

# CAN et CAN FD

Tout sur les protocoles  
et leur mise en œuvre avec Arduino

```

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  ACAN_ESP32_Settings
  const uint32_t errorCode = 0;
  if (0 != errorCode) {
    while (1) {
      delay(50);
      digitalWrite(LED, !digitalRead(LED_BUILTIN));
    }
  }
}

void loop() {
  if (int32_t data = gDataCAN) {
    digitalWrite(LED, !digitalRead(LED_BUILTIN));
  }
}

CANMessage msg;
if (ACAN_ESP32_Settings) {
  if (!message) {
    switch (errorCode) {
      case 0x00:
      case 0x01:
      default:
    }
  }
}
    
```



Pierre Molinaro





# CAN et CAN FD

Tout sur les protocoles  
et leur mise en œuvre avec Arduino



Pierre Molinaro

---

- **Droits de reproduction © 2024 – Publitronic - Elektor International Media**

- **Toute reproduction ou copie, même partielle, de ce livre, et sur quelque support que ce soit, sans l'accord écrit de l'éditeur, est interdite.**

- Le code de la propriété intellectuelle du 1er juillet 1992 interdit expressément la photocopie à usage collectif sans autorisation des ayants droit.

*No part of this book may be reproduced, in any form or means whatsoever, without permission in writing from the publisher. While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.*

La protection du droit d'auteur s'étend aux illustrations, y compris aux circuits imprimés et aux projets y relatifs. En conformité avec l'article 30 de la Loi sur les brevets, les circuits mentionnés ne peuvent être exécutés qu'à des fins particulières ou scientifiques et non pas dans ou pour une entreprise ; **ces exécutions et/ou applications se font en dehors de toute responsabilité de l'éditeur.**

Conformément au droit d'auteur, ce copyright ne s'applique pas à certains schémas reproduits dans ce livre à titre de citation et d'illustration des propos et de la démarche intellectuelle de l'auteur, avec l'aimable autorisation des ayants droit.

L'éditeur remercie d'avance le lecteur qui prendra la peine de lui signaler les erreurs éventuelles qui auront échappé à sa vigilance (écrire à [redaction@elektor.fr](mailto:redaction@elektor.fr)).

- **ISBN 978-2-86661-218-4      Version papier**  
**978-2-86661-219-1      Version numérique**

- 1<sup>e</sup> édition — 1<sup>er</sup> tirage - 03/2024

- Maquette : Pierre Molinaro

Imprimé aux Pays-Bas par Ipskamp Printing

# Table des matières

<b>Table des matières</b>	<b>5</b>
<b>Liste des tableaux</b>	<b>16</b>
<b>Liste des figures</b>	<b>19</b>
<b>Liste des algorithmes</b>	<b>25</b>
<b>Avant-propos</b>	<b>27</b>
<b>1 Lignes de transmission</b>	<b>32</b>
1.1 Un exemple illustratif : connexion directe d'un interrupteur	33
1.1.1 Connexion directe	33
1.1.2 Simulation Spice d'une connexion directe	35
1.1.3 Simulation Spice de la fermeture, avec diodes de clamping	37
1.1.4 Simulation Spice, diminution de $R_{pullup}$	39
1.1.5 Simulation Spice d'une connexion directe, avec une résistance en série	40
1.2 Connexion d'un interrupteur via une ligne différentielle	41
1.3 Théorie succincte des lignes de transmission	42
1.4 Ligne sans perte	44
1.5 Ligne avec perte	45
1.6 Simulations Spice d'une ligne sans perte	46
1.6.1 Ligne sans perte ouverte	46
1.6.2 Ligne sans perte terminée par $R_C$	47
1.6.3 Ligne sans perte terminée par une valeur différente de $R_C$	47
<b>2 Brève introduction aux réseaux CAN et CAN FD</b>	<b>49</b>
2.1 Cible des réseaux CAN	50
2.2 Quel type de réseau ?	51
2.3 Intérêt d'un réseau CAN	52
2.4 Bref historique du réseau CAN	53
2.5 La normalisation	54
2.6 Caractéristiques du réseau CAN	54
2.7 Protocoles applicatifs au dessus de CAN	55

<b>3</b>	<b>Réseau, bus, contrôleur et transcepteur</b>	<b>56</b>
3.1	Distinction entre réseau CAN et bus CAN	56
3.2	Contrôleurs CAN	57
3.3	Connexion d'un contrôleur CAN à un réseau	58
3.4	Fonction logique d'un réseau CAN : états <i>dominant</i> , <i>récessif</i>	58
3.4.1	Les signaux TxD et RxD des contrôleurs	59
3.4.2	La fonction logique du réseau	60
3.4.3	État du réseau, vu par un contrôleur	61
3.5	Des exemples de réseaux sans transcepteur	61
3.5.1	Logique du réseau réalisée avec une porte « <i>et</i> »	62
3.5.2	Logique du réseau réalisée par un « <i>et</i> » câblé	62
3.5.3	Logique du réseau réalisée par des diodes	63
3.5.4	Contrôleurs alimentés sous tensions différentes	64
3.6	Délai de boucle du réseau	64
<b>4</b>	<b>Transcepteur CAN</b>	<b>66</b>
4.1	Fonctionnement d'un transcepteur CAN	67
4.1.1	L'émetteur	68
4.1.2	Les deux états du bus CAN	69
4.1.3	Récepteur	69
4.1.4	Transcepteur CAN seul sur un bus	70
4.2	Exemples de transcepteurs CAN	70
4.2.1	Transcepteurs CAN alimentés sous 5 V	71
4.2.2	Transcepteurs CAN alimentés sous 3,3 V	72
4.2.3	Transcepteurs CAN bi-tension	73
4.3	Fonctionnalités des transcepteurs CAN	74
4.3.1	Tirage intégré sur T	74
4.3.2	Résistance équivalente à un bus CAN	75
4.3.3	Nombre maximum de transcepteurs sur un bus	76
4.3.4	Tensions de CANH et de CANL	77
4.3.4.1	Bus <i>dominant</i>	77
4.3.4.2	Bus <i>récessif</i>	78
4.3.4.3	Transitions entre états <i>dominant</i> et <i>récessif</i>	78
4.3.4.4	Détection de l'état du bus	79
4.3.4.5	Transcepteurs alimentés sous 3,3 V	79
4.3.5	Les différentes terminaisons de bus	81
4.3.5.1	La terminaison standard	81
4.3.5.2	La terminaison « SPLIT »	81
4.3.6	Protection tension d'alimentation basse et hors tension	83
4.3.7	Protection contre un état <i>dominant</i> permanent	83
4.3.8	Protection contre les surtensions	84
4.3.9	Contrôle de pente par résistance externe, mode de repos	85

4.3.10	Temps de boucle d'un transcepteur	85
4.4	Configurations particulières	86
4.4.1	Plusieurs contrôleurs CAN connectés au même transcepteur	86
4.4.2	Connexion avec isolation galvanique	86
<b>5</b>	<b>Bus, câble et connectique</b>	<b>89</b>
5.1	Bus et autres topologies	90
5.1.1	Disposition acceptable	90
5.1.2	Disposition recommandée	90
5.1.3	Longueur maximum d'un bus CAN	90
5.1.4	Autres topologies : à éviter	91
5.2	Simulations Spice de réseaux en étoile	92
5.2.1	Branches de même longueur, une résistance à l'extrémité de deux branches	93
5.2.1.1	Transition <i>récessif</i> → <i>dominant</i>	93
5.2.1.2	Transition <i>dominant</i> → <i>récessif</i>	95
5.2.2	Branches de même longueur, une résistance à l'extrémité de chaque branche	95
5.2.2.1	Transition <i>récessif</i> → <i>dominant</i>	96
5.2.2.2	Transition <i>dominant</i> → <i>récessif</i>	96
5.2.3	Deux terminaisons, branches inégales	97
5.2.3.1	Transition <i>récessif</i> → <i>dominant</i>	97
5.2.3.2	Transition <i>dominant</i> → <i>récessif</i>	97
5.2.4	Résistances sur toutes les extrémités, branches inégales	98
5.2.4.1	Transition <i>récessif</i> → <i>dominant</i>	98
5.2.4.2	Transition <i>dominant</i> → <i>récessif</i>	98
5.3	Câbles	100
5.3.1	Chute de tension dans le bus et section des conducteurs	101
5.3.2	Câble conçu pour le bus CAN	103
5.3.3	Câble en ruban	103
5.3.4	Câble plat téléphonique	104
5.4	Connectique	104
5.4.1	Connecteurs recommandés par le rapport technique CiA 106	105
5.4.2	La connectique RJ	106
5.4.2.1	Les connecteurs mâles <i>xPyC</i>	106
5.4.2.2	Les embases femelles <i>xPyC</i>	107
5.4.2.3	Les cordons Ethernet RJ45	107
5.4.2.4	Les cordons téléphoniques RJ12	109
5.4.2.5	Sertir ses propres câbles	109
5.4.3	La connectique IEEE 1394	110
5.4.3.1	Format des embases	111
5.4.3.2	Format des cordons	111
5.4.3.3	Adaptation à un équipement mobile	112
5.5	Choix des signaux véhiculés par les câbles réseau	113

5.5.1	Toutes les cartes ont la même référence de tension	113
5.5.2	Les cartes n'ont pas de référence commune	114
5.5.3	Les cartes ont des références de tension présentant un décalage	115
<b>6</b>	<b>Répéteur et passerelle</b>	<b>116</b>
6.1	Répéteur, passerelle	116
6.2	Intérêt d'un répéteur ou d'une passerelle	117
6.3	Principe d'une passerelle CAN	118
6.4	Principe d'un répéteur CAN	119
6.4.1	La fonction logique d'arbitrage n'est pas combinatoire	120
6.5	Exemples de répéteurs CAN	121
6.5.1	Répéteur réalisé avec des circuits intégrés logiques	122
6.5.2	Les AMIS-42700 et AMIS-42770 de ON Semiconductors	122
6.5.2.1	Description du AMIS-42700 et du AMIS-42770	124
6.5.2.2	Un répéteur entre deux bus CAN	124
6.5.2.3	Un répéteur entre deux bus CAN associé à un nœud CAN	124
6.5.2.4	Un répéteur entre quatre bus CAN	125
6.5.2.5	Un répéteur entre six bus CAN ou plus	126
<b>7</b>	<b>Protocole CAN 2.0B</b>	<b>128</b>
7.1	Les différents types de trame	129
7.2	Le réseau au repos	130
7.3	Format d'une trame de données et de requête	130
7.3.1	Champ SOF : début de trame	132
7.3.2	Champs ARBITRATION et CONTROL : arbitrage et contrôle	132
7.3.3	Champ DATA : données	134
7.3.4	Champ CRC : somme de contrôle	135
7.3.5	Champ ACK : acquittement	137
7.3.6	Champ EOF : fin de la trame	138
7.3.7	Champ IPS : espace inter-trames	138
7.3.8	Exemples de trames CAN	138
7.4	Validation de la transmission	140
7.5	Décision d'émettre : condition <i>bus libre</i>	140
7.6	Synchronisation forte	141
7.7	Synchronisation des contrôleurs en réception	141
7.8	Le <i>bit stuffing</i>	142
7.9	Cohabitation des trames standards et étendues	143
7.10	Résolution des collisions	144
7.10.1	CAN : collision non destructive	144
7.10.2	Re-émission des trames	146
7.10.3	Collision d'identificateurs	146
7.11	Activité d'un contrôleur CAN vis-à-vis du réseau	147
7.12	Longueur des trames CAN	148

7.13	Débit utile d'un réseau CAN . . . . .	150
<b>8</b>	<b>Détection et gestion des erreurs</b>	<b>151</b>
8.1	Les différents types d'erreurs d'un réseau CAN . . . . .	152
8.1.1	L'inversion de bit . . . . .	152
8.1.2	L'insertion et la suppression de bit . . . . .	153
8.1.3	L'erreur en rafale . . . . .	155
8.2	Les erreurs détectées par un contrôleur CAN . . . . .	155
8.2.1	Erreur de bit ( <i>bit error</i> ) . . . . .	156
8.2.2	Erreur de <i>bit stuffing</i> ( <i>stuff error</i> ) . . . . .	156
8.2.3	Erreur de forme ( <i>form error</i> ) . . . . .	156
8.2.4	Erreur de somme de contrôle ( <i>CRC error</i> ) . . . . .	156
8.2.5	Erreur d'acquiescement ( <i>acknowledgment error</i> ) . . . . .	157
8.3	Gestion des erreurs . . . . .	157
8.3.1	Compteurs d'erreurs . . . . .	157
8.3.2	Modes d'un contrôleur CAN vis-à-vis du réseau . . . . .	158
8.3.3	Comportement d'un contrôleur CAN vis-à-vis du réseau . . . . .	158
8.3.4	Champ SUSPEND TRANSMISSION . . . . .	159
8.3.5	Sortir du mode BUS OFF . . . . .	159
8.4	Signalement : les trames d'erreur . . . . .	160
8.4.1	Trame d'erreur active (ACTIVE ERROR FRAME) . . . . .	160
8.4.1.1	Algorithme . . . . .	160
8.4.1.2	Erreurs détectées durant l'émission . . . . .	161
8.4.1.3	Discussion . . . . .	161
8.4.2	Trame d'erreur passive (PASSIVE ERROR FRAME) . . . . .	162
8.4.2.1	Algorithme . . . . .	162
8.4.2.2	Erreurs détectées durant l'émission . . . . .	162
8.4.2.3	Discussion . . . . .	163
8.4.3	Trame de surcharge (OVERLOAD FRAME) . . . . .	163
8.4.3.1	Algorithme . . . . .	163
8.4.3.2	Erreurs détectées durant l'émission . . . . .	164
8.4.3.3	Discussion . . . . .	164
8.5	Addendum à la spécification CAN . . . . .	164
8.5.1	Champ DLC supérieur à 8 . . . . .	165
8.5.2	Bit SRR <i>dominant</i> . . . . .	165
8.5.3	Dernier bit du champ EOF reçu <i>dominant</i> . . . . .	165
8.5.4	Incrémentation du REC quand le mode ERROR PASSIVE est atteint . . . . .	166
8.5.5	Conditions complémentaires de prise en compte de la synchronisation forte . . . . .	166
8.5.6	Implémentation du compteur d'erreurs de transmission TEC . . . . .	167
<b>9</b>	<b>Fiabilité CAN 2.0B</b>	<b>168</b>
9.1	Duplication de l'émission d'une trame . . . . .	169
9.2	Inversions de bits dans les trames standards sans octet de données . . . . .	169

9.3	Inversions de bits dans les trames standards de requête . . . . .	170
9.4	Inversions de bits entre trames standard sans donnée et trames de requête . . . . .	171
9.5	Inversions de bits entre trames standard d'un octet de donnée . . . . .	171
9.6	Inversions de bits entre deux trames étendues sans octet de données . . . . .	173
9.7	Inversions de bits entre deux trames standard de deux octets de données . . . . .	175
9.8	Altération du champ DLC : exemple de double inversion de bits non détectée . . . . .	175
9.8.1	Construction du scénario . . . . .	175
9.8.2	Un exemple pratique . . . . .	177
<b>10</b>	<b>Décomposition du bit CAN</b>	<b>179</b>
10.1	Normalisation de la description du bit CAN . . . . .	179
10.2	Paramètres programmables . . . . .	180
10.3	Décomposition d'un bit CAN . . . . .	180
10.4	Synchronisation d'un récepteur, durée de propagation . . . . .	181
10.5	Resynchronisation et SJW . . . . .	183
10.6	Positions des points d'échantillonnage . . . . .	185
<b>11</b>	<b>Contraintes de paramétrage</b>	<b>186</b>
11.1	Tolérance des horloges . . . . .	186
11.1.1	Une seule horloge . . . . .	186
11.1.2	Systèmes à deux horloges . . . . .	187
11.2	Tolérance sur la durée nominale d'un bit . . . . .	188
11.3	Récapitulatif des contraintes de resynchronisation . . . . .	188
11.4	Tolérance maximum de l'horloge . . . . .	189
<b>12</b>	<b>Quelques contrôleurs CAN</b>	<b>190</b>
12.1	Quelques fonctionnalités des contrôleurs CAN . . . . .	190
12.1.1	Modes <i>loopback</i> , <i>listen only</i> , <i>self reception</i> . . . . .	190
12.1.2	Buffers d'émission . . . . .	191
12.1.3	Nombre de re-émissions . . . . .	192
12.1.4	Quitter le mode BUS OFF . . . . .	192
12.1.5	Filtrage des trames reçues . . . . .	192
12.2	Quelques contrôleurs CAN externes . . . . .	194
12.2.1	Anciens contrôleurs . . . . .	194
12.2.2	Le contrôleur SJA1000 . . . . .	195
12.2.3	Le contrôleur MCP2515 . . . . .	196
12.2.4	Le contrôleur HI-311x . . . . .	196
12.3	Quelques contrôleurs CAN intégrés dans un microcontrôleur . . . . .	197
12.3.1	Module TWAI de l'ESP32 . . . . .	197
12.3.2	Les <i>bxCAN</i> des STM32 . . . . .	197
12.3.3	FlexCAN de NXP . . . . .	198
12.3.4	Raspberry Pi RP2040 . . . . .	199

<b>13 Codes de calcul CAN</b>	<b>201</b>
13.1 Codes sources	201
13.2 Paramétrage pour atteindre un débit donné	202
13.2.1 Calcul de $D_{BRP}$ et de $N$ pour 375 kbit/s	203
13.2.2 Calcul de $N_{PROP\_SEG}$ , $N_{PS_1}$ , $N_{PS_2}$ et $S_{JW}$	206
13.2.3 Calcul de $D_{BRP}$ et de $N$ pour 250 kbit/s	208
13.3 Calcul de $D_{BRP}$ , de $N$ et de la fréquence du quartz pour 615 kbit/s	208
13.4 Vérifier la décomposition du bit CAN pour un MCP2515	209
13.5 Débits communs à deux contrôleurs CAN	211
13.5.1 Configurations du bit CAN du LPC 2294	211
13.5.2 Configurations du bit CAN du PIC 18F26K80	212
13.5.3 Construire la liste ordonnée des débits proches	212
13.5.4 Afficher les débits proches	212
13.6 Classe CANFrameGenerator	213
13.6.1 Constructeur : calcul de la trame	213
13.6.2 Obtention des caractéristiques d'une trame	214
13.6.3 Tester le générateur de trames	215
13.6.4 Affichage de deux trames CAN	218
13.6.5 Comparer deux trames CAN	218
13.7 Bornes effectives de la longueur des trames CAN	219
<b>14 Le protocole CAN FD</b>	<b>221</b>
14.1 Le protocole CAN FD en bref	222
14.2 Les types de trames CAN FD	222
14.3 Format d'une trame CAN FD	223
14.3.1 Champs ARBITRATION et CONTROL	223
14.3.2 Champ CRC	226
14.3.3 Insertion forcée d'un bit de <i>stuff</i>	229
14.3.4 Champ STUFF BIT COUNT	229
14.4 Exemples de trames CAN FD	230
14.5 Décomposition du bit CAN FD	233
14.6 Le changement de débit en détail	234
14.7 Compensation du délai de boucle du transmetteur	237
14.7.1 Trame transmise au débit d'arbitrage	237
14.7.2 Changement de débit, pas de compensation du délai de boucle	238
14.7.3 Changement de débit, avec compensation du délai de boucle	238
14.8 Longueur des trames CAN FD	240
14.8.1 Longueur des trames CAN FD, pas de changement de débit	240
14.8.2 Longueur des trames CAN FD, avec changement de débit	242
14.9 Débit utile d'un réseau CAN FD ISO	243
14.10 Activité d'un contrôleur CAN FD vis-à-vis du réseau	243
14.11 Recommandations pour la configuration du bit CAN FD	245

14.11.1	Les trois classes de réseau CAN FD . . . . .	245
14.11.2	Les recommandations CiA 601-3 . . . . .	247
<b>15</b>	<b>Codes de calcul CAN FD</b>	<b>249</b>
15.1	Classe CANFDFrameGenerator : générateur de trames CAN FD . . . . .	249
15.1.1	Constructeur : calcul de la trame . . . . .	249
15.1.2	Obtention des caractéristiques d'une trame CAN FD . . . . .	250
15.1.3	Tester le générateur de trames CAN FD . . . . .	251
15.2	La configuration du bit CAN FD en pratique . . . . .	254
<b>16</b>	<b>Fiabilité CAN FD</b>	<b>262</b>
16.1	Simple inversion de bit . . . . .	262
16.1.1	Validation de la transmission d'une trame . . . . .	263
16.1.2	Inversion du bit r0 . . . . .	263
16.2	Inversions de bits dans les trames de base sans octet de données . . . . .	263
16.3	Altération du champ DLC : double inversion de bit non détectée . . . . .	266
16.4	Altération du bit S0F : une vulnérabilité du protocole <i>non-ISO</i> . . . . .	266
<b>17</b>	<b>Transcepteurs CAN FD et CAN SIC</b>	<b>271</b>
17.1	Transcepteurs CAN FD . . . . .	271
17.2	Transcepteurs CAN SIC . . . . .	273
<b>18</b>	<b>Contrôleurs CAN FD</b>	<b>276</b>
18.1	Le MCP2518FD de Microchip . . . . .	276
18.2	Le TCAN4550-Q1 de Texas Instruments . . . . .	277
18.3	Le module M_CAN de Robert Bosch GmbH . . . . .	277
<b>19</b>	<b>Utiliser les services d'un pilote</b>	<b>281</b>
19.1	Pilote logiciel . . . . .	281
19.2	Structuration en couches . . . . .	282
19.3	Configuration du pilote . . . . .	282
19.4	Émission d'un message . . . . .	285
19.5	Réception d'un message . . . . .	286
<b>20</b>	<b>Concevoir une messagerie</b>	<b>287</b>
20.1	Éliminer les risques de collision fatale . . . . .	288
20.2	Rôle applicatif des identificateurs . . . . .	288
20.3	Choix des identificateurs . . . . .	288
20.4	Transmettre des états . . . . .	289
20.5	Données fragmentées . . . . .	290
20.6	Trafic périodique ou évènementiel? . . . . .	292
20.7	Évaluer la charge du bus . . . . .	293
20.8	Surveiller l'occupation des buffers du pilote . . . . .	294
20.9	Surveiller l'état du contrôleur vis-à-vis du réseau . . . . .	295

<b>21 Utiliser un analyseur logique</b>	<b>296</b>
21.1 Le signal affiché n'est pas le signal réel	296
21.2 Quel analyseur choisir ?	297
21.3 Exemples de décodage de trames CAN	298
21.4 Quels signaux observer ?	299
21.5 Exemples d'observation avec un seul contrôleur CAN	300
21.5.1 Signal RxD en permanence <i>récessif</i>	301
21.5.2 Signal RxD en permanence <i>dominant</i>	302
21.5.3 Contrôleur seul sur le réseau : non acquittement des trames	304
21.5.4 CAN FD : évaluer les positions des points d'échantillonnage	305
21.6 Exemples d'observation avec deux contrôleurs CAN	308
21.6.1 Acquittement de la réception d'une trame	308
21.6.2 La perte d'arbitrage	309
21.6.3 Collision entre trames identiques	310
21.6.4 Collision entre trames ayant même champ ARBITRATION	312
21.6.5 Exemple d'un <i>active error flag</i> de 12 bits	315
21.7 Observations du temps de réponse et de la perte d'arbitrage	316
21.7.1 Le matériel	316
21.7.2 Le code de $S_0$	317
21.7.3 Le code de $S_1$ , $S_2$ et $S_3$	319
21.7.4 La séquence des trames sur le réseau	320
<b>22 Exemples d'applications</b>	<b>322</b>
22.1 Commande d'une LED	323
22.1.1 Définition de la messagerie	325
22.1.2 Définition de l'état initial	325
22.1.3 Premier cahier des charges : la led reflète l'état du poussoir	325
22.1.3.1 Station LED	325
22.1.3.2 Station LED, variante si émission périodique	326
22.1.3.3 Station POUSSOIR, émission périodique	328
22.1.3.4 Station POUSSOIR, émission événementielle	329
22.1.3.5 Station POUSSOIR, émission événementielle, filtrage des rebonds	330
22.1.3.6 Station POUSSOIR, émission événementielle et périodique	332
22.1.4 Second cahier des charges : un appui complémente l'état de la led	332
22.2 Système d'acquisition	334
22.2.1 Définition de la messagerie	334
22.2.2 Émission spontanée	335
22.2.3 Émission sur demande	337
<b>23 Les évolutions futures</b>	<b>340</b>
23.1 CAN FD Light	340
23.2 CAN XL	341

<b>Annexes</b>	<b>342</b>
<b>A Le logiciel de simulation Spice</b>	<b>343</b>
A.1 Présentation succincte	343
A.2 Modèle d'une résistance, d'un condensateur	344
A.3 Modèle d'une source de tension fixe	345
A.4 Modèle d'une source de tension impulsionnelle	346
A.5 Modèle d'une ligne sans perte	346
A.6 Modèle d'une ligne avec perte	347
A.7 Modèle d'un interrupteur commandé en tension	347
<b>B Écrire un code Arduino</b>	<b>349</b>
B.1 Les types <code>int</code> et <code>unsigned</code>	349
B.2 Types d'entiers indépendants de la plateforme	350
B.3 Constantes littérales entières	350
B.4 La fonction de <code>lay</code> ne permet pas de respecter une période	351
B.5 Programmer des dates d'exécution, première écriture	352
B.6 Programmer des dates d'exécution, amélioration	353
B.7 Organisation d'un code Arduino	354
B.8 Discussion du débordement de <code>millis</code>	354
B.8.1 Rattrapage du dépassement d'échéance, débordement non géré	355
B.8.2 Rattrapage du dépassement d'échéance, débordement géré	356
B.8.3 Rattrapage du dépassement d'échéance, débordement mal géré	358
<b>C Contrôle de redondance cyclique</b>	<b>360</b>
C.1 Les opérations $\oplus$ et $\odot$ dans $\mathbb{N}$	361
C.2 $\mathbb{Z}/2\mathbb{Z}$	362
C.3 Les polynômes sur $\mathbb{Z}/2\mathbb{Z}$	362
C.4 Le point sur les notations	364
C.5 Notation polynomiale d'un nombre entier	364
C.6 Données, somme de contrôle, message	365
C.7 Un algorithme de calcul de la somme de contrôle	366
C.8 Représentations du polynôme	369
C.9 Influence de la valeur initiale <code>SEED</code>	370
C.10 Exemples	370
C.11 Vérification de la somme de contrôle à la réception	372
C.12 Distance et poids de Hamming d'un code	372
C.13 Propriétés des sommes de contrôle	373
C.13.1 Terme constant de <code>POLYNÔME</code>	373
C.13.2 Détection de toute erreur simple	374
C.13.3 Divisibilité d'un polynôme par $X + 1$	374
C.13.4 Détection d'un nombre impair d'erreurs	375
C.13.5 Détection d'une erreur en rafale	375

---

C.13.6	Borne supérieure de la distance de Hamming d'un code . . . . .	376
C.14	Rechercher un <i>bon</i> polynôme . . . . .	377
C.14.1	Logiciel d'évaluation d'un polynôme . . . . .	378
C.14.2	Choisir la valeur de SEED . . . . .	379
C.15	Les polynômes de CAN 2.0B et CAN FD . . . . .	380
C.15.1	CAN 2.0B, polynôme CRC15CAN . . . . .	380
C.15.2	CAN FD, polynôme CRC17CANFD . . . . .	380
C.15.3	CAN FD, polynôme CRC21CANFD . . . . .	381
<b>Bibliographie</b>		<b>382</b>
<b>Index</b>		<b>386</b>

# Liste des tableaux

3.1	État du réseau, vu par un contrôleur	61
4.1	Les deux états du bus CAN	69
4.2	Exemples de transcepteurs CAN 5 V	71
4.3	Exemples de transcepteurs CAN 3,3 V	73
4.4	Exemples de transcepteurs CAN bi-tension	74
4.5	Encadrement de $R_{DIFF}$	76
4.6	Transcepteurs 5 V et bi-tension, $V_{CANH}$ , $V_{CANL}$ et $V_{DIFF}$ délivrées dans l'état <i>dominant</i>	78
4.7	Transcepteurs 3,3 V, $V_{CANH}$ , $V_{CANL}$ et $V_{DIFF}$ délivrées dans l'état <i>dominant</i>	80
4.8	Contrôle de pente par résistance externe	85
5.1	Longueur maximum d'un bus CAN en fonction de son débit	91
5.2	Tableau de correspondance AWG	100
5.3	Caractéristiques résumées de quelques câbles CAN Helukabel	103
5.4	Caractéristiques résumées de quelques câbles en ruban	104
5.5	Affectation des broches d'une embase IEEE 1394	111
7.1	Nombre d'octets de la trame, en fonction du champ DLC	134
7.2	Deux exemples de trame CAN 2.0B	139
7.3	Longueur d'une trame CAN, en fonction de son type et du champ DLC	148
7.4	Calcul de la longueur d'une trame de données	149
7.5	Débit utile d'un réseau CAN, pour un débit de 1 Mbit/s	150
8.1	Compteurs d'erreurs	157
8.2	Actions sur les compteurs d'erreurs	157
8.3	Modes d'un contrôleur CAN vis-à-vis du réseau	158
8.4	Comportement d'un contrôleur CAN vis-à-vis du réseau, en fonction de son mode	158
8.5	Condition et instant d'émission d'une trame de surcharge	164
9.1	Distance de Hamming entre trames standard de données sans donnée	170
9.2	Exemple de quadruple inversion de bit non détectée entre deux trames standard sans octet de données	170
9.3	Distance de Hamming entre trames standard de requête	171
9.4	Exemple de triple inversion de bit non détectée entre deux trames standards de requête	172

9.5	Triples inversions de bit non détectées entre deux trames standards de requête et une trame standard sans octet de donnée . . . . .	172
9.6	Distance de Hamming entre trames standards d'un octet de donnée . . . . .	172
9.7	Exemple de double inversion de bit non détectée entre deux trames standards d'identificateur 0x65 contenant un octet de données . . . . .	173
9.8	Exemple de double inversion de bit non détectée entre deux trames étendues sans octet de données . . . . .	174
9.9	Doubles inversions de bit non détectées entre deux trames standards de deux octets . . . . .	175
9.10	Exemple de double inversion de bit non détectée . . . . .	178
10.1	Paramètres définissant le bit CAN . . . . .	180
11.1	Résumé des contraintes de paramétrage du bit CAN . . . . .	189
12.1	Principe du filtrage par <i>masque d'acceptation</i> et <i>filtre d'acceptation</i> . . . . .	193
12.2	Exemple d'acceptation par <i>masque d'acceptation</i> et <i>filtre d'acceptation</i> . . . . .	194
12.3	Exemple de refus par <i>masque d'acceptation</i> et <i>filtre d'acceptation</i> . . . . .	194
12.4	Cartes Nucleo . . . . .	197
12.5	Cartes Teensy 3.x . . . . .	198
13.1	Fichiers contenant les codes sources . . . . .	202
13.2	Plages des paramètres du bit CAN du MCP2515 . . . . .	203
13.3	Décomposition du bit CAN pour un MCP2515 . . . . .	210
13.4	Débits du LPC 2294 et PIC 18F26K80 proches . . . . .	212
13.5	Exemple d'affichage des champs d'une trame CAN . . . . .	215
13.6	Durées des calculs exhaustifs des longueurs de trames CAN . . . . .	219
13.7	Bornes des longueurs effectives de trames CAN . . . . .	219
13.8	Longueur effective de quelques trames CAN . . . . .	220
14.1	Nombre d'octets d'une trame CAN FD, en fonction du champ DLC . . . . .	226
14.2	Polynômes utilisés pour les sommes de contrôle CAN et CAN FD . . . . .	227
14.3	Composition du champ STUFF BIT COUNT . . . . .	230
14.4	Premier exemple de trames CAN FD . . . . .	231
14.5	Second exemple de trames CAN FD . . . . .	232
14.6	Paramètres définissant les bits CAN FD . . . . .	233
14.7	Plages des paramètres définissant les bits CAN FD . . . . .	234
14.8	Durée des bits d'une trame CAN FD . . . . .	236
14.9	Calcul de la longueur d'une trame CAN FD ISO sans changement de débit . . . . .	241
14.10	Longueur d'une trame CAN FD ISO sans changement de débit . . . . .	241
14.11	Calcul de la longueur d'une trame CAN FD ISO avec changement de débit . . . . .	242
14.12	Débit utile d'un réseau CAN FD ISO, débit d'arbitrage et de données de 1 Mbit/s . . . . .	244
14.13	Débit utile d'un réseau CAN FD ISO, débit d'arbitrage de 1 Mbit/s, de données de 8 Mbit/s . . . . .	245

15.1	Décomposition du bit CAN FD pour 500 kbit/s – 2 Mbit/s, échantillonnage à 80 % . . . . .	258
15.2	Décomposition du bit CAN FD pour 500 kbit/s – 5 Mbit/s, échantillonnage à 80 % . . . . .	261
16.1	Exemple de trames de base <i>non ISO</i> sans octet de données, distantes de 6 bits . . . . .	264
16.2	Exemple de trames de base <i>ISO</i> sans octet de données, distantes de 6 bits . . . . .	265
16.3	Exemple de double inversion de bits non détectée, protocole <i>non ISO</i> . . . . .	267
16.4	Exemple de double inversion de bits non détectée, protocole <i>ISO</i> . . . . .	268
16.5	Altération du bit S0F, exemple de vulnérabilité du protocole <i>non-ISO</i> . . . . .	270
17.1	Exemples de transcepteurs CAN SIC . . . . .	275
18.1	Taille d'un élément des zones <i>Tx FIFO 0, Tx FIFO 1, Rx Buffers</i> et <i>Tx Buffers</i> . . . . .	278
18.2	Composition d'un élément de la zone <i>11-bit Filters</i> . . . . .	279
18.3	Condition d'acceptation de la trame reçue d'identificateur <i>idf</i> . . . . .	279
18.4	Destination de la trame reçue d'identificateur <i>idf</i> . . . . .	280
18.5	Composition d'un élément de la zone <i>29-bit Filters</i> . . . . .	280
20.1	Exemple de description d'une trame d'une messagerie . . . . .	289
20.2	Fragments d'un bloc de 32 octets . . . . .	291
22.1	Trace de l'exécution de la station GESTIONNAIRE, émission spontanée . . . . .	337
22.2	Trace de l'exécution de la station GESTIONNAIRE, émission sur demande . . . . .	339
A.1	Suffixes multiplicateurs . . . . .	344
B.1	Types d'entiers indépendants de la plateforme . . . . .	350
B.2	Trace de l'exécution du croquis débordement–non–gere. <i>ino</i> . . . . .	356
B.3	Trace de l'exécution du croquis débordement–gere. <i>ino</i> . . . . .	357
B.4	Trace de l'exécution du croquis débordement–mal–gere. <i>ino</i> . . . . .	359
C.1	Exemple d'opérations + et $\oplus$ . . . . .	361
C.2	Exemple d'opérations $\times$ et $\odot$ . . . . .	361
C.3	Opérations booléennes $\boxplus$ et $\boxminus$ . . . . .	362
C.4	Représentation polynomiale de 10 et de 13 . . . . .	364
C.5	Représentation numérique « POLYNÔME » d'un polynôme . . . . .	369
C.6	Représentation « NORMAL » d'un polynôme . . . . .	369
C.7	Représentation « KOOPMAN » d'un polynôme . . . . .	369
C.8	Quelques exemples de sommes de contrôle pour $N = 16$ et $P = 8$ . . . . .	371
C.9	Exemple de calcul de distance de Hamming . . . . .	373
C.10	Quelques bons polynômes pour une somme de contrôle 8 bits ( $P = 8$ ) . . . . .	377
C.11	Sommes de contrôle pour POLYNÔME= 0x1F3 avec diverses valeurs de SEED . . . . .	379

# Liste des figures

1.1	Comment connecter un interrupteur à un microcontrôleur ? . . . . .	33
1.2	Connexion directe d'un interrupteur à un microcontrôleur . . . . .	33
1.3	Schéma Spice fermeture et ouverture d'un interrupteur . . . . .	35
1.4	Code de la simulation de la fermeture et de l'ouverture d'un interrupteur . . . . .	36
1.5	Résultats de la simulation de la fermeture et de l'ouverture d'un interrupteur . . . . .	37
1.6	Schéma Spice fermeture et ouverture d'un interrupteur, avec diodes de <i>clamping</i> . . . . .	38
1.7	Code Spice de la simulation de la fermeture d'un interrupteur, avec des diodes de <i>clamping</i> . . . . .	38
1.8	Simulation de la fermeture d'un interrupteur, prise en compte des diodes de <i>clamping</i> . . . . .	39
1.9	Résultats de la simulation de la fermeture et de l'ouverture d'un interrupteur, $R_{pullup} = 5\text{ k}\Omega$ . . . . .	39
1.10	Schéma Spice de la simulation fermeture et ouverture d'un interrupteur en série avec $R_c$ . . . . .	40
1.11	Code Spice de la simulation de la fermeture d'un interrupteur en série avec $R_c$ . . . . .	40
1.12	Code Spice de la simulation de l'ouverture d'un interrupteur en série avec $R_c$ . . . . .	41
1.13	Simulation fermeture et ouverture d'un interrupteur en série avec $R_c$ . . . . .	41
1.14	Connexion par ligne différentielle d'un interrupteur à un microcontrôleur . . . . .	42
1.15	Connexion par ligne différentielle d'un interrupteur à un microcontrôleur, schéma complet . . . . .	43
1.16	Ligne de transmission de longueur $l$ . . . . .	43
1.17	Décomposition d'une ligne de transmission . . . . .	43
1.18	Ligne sans perte de longueur infinie . . . . .	45
1.19	Ligne sans perte de longueur finie, terminée par $R_c$ . . . . .	45
1.20	Simulation Spice d'une ligne sans perte ouverte . . . . .	46
1.21	Simulation Spice d'une ligne sans perte terminée par $R_C$ . . . . .	47
1.22	Simulation Spice d'une ligne sans perte terminée par une résistance différente de $R_C$ . . . . .	48
2.1	Stations et bus CAN . . . . .	49
3.1	Schéma de principe d'un réseau CAN . . . . .	57
3.2	Réseau et bus CAN, détail de la connexion entre contrôleur et transcepteur . . . . .	57
3.3	Contrôleur CAN intégré ou externe . . . . .	58
3.4	Interface réseau d'un contrôleur CAN . . . . .	58
3.5	Schéma de principe d'un réseau CAN réunissant trois contrôleurs . . . . .	59
3.6	Contrôleur CAN, sortie TxD push/pull : active bas / active haut . . . . .	60
3.7	Contrôleur CAN, Sortie TxD collecteur ouvert : active bas / haute impédance . . . . .	60

3.8	La logique du réseau CAN et fonction « <i>et</i> »	60
3.9	Simple réseau avec une porte « <i>et</i> »	62
3.10	Simple réseau, réseau en logique câblée	62
3.11	Simple réseau, logique du réseau réalisée par des diodes	63
3.12	Contrôleurs alimentés sous des tensions différentes	64
3.13	Délai de boucle d'un réseau	64
3.14	Exemples de connexions donnant des temps de boucle différents	65
4.1	Transcepteurs et bus CAN (schéma de principe)	67
4.2	Schéma de principe d'un transcepteur CAN	67
4.3	Schéma de principe de la partie émetteur du transcepteur CAN	68
4.4	Les deux états de l'émetteur du transcepteur CAN	68
4.5	Principe de fonctionnement du récepteur du transcepteur CAN	69
4.6	Transcepteur CAN seul sur un bus	70
4.7	Disposition des broches des transcepteurs CAN	71
4.8	Connexion d'un transcepteur alimenté sous 5 V	72
4.9	Connexion d'un transcepteur alimenté sous 3,3 V	73
4.10	Schéma de connexion d'un transcepteur bi-tension	74
4.11	Tirage intégré sur l'entrée T	74
4.12	Résistance d'entrée différentielle	75
4.13	Tensions typiques de CANH et CANL, transcepteurs 5 V et bi-tension	77
4.14	Détection de l'état du bus, transcepteurs 5V et bi-tension	79
4.15	Tensions typiques de CANH et CANL, transcepteurs 3,3 V	80
4.16	Détection de l'état du bus, transcepteurs 3,3 V	80
4.17	Terminaison standard	81
4.18	Terminaison « SPLIT »	82
4.19	Bus à deux fils	83
4.20	Connexion de diodes de suppression de tensions transitoires bidirectionnelles	84
4.21	Connexion de contrôleurs CAN à un transcepteur via une porte « <i>et</i> »	86
4.22	Connexion de contrôleurs CAN dont la sortie est du type <i>collecteur ouvert</i> à un transcepteur	87
4.23	Connexion du transcepteur ISO 1050 de Texas Instruments	87
4.24	Isolation galvanique à l'aide d'optocoupleurs	88
5.1	Transcepteurs et bus CAN (schéma de principe)	89
5.2	Disposition acceptable	90
5.3	Disposition recommandée	90
5.4	Exemple de réseau CAN en étoile, terminaisons sur deux stations	92
5.5	Exemple de réseau CAN en deux étoiles reliées par un bus, terminaisons sur deux stations	92
5.6	Exemple de réseau CAN en deux étoiles reliées par un bus, terminaisons réparties	92
5.7	Schéma Spice du réseau en étoile, pour la transition <i>récessif</i> → <i>dominant</i>	94

5.8	Simulation Spice d'un réseau en étoile, résistance à une seule extrémité, <i>récessif</i> → <i>dominant</i> . . . . .	94
5.9	Schéma Spice du réseau en étoile, pour la transition <i>dominant</i> → <i>récessif</i> . . . . .	95
5.10	Simulation Spice d'un réseau en étoile, résistance à une seule extrémité, <i>dominant</i> → <i>récessif</i> . . . . .	96
5.11	Simulation Spice d'un réseau en étoile, résistances réparties, <i>récessif</i> → <i>dominant</i> . . . . .	97
5.12	Simulation Spice d'un réseau en étoile, résistances réparties, <i>dominant</i> → <i>récessif</i> . . . . .	98
5.13	Simulation Spice d'un réseau en étoile, résistance à une seule extrémité, branches inégales, <i>récessif</i> → <i>dominant</i> . . . . .	99
5.14	Simulation Spice d'un réseau en étoile, résistance à deux extrémités, branches inégales, <i>dominant</i> → <i>récessif</i> . . . . .	99
5.15	Simulation Spice d'un réseau en étoile, résistances réparties, branches inégales, <i>récessif</i> → <i>dominant</i> . . . . .	100
5.16	Simulation Spice d'un réseau en étoile, résistances réparties, branches inégales, <i>dominant</i> → <i>récessif</i> . . . . .	101
5.17	Tension aux extrémités du bus . . . . .	102
5.18	Câbles en ruban et connectique IDC . . . . .	103
5.19	Exemple de matériel pour fabriquer ses propres cordons téléphoniques . . . . .	104
5.20	Embase mâle D-SUB à 9 broches décrite dans le rapport technique CIA 106 . . . . .	105
5.21	Connectique <i>Mini-SPOX 2.50mm</i> de Molex . . . . .	106
5.22	Quelques borniers Degson, au pas de 3,81 mm . . . . .	106
5.23	Connecteurs mâles <i>xPyC</i> . . . . .	107
5.24	Cordons Ethernet droit et croisé . . . . .	108
5.25	Cordons plats RJ12 droit et croisé . . . . .	109
5.26	Cordons plats à 8 conducteurs droit et croisé . . . . .	110
5.27	Format des embases IEEE 1394A et IEEE 1394B . . . . .	111
5.28	Cordon IEEE 1394A . . . . .	111
5.29	Connexion d'un équipement mobile avec la connectique IEEE 1394A . . . . .	112
5.30	Toutes les cartes ont la même référence $\emptyset V$ . . . . .	113
5.31	Exemples d'agencement de CANH et CANL dans un câble plat . . . . .	113
5.32	Cartes sans référence commune (non recommandé) . . . . .	114
5.33	Cartes sans référence commune, le bus véhicule la référence . . . . .	114
5.34	Exemples d'agencement de CANH, CANL, $\emptyset V$ et Alimentation dans un câble plat . . . . .	115
6.1	Répéteur, passerelle . . . . .	116
6.2	Intérêt d'un répéteur ou d'une passerelle . . . . .	117
6.3	Schéma de principe d'une passerelle CAN . . . . .	118
6.4	Schéma de principe d'un répéteur CAN . . . . .	119
6.5	Circuit logique d'arbitrage combinatoire (défaillant) . . . . .	120
6.6	Circuit logique d'arbitrage combinatoire : mise en évidence d'une défaillance . . . . .	121
6.7	Ligne à retard . . . . .	122
6.8	Schéma du répéteur CAN . . . . .	123

6.9	Connexions et schéma-bloc d'un AMIS-427x0 . . . . .	124
6.10	Un répéteur entre deux bus CAN avec un AMIS-427x0 . . . . .	125
6.11	Un nœud CAN et un répéteur entre deux bus CAN avec un AMIS-42700 . . . . .	125
6.12	Un répéteur entre quatre bus CAN avec deux AMIS-427x0 . . . . .	126
6.13	Un répéteur entre six bus CAN avec trois AMIS-427x0 . . . . .	127
7.1	Fonction logique d'un réseau CAN : un <i>et</i> câblé . . . . .	129
7.2	Réseau CAN au repos . . . . .	130
7.3	Format d'une trame CAN de données ou de requête . . . . .	131
7.4	Début de trame . . . . .	132
7.5	Champ d'arbitrage et champ de contrôle . . . . .	133
7.6	Arbre de décision du décodage du type d'une trame CAN 2.0B . . . . .	134
7.7	Composition du champ <i>DATA</i> . . . . .	134
7.8	Composition du champ <i>CRC</i> . . . . .	135
7.9	Composition du champ <i>ACK</i> . . . . .	137
7.10	Composition du champ <i>END OF FRAME</i> . . . . .	138
7.11	Composition de l'espace inter-trames <i>INTERFRAME SPACE</i> , selon l'état du contrôleur . . . . .	138
7.12	Validation de la transmission . . . . .	140
7.13	Synchronisation des contrôleurs en réception . . . . .	141
7.14	Exemples d'insertion de bit par le mécanisme de <i>bit stuffing</i> . . . . .	142
7.15	Scénario d'insertion maximum de bits par le « <i>bit stuffing</i> » . . . . .	143
7.16	Exemple d'arbitrage entre trois contrôleurs . . . . .	145
7.17	Champ arbitrage des trames standards et étendues, de requête et de données . . . . .	146
7.18	Activité d'un contrôleur CAN vis-à-vis du réseau . . . . .	147
8.1	Exemple d'erreur d'inversion de bit . . . . .	152
8.2	Exemple d'erreur d'inversion de bit entraînant une violation du <i>bit stuffing</i> . . . . .	153
8.3	Exemple d'erreur d'inversion de bit entraînant une suppression de bit . . . . .	153
8.4	Exemple d'erreur d'inversion de bit entraînant une insertion de bit . . . . .	154
8.5	Exemple d'erreur de suppression de bit . . . . .	154
8.6	Composition de l'espace inter-trames <i>INTERFRAME SPACE</i> , selon le mode du contrôleur . . . . .	159
8.7	Trame d'erreur active ( <i>ACTIVE ERROR FRAME</i> ) . . . . .	160
8.8	Trame d'erreur passive ( <i>PASSIVE ERROR FRAME</i> ) . . . . .	162
8.9	Trame de surcharge ( <i>OVERLOAD ERROR FRAME</i> ) . . . . .	163
8.10	Fin d'une trame de données ou de requête . . . . .	166
9.1	Exemple de double erreur non détectée . . . . .	176
10.1	Normalisations de la décomposition d'un bit CAN . . . . .	180
10.2	Décomposition d'un bit CAN nominal (ISO 15765-4) . . . . .	181
10.3	Synchronisation d'un récepteur, durée de propagation . . . . .	182
10.4	Resynchronisation d'un récepteur . . . . .	183
10.5	Resynchronisations d'un récepteur rapide . . . . .	184

10.6	Resynchronisations d'un récepteur lent	184
10.7	Positions de l'unique ou triple échantillonnage	185
12.1	Variantes du mode « <i>Loop Back</i> » d'un contrôleur CAN	191
12.2	Principe du filtrage des trames reçues	193
12.3	Connexion d'un SJA1000 à un microcontrôleur	195
12.4	Connexion d'un MCP2515 à un microcontrôleur	196
12.5	Cheminement des messages dans le contrôleur FlexCAN	199
13.1	Connexion pour tester le générateur de trame CAN	216
14.1	Format d'une trame CAN FD	224
14.2	Champ arbitrage et champ de contrôle d'une trame CAN FD	225
14.3	Arbre de décision du décodage du type d'une trame CAN FD	227
14.4	Champs STUFF BIT COUNT et CRC d'une trame CAN FD	228
14.5	Zone où le <i>data bit rate</i> est actif	235
14.6	Décomposition du bit CAN et position du point d'échantillonnage	235
14.7	Décomposition du bit CAN FD et position des points d'échantillonnage	236
14.8	Décomposition des bits BRS et CRC DEL	236
14.9	Exemple de chronogramme des changements de débit	237
14.10	Pas de changement de débit, illustration du délai de boucle	238
14.11	Changement de débit, pas de compensation du délai de boucle	238
14.12	Mesure automatique du délai de boucle du transmetteur	239
14.13	Position du <i>point d'échantillonnage secondaire SSP</i>	240
14.14	Activité d'un contrôleur CAN FD vis-à-vis du réseau	246
15.1	Connexion pour tester le générateur de trames CAN FD	252
16.1	Altération du bit S0F	269
17.1	Disposition et signification des broches de quelques transcepteurs CAN FD	272
17.2	Délai de boucle d'un transcepteur CAN / CAN FD, état des sorties CANH / CANL	274
17.3	Délai de boucle d'un transcepteur CAN SIC, état des sorties CANH / CANL	274
18.1	Configuration de la <i>message RAM</i> d'un module M_CAN	278
19.1	Structure d'une station CAN	283
19.2	Cheminement des messages dans un contrôleur CAN	284
21.1	Affichage d'un signal échantillonné	296
21.2	Perte de la séquentialité entre deux évènements	297
21.3	Décodage d'une trame CAN standard, un octet de données	298
21.4	Inversion de bit aléatoire dans une trame CAN standard d'un octet de données	299
21.5	Connexions pour observer les effets d'un niveau <i>récessif</i> permanent sur RxD	301

21.6	Signal TxD, quand RxD est en permanence <i>récessif</i> . . . . .	302
21.7	Connexions pour observer les effets d'un niveau <i>dominant</i> permanent sur RxD . . . . .	303
21.8	Signal TxD, quand RxD devient en permanence <i>dominant</i> . . . . .	303
21.9	Connexions pour observer les effets d'un contrôleur seul sur le réseau . . . . .	304
21.10	Trame utilisée pour mettre en évidence l'erreur d'acquittement . . . . .	304
21.11	Erreur d'acquittement dans l'état <i>error active</i> . . . . .	305
21.12	Erreur d'acquittement dans l'état <i>error passive</i> . . . . .	305
21.13	Évaluation de la durée des bits BRS et CRC DEL . . . . .	306
21.14	Exemple de connexions pour des travaux pratiques . . . . .	308
21.15	Acquittement de la réception d'une trame . . . . .	309
21.16	Perte d'arbitrage lors de l'émission d'une trame . . . . .	310
21.17	Émission après la perte d'arbitrage . . . . .	310
21.18	Collision entre trames identiques, contrôleurs en <i>error active</i> . . . . .	311
21.19	Collision entre trames identiques, contrôleurs en <i>error passive</i> . . . . .	312
21.20	Collisions entre trames, émissions 1 à 15 . . . . .	313
21.21	Collisions entre trames, 16 <sup>e</sup> émission . . . . .	314
21.22	Collisions entre trames, 17 <sup>e</sup> émission . . . . .	314
21.23	Collisions entre trames, 18 <sup>e</sup> émission . . . . .	315
21.24	Collisions entre trames, 19 <sup>e</sup> émission . . . . .	315
21.25	Collisions entre trames, exemple d' <i>active error flag</i> de 12 bits . . . . .	316
21.26	Réseau entre les quatre modules Teensy 4.1 . . . . .	317
21.27	$M_0$ suivi de $M_2$ . . . . .	320
21.28	Impulsion au début du champ S0F . . . . .	321
22.1	Réseau et bus entre deux XIAO ESP32S3 . . . . .	324
22.2	Brochage d'un XIAO ESP32S3 (document Seeed Studio) . . . . .	324
22.3	Réseau et bus entre trois XIAO ESP32S3 . . . . .	334
A.1	Description d'une résistance et d'un condensateur en Spice . . . . .	345
A.2	Description d'une source de tension fixe . . . . .	345
A.3	Modèle Spice d'une source de tension impulsionnelle . . . . .	346
A.4	Modèle Spice d'une ligne sans perte . . . . .	346
A.5	Modèle Spice d'une ligne avec perte . . . . .	347
A.6	Interrupteur Spice commandé en tension . . . . .	347
A.7	Cycle d'hystérésis d'un interrupteur Spice commandé en tension . . . . .	348

# Liste des algorithmes

8.1	Émission de la trame d'erreur active	160
8.2	Émission de la trame d'erreur passive	162
8.3	Émission de la <i>trame de surcharge</i>	163
20.1	Algorithme de réception des fragments	292
21.1	Croquis, mise en évidence du comportement quand RxD est en permanence <i>récessif</i>	301
21.2	Croquis, mise en évidence du comportement quand RxD est en permanence <i>dominant</i>	303
21.3	Croquis, mise en évidence du comportement quand un contrôleur est seul sur le réseau	304
21.4	Croquis, mise en évidence de la perte de l'arbitrage	309
21.5	Émission de trames identiques	311
21.6	Émission de trames dont les champs ARBITRATION sont identiques	313
21.7	Émission de trames dont les champs ARBITRATION sont identiques	316
21.8	Code de la station $S_0$	318
21.9	Code de la station $S_1$	319
22.1	Code de la station LED	326
22.2	Code de la station LED, détection de l'absence d'émission	327
22.3	Code de la station POUSSOIR, émission périodique	328
22.4	Code de la station POUSSOIR, émission évènementielle	329
22.5	Code de la station POUSSOIR, émission évènementielle, prise en compte des rebonds	331
22.6	Code de la station POUSSOIR, émission évènementielle et périodique	332
22.7	Code de la station LED, inversion de l'état	333
22.8	Code de la station CAPTEUR0, émission spontanée	335
22.9	Code de la station GESTIONNAIRE, émission spontanée	336
22.10	Code de la station CAPTEUR0, émission sur demande	338
22.11	Code de la station GESTIONNAIRE, émission sur demande	338

---

B.1	Rattrapage du dépassement d'échéance, débordement non géré . . . . .	355
B.2	Rattrapage du dépassement d'échéance, débordement géré . . . . .	357
B.3	Rattrapage du dépassement d'échéance, débordement mal géré . . . . .	358

## Avant-propos

**Pourquoi utiliser un réseau?** Lorsqu'il conçoit un équipement, le concepteur peut être amené à connecter à l'informatique embarquée des actionneurs et des capteurs distants. Un signal numérique issu d'un microcontrôleur ou d'un circuit logique ne peut être transmis fiablement que sur quelques dizaines de centimètres. En cause, les parasites électromagnétiques et le fait qu'une ligne de transmission n'est pas une équipotentielle, même en l'absence de parasite. Une solution est d'adopter une *ligne différentielle*, mais cela double le nombre de fils. Ces points sont développés dans le [chapitre 1 à partir de la page 32](#).

**Intérêt d'un réseau.** Dès lors, le concepteur peut envisager d'adopter un réseau. Cette solution présente de nombreux avantages.

La transmission fiable des signaux sur le réseau est pris en charge par des composants dédiés à cette tâche – les transcepteurs.

Au lieu d'une (grande) carte informatique unique, l'équipement est constitué de plusieurs (petites) cartes séparées. Chacune d'elle est proche des actionneurs et des capteurs dont elle a la charge. Ces cartes sont plus faciles à concevoir et à maintenir et peuvent être en partie testées séparément, sans connexion au réseau. Comme un réseau effectue un multiplexage temporel de plusieurs signaux, on bénéficie d'une diminution du nombre de fils nécessaires à l'interconnexion, et d'une standardisation de celle-ci. En cas de panne, les cartes peuvent être remplacées indépendamment les unes des autres. Au fur et à mesure de l'évolution des prestations, de nouvelles cartes peuvent être intégrées au réseau. On peut donc concevoir son équipement progressivement, le rendant évolutif à la demande.

L'interface d'échange d'information est claire : c'est la messagerie du réseau.

**Inconvénients d'un réseau.** Chaque carte doit intégrer les composants d'interface, le contrôleur réseau, qui effectue la prise en charge du protocole réseau et le transcepteur chargé de la transmission fiable des signaux. S'il existe de nombreux microcontrôleurs intégrant des contrôleurs de divers types de réseau, le transcepteur est un composant externe. À cela s'ajoute le pilote, logiciel qui assure l'interface avec le contrôleur, dont le volume mémoire (code et mémoire vive) peut être important, relativement à la place disponible dans les microcontrôleurs.

Il faut aussi compter l'investissement intellectuel pour comprendre et apprivoiser le fonctionnement d'un réseau, pouvoir cerner la cause d'une défaillance ou d'un dysfonctionnement. Un des buts de ce livre est d'aider le concepteur à comprendre en détail le fonctionnement des réseaux CAN et CAN FD.

**Équipements électriques des véhicules routiers.** Les constructeurs automobiles sont confrontés à la progression constante des équipements électriques : d'abord, démarrage du moteur à combustion interne, éclairage de la route, puis les essuie-glaces, les clignotants, ... La liste est longue et augmente régulièrement. À la fin du 20<sup>e</sup> siècle<sup>1</sup>, une voiture haut de gamme peut embarquer jusqu'à 3 km de câbles, longueur cumulée de 1000 fils, représentant une masse non négligeable de 40 kg. L'interconnexion de ces équipements est un défi : complexité des faisceaux, coût, encombrement, fiabilité et maintenance.

À cela, s'ajoute un problème plus complexe, celui de l'évolution des prestations vers leur interdépendance. Au départ, les prestations sont indépendantes. Pour prendre un exemple, les freins sont actionnés uniquement par la pédale de frein, via un système hydraulique.

Actuellement, outre l'action du conducteur, les freins peuvent être commandés par :

- l'ABS (de l'allemand « *Antiblockiersystem* »), système anti-blocage des roues ;
- l'ESP (« *Electronic Stability Program* »), système d'anti-dérapiage pour éviter le sous-virage et le sur-virage ;
- l'ASR (« *Acceleration Slip Regulation* »), système d'anti-patinage des roues motrices ;
- l'EBD (« *Electronic Brakeforce Distribution* »), qui répartit le freinage sur les roues pour éviter la tête-à-queue.

Faire cohabiter ces différentes prestations ne peut plus être résolu par un câblage, aussi complexe soit-il. La solution adoptée est de disposer d'un grand nombre de calculateurs embarqués, interconnectés par un ou plusieurs réseaux informatiques. Le réseau CAN est l'un de ceux-là.

**Le réseau CAN.** Le protocole CAN (« *Controller Area Network* ») a été conçu par la société Bosch à partir de 1983 et la version 2.0B a été normalisée en 1993. Son but : proposer à l'industrie automobile un réseau *temps réel*, peu coûteux et fiable pour connecter toute l'instrumentation embarquée (capteurs, actionneurs). Une originalité du protocole CAN est qu'il inclut une gestion précise des erreurs qui est prise en charge automatiquement par le contrôleur, sans intervention du logiciel (ce qui a pour conséquence de le simplifier et de l'alléger). En 2012, l'extension CAN FD du protocole a été rendue publique, et a été normalisée après quelques modifications en 2015.

L'industrie automobile exige fiabilité, grandes séries et prix bas. En conséquence, de nombreux composants CAN / CAN FD sont disponibles, auprès de différents fournisseurs, tous étant parfaitement interopérables. De nouveaux marchés se créent alors, attirés par les faibles coûts. Ainsi fut créée en 1992 l'organisation internationale *CAN in Automation*<sup>2</sup> (CiA), qui regroupe les utilisateurs et fabricants de produits matériels et logiciels basés sur CAN.

L'industrie propose de nombreux modules microcontrôleurs dotés d'une interface CAN et / ou CAN FD, pour des prix allant de quelques euros à plusieurs dizaines d'euros. L'environnement de développement Arduino a démocratisé la programmation de ces modules et il existe des bibliothèques qui implémentent un pilote CAN et / ou un pilote CAN FD. Ceci rend accessible à l'amateur ou à l'ingénieur non spécialiste la conception et la maîtrise d'un réseau CAN / CAN FD.

<sup>1</sup> Déjà, en 1950, la Peugeot 203 embarquait un câblage de 50 fils pour une longueur cumulée de 125 m.

<sup>2</sup> <https://www.can-cia.org/about-us/>

**Plan du livre.** La première partie de ce livre est constituée de deux chapitres : le [chapitre 1 à partir de la page 32](#) est consacré aux lignes de transmission et le [chapitre 2 à partir de la page 49](#) est une brève introduction aux réseaux CAN et CAN FD.

La deuxième partie est consacrée au réseau CAN :

- le [chapitre 3 à partir de la page 56](#) présente la distinction entre réseau et bus et précise le rôle des contrôleurs et des transcepteurs ; la fonction logique d'un réseau CAN (une fonction « *et* ») est détaillée en s'appuyant sur les termes *dominant* et *récessif* ;
- le [chapitre 4 à partir de la page 66](#) détaille le fonctionnement des transcepteurs et leurs fonctionnalités ;
- le [chapitre 5 à partir de la page 89](#) présente la topologie la plus classique d'un réseau CAN qui est le bus, discute de la topologie en étoile et en propose des simulations, selon le type de terminaison choisie ; il se termine sur l'agencement des signaux sur un câble et cite différentes connectiques ;
- le [chapitre 6 à partir de la page 116](#) montre comment étendre un réseau CAN au moyen d'un *répéteur* ou d'une *passerelle* ;
- on remarquera que les chapitres précédents n'ont pas besoin de connaître le protocole CAN, seulement la fonction logique du réseau ; c'est pourquoi le protocole CAN n'est présenté qu'au [chapitre 7 à partir de la page 128](#) : le format d'une trame, le *bit stuffing*, la résolution non destructive des collisions ;
- le [chapitre 8 à partir de la page 151](#) liste les erreurs détectées par un contrôleur et comment il y réagit ;
- le protocole CAN est-il fiable ? Alors que la spécification CAN 2.0B annonce une distance de Hamming de 6 et la probabilité qu'une trame erronée vérifie la somme de contrôle inférieure à  $4,7 \times 10^{-11}$ , l'expérience a montré que ce n'était pas le cas ; le [chapitre 9 à partir de la page 168](#) donne quelques exemples qui montrent que la distance de Hamming est inférieure à 6 ;
- la configuration d'un contrôleur CAN fixe la valeur de différents paramètres qui définissent la composition temporelle d'un bit CAN et par conséquent le débit du réseau : le [chapitre 10 à partir de la page 179](#) explique leur signification et comment ils interviennent ;
- le paramétrage établi au chapitre précédent est-il valide pour un réseau donné ? Le [chapitre 11 à partir de la page 186](#) répond à cette question en listant les contraintes qui doivent être satisfaites ;
- le [chapitre 12 à partir de la page 190](#) présente succinctement quelques contrôleurs CAN ;
- le [chapitre 13 à partir de la page 201](#) présente différents codes écrits en C++ et destinés à être exécutés sur une machine de bureau : un générateur de la séquence de bits d'une trame CAN (qui a servi pour produire les exemples de trames de ce livre) et des logiciels utilitaires pour obtenir une décomposition du bit CAN pour un débit donné ou encore obtenir la liste des débits communs à plusieurs contrôleurs.

La troisième partie est consacrée au réseau CAN FD. On part de la connaissance des chapitres précédents :

- le [chapitre 14 à partir de la page 221](#) commence donc directement par l'exposé du protocole CAN FD, les nouveaux formats de trame ; la spécification d'origine du protocole – maintenant

appelée *CAN FD non ISO* – présentait des vulnérabilités dont certaines ont été corrigées lors de la normalisation du protocole – aboutissant à la version *CAN FD ISO*, incompatible avec la précédente; ce chapitre présente aussi la décomposition temporelle du bit CAN FD et le mécanisme de *changement de débit* propre à CAN FD;

- le [chapitre 15 à partir de la page 249](#) décrit un générateur de trames CAN FD et un algorithme de calcul de la décomposition des bits CAN FD en fonction des débits demandés et de la position des points d'échantillonnage;
- le protocole CAN FD est-il fiable? Plusieurs études ont mis en évidence que des faiblesses du protocole persistaient dans la version *CAN FD ISO* du protocole; le [chapitre 16 à partir de la page 262](#) en décrit quelques-unes, ainsi qu'une vulnérabilité propre à la version *CAN FD non ISO*: l'altération du bit SOF;
- les deux chapitres suivants sont courts: présentation des transcepteurs CAN FD ([chapitre 17 à partir de la page 271](#)) et leur version améliorée CAN SIC et des contrôleurs CAN FD ([chapitre 18 à partir de la page 276](#)).

La quatrième partie comporte deux chapitres qui s'appliquent aussi bien à CAN qu'à CAN FD:

- les services qu'offre un pilote logiciel dédié à CAN ou CAN FD font l'objet du [chapitre 19 à partir de la page 281](#);
- concevoir une messagerie CAN / CAN FD: [chapitre 20 à partir de la page 287](#); on y montre principalement qu'il vaut mieux transmettre un *état* plutôt qu'un *changement d'état* et que l'on peut organiser l'échange de messages sur le réseau soit en *trafic événementiel*, soit en *trafic périodique*;
- utiliser un analyseur logique ([chapitre 21 à partir de la page 296](#)): cet outil permet de visualiser le trafic d'un réseau et y déceler facilement des erreurs – typiquement des erreurs de débits entre les différents contrôleurs d'un réseau; un formateur pourra aussi y trouver des idées sur des travaux pratiques qui mettent en évidence différentes propriétés du protocole, comme le *bit stuffing*, la résolution non destructives des collisions, l'acquiescement des trames;
- le [chapitre 22 à partir de la page 322](#) présente plusieurs exemples d'applications.

Le [chapitre 23 à partir de la page 340](#) évoque rapidement les évolutions futures en cours de conception, *CAN FD Light* et *CAN XL*.

Le livre se termine par trois annexes:

- la première ([annexe A à partir de la page 343](#)) est une présentation rapide du logiciel Spice, qui a servi pour effectuer les simulations du [chapitre 1](#);
- la deuxième ([annexe B à partir de la page 349](#)) contient des réflexions sur l'écriture d'un code Arduino, notamment sur le respect d'une périodicité – ce que la fonction de `lay` ne permet pas – et sur le débordement de la fonction `millis`;
- la troisième ([annexe C à partir de la page 360](#)) est une présentation des mathématiques des CRC (« *Cyclic Redundancy Code*»), avec application aux sommes de contrôle spécifiées par les protocoles CAN et CAN FD.

Ce livre s'adresse aux amateurs et aux ingénieurs non spécialistes pour comprendre les possibilités qu'offre un réseau CAN et comment on le met en œuvre. Un enseignant trouvera des informations pour approfondir ses connaissances et pour concevoir des travaux pratiques. Une connaissance des