte);
    Porte_Cmde_ObstructionDetectionDriver(S_Porte);
    Porte_Cmde_TimeoutDetectionDriver(S_Porte);
    Porte_Cmde_ModesDriver(S_Porte);
    Porte_Cmde_CalculTransitionDriverMode(S_Porte);
    Porte_Cmde_MotorDriver(S_Porte);
    Porte_Cmde_IncrementationStep(S_Porte);
}

```

Porte_Cmde_DoorPositionDriver(S_Porte) : 2 modes de fonctionnement

- **mode normal** : La position de la porte est incrémentée ou décrémentée suivant que l'on est en ouverture ou en fermeture à partir des incréments définis dans le gabarit. Il s'agit donc d'une estimation de la position de la porte. Lorsque la porte est fermée, un écrêtage de la position à la valeur Zéro est réalisé. Une comparaison de la valeur de la position de la porte et d'une valeur maximale admissible (temporalPosCounterMaxPorte), permet de détecter la présence d'obstacle, et de prendre en compte l'état du moteur. Ce pilote fixe donc l'état (ouvert, fermé, non fermé) de la porte.
- **mode emergency** : la porte est forcée en ouverture.

Porte_Cmde_ObstructionDetectionDriver(S_Porte) : machines à états

Prend en compte la valeur du courant mesurée et la détection *bord sensible* et fixe la valeur de *Automate_Detection*, *Obstacle_Detection*

Porte_Cmde_TimeoutDetectionDriver(S_Porte) : vérifie les durées d'ouverture et de fermeture de la porte et compare à la valeur max (TimeOUT)

Porte_Cmde_ModesDriver(S_Porte) : permet de synchroniser et d'organiser les différents modes de fonctionnement

- teste s'il y a une transition de la variable I_Motor_Fault (détection d'un défaut moteur)
- puis exécute la machine à états (Automate_Driver)
- Initialise les valeurs des paramètres (exemple Motor_Supply_Region1_Open, Thresh_Overcurr_Mask_1) et les variables d'états (I_Door_Status, I_Obstacle_Status)

3 modes possibles

- PREDISPO_EMERGENCY_MODE : RG_E_Int_ExclusAlimFreinPorte actif I_Emergency_Switch
- NORMAL_MODE
- EMERGENCY_MODE

Porte_Cmde_CalculTransitionDriverMode(S_Porte) : teste si on passe d'un mode à un autre

Porte_Cmde_MotorDriver(S_Porte) : Gère le fonctionnement du moteur de porte : machine à états (Automate_Motor)

- fixe les variables I_Motor_Direction, Motor_Phase; I_Motor_Direction; Motor_Supply; Command_Motor; I_Motor_Direction, Motor_Final_Value

Porte_Cmde_DutyCycleCalculation(S_Porte)

- Transforme la tension en PWM,
- Fixe la valeur de I_Motor_Duty_Cycle à :
(CT_SUPPLY_STEP*Motor_Supply)/CT_THEORIC_VALUE_BATTERY

Porte_Cmde_IncrementationStep(S_Porte) : définit les valeurs des pas d'incrément en fonction de la phase du moteur (Door_Steady_Mode_Step ou Door_SoftStart_Mode_Step)

B ANNEXE B : EXTRAITS DE NOTICES TECHNIQUES DU SYSTÈME

B.1 DOCUMENTATION DES MATÉRIELS

B.1.1 MOTEUR

Les deux vantaux de la porte sont couplés mécaniquement. Un seul moteur est utilisé pour les piloter. Il est alimenté par un pont en H intégré dans une unité d'entrée sortie (IOU) ;

Caractéristiques du moteur :

Courant à vide de l'ordre de 0,03 A

Tension : 18 - 30 VDC (24 Vdc)

Courant : 2,5 à 8 A

Courant CC moteur bloqué 10 à 12 A

R : 1 à 10 ohm suivant le moteur

L : 500 μ H à 5 nH suivant le moteur

B.1.2 UNITÉS D'ENTRÉES SORTIES (IOU)

Caractéristiques matérielles d'une IOU

Chaque IOU dispose de 5 connecteurs : CN1 (Vams), CN2 et CN3 sont les connecteurs où l'on branche le réseau CAN (connecteurs de communication), CN4 (Entrées) et CN5 (Sorties + Vbat).

Glossaire

Vbat : Positive power supply coming directly from vehicle battery

V1 (=Vams=Vacb) : Positive power supply switched by the master switch

High side output : Vbat level when activated

Low side output : GND level when activated

High side input : the input is considered high level when Vbat is applied on its pin.

Low side input : the input is considered high level when GND is applied on its pin.

General Features

Voltage 24 V (from 16 to 32 V, operating during sharp fluctuation in voltage caused by actuation of the starter)

Protection against short circuit to ground or battery on each connector pin

Polarity inversion on the Vbat unswitched power supply inputs

Protection grade: IP655 (with the opposite connectors plugged)

Operating temperature range: -40°C to + 70°C

Storage temperature range: -40°C to + 85°C

Vibrations : 5 g from 10 Hz to 300 Hz in sinusoidal mode (15 hours on the 3 axes) and 0,02g_r/Hz from 10 Hz to 300 Hz in random mode (15 hours on the 3 axes)

EMC/EMI Immunity :

. Electrical disturbance on power leads and power active inputs ISO7637-2: class B

pulse 1 : -300 V; Ri = 10 Ω ; td = 2 ms

pulse 2 : +100 V; Ri = 10 Ω ; td = 0,2 ms

pulse 3a,3b : $\pm$ 200 V; Ri = 50 Ω ; td = 0,1 μ s

pulse 4 : Vs = - 16 V; Va = -12 V, t6 = 50 ms; t8 = 10 s (fluctuation in voltage caused by actuation of the starter)

pulse 5 : +120 V; Ri = 2,2 Ω ; td = 220 ms (the load dump protection needs the IOU to be connected to a master ECU)

. Electrical disturbance on signal lines ISO7637-3 : class B

pulse 2 : +100 V; Ri = 10 Ω ; td = 0,2 ms

pulse 3a,3b : $\pm$ 200 V; Ri = 50 Ω ; td = 0,1 μ s

Electrostatic discharges :

Direct discharges on the enclosure through 2 k Ω and 330 pF: $\pm$ 4 kV

Direct discharges on the connector pins through 2 k Ω and 330 pF: $\pm$ 2 kV

Air discharges through 2 k Ω and 330 pF : $\pm$ 8 kV

. Bulk current injection (BCI) on wiring harness (part 4) ISO 11452-4: 100 mA from 1 MHz to 400 MHz

Micro-controller Fujitsu MB90F545 (cf. documentation en annexe C)

CPU : 16 bits at 16 MHz avec CAN controllers : 1

RAM : 6 kbytes **FLASH EPROM** : 128 kbytes **E2PROM** : 1 kbytes

ENTREES

Les IOU possèdent 18 entrées réparties sur le connecteur CN104.

- 7 entrées logiques In0, In1, In2, In3, In4, In5, In6
- 6 entrées analogiques Ana0, Ana1, Ana2, Ana3, Ana4, Ana5
- 2 entrées de réveil WK0, WK1
- 3 entrées fréquentielles Freq0, Freq1, Freq2

Entrées logiques :

Pull up:	2,2k pull up to 20 V
Low voltage threshold (0 logic value):	2 V
High voltage threshold (1 logic value):	8 V
Maximum positive input voltage :	32 V
Maximum negative input voltage:	-5 V
Switch current (with 28V power supply)	9 mA $\pm$ 10%
Hardware filtering from pin to CPU:	0,8 ms

Un traitement interne au microcontrôleur de l'IOU permet par ailleurs de filtrer les entrées logiques grâce à l'utilisation d'un seuil de détection prenant une valeur entre 0 et 255.

Quand l'état de l'entrée logique ne change pas, un compteur est incrémenté toutes les 5ms.

Quand l'état de l'entrée change, le compteur est remis à zéro.

Lorsque la valeur du compteur est supérieure ou égale au seuil, l'entrée est considérée comme stable.

Entrées Analogiques :

Les entrées analogiques utilisent le convertisseur 10 bits multiplexé interne au microcontrôleur. Le temps de conversion est de 26,3 μ s par voie. Chaque voie est numérisée toutes les 10 ms.

Un dispositif externe au microcontrôleur permet l'utilisation d'un gain analogique.

Les entrées Ana0 à Ana3 peuvent être utilisées en mesure de tension ou bien en mesure de résistance.

	Ana0 to Ana3
Pull up/down	2,2 k Ω pull-up to 5 V
Voltage measurements:	
Voltage range	[0 – 5 V] (2% precision @2,5 V)
Resistance measurements:	
Resistance measurement range	[500 – 10 k Ω] with 5% precision

Quant aux entrées Ana4 et Ana5, elles ne peuvent être utilisées qu'en mesure de tension

	Ana4	Ana5
Voltage measurements:		
Voltage range	[0 – 32 V] (2 % precision @ 15 V)	[0 – 15 V] (2 % precision @ 7,5 V)

Toutes les entrées analogiques subissent un traitement numérique dans l'IOU avant de pouvoir être envoyées par le réseau CAN.

Filtre numérique

- K is the analog filter coefficient in the following relationship :
- $\text{new_filtered_value} = (((K-1) * \text{old_filtered_value}) + \text{new_analog_input_value}) / K$
- New_filtered_value is calculated every 10ms.
- Possible values for K are: **1, 2, 4, 8, 16, 32** or **64**.

Correction numérique

Suppression d'un éventuel offset et gain parasite dans la chaîne d'acquisition.

- Corrected value = (Filtered value + B)*A / 32
- A parameters are **char unsigned** (A = stored value * 32 = lxxxx_A * 32)
- B parameters are **char signed**

Note : These the parameters are used to correct direct (non-amplified) inputs only. For indirect (amplified) inputs, calculations are done with parameters from the amplifier, in order to obtain the true correction parameters.

Conversion Linéaire

- The following parameters are used in the analog conversion formula :
- Converted value = $((C1/C2) * \text{Corrected value}) + D$
- All parameters are **char signed**, therefore their values are between -128 and +127.

SORTIES

Les IOU possèdent 18 sorties sur le connecteur CN105

- 1 sortie High side 9 A (Out0)
- 1 sortie High side 7 A (Out5)
- 6 sorties High side outputs 2 A (OUT4, OUT6, OUT7, OUT8, OUT9, OUT10)
- 3 sorties High side outputs 3,2 A (OUT1, OUT2, OUT3)
- 4 sorties Logic HS/LS/Half Bridge/Push-Pull outputs
 - 9 A HIGH SIDE BRANCH (Out12)
 - 7 A HIGH SIDE BRANCH (Out11, Out13 et Out14)
 - 5 A LOW SIDE BRANCH (OUT11, OUT12, OUT13, OUT14)

Toutes ces sorties sont conçues pour pouvoir piloter des ampoules, des charges résistives, des relais des moteurs ou des électrovannes. Les sorties OUT11/OUT12 et OUT13/OUT14 peuvent former deux ponts en H (mode demi-pont). Avec chacun des ponts, il est possible de piloter un moteur ou toutes autres charges bidirectionnelles.

Toutes les sorties sont protégées contre les courts-circuits à Vbat (ou GND), contre les surintensités et les surchauffes. Sur les sorties HS, la détection d'un court-circuit ou d'un circuit ouvert est réalisé.

Remarque : la sortie OUT12D est réservée pour une utilisation ultérieure.

POWER SUPPLY

The I/OU has 2 power supplies :

- * Direct battery power supply (+ BAT line)
- * Switched battery power supply (Vbat Switched)
- Nominal operating voltage : 24 Vdc
- Min operating voltage : 16 Vdc
- Max operating voltage : 32 Vdc
- Sleep mode supply current on +BAT line : < 3 mA
- Max continuous current on +BAT line (normal mode) : 15 A
- Max continuous current on Vbat Switched (normal mode) : 40 A

The sum of the permanent power dissipated on the body must be less than 7W with these locals constraints :

- The sum of permanent power dissipated on OUT0, OUT5, OUT11, OUT12, OUT13, OUT14 < 7 W
- The sum of permanent power dissipated on OUT2 to OUT3 < 1,3 W
- The sum of permanent power dissipated on OUT4, OUT6, OUT7, OUT8 < 1,3 W
- The sum of permanent power dissipated on OUT1, OUT9, OUT10 < 1,3 W

Moreover, to avoid internal overheat, an internal temperature measurement is done inside the ECU. If the internal temperature is more then 85°C the application software will enter in degraded mode and cut off some critical output in order to reduce the internal dissipation.

OPERATING MODES

The IOU has 3 operating modes :

- * Sleep mode (Vbat Switched OFF): the IOU does not operate.
- * Normal mode without Vbat switched line : the system is able to read all the inputs. The IOU is working and is able to drive loads connected on the Vbat unswitched.
- * Normal mode (Vbat Switched ON) : all the functions operate normally.

There are 3 ways of leaving sleep mode :

- * EREV0 inputs with wake-up capability grounded.
- * Low level on the CAN address input lines (these inputs are reserved by the system).
- * Activation of the Vbat Switched

CAN CONTROLLER /INTERFACES

The physical layer is fully compatible with the ISO 11898 standard.

Absolute maximum ratings (no operation) :

- Voltage on CAN H : -3 / +32 V
- Voltage on CAN L : -3 / +32 V

Operating ratings :

- Recessive state output voltage on CAN H : 2,5 V
- Recessive state output voltage on CAN L : 2,5 V
- Dominant state output voltage on CAN H : 3,5 V (load : 60 W between CAN H and CAN L)
- Dominant state output voltage on CAN L : 1,5 V (load : 60 W between CAN H and CAN L)
- Recessive state differential voltage : 0,1 V max (no load)
- Dominant state differential voltage : 1,5 V min

The IOU supports the extended message frames in CAN specification 2.0 part B. The bit rate is 250 Kbits/s.

B.1.3 UNITÉ DE SUPERVISION (CMU)

L'architecture matérielle des unités de supervision est similaire à celle des IOU. Toutefois, la mémoire, le microcontrôleur utilisé, le nombre d'entrées sorties diffèrent.

MICRO-CONTROLLER AND MEMORY FEATURES

CPU : 16 bits at 20 MHz with 2 CAN controllers and 1 UART :ISO 9141/ ISO14230

RAM : 512 kbytes **FLASH** : 1Mbytes **E2PROM** : 2 kbytes

The General features, EMC/EMI Immunity:Electrostatic discharges, power supply and operating modes are similar with IOU.

INPUTS. The CMU provides 24 inputs as follows:

- 13 logic inputs (low-side inputs). Same as IOU, Hardware filtering: 1,5 ms
- 5 Analog /logic inputs (low-side inputs). Same as IOU
- 2 diagnostic lines (high-side input).
- 2 wake-up inputs (low-side logical inputs with wake-up capability)
- 2 frequency/PWM or logic low side inputs.

OUTPUTS : the CMU provides 23 outputs as follow:

- 14 logic high-side HS1 output.
- 2 logic high-side HS7 output.
- 4 logic low side LS5 output.
- 1 Frequency/PWM voltage galvanometer driver (0..10 V).
- 1 Frequency/PWM voltage galvanometer driver (0..15 V).
- 1 Voltage galvanometer driver (0 V - 10 V).

CAN CONTROLLER /INTERFACES : two CAN bus are available on the CMU. Same Features as IOU

B.2 COMMUNICATION ENTRE LE CMU ET LES AUTRES ÉQUIPEMENTS

Il y a trois types de fonctionnement pour le réseau Can MULTIBUS de la société ACTIA

- La phase d'adressage et d'initialisation du réseau.
- Le fonctionnement normal
- Le mode de test

B.2.1 GESTION DU RÉSEAU

De par sa qualité de calculateur maître, le CMU doit connaître tous les paramètres du réseau. Il doit connaître la forme du réseau, combien d'esclaves sont connectées et leurs adresses. Il doit connaître également toutes les trames circulant sur le réseau. Pour chaque trame, il doit savoir son type (TXD, RXD, RRXD), son identificateur, le calculateur esclave auquel elle appartient; et pour une trame en émission, le nombre d'octets à émettre. Toutes ces informations sont codées en ROM dans la mémoire du CMU.

L'identification d'un boîtier IOU se fait par sa fonctionnalité; par exemple « boîtier porte avant » ou « boîtier carrosserie arrière droit ». L'adresse d'un boîtier est liée à sa fonctionnalité et non à sa position sur le réseau. Pour la phase d'adressage, le CMU utilise un tableau lui indiquant l'adresse d'un boîtier en fonction de sa position. En effet, pendant cette phase, le CMU communique avec un boîtier au moyen des lignes d'adressages (ADCAN0, ADCAN1). Si l'ordre de connexion des boîtiers change, leurs adresses ne changent pas mais il faut mettre à jour le tableau. En phase de fonctionnement normal, l'ordre de connexion est sans importance car le CMU communique avec un boîtier esclave au moyen de son adresse qui est intégrée dans les identificateurs de trames.

B.2.2 STRUCTURE DU CÂBLAGE

Le système MULTIBUS est un système avec un maître unique, et un certain nombre d'esclaves. Chaque calculateur est relié à son ou ses voisins par 3 câbles :

- Deux pour le bus CAN (signaux CAN H et CAN L)
- une ligne ADCAN ou ligne d'adressage.

B.2.3 PHASE DE RÉVEIL

Le système est dit « **en veille** » lorsqu'il est alimenté en +BAT mais qu'il ne fonctionne pas. La consommation électrique est alors très faible. Le but de la phase de réveil est de réveiller tous les calculateurs. Pour cela, lorsqu'un calculateur se réveille, il positionne ses 2 lignes d'adressage (ADCAN0 et ADCAN1) à la masse, ce qui lui permet de réveiller ses voisins, et donc tout le système par propagation. Cette phase se termine lorsque tout le système est réveillé. Cette information est gérée par le calculateur maître selon la structure du réseau.

Lorsque tout le système est réveillé, le maître prend la main et commence la **phase d'adressage**.

Un calculateur peut être réveillé de 3 façons (par ordre de priorité) :

- présence de la tension +ACB (Après Coupe-Batterie)
- activation d'une ou de plusieurs entrée réveil
- activation (mise à la masse) d'une ligne ADCAN0 ou ADCAN1.

ATTENTION, dans le cas d'un IOU, les priorités sont inversées.

Réveil d'un calculateur IOU

2 cas peuvent se présenter :

- Réveil par une entrée réveil ou par la présence de la tension +ACB. Cette origine de réveil du boîtier correspond alors à l'origine de réveil du système.
- Réveil par une ligne d'adressage. Cela signifie que le réveil est provoqué par un boîtier voisin.

Dans les 2 cas, le calculateur doit réveiller ses voisins afin de propager le réveil. Il active donc ses 2 lignes d'adressage ADCAN0 et ADCAN1, et attend un ordre CAN venant du maître pour passer en phase d'adressage. S'il ne trouve aucune origine de réveil, l'IOU n'active pas ses lignes d'adressages et se rendort. C'est un cas de réveil intempestif.

B.2.4 PHASE D'ADRESSAGE

Il s'agit de l'adressage de boîtiers **IOU** par un boîtier **maître** (CMU)

La phase d'adressage a pour but de :

- vérifier le bon fonctionnement de tous les calculateurs.
- vérifier la cohérence du système : adresses et paramètres des esclaves.
- Identifier l'origine du réveil du système.
- Si besoin, reprogrammer les paramètres des esclaves.

B.3 SPÉCIFICITÉS DU RÉSEAU CAN (MULTIBUS ACTIA)

Dans les documentations les trames utilisées pour chaque mode de fonctionnement sont repérées par des sigles différents :

AdrX : frame number X for the addressing phase
NormX : frame number X for the normal running phase
TstX : frame number X for the functional testing phase

B.3.1 TRAMES ÉCHANGÉES ENTRE LES IOU ET LE CMU EN FONCTIONNEMENT NORMAL

The frames of this phase are named **NormX**.

You can access the bytes all the frames of all the IOU through the table RG_MON_CanData[Max_IOU][8]. Refer to RGMON.H listing in Appendix A its use.

XI.1 FRAME NORM1

Way : IOU → CMU

Function : State of IOU (part 1/3)

Remark : This frame is sent in reply to a remote frame.

Data :

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Freq input 0 failure	Freq input 1 failure	Freq input 2 failure	Frequency input 0 (MSB)				
Byte 1	Frequency input 0 (LSB)							
Byte 2	Frequency input 1							
Byte 3	Frequency input 2							
Byte 4	Wakeup input 0	Logical input 6	Logical input 5	Logical input 4	Logical input 3	Logical input 2	Logical input 1	Logical input 0
Byte 5	not used				Bridge 3 failure	Bridge 2 failure	Bridge 1 failure	Wakeup input 1
Byte 6	Analogical input 0							
Byte 7	Analogical input 1							

Remarks :

- * if the frequency input 0 is used as a logical input, its state is placed in the bit 0 of the byte 1.
- * if the frequency input 1 is used as a logical input, its state is placed in the bit 0 of the byte 2.
- * if the frequency input 2 is used as a logical input, its state is placed in the bit 0 of the byte 3.
- * if the analogical input 0 is used as a logical input, its state is placed in the bit 0 of the byte 6.
- * the not used bits are reserved for future use.

XI.2 FRAME NORM2

Way : IOU → CMU

Function : State of IOU (part 2/3)

Remark : This frame is sent in reply to a remote frame.

Data :

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Analog input 5 fail.	Analog input 4 fail.	Analog input 3 fail.	Analog input 2 fail.	Analog input 1 fail.	Analog input 0 fail.	Analog input 2 (MSB)	
Byte 1	Analogical input 2 (LSB)							
Byte 2	E2PROM failure	Wiper failure	Sensors supply fail.	Internal temp failure	Analog input 4 (MSB)		Analog input 3 (MSB)	
Byte 3	Analogical input 3 (LSB)							
Byte 4	Analogical input 4 (LSB)							
Byte 5	Analogical input 5							
Byte 6	OUT7 failure	OUT6 failure	OUT5 failure	OUT4 failure	OUT3 failure	OUT2 failure	OUT1 failure	OUT0 failure
Byte 7	ECU alive	OUT14 fail.	OUT13 fail.	OUT12 fail.	OUT11 fail.	OUT10 fail.	OUT9 failure	OUT8 failure

Remarks :

- * if the analogical input 2 is used as a logical input, its state is placed in the bit 0 of the byte 1.
- * if the analogical input 3 is used as a logical input, its state is placed in the bit 0 of the byte 3.
- * if the analogical input 4 is used as a logical input, its state is placed in the bit 0 of the byte 4.
- * if the analogical input 5 is used as a logical input, its state is placed in the bit 0 of the byte 5.
- * ECU alive : this bit is inverted each 50ms to show the ECU is still running correctly.

XL3 FRAME NORM3

Way : IOU → CMU

Function : State of IOU (part 3/3)

Remark : This frame is sent in reply to a remote frame.

Data :

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	bridge 1 current							
Byte 1	bridge 2 current							
Byte 2	bridge 3 current							
Byte 3	internal temperature							
Byte 4	+ACB tension							

XL4 FRAME NORM4

Way : CMU → IOU

Function : commands to drive the IOU's outputs

Data :

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	sensors supply tension (Unit: 0.1V)							
Byte 1	OUT8 on/off	OUT7 on/off	OUT6 on/off	OUT4 on/off	OUT3 on/off	OUT2 on/off	OUT1 on/off	OUT0 on/off
Byte 2	Bridge 2 direction	Bridge 2 on/off	Bridge 1 direction	Bridge 1 on/off	not used	Screen Wiper high speed	Screen Wiper low speed	Power down
Byte 3	not used						Bridge 3 direction	Bridge 3 on/off
Byte 4	not used				OUT5 Stat/Dyn			
Byte 5	OUT10 Stat/Dyn				OUT9 Stat/Dyn			
Byte 6	OUT12 Stat/Dyn				OUT11 Stat/Dyn / pont2Dyn			
Byte 7	OUT14 Stat/Dyn				OUT13 Stat/Dyn / pont3Dyn			

Remarks :

The conversion parameters of the sensor supply must be set to obtain 0.1V unit.

* Screen Wiper Stop = NOT(high speed) ET NOT(Low Speed)

* OUT5, OUT9, OUT10, OUT11, OUT12, OUT13, OUT14 drive : these outputs can work as PWM, frequency or logical outputs.

When driven in logical : The Output is ON if one of the 4 bits is ON

When driven in **logical** : The Output is ON if one of the 4 bits is ON

When used in **PWM** :

0000 = 0x0 -> OFF
 0001 = 0x1 -> Duty cycle = 10%
 0010 = 0x2 -> Duty cycle = 20%
 0011 = 0x3 -> Duty cycle = 30%
 0100 = 0x4 -> Duty cycle = 40%
 0101 = 0x5 -> Duty cycle = 50%
 0110 = 0x6 -> Duty cycle = 60%
 0111 = 0x7 -> Duty cycle = 70%
 1000 = 0x8 -> Duty cycle = 80%
 1001 = 0x9 -> Duty cycle = 90%
 1010 = 0xA -> Duty cycle = 100%

When used in **frequency** :

0000 = 0x0 -> OFF
 0001 = 0x1 -> Frequency = 50 Hz
 0010 = 0x2 -> Frequency = 100 Hz
 0011 = 0x3 -> Frequency = 150 Hz
 0100 = 0x4 -> Frequency = 200 Hz
 0101 = 0x5 -> Frequency = 250 Hz
 0110 = 0x6 -> Frequency = 300 Hz
 0111 = 0x7 -> Frequency = 350 Hz
 1000 = 0x8 -> Frequency = 400 Hz
 1001 = 0x9 -> Frequency = 450 Hz
 1010 = 0xA -> Frequency = 500 Hz

For a same frequency group, it's the last parameter in the frame that is used. For example, if Out5 and Out10 are dynamic outputs, they will both use the parameter from OUT10.

Bridge 1 = Out9 and Out10

Bridge 2 = Out11 and Out12 (dynamic parameter in OUT11)

Bridge 3 = Out13 and Out14 (dynamic parameter in OUT13)

Frequency group 1 = Out5, Out9 and Out10

Frequency group 3 = Out11 and Out12

Frequency group 4 = Out13 and Out14

Remarks :

- * if the analogical input 2 is used as a logical input, its state is placed in the bit 0 of the byte 1.
- * if the analogical input 3 is used as a logical input, its state is placed in the bit 0 of the byte 3.
- * if the analogical input 4 is used as a logical input, its state is placed in the bit 0 of the byte 4.
- * if the analogical input 5 is used as a logical input, its state is placed in the bit 0 of the byte 5.
- * ECU alive : this bit is inverted each 50ms to show the ECU is still running correctly.

B.3.2 DÉFINITION DES IDENTIFIANTS DES TRAMES POUR LE RÉSEAU CAN1 ENTRE LE CMU ET LES IOU.

Frames identifiers values are given in hexadecimal.

The ECU must use CAN 2.0B (29-bits identifiers) protocol, except of the SLU, which use CAN 2.0A (11-bits identifiers) protocol because of the used micro-controller.

VIII.1 EXTENDED FRAMES IDENTIFIER FORMAT

The frame identifiers have to be built using the following structure.

bit 28 - bit 24	bit 23 - bit 19	bit 18 - bit 14	bit 13 - bit 9	bit 8 - bit 0
Source Id.	Source number	Destination Id.	Destination number	Frame Id.

Source Id. : identification of the ECU that emits the frame.

bit 28	bit 27	bit 26	bit 25	bit 24	ECU code
0	0	0	0	0	MASTER Principal
1	0	1	0	0	IOU
0	0	1	0	1	MASTER Secondary
0	0	1	1	0	ALADIN
1	1	1	0	0	SCU/GMU Functional Test Unit
1	1	1	0	1	CAMU Functional Test Unit
1	1	1	1	0	IOU Functional Test Unit
1	1	1	1	1	RESERVED. DO NOT USE FOR SOURCE ID.

Source number :

ECU Id	Value
IOU	IOU address
other	00000

Destination Id. : identification of the ECU (see Source Id) which receives the frame. The code 11111 is used in case of broadcast frames (frames that have to be received by all ECUs).

Destination number : see **Source number**.

Frame Id :

bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Function code		Frame number					Packet number	

Function code : indicates in which operating mode the frame is used.

bit 8	bit 7	Function Code
0	0	Normal running frames
0	1	Addressing frames
1	0	not used
1	1	Functional testing frames

Frame number : codes the number X in TstX / AdrX / NormX

bit 6	bit 5	bit 4	bit 3	bit 2	Frame
0	0	0	0	1	Tst1 / Adr1 / Norm1
0	0	0	1	0	Tst2 / Adr2 / Norm2
0	0	0	1	1	Tst3 / Adr3 / Norm3
0	0	1	0	0	Tst4 / Adr4 / Norm4
					...

Packet number : in case of there are more than 8 data bytes, the frame can be divided in 4 8-bytes packets, using the packet number

VIII.3 IOU IDENTIFIERS

Some frames are depending on the IOU's number. They are Adr7, Adr10, Adr11, Adr13, Norm1, Norm2, Norm3 and Norm4. The other frames are not depending on the IOU's number.

IOU number	Frame Adr7	Frame Adr10	Frame Adr11	Frame Adr13	Frame Norm1	Frame Norm2	Frame Norm3	Frame Norm4
0	1400009C	000500A8	000500A C	000500B4	14000004	14000008	1400000C	00050010
1	1408009C	000502A8	000502A C	000502B4	14080004	14080008	1408000C	00050210
2	1410009C	000504A8	000504A C	000504B4	14100004	14100008	1410000C	00050410
3	1418009C	000506A8	000506A C	000506B4	14180004	14180008	1418000C	00050610
4	1420009C	000508A8	000508A C	000508B4	14200004	14200008	1420000C	00050810
5	1428009C	00050AA8	00050AAC	00050AB4	14280004	14280008	1428000C	00050A10
6	1430009C	00050CA 8	00050CAC	00050CB4	14300004	14300008	1430000C	00050C10
7	1438009C	00050EA8	00050EAC	00050EB4	14380004	14380008	1438000C	00050E10
8	1440009C	000510A8	000510A C	000510B4	14400004	14400008	1440000C	00051010
9	1448009C	000512A8	000512A C	000512B4	14480004	14480008	1448000C	00051210
10	1450009C	000514A8	000514A C	000514B4	14500004	14500008	1450000C	00051410
11	1458009C	000516A8	000516A C	000516B4	14580004	14580008	1458000C	00051610
12	1460009C	000518A8	000518A C	000518B4	14600004	14600008	1460000C	00051810

C ANNEXE C : DOCUMENTATION DE COMPOSANTS

C.1 COMPOSANTS ÉLECTRONIQUES

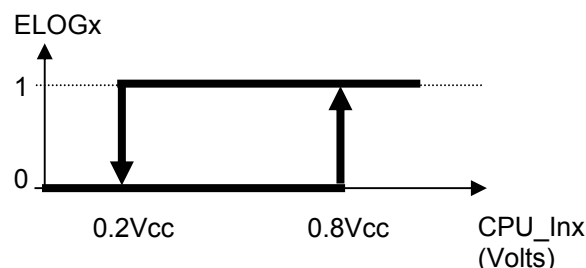
C.1.1 MICROCONTRÔLEUR MB90F540 /545

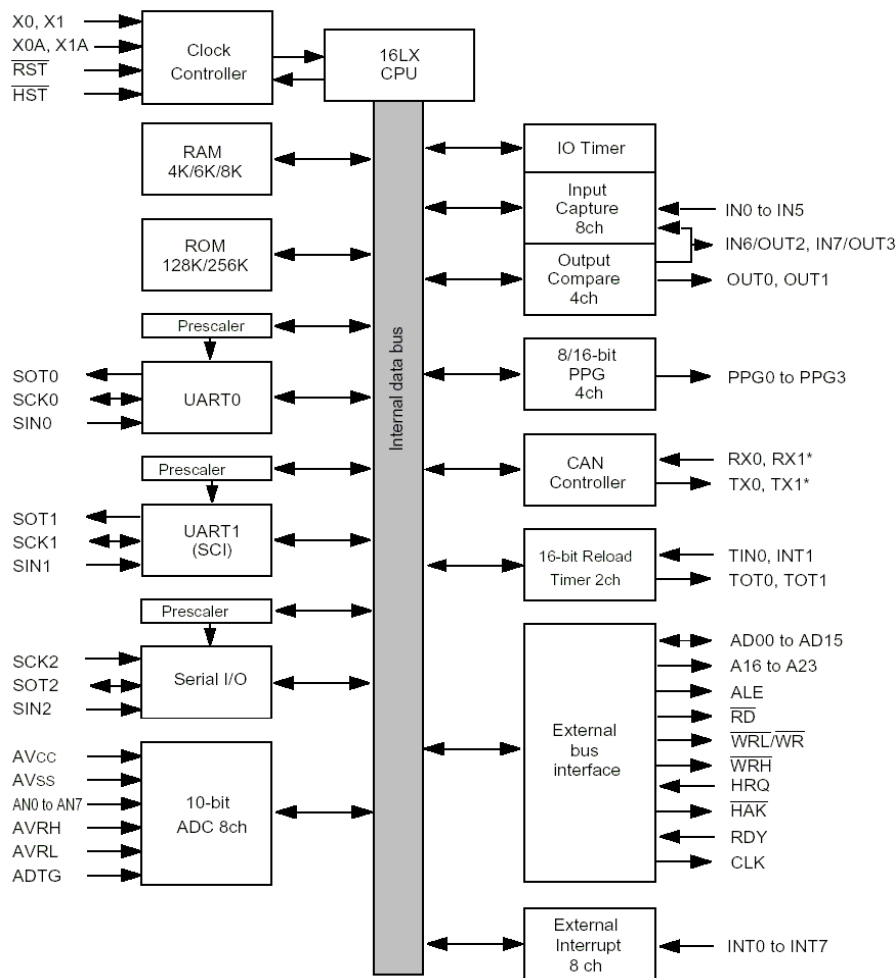
Function	Feature
UART0	Full duplex double buffer Supports asynchronous/synchronous (with start/stop bit) transfer Baud rate: 4808/5208/9615/10417/19230/38460/62500/500000bps (asynchronous) 500K/1M/2Mbps (synchronous) at System clock = 16MHz
UART1 (SCI)	Full duplex double buffer Asynchronous (start-stop synchronized) and CLK-synchronous communication Baud rate: 1202/2404/4808/9615/192301/31250/38460/62500bps (asynchronous) 62.5K/125K/250K/500K/1Mbps (synchronous) at 6,8,10,12,16 MHz
Serial IO	Transfer can be started from MSB or LSB Supports internal clock synchronized transfer and external clock synchronized transfer Supports positive-edge and negative-edge clock synchronization Baud rate : 31.25K/62.5K/125K/500K/1M/2M bps at System clock = 16MHz
A/D Converter	10-bit or 8-bit resolution 8 input channels Conversion time: 26.3 μ s (per one channel)
16-bit Reload Timer (2 channels)	Operation clock frequency: $f_{sys}/2^1$, $f_{sys}/2^3$, $f_{sys}/2^5$ (f_{sys} = System clock frequency) Supports External Event Count function
16-bit I/O Timer	Signals an interrupt when overflow Supports Timer Clear when a match with Output Compare(Channel 0) Operation clock freq.: $f_{sys}/2^2$, $f_{sys}/2^4$, $f_{sys}/2^6$, $f_{sys}/2^8$ (f_{sys} = System clock freq.)
16-bit Output Compare (4 channels)	Signals an interrupt when a match with 16-bit IO Timer Four 16-bit compare registers A pair of compare registers can be used to generate an output signal
16-bit Input Capture (8 channels)	Rising edge, falling edge or rising & falling edge sensitive Four 16-bit Capture registers Signals an interrupt upon external event

Function	Feature
8/16-bit Programmable Pulse Generator (4 channels)	Supports 8-bit and 16-bit operation modes Eight 8-bit reload counters Eight 8-bit reload registers for L pulse width Eight 8-bit reload registers for H pulse width A pair of 8-bit reload counters can be configured as one 16-bit reload counter or as 8-bit prescaler plus 8-bit reload counter 4 output pins Operation clock freq.: f_{sys} , $f_{sys}/2^1$, $f_{sys}/2^2$, $f_{sys}/2^3$, $f_{sys}/2^4$ or $128\mu s @ f_{osc}=4MHz$ (f_{sys} = System clock frequency, f_{osc} = Oscillation clock frequency)
CAN Interface MB90540 series: 2 channels MB90545 series: 1 channel	Conforms to CAN Specification Version 2.0 Part A and B Automatic re-transmission in case of error Automatic transmission responding to Remote Frame Prioritized 16 message buffers for data and ID's Supports multiple messages Flexible configuration of acceptance filtering: Full bit compare / Full bit mask / Two partial bit masks Supports up to 1Mbps
External Interrupt	Can be programmed edge sensitive or level sensitive
External bus interface	External access using the selectable 8-bit or 16-bit bus is enabled (external bus mode).
IO Ports	Virtually all external pins can be used as general purpose IO All push-pull outputs and schmitt trigger inputs Bit-wise programmable as input/output or peripheral signal
32 kHz subclock	Subclock for low-power operation
Flash Memory	Supports automatic programming, Embedded Algorithm™ *1 Write/Erase/Erase-Suspend/Resume commands A flag indicating completion of the algorithm Number of erase cycles: 10,000 times Data retention time: 10 years Boot block configuration Erase can be performed on each block Block protection by externally programmed voltage

*1: Embeded Algorithm is a trade mark of Advanced Micro Devices Inc.

Hystérésis sur entrées logiques





DC Characteristics :

Parameter	Symbol	Value		Units	Remarks
		Min	Max		
Power supply voltage	V _{CC}	V _{SS} - 0.3	V _{SS} + 6.0	V	
	AV _{CC}	V _{SS} - 0.3	V _{SS} + 6.0	V	V _{CC} = AV _{CC} *1
	AV _{RH} , AV _{RL}	V _{SS} - 0.3	V _{SS} + 6.0	V	AV _{CC} ≥ AV _{RH} /AV _{RL} , AV _{RH} ≥ AV _{RL} *1
Input voltage	V _I	V _{SS} - 0.3	V _{SS} + 6.0	V	*2
Output voltage	V _O	V _{SS} - 0.3	V _{SS} + 6.0	V	*2
Maximum clamp current	I _{CLAMP}	- 2.0	+ 2.0	mA	*6
Total maximum clamp current	Σ I _{CLAMP}	—	20	mA	*6
"L" level max output current	I _{OL}	—	15	mA	*3
"L" level avg. output current	I _{OLAV}	—	4	mA	*4
"L" level max overall output current	ΣI _{OL}	—	100	mA	
"L" level avg. overall output current	ΣI _{OLAV}	—	50	mA	*5
"H" level max output current	I _{OH}	—	-15	mA	*3
"H" level avg. output current	I _{OHAV}	—	-4	mA	*4
"H" level max overall output current	ΣI _{OH}	—	-100	mA	
"H" level avg. overall output current	ΣI _{OHAV}	—	-50	mA	*5
Power consumption	P _D	—	500	mW	Flash device
		—	400	mW	Mask ROM
Operating temperature	T _A	-40	+85	°C	MB90F543/F549
		-40	+105	°C	Other than MB90F543/F549
Storage temperature	T _{STG}	-55	+150	°C	

C.1.2 DOCUMENTATION CONSTRUCTEUR VNB20N07



VNP20N07FI
VNB20N07/VNV20N07

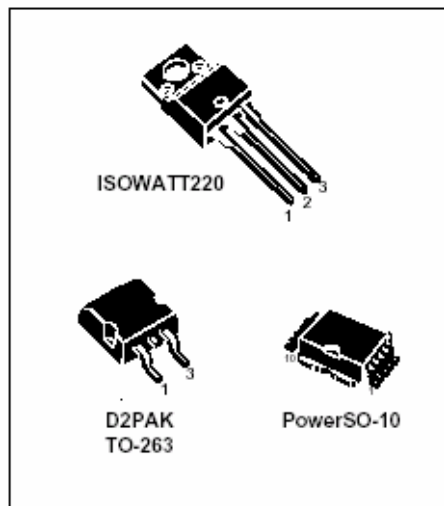
"OMNIFET":
FULLY AUTOPROTECTED POWER MOSFET

TYPE	Vclamp	Rds(on)	Ilim
VNP20N07FI	70 V	0.05 Ω	20 A
VNB20N07	70 V	0.05 Ω	20 A
VNV20N07	70 V	0.05 Ω	20 A

- LINEAR CURRENT LIMITATION
- THERMAL SHUT DOWN
- SHORT CIRCUIT PROTECTION
- INTEGRATED CLAMP
- LOW CURRENT DRAWN FROM INPUT PIN
- DIAGNOSTIC FEEDBACK THROUGH INPUT PIN
- ESD PROTECTION
- DIRECT ACCESS TO THE GATE OF THE POWER MOSFET (ANALOG DRIVING)
- COMPATIBLE WITH STANDARD POWER MOSFET

DESCRIPTION

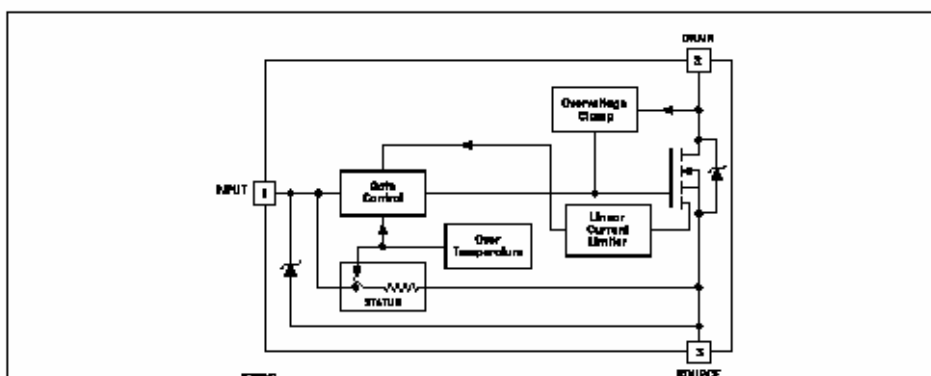
The VNP20N07FI, VNB20N07 and VNV20N07 are monolithic devices made using STMicroelectronics VIPower M0 Technology, intended for replacement of standard power MOSFETs in DC to 50 KHz applications. Built-in thermal shut-down, linear current limitation and overvoltage clamp protect the chip in harsh



environments.

Fault feedback can be detected by monitoring the voltage at the input pin.

BLOCK DIAGRAM (*)



PROTECTION FEATURES

During normal operation, the Input pin is electrically connected to the gate of the internal power MOSFET. The device then behaves like a standard power MOSFET and can be used as a switch from DC to 50 KHz. The only difference from the user's standpoint is that a small DC current (I_{iss}) flows into the Input pin in order to supply the internal circuitry.

The device integrates:

- **OVERVOLTAGE CLAMP PROTECTION:** internally set at 70V, along with the rugged avalanche characteristics of the Power MOSFET stage give this device unrivalled ruggedness and energy handling capability. This feature is mainly important when driving inductive loads.
- **LINEAR CURRENT LIMITER CIRCUIT:** limits the drain current I_d to I_{lim} whatever the Input pin voltage. When the current limiter is active, the device operates in the linear region, so power dissipation may exceed the capability of the heatsink. Both case and junction temperatures increase, and if this phase lasts long enough, junction temperature may reach the overtemperature threshold T_{jsh} .

- **OVERTEMPERATURE AND SHORT CIRCUIT PROTECTION:** these are based on sensing the chip temperature and are not dependent on the input voltage. The location of the sensing element on the chip in the power stage area ensures fast, accurate detection of the junction temperature. Overtemperature cutout occurs at minimum 150°C. The device is automatically restarted when the chip temperature falls below 135°C.

- **STATUS FEEDBACK:** In the case of an overtemperature fault condition, a Status Feedback is provided through the Input pin. The internal protection circuit disconnects the input from the gate and connects it instead to ground via an equivalent resistance of 100 Ω . The failure can be detected by monitoring the voltage at the Input pin, which will be close to ground potential.

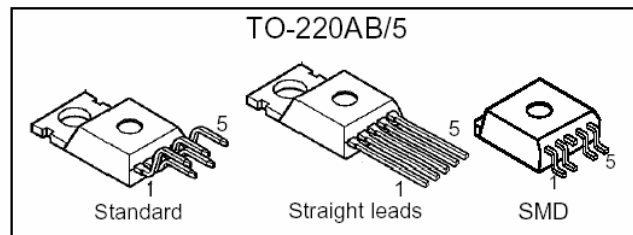
Additional features of this device are ESD protection according to the Human Body model and the ability to be driven from a TTL Logic circuit (with a small increase in $R_{DS(on)}$).

Smart Highside Power Switch**Features**

- Overload protection
- Current limitation
- Short circuit protection
- Thermal shutdown
- Overvoltage protection (including load dump)
- Fast demagnetization of inductive loads
- Reverse battery protection¹⁾
- Undervoltage and overvoltage shutdown with auto-restart and hysteresis
- Open drain diagnostic output
- Open load detection in ON-state
- CMOS compatible input
- Loss of ground and loss of V_{bb} protection
- **Electrostatic discharge (ESD)** protection

Product Summary

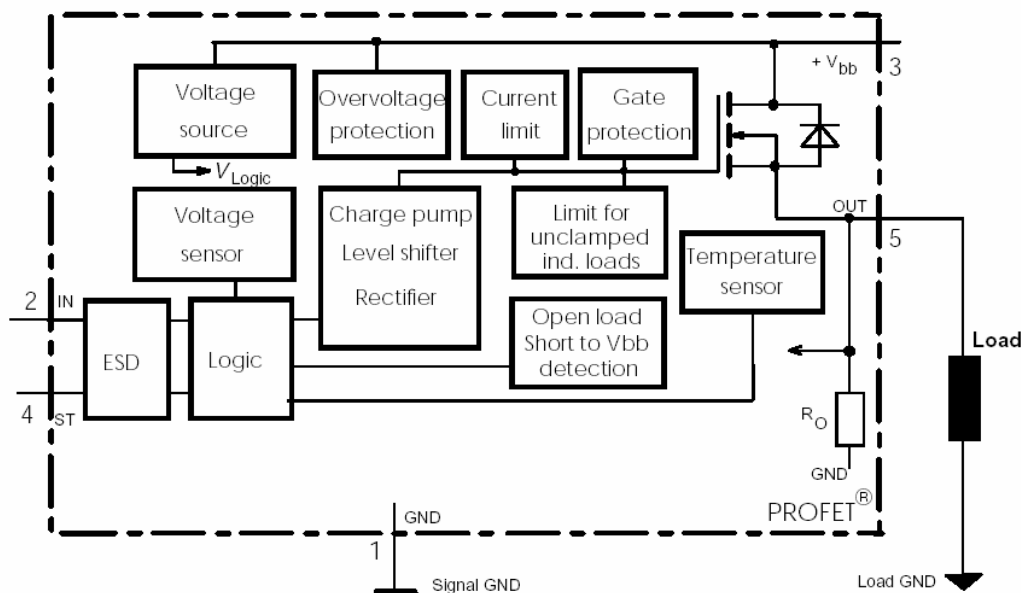
Overvoltage protection	$V_{bb(AZ)}$	43	V
Operating voltage	$V_{bb(on)}$	5.0 ... 34	V
On-state resistance	R_{ON}	60	m Ω
Load current (ISO)	$I_L(ISO)$	7.0	A
Current limitation	$I_L(SCr)$	16	A

**Application**

- μ C compatible power switch with diagnostic feedback for 12 V and 24 V DC grounded loads
- All types of resistive, inductive and capacitive loads
- Replaces electromechanical relays, fuses and discrete circuits

General Description

N channel vertical power FET with charge pump, ground referenced CMOS compatible input and diagnostic feedback, monolithically integrated in Smart SIPMOS® technology. Fully protected by embedded protection functions.



1) With external current limit (e.g. resistor $R_{GND}=150\ \Omega$) in GND connection, resistor in series with ST connection, reverse load current limited by connected load.

V_{out}

Turn-on time with $R_L = 10\ \Omega$ min : 80 μ s ; typ : 200 μ s ; Max : 400 μ s

Turn-off time with $R_L = 10\ \Omega$ min : 80 μ s ; typ : 230 μ s ; Max : 450 μ s

Pin	Symbol		Function
1	GND	-	Logic ground
2	IN	I	Input, activates the power switch in case of logical high signal
3	V _{bb}	+	Positive power supply voltage, the tab is shorted to this pin
4	ST	S	Diagnostic feedback, low on failure
5	OUT (Load, L)	O	Output to the load

Truth Table

	Input- level	Output level	Status 425L1 426L1
Normal operation	L H	L H	H H
Open load	L H	¹²⁾ H	H (L ¹³⁾ L
Short circuit to V _{bb}	L H	H H	L ¹⁴⁾ H (L ¹⁵⁾
Overtem- perature	L H	L L	H L
Under- voltage	L H	L L	H H
Overvoltage	L H	L L	H H

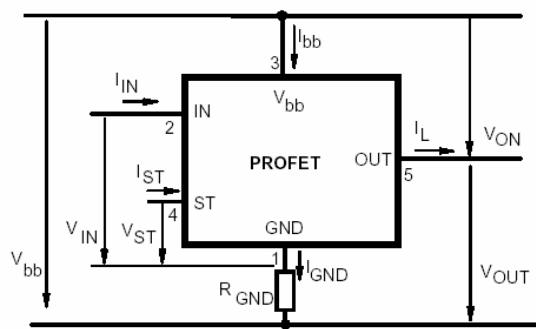
L = "Low" Level
H = "High" Level

X = don't care

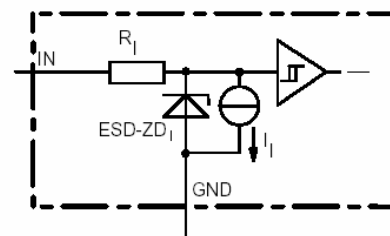
Z = high impedance, potential depends on external circuit

Status signal after the time delay shown in the diagrams (see fig 5. page 11...12)

Terms



Input circuit (ESD protection)



ESD zener diodes are not to be used as voltage clamp at DC conditions. Operation in this mode may result in a drift of the zener voltage (increase of up to 1 V).

12) Power Transistor off, high impedance

13) with external resistor between pin 3 and pin 5

14) An external short of output to V_{bb}, in the off state, causes an internal current from output to ground. If R_{GND} is used, an offset voltage at the GND and ST pins will occur and the V_{ST low} signal may be errorious.

15) Low resistance to V_{bb} may be detected in ON-state by the no-load-detection

8-channel analog multiplexer/demultiplexer

74HC/HCT4051

FEATURES

- Wide analog input voltage range: ± 5 V.
- Low "ON" resistance:
80 Ω (typ.) at $V_{CC} - V_{EE} = 4.5$ V
70 Ω (typ.) at $V_{CC} - V_{EE} = 6.0$ V
60 Ω (typ.) at $V_{CC} - V_{EE} = 9.0$ V
- Logic level translation:
to enable 5 V logic to communicate with ± 5 V analog signals
- Typical "break before make" built in
- Output capability: non-standard
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT4051 are high-speed Si-gate CMOS devices and are pin compatible with the "4051" of the

"4000B" series. They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT4051 are 8-channel analog multiplexers/demultiplexers with three digital select inputs (S_0 to S_2), an active LOW enable input ($\overline{E}$), eight independent inputs/outputs (Y_0 to Y_7) and a common input/output (Z).

With $\overline{E}$ LOW, one of the eight switches is selected (low impedance ON-state) by S_0 to S_2 . With $\overline{E}$ HIGH, all switches are in the high impedance OFF-state, independent of S_0 to S_2 .

V_{CC} and GND are the supply voltage pins for the digital control inputs (S_0 to S_2 , and $\overline{E}$). The V_{CC} to GND ranges are 2.0 to 10.0 V for HC and 4.5 to 5.5 V for HCT. The analog inputs/outputs (Y_0 to Y_7 , and Z) can swing between V_{CC} as a positive limit and V_{EE} as a negative limit. $V_{CC} - V_{EE}$ may not exceed 10.0 V.

For operation as a digital multiplexer/demultiplexer, V_{EE} is connected to GND (typically ground).

QUICK REFERENCE DATA

$V_{EE} = \text{GND} = 0$ V; $T_{\text{amb}} = 25$ °C; $t_r = t_f = 6$ ns

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t _{PZH} / t _{PZL}	turn "ON" time E̅ to V _{OS} S _n to V _{OS}	C _L = 15 pF; R _L = 1 kΩ; V _{CC} = 5 V	22	22	ns
			20	24	ns
t _{PHZ} / t _{PLZ}	turn "OFF" time E̅ to V _{OS} S _n to V _{OS}		18	16	ns
			19	20	ns
C _I	input capacitance		3.5	3.5	pF
C _{PD}	power dissipation capacitance per switch	notes 1 and 2	25	25	pF
C _S	max. switch capacitance independent (Y) common (Z)		5	5	pF
			25	25	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in μ W):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum \{ (C_L + C_S) \times V_{CC}^2 \times f_o \} \text{ where:}$$

f_i = input frequency in MHz

f_o = output frequency in MHz

$\sum \{ (C_L + C_S) \times V_{CC}^2 \times f_o \}$ = sum of outputs

C_L = output load capacitance in pF

C_S = max. switch capacitance in pF

V_{CC} = supply voltage in V

2. For HC the condition is $V_I = \text{GND}$ to V_{CC}
For HCT the condition is $V_I = \text{GND}$ to $V_{CC} - 1.5$ V

C.1.5 DOCUMENTATION CONTRÔLEUR CAN

Philips Semiconductors

PCA82C250 / 251 CAN Transceiver

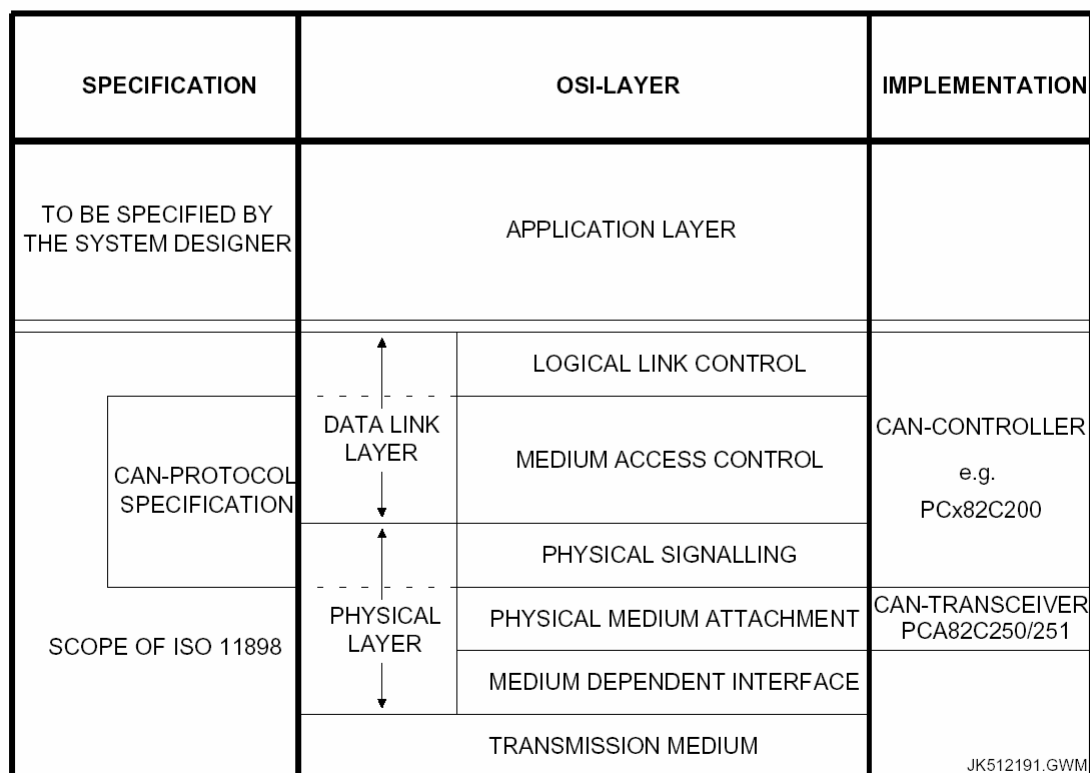
Application Note AN96116

1. INTRODUCTION

ISO 11898 [3] is the international standard for in-vehicle high-speed communication using the Controller Area Network (CAN) bus protocol. The scope of this standard essentially is to specify the so-called data link layer and physical layer of the communication link. The physical layer is subdivided into three sublayers as shown in Fig. 1. These are

- Physical Signalling bit coding, timing and synchronization
- Physical Medium Attachment driver and receiver characteristics
- Medium Dependent Interface bus connector

This report focuses on the implementation of the Physical Medium Attachment sublayer using the transceivers PCA82C250 [1] and PCA82C251 [2] from Philips Semiconductors. The implementation of the Physical Signalling sublayer and the Data Link Layer is typically performed by integrated protocol controller products, like the PCx82C200 from Philips Semiconductors. Connection to the transmission medium is provided via the Medium Dependent Interface i.e. a connector used to attach bus nodes to the bus line.



Note: OSI = Open Systems Interconnection (see ISO 7498)

Fig. 1 Layered architecture of CAN

2. APPLICATION OF THE PCA82C250 AND PCA82C251

The PCA82C250/251 transceiver products basically provide interfacing between a protocol controller and a physical transmission line. They are designed to transmit data with a bit rate of up to 1 Mbit/s over a two-wire differential voltage bus line as described in the ISO 11898 standard. Their general features are listed in the data sheets (see [1] and [2]).

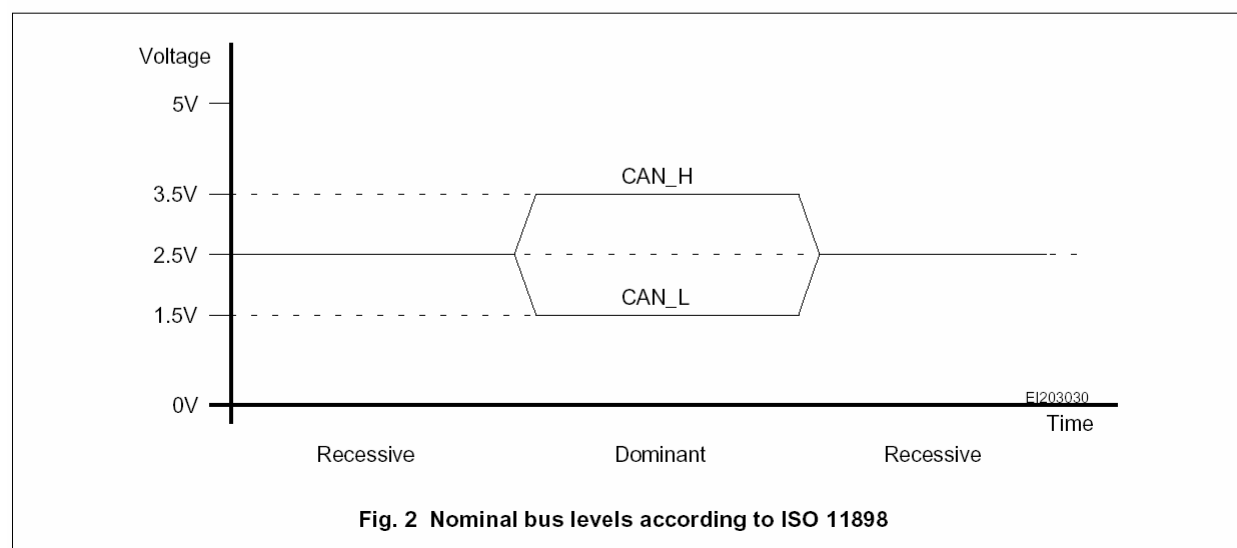
Both devices are designed for the use in CAN bus systems with a nominal supply voltage of 12 V (PCA82C250) and 24 V (PCA82C251) respectively. They are functionally identical and can be used in automotive and general industrial applications according to the relevant standards e.g. the ISO 11898 standard [3] and the DeviceNet™ Specification [5]. Both the PCA82C250 and the PCA82C251 can communicate to one another in one network. Moreover they are pin- & function-compatible i.e. they can be used with identical printed circuit boards.

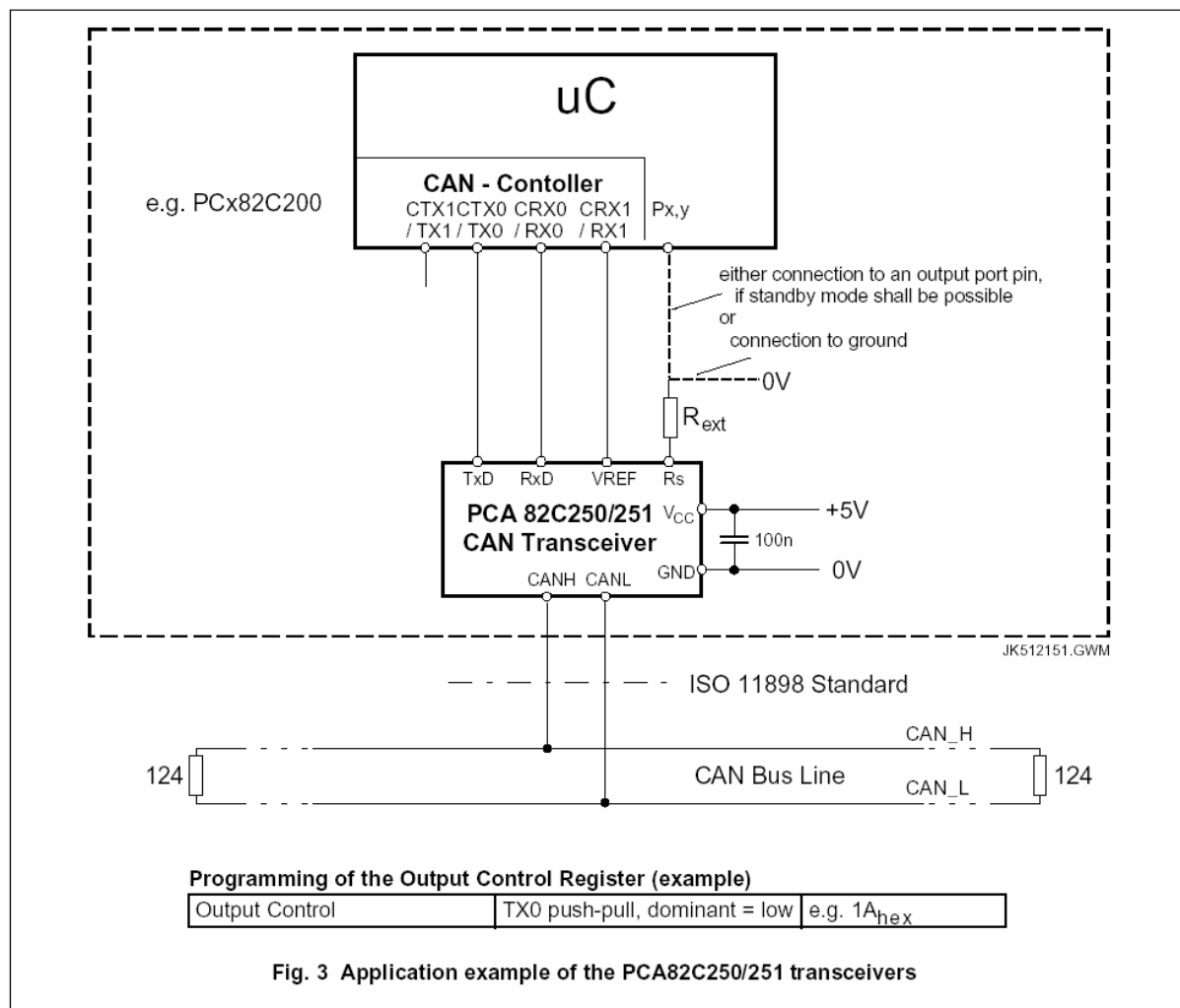
Some main differences between both products are listed in Table 1.

Table 1 Main differences between PCA82C250 and PCA82C251

	PCA82C250	PCA82C251
Nominal system supply voltage	12 V	12V and/or 24 V
Maximum bus terminal DC voltage ($0\text{ V} < V_{CC} < 5.5\text{ V}$)	$-8\text{ V} < V_{CANL,H} < +18\text{ V}$	$-40\text{ V} < V_{CANL,H} < +40\text{ V}$
Maximum transient bus terminal voltage (ISO 7637)	$-150\text{ V} < V_{tr} < +100\text{ V}$	$-200\text{ V} < V_{tr} < +200\text{ V}$
Minimum transceiver supply voltage for extended fan out applications ($R_L = 45\ \Omega$)	$V_{CC} > 4.9\text{ V}$	$V_{CC} > 4.5\text{ V}$

For general industrial applications the PCA82C251 is recommended to be employed as to e.g. its higher break-down voltage and its capability to drive loads down to $45\ \Omega$ over the whole supply voltage range. Also the PCA82C251 draws less supply current in the recessive state and provides an enhanced bus output behaviour in power-fail situations.

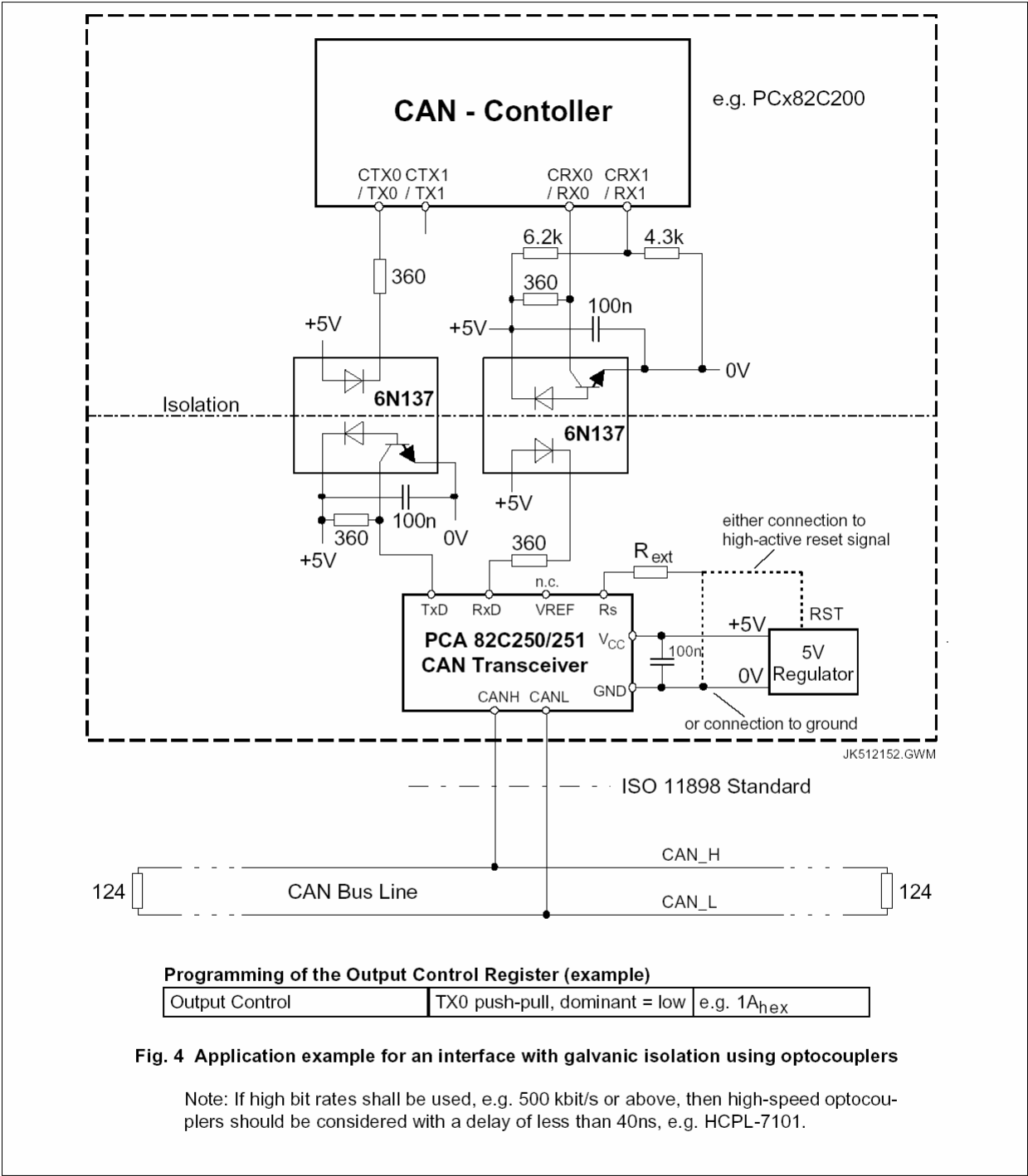




2.1 Application Examples

A typical application of the PCA82C250/251 transceiver is shown in Fig. 3. A protocol controller is connected to the transceiver via a serial data output line (TX) and a serial data input line (RX). The transceiver is attached to the bus line via its two bus terminals CANH and CANL, which provide differential receive and transmit capability. The input Rs is used for mode control purpose. The reference voltage output V_{REF} provides an output voltage of $0.5 \times V_{CC}$ nominal. Both transceiver products are powered with a nominal supply voltage of +5 V.

The protocol controller outputs a serial transmit data stream to the TxD input of the transceiver. An internal pull-up function sets the TxD input to logic HIGH i.e. the bus output driver is passive by default. In this so-called recessive state (see Fig. 2) the CANH and CANL inputs are biased to a voltage level of 2.5 V nominal via receiver input networks with an internal impedance of 17 kΩ typical. Otherwise if a logic LOW-level is applied to TxD, this activates the bus output stage, thus generating a so-called dominant signal level on the bus line (see Fig. 2). The output driver consists of a source and a sink output stage. CANH is attached to the source output and CANL to the sink output stage. The nominal voltage in the dominant state is 3.5 V for the CAN_H line and 1.5 V for the CAN_L line.



The bus line is in recessive state if no bus node transmits a dominant bit, i.e. all TxD inputs in the network are logic HIGH. Otherwise if one or multiple bus nodes transmit a dominant bit, i.e. at least one TxD input is logic LOW, then the bus line enters the dominant state thus overriding the recessive state (wired-AND characteristic). For general industrial applications the PCA82C251 is recommended to be employed as to e.g. its higher break-down voltage and its capability to drive loads down to 45 Ω over the whole supply voltage range. Also the PCA82C251 draws less supply current in the recessive state and provides an enhanced bus output behaviour in power-fail situations.

The receiver comparator converts the differential bus signal to a logic level signal which is output at RxD. The serial receive data stream is provided to the bus protocol controller for decoding. The receiver comparator is always active i.e. it monitors the bus while the bus node is transmitting a message. This is required e.g. for safety reasons and to support the non-destructive bit by bit contention scheme of CAN. Some controller products provide an analog receive interface (RX0, RX1). In that case RX0 usually needs to be connected to the RxD output and RX1 needs to be biased to an appropriate voltage level. This can be done e.g. by using the V_{REF} output (see Fig. 3) or by using a resistive voltage divider (see Fig. 4).

In Fig. 3 the transceiver is directly connected to the protocol controller and its application circuitry. In cases where galvanic isolation is desired, optocouplers can be placed e.g. between the transceiver and the protocol controller as shown in Fig. 4. When using optocouplers one has to pay attention to choose the right default state when the circuitry at the protocol controller side of the isolation is not powered. In such a case the optocoupler being attached to TxD will be "dark" i.e. LED switched off. When this optocoupler is off/dark, then a logic HIGH-level has to be output to the TxD input of the transceiver for fail-safe purpose. Also if using optocouplers one may consider to attach the Rs mode control input to an active-high reset signal, e.g. to disable the transceiver when the local transceiver supply voltage is not OK e.g. during ramp-up and -down.

However using optocouplers generally increases the so-called loop delay of a bus node, if placed between the transceiver and the protocol controller. The signal has to pass these devices twice per node, i.e. transmit and receive path, which effectively decreases the maximum achievable bus length at a given bit rate. This fact has to be considered when calculating the maximum achievable bus length due to propagation delays in a CAN network. For more details please refer e.g. to [4].

2.2 Reference Voltage Output

The PCA82C250/251 provides a reference voltage output V_{REF} , which may be used e.g. to bias one of the inputs of a CAN protocol controller's differential input comparator as shown in Fig. 3. In other cases a reference voltage may be generated locally at the protocol controller input as shown e.g. in Fig. 4. Which solution is appropriate in a system depends on the application and the bus input structure of the protocol controller product.

3. OPERATION MODES

The PCA82C250 and PCA82C251 provide three different operation modes. Mode control is being provided through the Rs control input.

The first mode is the high-speed mode supporting maximum bus speed and/or length.

The second mode is the so-called slope control mode which should be considered if unshielded bus wires shall be used. In this mode the output slew rate can be decreased intentionally, e.g. to reduce electromagnetic emission.

The third mode is the stand-by mode being of interest especially in battery powered applications, when the system power consumption needs to be very low. System reactivation is performed through transmission of a message. Fig. 3 gives an example for switching the transceiver between stand-by mode and normal operating mode.

- $P_{x,y} = \text{HIGH}$: the PCA82C250/251 is switched to stand-by mode ($V_{RS} > 0.75 \times V_{CC}$)
- $P_{x,y} = \text{LOW}$: the PCA82C250/251 is switched to normal operating mode, which is either high-speed mode or slope control mode, depending essentially upon the resistance connected to Rs.

Usually the following resistance values for the slope-control resistor R_{ext} are suitable:

- $0 \Omega < R_{ext} < 1.8 \text{ k}\Omega$ high-speed mode ($V_{RS} < 0.3 \times V_{CC}$)
- $16.5 \text{ k}\Omega < R_{ext} < 140 \text{ k}\Omega$ slope control mode ($10 \mu\text{A} < -I_{RS} < 200 \mu\text{A}$)

In the following these three operation modes shall be discussed in more detail.

3.1 High-Speed Mode

This mode is suitable to achieve a maximum bit rate and/or bus length. The high-speed mode is commonly employed in general industrial applications such as the CAN based system DeviceNet™. In this mode the bus output signals are switched as fast as possible and therefore a shielded bus cable usually would be appropriate to prevent a possible disturbance of e.g. a car radio by the bus signal.

The high-speed mode is selected with $V_{RS} < 0.3 \times V_{CC}$. This can be achieved with a direct connection of the Rs control input to an output port of a microcontroller or ground potential or an active-high reset signal (see Fig. 3 and Fig. 4).

In high-speed mode the transceivers provide an effective loop delay of as low as 145 ns max. (155 ns for $T_{amb} > 85^\circ\text{C}$). With view to the CAN bit timing requirements, the effective loop delay is the maximum of the dominant edge loop delay and the average value of dominant and recessive edge loop delay.

$$t_{loop,eff} = \max\{0.5 \times (t_{onRxD} + t_{offRxD}), t_{onRxD}\}$$

3.2 Slope Control Mode

In several applications the use of an unshielded bus cable will be desirable e.g. for system cost reasons. However using an unshielded cable implies additional requirements to be met by the transceiver product e.g. with view to electromagnetic compatibility (EMC). Using the PCA82C250/251 the slew rate of the bus signal can be decreased intentionally, which is recommended if an unshielded bus cable shall be used. The slew rate can be set via a series resistance value R_{ext} being connected to the control pin Rs. With respect to the CAN bit timing requirements a decreased slew rate implies an increase of the bus node loop delay and thus a lower bus length at a given bit rate or alternatively a lower bit rate at a given bus length. In slope control mode the bus output slew rate is basically proportional to the current flow out of pin Rs in the range of $10 \mu\text{A} < -I_{RS} < 200 \mu\text{A}$ (see data sheets [1], [2]). If the Rs output current is in that range, then a voltage of approximately $0.5 \times V_{CC}$ will be output at the pin Rs. The transceiver is set to slope control mode when an appropriate resistance value is applied between the Rs pin and ground potential. As a rule of thumb the resistance value should be in the range of $16.5 \text{ k}\Omega < R_{ext} < 140 \text{ k}\Omega$ to meet the above-mentioned range for the Rs output current.

The suitable range for R_{ext} can be calculated using the limits for slope control mode

$$10 \mu\text{A} < -I_{\text{Rs}} < 200 \mu\text{A} \quad \text{and} \quad 0.4 \times V_{\text{CC}} < V_{\text{Rs}} < 0.6 \times V_{\text{CC}}.$$

The minimum and maximum values for R_{ext} can be calculated with the following two relations (for the explanation of the symbols see APPENDIX 1).

$$R_{\text{ext}} > \frac{0.6V_{\text{CC,max}}}{I_{\text{Rs,max}}} = \frac{0.6V_{\text{CC,max}}}{200\mu\text{A}} \quad (1)$$

$$R_{\text{ext}} < \frac{0.4V_{\text{CC,min}} - V_{\text{OL,max}}}{I_{\text{Rs,min}}} = \frac{0.4V_{\text{CC,min}} - V_{\text{OL,max}}}{10\mu\text{A}} \quad (2)$$

If the slope control resistor R_{ext} is connected to ground, then the value $V_{\text{OL,max}}$ is zero volts. The relation between the R_{s} output current and the bus signal slew rate is discussed in chapter 4.1.

3.3 Stand-by Mode

This mode is to be used when the power consumption needs to be minimized e.g. temporarily. The stand-by mode is selected with $V_{\text{Rs}} > 0.75 \times V_{\text{CC}}$.

Using the stand-by mode, the system power consumption can be reduced drastically. This mode is primarily intended for battery powered applications for example when a vehicle is parked. To enter stand-by mode a logic HIGH-level has to be applied to the transceiver's control input R_{s} . This can be done either by direct connection of an output port pin to R_{s} or via any suitable slope control resistor R_{ext} . In stand-by mode the transmitter function and the receiver input bias network are switched off to reduce power consumption. The reference voltage output and a basic receive function will remain active and work with very low power consumption. This allows to reactivate the system via the bus line by transmission of a message. Upon detection of a dominant bus condition of at least 3 μs length, the transceiver will provide a wake-up interrupt signal to the protocol controller via its $R_{\text{x}}\text{D}$ output. Upon detection of a falling edge on $R_{\text{x}}\text{D}$ the controller should set the R_{s} pin to logic LOW-level in order to switch the transceiver back to normal transmission mode. As the receiver is slower in stand-by mode, it essentially depends on the delay time of the logic (falling edge on R_{s}) when the transceiver is back to normal reception speed. At high bus speeds the transceiver may not be able to correctly receive messages in stand-by mode i.e. while the R_{s} pin is still HIGH.

An alternative application is to connect the R_{s} input to an active-high reset signal. This can be done for example with view to the case of the transceiver and the protocol controller being supplied by different supply sources, e.g. if optocouplers are used (see Fig. 4).

4. SLOPE CONTROL FUNCTION

[...]

5. MAXIMUM BUS LINE LENGTH

The maximum achievable bus line length in a CAN bus network is determined essentially by the following physical effects:

1. The loop delays of the connected bus nodes (CAN controller, transceiver etc.) and the delay of the bus line
2. The differences in bit time quantum length due to the relative oscillator tolerance between nodes
3. The signal amplitude drop due to the series resistance of the bus cable and the input resistance of bus nodes

The effect 3. is discussed below.

The effects 1. and 2. are not discussed in this document (please refer e.g. to [4]). However as a rule of thumb the following bus line length can be achieved with the PCA82C250 and PCA82C251 in high-speed mode and with CAN bit timing parameters being optimized for maximum propagation delay:

Table 3 Bit Rate / Bus Length Relation

Bit Rate (kbit/s)	Bus Length (m)
1000	30
500	100
250	250
125	500
62.5	1000

5.1 Impact of the Bus Cable Resistance

The ISO 11898 Standard [3] assumes the network wiring topology to be close to a single line structure in order to minimize reflection effects on the bus line (Fig. 6).

At static conditions the differential input voltage at a bus node is determined by the current flowing through the differential input resistance of that node. In case of a dominant bit the output transistors of the transmitting node are switched on, causing a current flow, whereas the transistors are switched off for a recessive bit.

Thus the generated differential voltage at the input of a node ($V_{\text{diff.in}}$) depends on (see Fig. 7)

- The differential output voltage of the transmitting nodes ($V_{\text{diff.out}}$)
- The resistance of the bus cable ($R_W = \rho \times L$) with
 ρ = specific resistance per length unit and
 L = length of the bus line
- The differential input resistance of receiving nodes (R_{diff})

The worst case situation is given for one transmitting node at one end of the bus wire and a receiving node at the other end.

For this case the differential input voltage at the receiving node is calculated using Fig. 7 (see APPENDIX 2).

$$V_{\text{diff.in}} = \frac{V_{\text{diff.out}}}{1 + 2R_W \times \left(\frac{1}{R_T} + \frac{n-1}{R_{\text{diff}}} \right)} \quad (8)$$

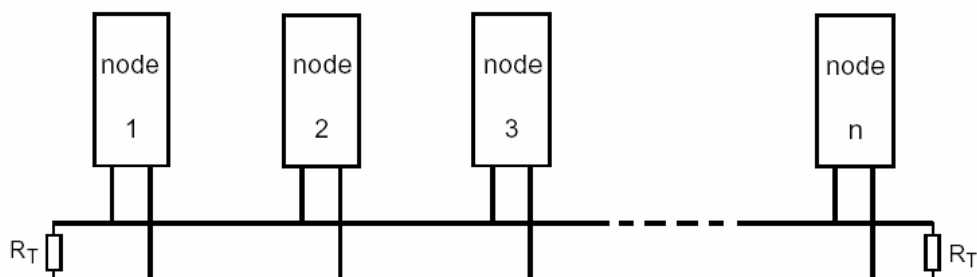


Fig. 6 Basic setup of a bus system (ISO 11898)

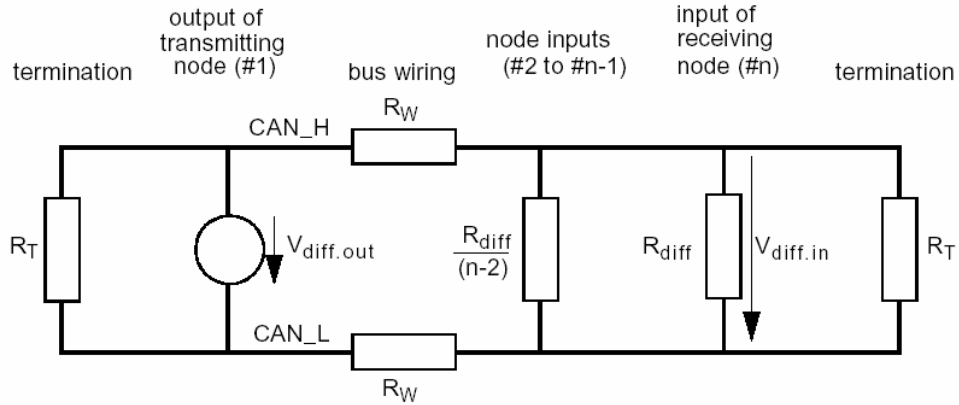


Fig. 7 Circuit diagram for the system setup of Fig. 6

A receiver recognizes a

recessive bit if the differential input voltage is below a level of 0.5V or 0.4V (see [1] and [2])

dominant bit if the differential input voltage is above a level of 0.9V or 1.0V (see [1] and [2])

The recessive level is generated by the bias network of the bus nodes and the termination resistors. The dominant level is determined by the drive capability of the transmitting node and the total network load resistance. Thus for proper detection of a dominant bit, a differential input voltage at the receiving node is requested ($V_{diff.in.req}$), which is given by the dominant threshold voltage of the receiver (V_{th}) and a user-defined safety margin. This safety margin can be considered as a fraction (k_{sm}) of the difference between the output level at the transmitting node and the receiver input threshold for detection of a dominant bit as shown in equation (9).

$$V_{diff.in.req} = V_{th} + k_{sm} \times (V_{diff.out} - V_{th}) \quad \text{with } k_{sm} = 0 \dots 1 \quad (9)$$

From equation (8) it is evident that the value of $V_{diff.in}$ for a dominant level is restricted by

- the minimum value of the differential output voltage for a dominant level ($V_{diff.out.min}$)
- the maximum value of the bus wire resistance ($R_{W.max}$)
- the minimum value of the termination resistors ($R_{T.min}$)
- the minimum differential input resistance of the nodes ($R_{diff.min}$)
- the maximum number of connected bus nodes (n_{max}).

This leads to the following relation

$$V_{diff.in.min} = \frac{V_{diff.out.min}}{1 + 2R_{W.max} \times \left(\frac{1}{R_{T.min}} + \frac{n_{max}-1}{R_{diff.min}} \right)} \geq V_{diff.in.req} \quad (10)$$

Equations (10) and (9) are the basis for calculating the maximum bus line length (see APPENDIX 3) dependent on

- the maximum number of nodes in a system (n_{max})
- the desired safety margin for detecting a dominant bit (k_{sm})
- the maximum specific resistance per length unit (cross section) of the used cable (ρ_{max}).

$$L_{max} \leq \frac{1}{2 \times \rho_{max}} \times \left(\frac{V_{diff.out.min}}{V_{th.max} + k_{sm} \times (V_{diff.out.min} - V_{th.max})} - 1 \right) \times \frac{R_{T.min} \times R_{diff.min}}{R_{diff.min} + (n_{max}-1) \times R_{T.min}} \quad (11)$$

Using this equation the maximum bus line length for different wire types and a different number of connected nodes can be calculated. Some examples are given in Table 6.

5.2 Maximum Number of Nodes

The transceivers PCA82C250 and PCA82C251 provide an output drive capability down to a minimum load of $R_{L.min} = 45 \Omega$. If the PCA82C250 is used, a supply voltage of $V_{CC} > 4.9 \text{ V}$ is needed for driving a load of $R_L = 45 \Omega$ (see Table 1). The number of nodes which can be connected to a network depends e.g. on the minimum load resistance a transceiver is able to drive. This maximum number of nodes can be calculated using the circuit diagram of Fig. 7. For worst case consideration the bus line resistance R_W is considered to be zero.

This leads to the following relations for calculating the maximum number of nodes:

$$\frac{R_{T.min} \times R_{diff.min}}{(n_{max} - 1) \times R_{T.min} + 2R_{diff.min}} > R_{L.min} \implies n_{max} < 1 + R_{diff.min} \times \left(\frac{1}{R_{L.min}} - \frac{2}{R_{T.min}} \right)$$

As the minimum differential input resistance of the PCA82C250/251 transceivers is $R_{diff.min} = 20 \text{ k}\Omega$, the following maximum number of bus nodes can be connected:

106 nodes for $R_T = 118 \Omega$ and $R_L = 45 \Omega$; ($V_{CC} > 4.9 \text{ V}$ if 82C250 is used)

112 nodes for $R_T = 120 \Omega$ and $R_L = 45 \Omega$; ($V_{CC} > 4.9 \text{ V}$ if 82C250 is used)

5.3 Examples

Table 4 provides a first indication on which kind of wire cross section should be considered for the signal pair of the bus trunk cable.

Table 4 Minimum recommended bus wire cross-section for the trunk cable¹

Bus Length / Number of Nodes	32	64	100
100 m	0.25 mm ² or AWG 24	0.25 mm ² or AWG 24	0.25 mm ² or AWG 24
250 m	0.34 mm ² or AWG 22	0.5 mm ² or AWG 20	0.5 mm ² or AWG 20
500 m	0.75 mm ² or AWG 18	0.75 mm ² or AWG 18	1.0 mm ² or AWG 18

1. Assumptions for Table 4: 32 nodes : $R_W < 21 \Omega$
64 nodes : $R_W < 18.5 \Omega$
100 nodes: $R_W < 16 \Omega$

For the drop cables a wire cross section of 0.25 to 0.34 mm² (or AWG 24, AWG 22) would be an appropriate choice in many cases.

Based on the discussion in chapter 5.1 and chapter 5.2, the following examples list the maximum achievable bus line length, calculated for bus cables being specified in the ISO 11898 standard [3] and in the Device NetTM specification [5]. The specific cable resistance for the cables used are given in Table 5.

Table 5 Specific resistance of different cables (1 km = 3280.84 ft., 1 ft. = 0.3048 m)

Cable type	Specific cable resistance	
	$\rho_{nom} [\Omega/\text{km}]$	$\rho_{max} [\Omega/\text{km}]$
ISO 11898 (automotive): 0.25 mm ² (or AWG23)	70	90 ¹
Device Net TM thin cable	69	92
Device Net TM thick cable	18	23
0.5 mm ² (or AWG20)	37	50 ¹
0.75 mm ² (or AWG18)	26	33 ¹

1. Assumed value

With the known values for

the minimum dominant value	: $V_{\text{diff.out.min}} = 1.5 \text{ V}$	see [1] and [2]
the minimum differential input resistance	: $R_{\text{diff.min}} = 20 \text{ k}\Omega$	see [1] and [2]
the requested differential input voltage	: $V_{\text{th.max}} = 0.9 \text{ V or } 1.0 \text{ V}$	see [1] and [2]
the minimum termination resistance of	: $R_{\text{T.min}} = 118 \text{ }\Omega$	see [3]

the maximum wiring length is calculated for different bus cable types and a different number of connected bus nodes using equation (11) on page 19. The result is listed in Table 6.

Table 6 Maximum bus cable length for different cables and number of nodes (n)

Cable Type	$L_{\text{max}} (k_{\text{sm}} = 0,2)^1$			$L_{\text{max}} (k_{\text{sm}} = 0,1)^2$		
	n = 32	n = 64	n = 100	n = 32	n = 64	n = 100
DeviceNet™ (thin cable) and/or ISO 11898 cable	200 m	170 m	150 m	230 m	200 m	170 m
DeviceNet™ (thick cable)	800 m	690 m	600 m	940 m	810 m	700 m
0.5 mm ² (or AWG 20)	360 m	310 m	270 m	420 m	360 m	320 m
0.75 mm ² (or AWG 18)	550 m	470 m	410 m	640 m	550 m	480 m

1. Calculated with $V_{\text{th.max}} = 1.0 \text{ V}$ and a safety margin of $k_{\text{sm}} = 0.2$

2. Calculated with $V_{\text{th.max}} = 1.0 \text{ V}$ and a safety margin of $k_{\text{sm}} = 0.1$

Note: If driving more than 64 bus nodes and/or more than 250 m bus length the accuracy of the V_{CC} supply voltage for the PCA82C251 is recommended to be 5% or better. The PCA82C250 needs a supply voltage of at least 4.75V when driving 50 Ω load, i.e. 64 bus nodes, and at least 4.9V when driving 45 Ω load, i.e. 100 bus nodes.

6. BUS TERMINATION AND TOPOLOGY ASPECTS

Generally the CAN high-speed standard ISO 11898 provides a single line structure as network topology. The bus line is terminated at both ends with a single termination resistor. However in practice some deviation from that topology may be needed to accommodate appropriate drop cable length of e.g. a few meters. Also a modified termination network may be desirable in some applications e.g. for EMC related considerations. In this chapter some modified bus termination concepts as well as topology aspects shall be discussed.

6.1 Split Termination Concept

This is an option intended to provide enhanced EMC characteristics without changing the DC characteristics of the terminated line. Basically each of the termination resistors is split into two resistors of equal value, i.e. two resistors of 62 Ω instead of one resistor of 124 Ω (see Fig. 8). The special characteristic of this approach is, that the so-called common-mode signal is available at the centre tap of the termination. As the common-mode signal is simply a DC voltage in the ideal case, this centre tap can be grounded via a capacitor of e.g. 10 nF to 100 nF. However it is obvious, that the capacitor should be connected to a "quiet" ground level. For example a separate ground lead to the connector's ground pin is recommended, if termination is placed inside of bus nodes.

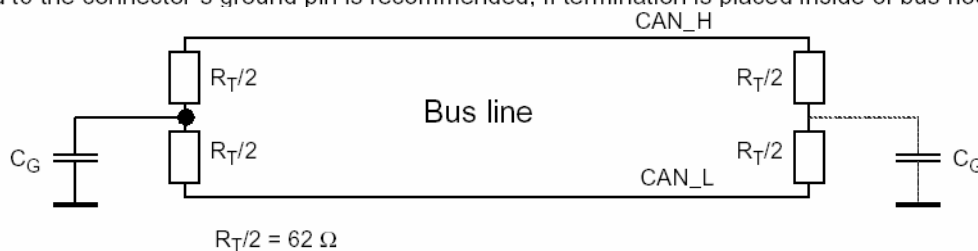


Fig. 8 Split termination concept

Basically there are two options with different advantages and disadvantages. The first option includes both termination resistors to be split and grounded. This is the preferred approach to optimize the characteristic in the higher frequency range. However there is a chance that there are unwanted loop currents via ground potential, as both termination resistors are grounded. In that case one may consider to ground only one of the termination resistors. This can provide a better characteristic in the medium to low frequency range. As mentioned above, the DC characteristics of the terminated bus line is not changed.

6.2 Multiple Termination Concept

This concept can be used in combination with the split termination concept above and targets at network topologies which differ from a single line structure.

In some applications a topology different from a single line structure is needed, e.g. a star topology with three branches (see Fig. 9). To accommodate such a topology, the multiple termination concept may be considered. Essentially this approach suggests, that the total termination resistance, i.e. $62\ \Omega$, is being distributed over more than two resistors. If for example a star topology is needed with three branches, then one may consider to terminate each branch with about three times the total termination resistance, i.e. $180\ \Omega$. With this approach it is essential that the total termination resistance (i.e. all termination resistors in parallel) does suit the transceiver's output drive capability. If one of the branches is optional, e.g. for temporary attachment of diagnostic equipment, then the trunk line would be terminated via two resistors of $180\ \Omega$ and the optional branch would be terminated via another termination resistor of $180\ \Omega$. It is obvious that this concept implies some mismatch between characteristic line impedance and termination resistance. However this is not considered to be critical, provided there is a sufficient safety margin left with view to the CAN bit timing parameters.

As a rule of thumb the total bus length including all branches shall be less than the suitable bus length for the single line structure in a given configuration. For example instead of a single line structure of 100 m length a 3-branch star topology may be considered with each branch being terminated with $180\ \Omega$ and a drop length of less than 33 m each. The basic network is recommended to be terminated with at least 50% of the nominal termination resistance i.e. when all optional parts are disconnected, the remaining "basic" termination resistance is recommended to be less than $120\ \Omega$ (e.g. $2 \times 180\ \Omega$ or $3 \times 240\ \Omega$, etc.).

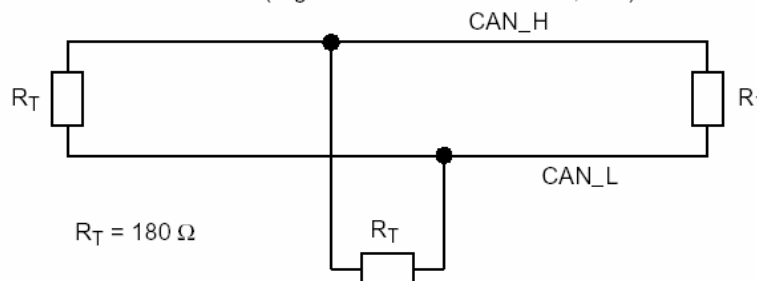


Fig. 9 Multiple termination concept (example)

7. CONCLUSION

The PCA82C250 and PCA82C251 are advanced transceiver products being suitable for usage in automotive as well as general industrial applications with bit rates up to 1 Mbit/s. They support a differential bus signal representation as described in the international standard for in-vehicle high-speed applications (ISO 11898) using the Controller Area Network (CAN) protocol.

Enhanced electromagnetic compatibility (EMC) performance is provided through an extended common-mode range of -7V to +12V and the slope-control function, where the slew rate of the bus signal can be adjusted via a resistance value. For battery powered applications a stand-by mode is provided to drastically reduce power consumption of the network, e.g. when a vehicle is parked. In stand-by mode the network is being activated via the bus lines upon detection of a message.

The PCA82C250 and PCA82C251 are proof against short-circuit conditions on the bus outputs and usual transients in an automotive environment (ISO 7637). Moreover a thermal shutdown function protects the devices against thermal overload e.g. due to short-circuit conditions. Both products are designed for connection to the protocol controller or bus line with a minimum number of external components.

Also both products are capable of driving a large number of bus nodes i.e. 64 to 100 per network, and bus length of up to about 0.5 to 1 km, which is advantageous primarily in general industrial applications such as the CAN based system DeviceNet™.

The PCA82C250 and PCA82C251 are pin- and function compatible and operate in a wide supply voltage range of $5\text{ V} \pm 10\%$. For general industrial applications the PCA82C251 should be used because of e.g. its larger drive capability and higher breakdown voltage protection at the bus outputs.

The advanced functionality being described above makes the PCA82C250 and PCA82C251 an attractive choice in many automotive and general industrial applications.

APPENDIX 1 ABBREVIATIONS AND DEFINITIONS

SR	single-ended slew rate of the signal transition at CANL and/or CANH
kSE	single-ended slew rate constant (in slope control mode the slew rate is proportional to the current out of pin Rs of the transceiver)
IRs, IRs.min, IRs.max	current (minimum, maximum) at pin Rs of the transceiver
VRs	typical voltage at pin Rs of the transceiver
Px,y	digital output port pin of a controller IC for transceiver mode control
VOL, VOL.max	nominal (maximum) LOW-level output voltage of the controller output port
Rext, Rext.min, Rext.max	nominal (minimum, maximum) value of the slope-control resistor at pin Rs of the transceiver. This value determines the current and thus the slew rate at CANL and CANH
VCC, VCC.min, VCC.max	nominal (minimum, maximum) value of the transceiver supply voltage
tprop	effective transceiver (other components) loop delay
tp	available two-way propagation delay (limited by CAN bit timing parameters)
tonRxD, toffRxD	specific line delay per length unit (e.g. 5 ns/m)
Vdif f.in, Vdiff.in.min, Vdiff.in.req	loop delay of the transceiver between pin TxD (transmit data input) and RxD (receive data output) at switching from recessive to dominant (dominant to recessive) state (see [1] and [2])
Vdif f.out, Vdiff.out.min	nominal (minimum, requested) differential input voltage for reception
Vth, Vth.max	nominal (minimum) differential output voltage at the transmitting node
ksm	nominal (maximum) differential input threshold voltage for detection of a dominant bus condition
Rdiff, Rdiff.min	a factor indicating the safety margin for the differential input voltage for detecting a dominant bit at reception ($0 < \text{ksm} < 1$)
n, nmax	nominal (minimum) differential input resistance of a bus node in recessive state (TxD = HIGH)
RT, RT.min	number (maximum number) of bus nodes in the network
RW, RW.max	nominal (minimum) value for the bus termination resistors
L, Lmax	nominal (maximum) series resistance of the bus wires
Lu	length (maximum length) of the bus wires between any two bus nodes
p, ptyp, pmax	length of the unterminated cable stub.
RL, RL.min	specific (typical, maximum) resistance per length unit of the bus wires
	total (minimum total) differential resistive bus load as seen by the transmitting node

APPENDIX 2 CALCULATION OF THE VOLTAGE AT THE INPUT OF A NODE

For the calculation of the worst case (i.e. minimum) differential input voltage at the receiving node the following assumptions or simplifications are made (see also Fig. 7 and Fig. 10):

- The termination resistors (R_T) are located at the output of the transmitting and at the input of the receiving node.
- The resistance of the lines between the transmitting and the receiving node is represented by the resistors R_W .
- All other nodes are at the same end of the transmission line as the receiving node resulting in the minimum differential input voltage.
- The output voltage of the transmitting node ($V_{diff.out}$) is supposed to be generated by a voltage source.

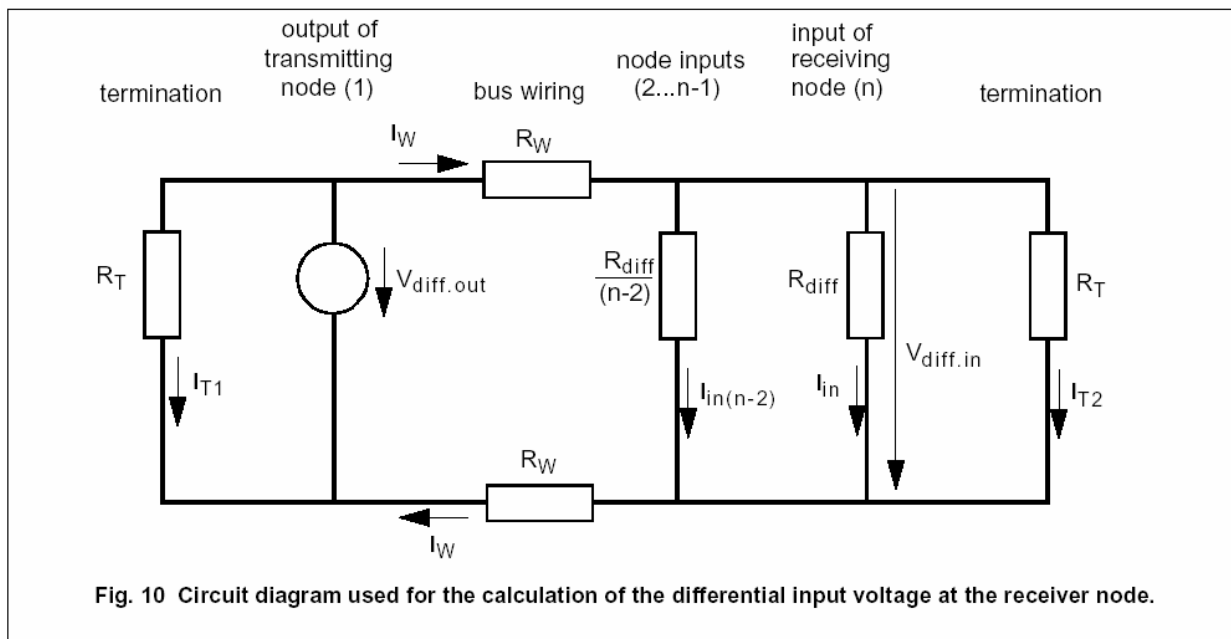


Fig. 10 Circuit diagram used for the calculation of the differential input voltage at the receiver node.

Thus the relation between the achievable differential input voltage at the receiving node and the differential output voltage of the transmitting node is given by:

$$V_{diff.out} = V_{diff.in} + 2 \times R_W \times I_W \quad (14)$$

The current I_W flowing through the bus lines splits up in $I_{in(n-2)}$ (input current of the connected nodes without the transmitting and receiving one), I_{T2} (current flowing through the termination resistor) and I_{in} (input current of the receiving node) as shown in Fig. 10.

With $I_W = I_{in(n-2)} + I_{T2} + I_{in}$ and

$$V_{diff.in} = I_{in(n-2)} \times \frac{R_{diff}}{n-2} \quad \text{and} \quad V_{diff.in} = I_{T2} \times R_T \quad \text{and} \quad V_{diff.in} = I_{in} \times R_{diff}$$

the relation between $V_{diff.out}$ and $V_{diff.in}$ is calculated from equation (14)

$$V_{\text{diff.out}} = V_{\text{diff.in}} + 2 \times R_W \times V_{\text{diff.in}} \times \left(\frac{1}{R_T} + \frac{n-1}{R_{\text{diff}}} \right) \quad (15)$$

Equation (8) on page 17 is derived from equation (15).

APPENDIX 3 CALCULATION OF THE MAXIMUM BUS LINE LENGTH

Under worst case conditions the minimum differential input voltage of a dominant level at the receiving node must be higher than the sum of the worst case switching threshold of the input transistor and a certain safety margin, which is requested in the system. The requested input voltage was given by equation (9) on page 18. The worst case value is given by

$$V_{\text{diff.in.req}} = V_{\text{th.max}} + k_{\text{sm}} \times (V_{\text{diff.out.min}} - V_{\text{th.max}}) \quad \text{with } k_{\text{sm}} = 0 \dots 1.$$

Thus the relation (10) on page 19 is changed to

$$V_{\text{diff.in.min}} = \frac{V_{\text{diff.out.min}}}{1 + 2R_{W.\text{max}} \times \left(\frac{1}{R_{T.\text{min}}} + \frac{n_{\text{max}} - 1}{R_{\text{diff.min}}} \right)} \geq V_{\text{th.max}} + k_{\text{sm}} \times (V_{\text{diff.out.min}} - V_{\text{th.max}}) \quad (16)$$

With the definition

$$R_{W.\text{max}} = \rho_{\text{max}} \times L_{\text{max}}$$

the maximum wiring length is determined using equation (16):

$$L_{\text{max}} \leq \frac{1}{2 \times \rho_{\text{max}}} \times \left(\frac{V_{\text{diff.out.min}}}{V_{\text{th.max}} + k_{\text{sm}} \times (V_{\text{diff.out.min}} - V_{\text{th.max}})} - 1 \right) \times \frac{R_{T.\text{min}} \times R_{\text{diff.min}}}{R_{\text{diff.min}} + (n_{\text{max}} - 1)R_{T.\text{min}}}$$

C.2 GÉNÉRALITÉS SUR LE RÉSEAU DE TERRAIN CAN

CAN (Controller Area Network) est un bus de communication série. Son architecture générale ainsi que les couches liaison de données et physique sont définies par les normes ISO 11519-1 pour les débits jusqu'à 125 kbit/s (CAN basse vitesse) et ISO 11898 pour les débits jusqu'à 1Mbit/s (CAN haute vitesse).

L'architecture CAN est conforme au modèle de référence pour l'interconnexion des systèmes ouverts (OSI) de l'ISO en ce qui concerne la décomposition en couches. Elle propose des services de transferts de données et de demandes de transmission distante sans connexion et acquittés. Tous les échanges sont réalisés en diffusion.

Les messages de données peuvent être émis depuis n'importe quelle station. Le message ne contient ni adresse de destinataire et ni l'adresse de l'adresse de l'émetteur, en effet le mécanisme d'adressage sur le bus n'est pas associé à une station mais associé à une donnée. Les données échangées sont identifiées par une adresse logique unique sur tout le réseau appelée « l'identificateur ». Chaque nœud du réseau reçoit ou filtre la trame suivant l'identificateur (nature de la donnée transmise). Cet identificateur détermine la priorité des messages. La valeur la plus basse de l'identificateur correspond à une donnée de priorité la plus élevée.

Lors de transmissions simultanées, l'ordre d'accès au bus est réalisé par un mécanisme d'arbitrage bit à bit basé sur l'identificateur et la nature de la trame (une trame de requête est moins prioritaire qu'une trame de transmission de donnée).

CAN est un protocole multi-maîtres qui peut utiliser plusieurs médias de transmission. Elle utilise en particulier comme support de transmission une paire blindée ou non blindée. Les stations sont reliées entre elles par l'intermédiaire d'un bus série équipé de terminateurs de lignes. Les interfaces physiques sont de type différentiel en mode tension, proche du principe de la liaison RS485.

Le codage utilisé est de type NRZ avec bits de stuffing, de telle sorte qu'il y aura un maximum de 5 bits consécutifs identiques.

Les messages peuvent transporter jusqu'à 8 octets de données utiles.

La méthode d'accès proposée par la norme est de type aléatoire (CSMA/CR) avec détection de collisions et arbitrage bit à bit, non destructif pour le message le plus prioritaire. La notion de priorité de message est inhérente au principe d'accès.

La norme établit une distinction entre les erreurs temporaires et les erreurs permanentes sur le bus. Elle préconise une déconnexion logique automatique des modules en erreur. L'émission automatique de trames de surcharges par les modules saturés assure la gestion du contrôle de flux.

Diverses couches application (niveau 7 du modèle OSI) orientées systèmes d'automatismes industriels et s'appuyant sur CAN sont disponibles (Canopen, Devicenet,...).

Topologie	Bus pour CAN High Speed Norme ISO 11898 Length-Related Resistance (r): 70 mΩ /m Termination Resistor (Rt): nominal 120 Ω (min. 108 Ω, max. 132 Ω) Impedance (Z): nominal 120 Ω (min. 108 Ω, max. 132 Ω) Specific Line Delay: 5 ns/m All these mentioned AC and DC parameters are suitable for a 1 Mbit/s transmission rate.		
Media supportés	Paires différentielles High Speed normes ISO 11898 Low Speed norme ISO 11519-2 RS485, Infra rouge, CPL, Fibres optiques		
débit maximal	1 Mbits/s pour longueur maxi de 30 m (High Speed) 125 kbits/s pour Cmaxi 30 pF/m (Low Speed)		
Longueur maximale	Bit Rate	Bus Length	Nominal Bit-Time
	1 Mbit/s	30 m	1 µs
	800 kbit/s	50 m	1,25 µs
	500 kbit/s	100 m	2 µs
	250 kbit/s	250 m	4 µs
	125 kbit/s	500 m	8 µs
	62,5 kbit/s	1 000 m	20 µs
	20 kbit/s	2 500 m	50 µs
	10 kbit/s	5 000 m	100 µs
Nombre de répéteurs	non spécifié, fonction de l'implémentation		
Nombre maximal d'équipements sur le réseau	dépend du média et des drivers de ligne utilisés. dépend du nombre de messages différents associés à chaque équipement		
Spécifications générales	Filtrage des messages Traitement des message et status Détection d'erreur Validation des messages Acquittement des messages Gestion arbitrage des conflits de bus avec la notion de messages prioritaires Gestion et contrôle de la mise en forme trame, du débit, et de la conformité temporelle du transfert (timing)		
Type de trames	Trame de données (Data Frame) Trame de requête (Remote Frame) Trame d'erreurs (Error Frame) Trame de surcharge (Overload Frame)		
Détection d'erreur	Données protégés par un checksum codé sur 15 bits. Signalement type d'erreur et faute de confinement (distinction entre les erreurs temporaires et permanentes). Validation du format des messages (Format Check) Ecoute et surveillance du bit émis sur le bus (Bit Check) Signalement type d'erreur et faute de confinement (distinction entre les erreurs temporaires et permanentes).		
Débit maximal	1 Mbits/s		
Nombre d'octets de données	0 à 8 Octets par message		
Mode d'adressage	Orienté message (adressage des messages) Adressage par identificateur associé aux messages Identificateur 11bits en mode standard (Format 2.0A) Identificateur (29 bits) étendu (Format 2.0B)		
Hiérarchie	Multimaître Priorité définie par les identificateurs de messages (2 048 priorités en mode standard)		
Mode d'accès au réseau (arbitrage)	Abritage bit à bit (non destructif) CSMA/CR (R=Resolve) arbitrage pendant le champ d'identification des messages		
Trame de donnée (Data FRAME)	Données (0 à 8 Octets) Encapsulation des données 47 Bits pour le format standard 2.0A Encapsulation des données 64 Bits pour le format étendu 2.0B		

CONTROLLER AREA NETWORK – HOW CAN WORKS

Principle

Data messages transmitted from any node on a CAN bus do not contain addresses of either the transmitting node, or of any intended receiving node.

Instead, the content of the message is labelled by an identifier that is unique throughout the network. All other nodes on the network receive the message and each performs an acceptance test on the identifier to determine if the message, and thus its content, is relevant to that particular node.

If the message is relevant, it will be processed; otherwise it is ignored.

Identifiers

The unique identifier also determines the priority of the message. The lower the numerical value of the identifier, the higher the priority.

The higher priority message is guaranteed to gain bus access as if it were the only message being transmitted. Lower priority messages are automatically re-transmitted in the next bus cycle, or in a subsequent bus cycle if there are still other, higher priority messages waiting to be sent.

Bit encoding

CAN uses Non Return to Zero (NRZ) encoding (with bit-stuffing) for data communication on a differential two wire bus. The use of NRZ encoding ensures compact messages with a minimum number of transitions and high resilience to external disturbance.

The physical bus

The two wire bus is usually twisted pair (shielded or unshielded). Flat pair (telephone type) cable also performs well but generates more noise itself, and may be more susceptible to external sources of noise.

The bus nodes shall detect a recessive bus condition if the voltage of CAN_H is not higher than the voltage of CAN_L plus 0.5 V. If the voltage of CAN_H is at least 0.9 V higher than CAN_L, then a dominant bus condition shall be detected. The nominal voltage in the dominant state is 3.5 V for the CAN_H line and 1.5 V for the CAN_L line.

Robustness

CAN will operate in extremely harsh environments and the extensive error checking mechanisms ensure that any transmission errors are detected. The ISO11898 standard "Recommends" that bus interface chips be designed so that communication can still continue (but with reduced signal to noise ratio) even if:

- Either of the two wires in the bus is broken
- Either wire is shorted to power
- Either wire is shorted to ground

Depending on the design and configuration of the system, if both wires are broken at the same location, some limited functionality may still be achievable in each of the sub-systems created by the breaks.

How Arbitration Works on the CAN Bus

In any system, some parameters will change more rapidly than others. For example - parameters that change quickly could be the RPM of a car engine, or the current floor level of an elevator - lift. Slower changing parameters may be the temperature of the coolant of a car engine.

It is likely that the more rapidly changing parameters need to be transmitted more frequently and, therefore, must be given a higher priority.

To cater for real-time data communication, this requires not only a fast data transmission rate, but also a rapid bus allocation mechanism to deal with occasions when more than one node may be trying to transmit at the same time.

To determine the priority of messages, CAN uses the established method known as Carrier Sense, Multiple Access with Collision Detect (CSMA/CD) but with the enhanced capability of non-destructive bitwise arbitration to provide collision resolution, and to deliver maximum use of the available capacity of the bus.

Non-Destructive Bitwise Arbitration

The priority of a CAN message is determined by the binary value of its identifier. The numerical value of each message identifier (and thus the priority of the message) is assigned during the initial phase of system design.

The identifier with the lowest numerical value has the highest priority. Any potential bus conflicts are resolved by bitwise arbitration in accordance with the wired-and mechanism, by which a dominant state (logic 0) over writes a recessive state (logic 1).

The overall result is the same as if the highest priority message was the only message being transmitted. As soon as any lower priority transmitter loses control of the bus via the arbitration mechanism, it automatically becomes a receiver of the message with the highest priority and will not attempt re-transmission until the bus becomes available again.

The Benefits

Non-destructive bitwise arbitration provides bus allocation on the basis of need, and delivers efficiency benefits that can not be gained from either fixed time schedule allocation (e.g. Token ring) or destructive bus allocation (e.g. Ethernet.)

With only the maximum capacity of the bus as a speed limiting factor, CAN will not collapse or lock up. Outstanding transmission requests are dealt with, in their order of priority, with minimum delay, and with maximum possible utilisation of the available capacity of the bus.

CAN Message Format

Message Frames

In a CAN system, data is transmitted and received using Message Frames. Message Frames carry data from a transmitting node to one, or more, receiving nodes.

The Standard CAN protocol (version 2.0A) supports messages with 11 bit identifiers.

The Extended CAN protocol (version 2.0B) supports both 11 bit and 29 bit identifiers.

2.0A Format

A Standard CAN (Version 2.0A) Message Frame consists of seven different bit fields :

A Start of Frame (SOF) field - which indicates the beginning of a message frame.

An Arbitration field, containing a message identifier (11 bits) and the Remote Transmission Request (RTR) bit. The RTR bit is used to discriminate between a transmitted Data Frame and a request for data from a remote node.

A Control Field containing six bits:

- two dominant bits (r0 and r1) that are reserved for future use, and
- a four bit Data Length Code (DLC). The DLC indicates the number of bytes in the Data Field that follows

A Data Field, containing from zero to eight bytes.

- The CRC field, containing a fifteen bit cyclic redundancy check code and a recessive delimiter bit.
- The ACKnowledge field, consisting of two bits. The first is the Slot bit which is transmitted as recessive, but is subsequently over written by dominant bits transmitted from any node that successfully receives the transmitted message. The second bit is a recessive delimiter bit
- The End of Frame field, consisting of seven recessive bits.

Following the End Of Frame is the intermission field consisting of three recessive bits. After the three bit Intermission period the bus is recognised to be free. Bus Idle time maybe of any arbitrary length including zero.

2.0B Format

The CAN 2.0B format provides a twenty nine (29) bit identifier as opposed to the 11 bit identifier in 2.0A.

Version 2.0B evolved to provide compatibility with other serial communications protocols used in automotive applications. To cater for this, and still provide compatibility with the 2.0A format, the Message Frame in Version 2.0B has an extended format.

The differences are:

- - In Version 2.0B the Arbitration field contains two identifier bit fields. The first (the base ID) is eleven (11) bits long for compatibility with Version 2.0A. The second field (the ID extension) is eighteen (18) bits long, to give a total length of twenty nine (29) bits.
- - The distinction between the two formats is made using an Identifier Extension (IDE) bit.
- - A Substitute Remote Request (SRR) bit is also included in the Arbitration Field. The SRR bit is always transmitted as a recessive bit to ensure that, in the case of arbitration between a Standard Data Frame and an Extended Data Frame, the Standard Data Frame will always have priority if both messages have the same base (11 bit) identifier.

All other fields in a 2.0B Message Frame are identical to those in the Standard format.

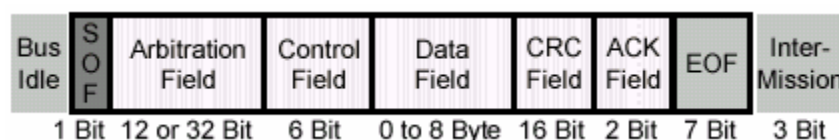


Figure 1: DATA FRAME

D ANNEXE D: EXTRAITS DE RÉFÉRENTIELS ET PROGRAMMES

D.1 EXTRAITS DU PROGRAMME DU BACCALAURÉAT GÉNIE ÉLECTRONIQUE :

D.1.1 EXTRAITS DES COMPÉTENCES (COMPÉTENCES D ET E)

D. D'identifier à une fonction la ou les structures participant à sa réalisation.

Mise en situation, le candidat dispose :

- du schéma fonctionnel relatif aux fonctions de technologie électronique ;
- de l'identification des grandeurs d'entrées-sorties de chacune des fonctions ;
- du schéma structurel (avec le repérage total ou partiel des grandeurs d'entrées-sorties).

Comportement vérifiable :

Le candidat doit, par exemple, entourer sur le schéma structurel, l'ensemble des composants, éléments de la structure associée à chacune des fonctions.

E. D'analyser l'organisation structurelle d'une fonction afin : .

- **E1** d'établir les relations entre grandeurs d'entrée et grandeurs de sortie qui caractérisent cette fonction ;
- **E2** de justifier le dimensionnement d'un composant ;
- **E3** de choisir un composant ;
- **E4** d'évaluer que la fonction requise est assurée.

Mise en situation, le candidat peut disposer :

- des spécifications du cahier des charges ;
- des schémas fonctionnels de l'objet technique ;
- du schéma structurel,
- de la nomenclature des composants figurant sur le schéma structurel
- d'un poste de simulation informatique et de son mode d'emploi.

Comportement vérifiable, le candidat peut être amené à :

- déterminer ou utiliser l'expression de la fonction de transfert (E1, E4)
- établir, utiliser le graphe représentatif de la fonction de transfert (E1,
- établir, compléter la table de vérité ou les chronogrammes associant les grandeurs d'entrées - sorties (E1, E4) ;
- utiliser un logiciel de simulation du fonctionnement de la structure (E4) ;
- utiliser les relations établies pour choisir un composant (E3), le dimensionner, mettre en évidence ses limites d'emploi (E2).

D.1.2 EXTRAITS DES SAVOIRS

Traitement des signaux analogiques

FONCTION	Eléments de structure mis en œuvre	Durée	Commentaires
Amplification	amplificateur de tension, amplificateur de courant, amplificateur de puissance: - aspects technologiques - problèmes thermiques - rendement	60 h	Les principes physiques mis en oeuvre dans l'amplification étant abordés en physique appliquée l'activité consistera surtout à travers les documents constructeurs à mettre en évidence les caractéristiques et limites d'emploi des composants. En ce qui concerne les amplificateurs de tension et de courant les composants intégrés seront de préférence utilisés. L'étude des différentes structures est abordée dans le contexte fonctionnel. L'étude de l'amplificateur de puissance sera l'occasion de mettre en oeuvre les solutions de dissipation thermique des composants de puissance. La notion de rendement étudiée en physique sera abordée en relation avec le contexte fonctionnel.
Opérations algébriques	comparateur, additionneur/soustracteur, diviseur/multiplicateur.	30 h	Les opérations de multiplication et de division ne seront envisagées que s'il est possible de limiter la complexité de la structure. On s'orientera vers l'exploitation des notices constructeurs des composants spécialisés.
Opérations mathématiques Filtrage	intégrateur/dérivateur. filtre passe-bas, filtre passe-haut, (filtres actifs du premier ordre utilisant un amplificateur intégré).	40 h	A l'occasion d'une étude il sera possible de rencontrer une structure d'un ordre supérieur à 1. Les documents résumant les calculs théoriques seront remis aux élèves. Éventuellement la validation de sera faite de façon expérimentale.

Traitement des signaux logiques et/ou numériques

FONCTION	Eléments de structure mis en œuvre	Durée	Commentaires
Commutation	Opérateurs logiques de base: - Et/Ou/Non, - Et-Non/Ou-Non, - Ou exclusif, multiplexeur/dé multiplexeur.	30 h	Lors de l'utilisation et de l'assemblage numériques d'opérateurs logiques intégrés, les caractéristiques technologiques seront mises en évidence à partir de l'analyse des notices constructeurs (niveaux logiques, entrance, sortance, marge de bruit, caractéristiques temporelles...) en excluant l'analyse des structures internes.
Codage Transformation de code	codeur, décodeur. transcodeur.	10 h	On se limitera à l'utilisation des codes usuels en employant des circuits intégrés spécifiques.
Traitement programmé de l'information - Organisation matérielle : architecture fonctionnelle d'un dispositif utilisant la logique programmée - séquençement, - mémorisation, - échanges des informations entre les entités fonctionnelles. - Organisation logicielle :	structures de bus, interfaces. structure algorithmiques de base: - linéaire, - alternative, - itérative, - sous-programmes, - éléments de programmation permettant la réalisation de fonctions relatives à l'objet technique étudié.	50 h	Lors de l'analyse du fonctionnement d'un objet technique programme, il est important de montrer comment l'association des structures et du programme réalisé la (ou les) fonction requise par l'objet technique. Après étude des algorithmes et analyse du programme en langage assembleur ou en langage évolué, les élèves seront amenés à modifier certains segments de programme en relation avec l'évolution des contraintes fonctionnelles. A cet effet les élèves recevront les informations et documents strictement nécessaires. Les langages de programmation ne pourront faire l'objet d'une étude intrinsèque. La maîtrise des segments de programme relatifs à l'initialisation des périphériques ne pourra être exigée des élèves.

Transmission de l'information

FONCTION	Eléments de structure mis en œuvre	Durée	Commentaires
Conversion de données	convertisseur analogique/numérique, convertisseur numérique/analogique, convertisseur fréquence/tension, convertisseur tension/fréquence.	30 h	Seules les structures intégrées seront utilisées et on s'intéressera plus particulièrement aux signaux de contrôle, de commande et de sortie des convertisseurs. L'étude exhaustive des divers principes de conversion est exclue.
Codage des informations	modulateur à déplacement de fréquence.	10 h	Seule sera abordé le principe de la modulation de type FSK. Les structures seront validées par voie expérimentale uniquement.
Transmission non galvanique	coupleur optique, transformateur, relais.	20 h	La nécessité de réaliser des transmissions non galvaniques d'information dans le cas de la protection des utilisateurs et des matériels permet la mise en évidence des caractéristiques d'utilisation et des limites d'emploi des coupleurs optiques magnétiques et inductifs dont la structure est justifiée lors de l'analyse fonctionnelle de l'objet technique.
Transmission de données numériques	liaison série, liaison parallèle.	30 h	L'étude sera orientée vers les protocoles et les structures indispensables à la réalisation de liaisons série ou parallèle.

D.2 EXTRAITS DU RÉFÉRENTIEL DE BTS IRIS

D.2.1 EXTRAITS DES COMPÉTENCES

CAPACITÉ	Compétence terminale	
C3 : CONCEVOIR	C3.8 :	Caractériser les contraintes principales d'un système de transmission de l'information
Ressources mises à disposition		
Le cahier des charges client avec toutes ses contraintes (géographiques, temporelles, opérationnelles, ...). Dossier technique relatif à l'existant éventuel. Documentation technique sur le sujet (sur support papier et sous forme numérique).		
Compétences composantes	Critères d'évaluation	Niveaux de performance
- Identifier et caractériser toutes les informations de l'application qui seront transportées par le système. - Énumérer et classer les caractéristiques physiques de l'application qui vont conditionner les choix techniques pour le système de transmission. - Calculer les différentes caractéristiques physiques de l'application qui vont conditionner les choix techniques pour le système de transmission. - Déterminer les caractéristiques des moyens possibles d'interfaçage avec les systèmes de communication déjà existants.	- Précision des définitions des informations. - Pertinence des choix des outils de formalisation utilisés. - Pertinence des questions posées au client. - Précision de la liste classée des caractéristiques déterminantes. - Exactitude des valeurs des grandeurs typiques avec leurs unités. - Précision des caractéristiques des éléments d'interfaçage.	- Les informations sont définies, classées et quantifiées. - Au moins un outil de formalisation sera utilisé. - Les caractéristiques (spatiales, temporelles, environnementales...) sont définies sans ambiguïté par un vocabulaire technique adapté. On pourra tolérer quelques oublis s'ils sont dus à des caractéristiques « métier » très pointues. - Toutes les grandeurs sont quantifiées sans ambiguïtés. - Aucune erreur d'unité n'est tolérée. On peut tolérer des erreurs de 30% maximum sur la justesse. - Les informations tirées du document d'analyse de l'existant sont suffisamment pertinentes et complètes pour pouvoir proposer des choix techniques d'interfaçage argumentés.

Savoirs et savoir-faire associés :

Les réseaux et les modes de transmission :

Notions fondamentales.

Principes de base de la transmission au niveau 1.

Applications utilisateurs.

Les systèmes d'exploitation et les systèmes temps réel :

Spécificités temps réel.

Contraintes de temps d'un système de contrôle/commande.

Développement logiciel :

Structure et gestion des données.

Codage, cryptage et compression des données.

CAPACITÉ	Compétence terminale	
C3 : CONCEVOIR	C3.9 :	Décrire les tâches d'une application de communication
Ressources mises à disposition		
Dossier de spécification. Dossier de conception préliminaire. Documentations techniques du réseau utilisé. Documentations techniques de la machine cible. Documentation du système d'exploitation de la machine cible. Outils informatiques d'analyse et de production de documents.		
Compétences composantes	Critères d'évaluation	Niveaux de performance
- Identifier et Interpréter en les classant en groupes typologiques les éléments pertinents des dossiers fournis. - Mettre en relation les différentes composantes de l'application avec le modèle OSI. - Identifier les caractéristiques de l'application qui vont guider les choix des techniques de réalisation. - Valider l'organisation de l'application sous forme d'un ensemble de tâches communicantes. - Compléter le dossier de conception détaillée.	- Cohérence des groupes d'éléments caractéristiques. - Pertinence de la traduction des éléments identifiés dans les dossiers fournis. - Justesse de l'identification du système étudié avec le modèle OSI. - Reconnaissance des protocoles de communication imposés. - Localisation de l'interface du système de communication avec l'application. - Reconnaissance des modes de fonctionnements particuliers du matériel. - Conformité de l'architecture proposée avec les contraintes spécifiées. - Respect des règles de conception d'une application multitâches. - Respect des protocoles de communication. - Formalisation des éléments de conception. - Qualité du dossier produit (contenu et forme)	- Tous les éléments nécessaires à la conception sont identifiés et classés. - Les différentes couches du système et leurs caractéristiques sont identifiées. - Les caractéristiques spécifiques du niveau applicatif sont listées (<i>exemple : contraintes de temps, de débit d'information</i>) ainsi que les interfaces avec les couches inférieures. - Toutes les caractéristiques matérielles qui influent directement sur la conception de l'application sont caractérisées. - Les solutions envisagées sont viables. - Exploitation correcte des documents générés par l'atelier de génie logiciel. - Les techniques de base de la programmation multitâches sont connues et mises en pratique - Le dossier est lisible et exploitable, en particulier par le programmeur chargé de la réalisation.

Savoirs et savoir-faire associés :

Les réseaux et les modes de transmission :

Notions fondamentales.

Programmation réseau.

Systèmes d'exploitation réseau.

Savoirs et savoir-faire associés :

- Les réseaux et les modes de transmission :**
 - Notions fondamentales.
 - Spécificités des réseaux locaux, industriels, et d'instrumentation.
 - Réseaux locaux industriels.
 - Équipements réseau.
 - Systèmes d'exploitation réseau.
 - Sécurisation du réseau.
- Architecture matérielle des systèmes informatiques :**
 - Coupleurs d'entrées/sorties.
 - Environnement matériel des systèmes informatiques.

LES RÉSEAUX ET LES MODES DE TRANSMISSION

	X		X	- pour le câble coaxial (connecteurs et bouchons BNC), - pour le câble à paires torsadées (<i>par exemple : connecteur RJ45, baie de brassage, concentrateur de câblage</i>).
		X	X	Transmission des signaux (transmission à bande de base et à large bande). Liaisons séries : - les liaisons (<i>par exemple : RS232, RS485, RS422, boucle 20mA, USB, IEEE1394, Ethernet niveau 1</i>), - protocoles bipoints (<i>par exemple : RTS/CTS, XON/XOFF</i>). Carte réseau : - configuration matérielle et logicielle, - compatibilité avec le bus de données et avec les connecteurs BNC, RJ45 ou AUI, - transceiver (transmitter/receiver) externe et interne, - architecture fonctionnelle et caractéristiques des E/S des cartes : + pour réseaux sans fil , + pour fibre optique.
	X			Protocoles de liaison et méthodes d'accès au support <i>(HDLC, Ethernet niveau 2, 802.2, 802.3, 802.5, PPP...)</i> Principales méthodes d'accès : - méthode CSMA avec détection de collision (CSMA/CD) ou avec prévention des collisions (CSMA/CA), - méthode fondée sur le passage d'un jeton, - méthode à priorité de demande (IEEE 802.12). Sous-couches LLC et MAC (IEEE 802.2) du modèle OSI.
	X			Spécificités des réseaux locaux, industriels, et d'instrumentation <i>(Fast Ethernet, FDDI, MODBUS, CAN, I2C, FIP, IEEE488 ...)</i> Réseaux sans fil (IEEE 802.16 et 802.11) : - les types de réseaux sans fil, - techniques de transmission (infrarouge, laser, radio). Ethernet (IEEE 802.3) : - caractéristiques générales (topologies, type d'architecture, méthodes d'accès, vitesses de transmission, types de câbles), - format de la trame, - normes 100BaseTX, 100BaseFX, - équipements (concentrateur, commutateur).
		X		Réseaux locaux industriels : - spécificités des réseaux locaux industriels, - Ethernet dans un contexte industriel, - les réseaux de terrain FIP, CAN.
		X		Les équipements réseau <i>(carte réseau, modem, concentrateur, commutateur, routeur...)</i> Modems : - caractéristiques, normes ITU V22bis à V90, - communications asynchrones, synchrones, - ADSL. Répéteurs (fonctionnement, mise en œuvre). Concentrateurs de câblage. Commutateurs. Ponts (fonctionnement, tables de routage, installation). Routeurs : - fonctionnement, - protocoles routables, - sélection des chemins et algorithme de routage (RIP, NLSP, OSPF), - routeur statique, routeur dynamique.

ARCHITECTURE MATÉRIELLE DES SYSTÈMES INFORMATIQUES

Niveaux				
1	2	3	4	
	X			Microprocesseur
	X			Principe de fonctionnement.
	X			Structure interne (<i>par exemple : registres, mémoire cache, pipeline d'instructions, pagination</i>).
	X			Communication avec l'extérieur : bus, dynamique de la communication.
				Notions de langage d'assemblage.
	X			Mémoires
	X			Principaux types de mémoires.
		X		Structure interne et organisation.
				Couplage avec un microprocesseur et accès (adressage).
	X			Bus
	X			Architecture et fonctionnalités d'un bus système (VME, PCI).
				Architecture et fonctionnalités d'un bus d'extension (IDE, SCSI, AGP, USB, Serial ATA).
	X			Coupleurs d'entrées/sorties
	X			Fonctionnalités.
	X			Structure générale interne (lien avec un modèle générique objet).
		X		Couplage avec un microprocesseur.
		X		Mode d'accès : par scrutation, par interruption, DMA.
	X			Liaisons asynchrones.
		X		Liaisons synchrones.
		X		Liaisons parallèles (Centronics, IEEE488).
		X		Pilotes de commande (utilisation, adaptation, choix d'un mode de fonctionnement).
		X		Bibliothèques de composants réutilisables.
		X		Architecture fonctionnelle et caractéristiques d'entrées/sorties des cartes industrielles :

		X	
		X	
		X	
		X	
	X		

- TOR,
- de conversion NA, AN,
- de comptage,
- de commande de moteurs (pas à pas, continu),
- modules mezzanines (IP).

Environnement matériel des systèmes informatiques

Caractéristiques d'entrées/sorties de capteurs (position, vitesse, pression, effort) et des préactionneurs (pour moteurs pas à pas et à courant continu)

Architecture fonctionnelle, caractéristiques d'entrées/sorties et utilisation de périphériques :

- multimédias (son, image),
- de lecteurs de codes,
- de cartes à puce,
- de périphériques de stockage de données par procédés magnétiques et optiques.

Systèmes embarqués et dédiés

LES SYSTÈMES D'EXPLOITATION

Niveaux			
1	2	3	4
			X
			X
		X	
		X	
		X	
		X	
		X	

Systèmes d'exploitation et systèmes temps réel

Règles d'échange des données :

- libre au sein d'une section critique verrouillée,
- modèle producteur - consommateur sans et avec stockage de l'information,
- modèle lecteur - rédacteur,

Communication inter - processus (locaux et distants).

Exemple : fichiers mappés en mémoire, mémoire partagée, boîtes aux lettres, tubes.

Identification du modèle d'échange dans un mécanisme de communication.

Gestion des entrées/sorties.

A.1 EXTRAITS DU RÉFÉRENTIEL DE BTS SE

Capacité A : ANALYSER l'existant	BTS systèmes électroniques
Compétence terminale A1 : expliciter un schéma fonctionnel	

Savoir-faire Être capable de :	Conditions de réalisation		Niveau de performance (On se limite à ...)
	On donne	On demande	
Le candidat doit être capable de : - situer l'objet technique au sein du système auquel il appartient, - d'énoncer la fonction d'usage de l'objet technique - d'associer les spécifications du cahier des charges aux caractéristiques des fonctions et des grandeurs traitées - d'expliquer comment l'organisation du schéma fonctionnel de l'objet permet la réalisation de la fonction d'usage de cet objet.	Le candidat dispose de tout ou partie d'un dossier comportant : - la fonction d'usage et la fonction globale du système technique, - le schéma fonctionnel associé à la fonction d'usage - la mise en situation topographique de l'objet au sein du système auquel il appartient, - le diagramme sagittal du système technique, - un descriptif des milieux associés à l'objet technique, - les notices de maintenance et d'utilisation, - les schémas fonctionnels complets ou partiels de l'objet, - les schémas structurels complets des structures abordées, - la partie du logiciel étudiée dans le cas d'un objet technique à technologie programmée, - des spécifications techniques extraites du cahier des charges. Le schéma fonctionnel temporel précise les fonctions et les liaisons actives pour la phase de fonctionnement validée à une date donnée. Il est associé à l'algorithme de fonctionnement de l'objet.	Le candidat est amené à : - identifier les éléments du système technique, en particulier l'objet technique étudié, - situer dans l'espace les différents éléments participant au système, - expliciter les différentes relations entre les éléments composant le système, - préciser le type de flux traité par les différents composants du système (matière, énergie, information), - exploiter l'algorithme de fonctionnement du système - exploiter un diagramme sagittal en montrant notamment les boucles de régulation, les stockages, les circulations des flux. Le candidat doit : - compléter les schémas fonctionnels associés à la fonction d'usage de l'objet technique en indiquant la nature des flux traités (matérielle logicielle ou mixte) - exploiter ou donner les caractéristiques des grandeurs d'entrée et de sortie des fonctions du schéma fonctionnel associé à la fonction d'usage. Pour chaque sortie de l'objet technique, le candidat : - identifie le verbe d'action contenu dans l'expression de la fonction	L'objet technique étudié est inséré dans un système dont le nombre d'éléments sera limité. L'appropriation du fonctionnement du système ne doit pas prendre plus d'une heure d'épreuve. Le dossier fourni au candidat au cours de l'épreuve U 4.1 ne doit retenir que le fonctionnement prioritaire de l'objet technique. Les schémas descriptifs du système ainsi que les textes associés précisent parfaitement les limites de l'étude.

Savoir-faire Être capable de :	Conditions de réalisation		Niveau de performance (On se limite à ...)
	On donne	On demande	
		d'usage. Celui-ci produit le résultat qui est fourni à la sortie considérée. • associe ce verbe d'action à la fonction correspondante, • identifie la nature des données présentes en entrée nécessaire à la réalisation de cette fonction, • recherche pour chacune des entrées, • si la donnée correspondante existe à l'extérieur de l'objet technique. Cette donnée est alors une entrée pour l'objet technique étudié. Le traitement de cette sortie est alors terminé. • si la donnée n'existe pas, elle se trouve alors élaborée par une autre fonction interne correspondante à un autre verbe d'action de l'expression de la fonction d'usage. Il est alors nécessaire d'identifier ce nouveau verbe d'action puis de reprendre les opérations jusqu'à épuisement de toutes les données.	

Compétence terminale **M1** : Installer et configurer un nouvel équipement ou produit

Savoir-faire Être capable de ..	Conditions de réalisation		Niveau de performance (On se limite à ...)
	On donne	On demande	
Le candidat doit être capable d'installer un équipement et de le configurer dans le mode d'exploitation demandé. Cela implique qu'il doit - analyser la documentation d'installation de l'équipement - établir, puis préparer un plan d'action - préparer le matériel nécessaire notamment les éléments de connexion - réaliser les interconnexions nécessaires y compris la connexion à l'alimentation en énergie dans le respect des règles de sécurité définies par la législation en vigueur, - valider la continuité électrique des connexions entre l'équipement installé et les autres éléments du système. - placer les programmes qui permettront à l'équipement de fonctionner conformément aux attentes du client - vérifier la conformité du fonctionnement.	Le candidat dispose - du dossier d'installation de l'équipement. - du plan architectural du site et du schéma de câblage du site (si nécessaire) - de la configuration initiale du système sur le quel il intervient - des éléments d'environnement Le candidat dispose également ; - du système technique sur lequel il doit installer un nouvel équipement, - des outils de configuration. (à distance si nécessaire ...), - un ensemble de composants de connexion - des outils de tests et câblage Le candidat dispose éventuellement - de la liste des moyens matériels et humains nécessaires à l'exécution des travaux, - des plages horaires et des durées d'intervention, - de la détermination de la dégradation de service durant l'intervention.	Le candidat fournit un compte rendu d'installation et de mise en service de l'équipement. Ce dossier contient ; - l'algorithme d'installation du nouvel équipement - le plan d'implantation et le schéma de câblage avec les modifications, - l'exécution et l'enchaînement des travaux, Le candidat réalise, selon les normes et les règles de l'art, les connexions nécessaires en adaptant les éléments de connectique au support de transmission Le candidat montre le nouvel équipement en état de fonctionnement inséré dans le système. Il indique les ajouts effectués et les nouvelles fonctionnalités apportées au système par la mise en service de ce nouvel équipement.	Au cours de la validation de cette compétence, on se limite à l'installation, la configuration et la mise en service d'un nouveau produit sur l'équipement existant. Le candidat se limite à fournir par écrit la liste des éléments nécessaires à l'installation du produit, les schémas d'installation, l'algorithme d'installation, l'algorithme de mise en service, et si l'intervention nécessite une interruption de service, la durée et les consignes liés à mettre de mise hors service de l'installation. Au cours de la mise en service, le candidat note toutes les actions et les effets de ces actions. Celles-ci sont consignées dans le rapport d'installation évalué par la commission d'interrogation NB: Il s'agit de valider également la démarche qualité : Je dis ce que je fais, je fais ce que je dis. NB: Les notices fournies seront celles du concepteur du produit. Elles pourront être en langue anglaise. Toutefois, les productions du candidat sont en langue française.

Savoir-faire Être capable de ..	Conditions de réalisation		Niveau de performance (On se limite à ...)
	On donne	On demande	
Le candidat doit être capable valider le bon fonctionnement du produit ou de l'équipement. Cela implique qu'il doit • comparer le fonctionnement réel et le fonctionnement attendu, • interpréter les indicateurs de fonctionnement • diagnostiquer d'un dysfonctionnement éventuel en vérifiant le fonctionnement des matériels et logiciels • conclure quant au bon fonctionnement de l'équipement ou du produit.	Le candidat dispose • des dossiers d'installation de l'équipement. • du plan architectural du site et du schéma de câblage du site (si nécessaire) • de la configuration initiale du système sur le quel il intervient • des éléments d'environnement • des outils de tests et câblage • des spécifications du cahier des charges du sous-ensemble à mettre en conformité et de son cahier technique ; • des résultats des analyses fonctionnelles et structurelles de l'objet technique ; • d'un guide (complet ou partiel) décrivant les processus de test et de maintenance. • des règlements d'hygiène et sécurité. • normes et consignes de sécurité. Le candidat dispose également de l'équipement ou du produit ainsi que de l'ensemble des appareils de mesures spécifiques permettant d'effectuer les tests nécessaires	Le candidat consigne les tests effectués dans un court rapport de tests qui doit l'amener à conclure au bon fonctionnement ou non de l'équipement ou du produit. Le rapport contient ; : • le diagnostic du fonctionnement établi en comparant le fonctionnement réel et le fonctionnement attendu, • les tests et les essais effectués en précisant le type de mesures, les appareils utilisés, les conditions de la mesure, • l'interprétation des tests effectués • la liste des éléments qui permet de caractériser le bon ou le mauvais fonctionnement de l'équipement ou du produit • la conclusion quant au bon ou mauvais fonctionnement du produit ou de l'équipement testé.	Le candidat se limite à appliquer des procédures de tests. Le résultat des mesures est consigné dans un court rapport Le candidat se limite à analyser ses résultats des mesures en regard de ceux obtenus en simulation ou d'autres résultats fournis Le candidat doit conclure quant à la validation du produit ou de l'équipement. Au cours de ces mesures, le candidat note toutes les actions et les effets de ces actions. Celles-ci sont consignées dans le rapport d'installation évalué par le jury NB: Il s'agit de valider également la démarche qualité : Je dis ce que je fais, je fais ce que je dis. NB: Les procédures de test fournies sont celles du concepteur du produit. Elles pourront être en langue anglaise. Toutefois, les productions du candidat sont uniquement en langue française.

D.3.1 EXTRAITS DES CHAMPS DES SAVOIRS ASSOCIÉS AUX COMPÉTENCES

L'inventaire des "savoirs" et "savoirs faire" que doit mobiliser le titulaire du BTS pour exercer les compétences dans ses futures activités professionnelles est issu des champs technologiques du RAP affectés du niveau de taxonomie 3 et 4. Ces savoirs sont classés en huit groupes.

Repères	Fonctions
S01	Acquisition des grandeurs physiques.
S02	Traitement analogique de l'information captée
S03	Traitement numérique de l'information et stockage des données
S04	Transmission et transport de l'information
S05	Connexions entre constituants électroniques ou connexions inter-systèmes
S06	Commande des actionneurs
S07	Fabrication industrielle d'un produit électronique
S08	Utilisation de l' outil informatique

Repères	Fonctions	% de l'horaire global en électronique	% de l'horaire global en physique appliquée
S01	Acquisition des grandeurs physiques	5%	10%
S02	Traitement analogique de l'information captée	20%	27%
S03	Traitement numérique de l'information et stockage des données	20%	18%
S04	Transmission et transport de l'information	25%	30%
S05	Connexions entre constituants électroniques ou connexions inter systèmes	5%	6%
S06	Commande des actionneurs	2,5%	5%
S07	Fabrication industrielle d'un produit électronique	20%	2%
S08	Utilisation de l'outil informatique	2,5%	2%
		100%	100%

S04 : Transmission et transport de l'information	
Savoirs	On se limite à:
- Transmission de signaux analogiques et numériques en bande de base. - Transmission de signaux analogiques et numériques en bande transposée (fréquence porteuse, ...) - Modulation et démodulation analogique (AM, FM, PM) et numériques (FSK, PSK....) - Encombrement spectral	- déterminer la relation entre la grandeur d'entrée et la grandeur de sortie des modulations de type AM, FM, PM et FSK. Les autres types de modulation seront abordés expérimentalement en fonction du projet et des systèmes étudiés - valider le type de modulation au regard du cahier des charges de la transmission en terme de bande passante, de débit ou de rapport signal sur bruit. Il est souhaitable que cette validation s'effectue expérimentalement.
Transmission numérique de l'information point à point sous forme série (RS232, RS 485,...) ou parallèle	vérifier le type de connexion est conforme à la norme ou à la convention de connexion
Architecture et organisation en couches des réseaux de transmissions et mode de gestion (protocoles) de ces réseaux (CAN, Ethernet...) bus de terrain...)	identifier le protocole utilisé par la liaison et vérifier que celui-ci répond au cahier des charges de l'équipement.

S05 : Connexions entre constituants électroniques ou connexions inter systèmes

Savoirs	On se limite à:
Normalisation: - Les organismes de normalisation - Symboles graphiques de schéma d'interconnexions et structurels. Normalisations associées à la connexion des équipements: - Liaisons série et parallèle, bus d'instrumentation, signaux, brochage, protocoles associés et structures. - Liaisons point à point, multipoint, réseaux.	- utiliser la norme dans le domaine électronique et informatique - donner le rôle des couches du modèle OSI - utiliser les normes de connectique les plus courantes - utiliser les normes pour réaliser une liaison simple
Distribution électrique et sécurité. - Notions générales sur le transport et la distribution de l'énergie électrique ; rôles d'un transformateur. - Sécurité : danger d'électrocution ; limites des domaines de tension ; régime de liaison à la terre.	- donner le rôle des constituants qui interviennent dans la protection des personnes qui interviennent à proximité du courant électrique. - appliquer la réglementation relative à la sécurité des opérations en présence du courant électrique.
Liaison hertzienne Propagation, puissance rayonnée, ouverture, bilan d'une liaison	justifier les caractéristiques d'une antenne à partir des exigences d'un cahier des charges.
Liaison filaire : - Paire torsadée, câble coaxial, adaptation d'impédance, atténuation, dBm, normalisation des câbles, des connecteurs. Liaison par fibre optique: - Fibre mono-mode, saut d'indice et gradient d'indice - Longueur d'onde, bande passante, atténuation - Structure du câble, connectique. - Transducteurs. Diode PIN Laser Liaison infra rouge: - Principes mis en jeu, définition des unités associées, transducteurs.	- justifier les caractéristiques d'un câble, d'une liaison infra-rouge à partir des exigences d'un cahier des charges. - justifier la liaison à partir de la documentation technique en anglais ou en français - exploiter de la documentation technique en anglais ou en français dans le but de choisir le support filaire. - utiliser les fibres optiques dans le domaine des télécommunications

D.3.2 RELATIONS SAVOIRS COMPÉTENCES

		S01 : Acquisition et restitutions des grandeurs physiques	S02 : Traitement analogique de l'information captée	S03 : Traitement numérique de l'information et stockage des données	S04 : Transmission et transport de l'information	S05 : Connexions entre composants électroniques ou connexions inter-systèmes	S06 : Commande des actionneurs	S07 : Fabrication industrielle d'un produit électronique	S08 : Utilisation de l'outil informatique
ANALYSER l'existant	A1 : Expliciter un schéma fonctionnel	X	X	X	X		X		X
	A2 : Analyser un schéma structurel	X	X	X	X	X	X		X
	A3 : Expliciter une structure logicielle commentée	X	X	X	X		X		X
TESTER et VALIDER un équipement ou un produit électronique	T1 : Effectuer des tests en conformité avec une procédure établie sur une maquette	X	X	X	X	X	X	X	X
	T2 : Établir des procédures de tests sur une maquette	X	X	X	X	X	X	X	X
MAINTENIR et INSTALLER un équipement ou un produit en fonctionnement	M1 : Installer et configurer un nouvel équipement ou produit					X			X
	M2 : Valider le bon fonctionnement de l'équipement ou du produit					X			X
	M3 : Détecter le (ou les) composants ou le (ou les) constituants défectueux	X	X	X	X	X	X	X	X
	M4 : Remplacer le ou les constituants défectueux							X	
APPLIQUER des nouvelles solutions technologiques à partir de l'existant et d'un nouveau cahier des charges	C1 : Adapter le schéma structurel existant	X	X	X	X		X		X
	C2 : Adapter le logiciel à un nouveau cahier des charges			X	X		X		X
	C3 : Élaborer une nouvelle maquette							X	X
ÉCHANGER des connaissances électroniques	E1 : Exploiter une documentation technique en Français et en Anglais	X	X	X	X	X	X	X	X
	E2 : Rédiger un rapport d'activité, une procédure de test ou un dossier de fabrication en Français et en Anglais	X	X	X	X	X	X	x	X
	E3 : Transférer les acquis vers d'autres systèmes, objets ou structures.	X	X	X	X	X	X	X	X

(a) : Travaux pratiques d'atelier