MEDIAN SM 57 - Printemps 2014

Nom :

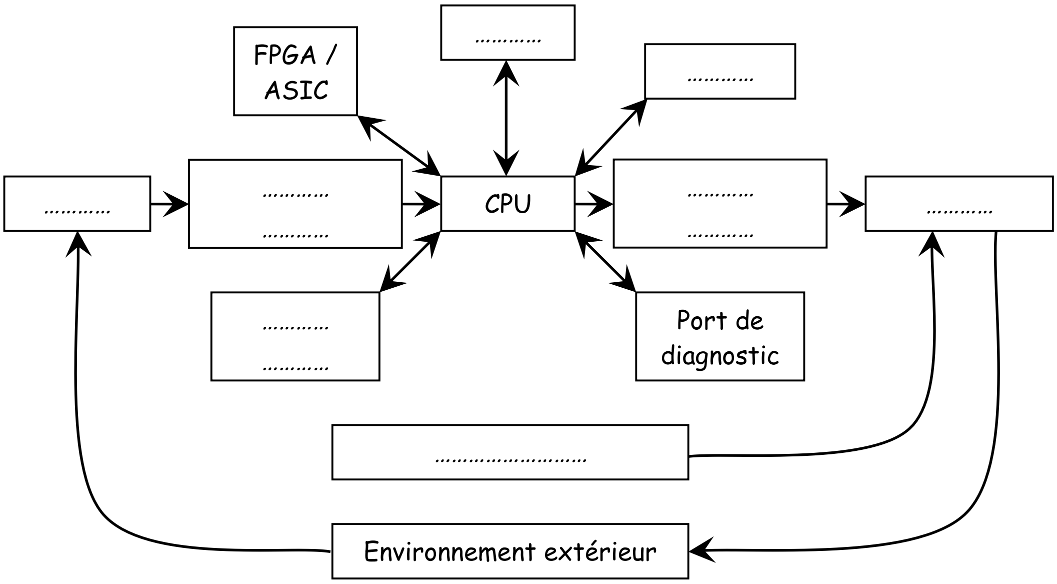
Prénom :

Aucun document autorisé

Calculatrice autorisée

I. Questions de cours

1. Comment définiriez-vous un système embarqué ?
2. Compléter sur votre copie le schéma de l'architecture générale d'un système embarqué.

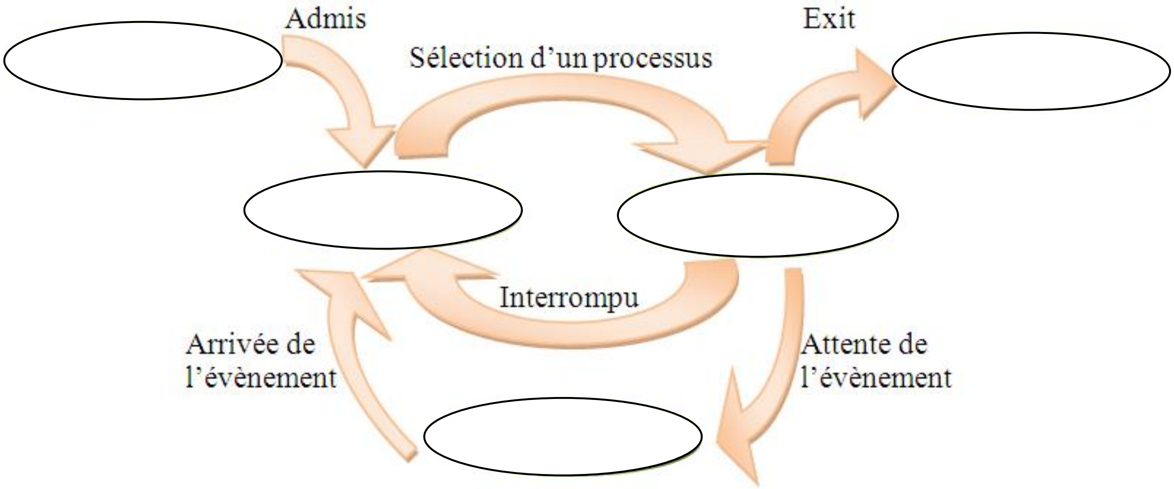


3. Quelle est la différence entre un système temps réel à contrainte stricte et un système temps réel à contraintes souples ? Argumenter par des exemples

4. Dans un réseau CAN low speed, parmi les trois messages suivants, lequel serait le plus prioritaire s'ils venaient à être émis en même temps par trois nœuds différents ? Justifiez.

message n°1 :	ID = 0x64C	Data
message n°2 :	ID = 0x740	Data
message n°3 :	ID = 0x631	Data

5. Quel est le rôle des résistances de terminaison sur le bus CAN high speed ?
6. Quel est le rôle du « bit stuffing » ? Quel est son principe de fonctionnement ?
7. Quel est le rôle de l'ordonnanceur dans un système d'exploitation temps réel ?
8. Compléter le schéma suivant de gestion des processus par l'ordonnanceur



II. Problème : véhicule hybride pile à combustible

Un véhicule électrique est hybridé par l'ajout d'un système pile à combustible afin d'améliorer son autonomie. Le partage de la puissance entre les différentes sources d'énergie (batterie et pile à combustible) est gérer par un calculateur automobile. La communication entre les composants se fait par un réseau CAN 2.0A ayant les caractéristiques suivantes :

- Nœud 1 : contrôleur pile à combustible assurant la génération du courant (injection d'hydrogène et d'oxygène dans le stack (cœur de pile)) et le refroidissement du système à l'aide de ventilateurs
- Nœud 2 : Battery Management System (BMS), estimant l'état de charge des différentes cellules et gérant les différents modes de fonctionnement (Standby, Drive et Charge).
- Nœud 3 : Electronic Control Unit (ECU) assurant la gestion de la chaine de traction du véhicule complet et définissant le courant de consigne de la pile à combustible en fonction de l'état de charge des batteries.

Voici une liste de quelques paramètres circulant sur le réseau :

- Point de fonctionnement de la pile à combustible (identifiant 0x109 : gamme 0..255 A sur 8 bits).
- Etat de charge du pack batterie (identifiant 0x200 : gamme 0..100 % sur 8 bits).
- Température du stack pile à combustible (identifiant 0x108 : gamme -40..90 °C sur 16 bits signés, le bit de poids le plus fort étant utilisé comme bit de signe).

Première partie : Gestion de l'énergie du véhicule : Etude des trames CAN

1.1 Quelles sont les résolutions des variables numériques traduisant le point de fonctionnement de la pile, l'état de charge des batteries et la température du stack? **Détailler les calculs.**

1.2 Le contrôleur pile à combustible envoie l'information de température du stack pile sur le bus. La valeur relevée par le capteur est de 35°C. La donnée est découpée en 2 octets, le premier donnant les 8 bits de poids le plus faible, le deuxième donnant les 8 bits de poids le plus fort. L'envoi se fait sous le format Little Endian (LSB en premier). Compléter le document réponse en donnant la trame avec les bits de stuffing; le champ CRC sera rempli par des croix (X). On part du principe qu'il n'y a jamais d'erreur de transmission**. Détailler les calculs.**

1.3 Compléter également la trame en tension en y indiquant les plages de tensions minimum et maximum ainsi que la partie champs en précisant le nom de chaque champ de la trame.

Deuxième partie : Etude de l'ECU

Le circuit électronique de l'ECU est composé d'un dsPIC30F4011 et du driver CAN MCP2551 dont les extraits des documentations sont donnés ci-après.

2.1 Déterminer les différents paramètres des registres de configuration du Bit Timing pour définir la configuration suivante :

- Fcy = 29.4912 MHz
- Baud Rate = 750kbits/s
- Nominal Bit Time = 20 TQ
- Segment de propagation = 7 TQ
- Segment de phase 1 = 6 TQ
- Segment de phase 2 = 6 TQ
- Sauts de resynchronisation = 2 TQ
- Echantillonnage en 3 points.

2.2 Calculer la fréquence de transmission exacte compte tenu des valeurs retenues après arrondi.

2.3 Déterminer la durée maximale de transmission de la trame de température du stack pile définis à la question 1.2.

2.4 On désire permettre à l'ECU de réceptionner les messages provenant du contrôleur de la pile à combustible (identifiant compris entre 0x100 et 0x10F inclus) ainsi que du BMS (identifiants compris entre 0x200 et 0x20F inclus). Déterminer les registres de filtrage et les masques de filtrage pour affecter ces groupes d'identificateurs respectivement aux deux boîtes de réception RXB0 et RXB1.

Pour faciliter la manipulation en langage C des messages échangés par le bus CAN, on définit une structure et un type de donnée propres aux messages CAN de la manière suivante :

```
typedef struct {
  unsigned int STDID; // ID standard (11 bits)
  unsigned char length;// nombre d'octets de données utiles
  unsigned char data[8];          // données
} CAN_frame;
```

Il est ainsi possible de déclarer des variables de type ***CAN_frame*** (trame CAN) :

```
CAN_frame CAN_message;
```

2.5 Ecrire l'organigramme donnant les différentes étapes de l'envoi du message CAN du courant de consigne de la pile à combustible (un seul octet de donnée) sur la boîte TXB0 sans utiliser les interruptions.

2.6 Ecrire en langage C la fonction **unsigned char envoi_consigne_courant_CAN(CAN_frame CAN_mes)** correspondant à l'organigramme de la question précédente qui retourne la valeur 1 si le message à bien été transmis, et 0 sinon.

La réception des trames CAN provenant du BMS concernant l'estimation de l'état de charge batteries sont faites par interruptions et stockage par buffer circulaire avec gestion des erreurs. La fonction **unsigned char CAN_recu(void)** test la présence d'au moins un message dans le buffer et la fonction **void lecture_CAN(CAN_frame *t)** permet la lecture d'un message dans le buffer.

2.7 En vous aidant de ces fonctions et des informations des données courant pile et état de charge de l'énoncé, écrire en langage C la fonction **void gestion_energie(void)** qui gère le courant de la pile à combustible en fonction de l'état de charge des batteries de la manière suivante :

$$Si\ SoC_{batt} > 70\%\ alors\ I_{pac} = 0A$$

$$Si\ SoC_{batt} \leq 70\%\ alors\ I_{pac} = 100A$$

Avec SoC_{batt} l'état de charge du pack batteries et I_{pac} le courant de consigne de la pile à combustible.

Troisième partie : Utilisation de la pile à combustible en co-génération

Dans le but de tirer parti au maximum des capacités de la pile à combustible, un système de chauffage de l'habitacle est mise en place en récupérant la chaleur produite par la pile lors de son fonctionnement par l'intermédiaire d'un système de tuyauterie et de ventilateurs de brassage. L'utilisateur peux faire varier la vitesse des ventilateurs à l'aide d'un commutateur 4 positions :

- Arrêt
- Marche 25%
- Marche 50%
- Marche 100%

Une deuxième interface utilisateur composé d'une LED s'allume si la température que fournis la pile est inférieure à 25°C (on suppose la température connue par l'ECU via le bus CAN).

La carte électronique regroupant le commutateur 4 positions, la deuxième interface utilisateur et le système de ventilation sont équipés d'une interface LIN. Un driver LIN est ajouté au circuit électronique de l'ECU et connecté aux broches TX et RX de son UART. On définit l'ECU comme le maitre du bus.

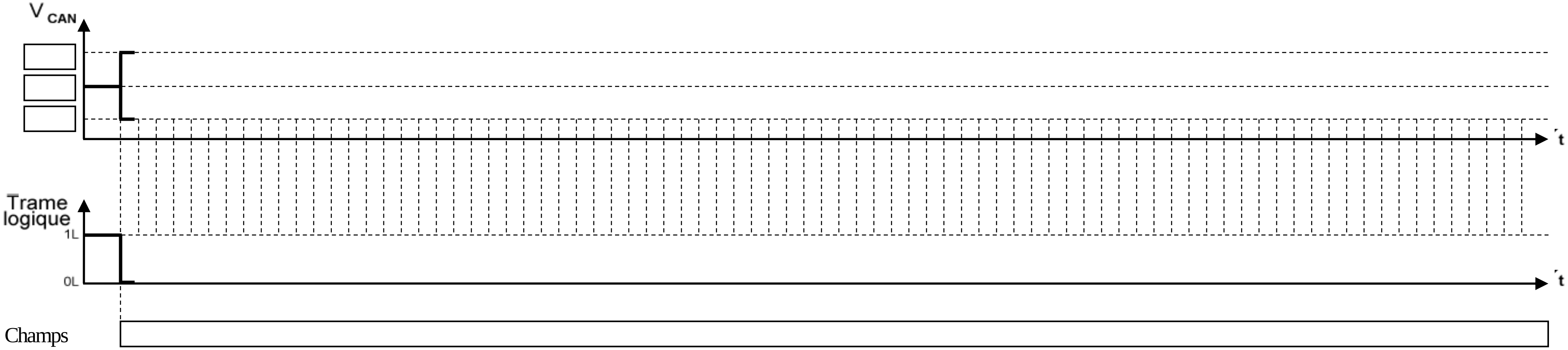
3.1 Dessiner un schéma représentant le bus LIN avec les 4 nodes.

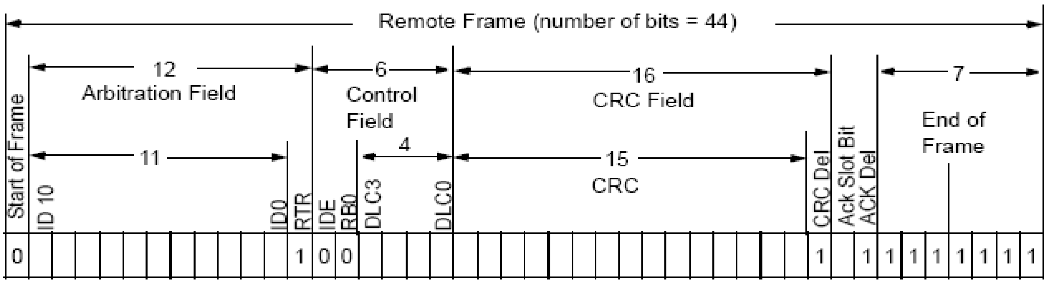
3.2 Remplir le tableau ci-dessous afin de déterminer pour chaque node esclaves : Le type de communication avec le maitre (réception et émission) et attribuer un identifiant d'émission ou de réception si besoin.

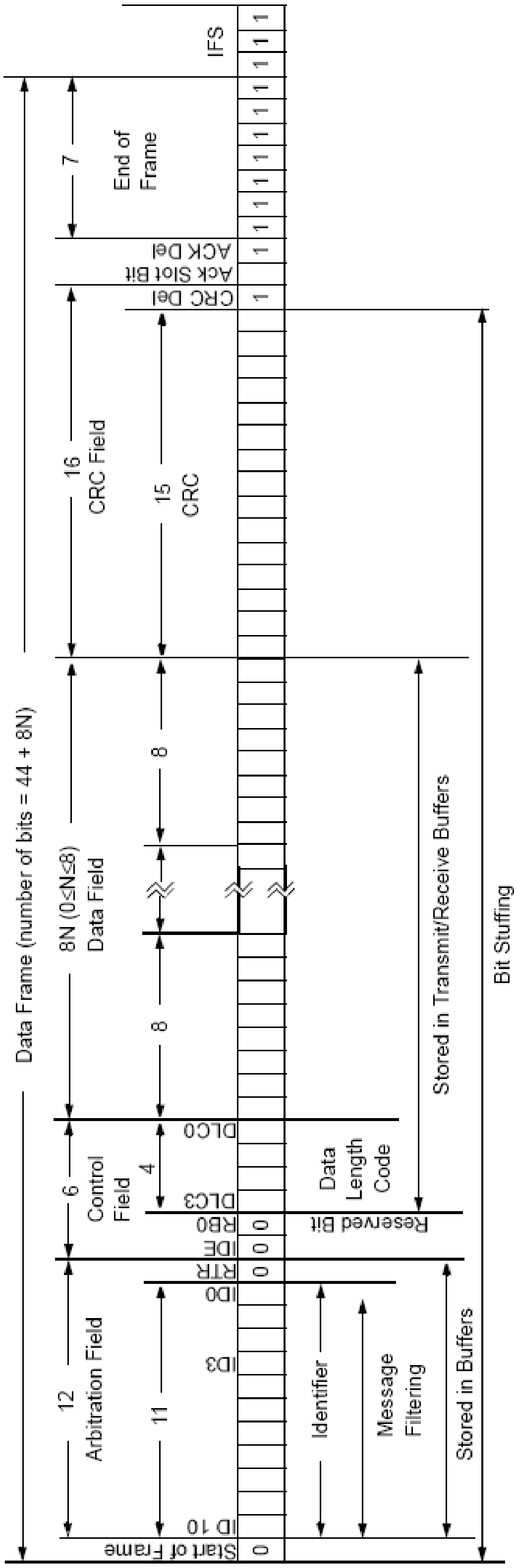
node	fonction de réception	fonction d'émission	Identifiant d'émission	Identifiant de réception
LED	Reçois de l'ECU un niveau logique pour allumer ou éteindre la led en fonction de la température	Ne renvoie pas d'informations à l'ECU	╳	0x10
commutateur 4 positions				
ventilation				

3.3 Quels identifiants LIN ne peuvent pas être utilisés ? Pourquoi ?

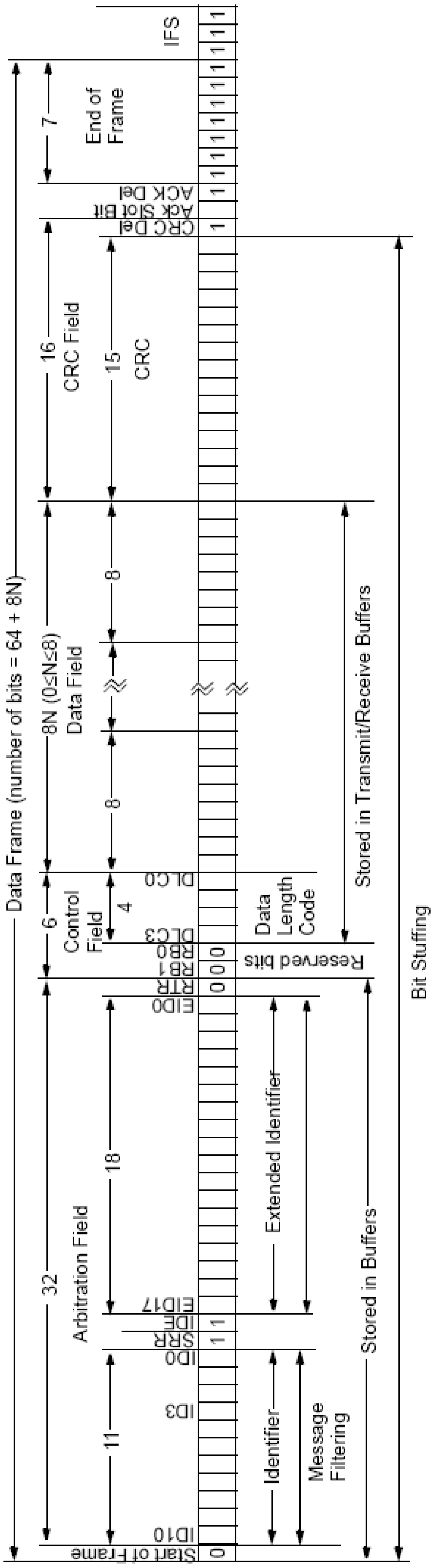
Document réponse : problème, questions 1.2 / 1.3







STANDARD DATA FRAME



EXTENDED DATA FRAME

23.3 Register Maps

Table 23-1: CAN1 Register Map

File Name	ADR	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset
C1RXF0SID	300	—	—	—	SID<10:6>					SID<5:0>						—	EXIDE	xxxx
C1RXF0EIDH	302	—	—	—	—	EID<17:14>				EID<13:6>								xxxx
C1RXF0EIDL	304	EID<5:0>						—	—	—	—	—	—	—	—	—	—	xxxx
unused	306	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
C1RXF1SID	308	—	—	—	SID<10:6>					SID<5:0>						—	EXIDE	xxxx
C1RXF1EIDH	30A	—	—	—	—	EID<17:14>				EID<13:6>								xxxx
C1RXF1EIDL	30C	EID<5:0>						—	—	—	—	—	—	—	—	—	—	xxxx
Unused	30E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
C1RXF2SID	310	—	—	—	SID<10:6>					SID<5:0>						—	EXIDE	xxxx
C1RXF2EIDH	312	—	—	—	—	EID<17:14>				EID<13:6>								xxxx
C1RXF2EIDL	314	EID<5:0>						—	—	—	—	—	—	—	—	—	—	xxxx
Unused	316	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
C1RXF3SID	318	—	—	—	SID<10:6>					SID<5:0>						—	EXIDE	xxxx
C1RXF3EIDH	31A	—	—	—	—	EID<17:14>				EID<13:6>								xxxx
C1RXF3EIDL	31C	EID<5:0>						—	—	—	—	—	—	—	—	—	—	xxxx
Unused	31E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
C1RXF4SID	320	—	—	—	SID<10:6>					SID<5:0>						—	EXIDE	xxxx
C1RXF4EIDH	322	—	—	—	—	EID<17:14>				EID<13:6>								xxxx
C1RXF4EIDL	324	EID<5:0>						—	—	—	—	—	—	—	—	—	—	xxxx
Unused	326	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
C1RXF5SID	328	—	—	—	SID<10:6>					SID<5:0>						—	EXIDE	xxxx
C1RXF5EIDH	32A	—	—	—	—	EID<17:14>				EID<13:6>								xxxx
C1RXF5EIDL	32C	EID<5:0>						—	—	—	—	—	—	—	—	—	—	xxxx
Unused	32E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
C1RXM0SID	330	—	—	—	SID<10:6>					SID<5:0>						—	MIDE	xxxx
C1RXM0EIDH	332	—	—	—	—	EID<17:14>				EID<13:6>								xxxx
C1RXM0EIDL	334	EID<5:0>						—	—	—	—	—	—	—	—	—	—	xxxx
Unused	336	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
C1RXM1SID	338	—	—	—	SID<10:6>					SID<5:0>						—	MIDE	xxxx
C1RXM1EIDH	33A	—	—	—	—	EID<17:14>				EID<13:6>								xxxx
C1RXM1EIDL	33C	EID<5:0>						—	—	—	—	—	—	—	—	—	—	xxxx
Unused	33E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx

Legend: x = Unknown

Table 23-1: CAN1 Register Map (Continued)

File Name	ADR	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset
C1TX2SID	340	SID<10:6>					—	—	—	SID<5:0>						SRR	TX IDE	xxxx
C1TX2EID	342	EID<17:14>				—	—	—	—	EID<13:6>								xxxx
C1TX2DLC	342	EID<5:0>						TX RTR	TX RB1	TX RB0	DLC<3:0>				—	—	—	xxxx
C1TX2B1	346	Transmit Buffer 0 Byte 1								Transmit Buffer 0 Byte 0								xxxx
C1TX2B2	348	Transmit Buffer 0 Byte 3								Transmit Buffer 0 Byte 2								xxxx
C1TX2B3	34A	Transmit Buffer 0 Byte 5								Transmit Buffer 0 Byte 4								xxxx
C1TX2B4	34C	Transmit Buffer 0 Byte 7								Transmit Buffer 0 Byte 6								xxxx
C1TX2CON	34E	—	—	—	—	—	—	—	—	—	TX ABT	TX LARB	TX ERR	TX REQ	—	TXPRI<1:0>		0000
C1TX1SID	350	SID<10:6>					—	—	—	SID<5:0>						SRR	TX IDE	xxxx
C1TX1EID	352	EID<17:14>				—	—	—	—	EID<13:6>								xxxx
C1TX1DLC	352	EID<5:0>						TX RTR	TX RB1	TX RB0	DLC<3:0>				—	—	—	xxxx
C1TX1B1	356	Transmit Buffer 0 Byte 1								Transmit Buffer 0 Byte 0								xxxx
C1TX1B2	358	Transmit Buffer 0 Byte 3								Transmit Buffer 0 Byte 2								xxxx
C1TX1B3	35A	Transmit Buffer 0 Byte 5								Transmit Buffer 0 Byte 4								xxxx
C1TX1B4	35C	Transmit Buffer 0 Byte 7								Transmit Buffer 0 Byte 6								xxxx
C1TX1CON	35E	—	—	—	—	—	—	—	—	—	TX ABT	TX LARB	TX ERR	TX REQ	—	TXPRI<1:0>		0000
C1TX0SID	360	SID<10:6>					—	—	—	SID<5:0>						SRR	TX IDE	xxxx
C1TX0EID	362	EID<17:14>				—	—	—	—	EID<13:6>								xxxx
C1TX0DLC	362	EID<5:0>						TX RTR	TX RB1	TX RB0	DLC<3:0>				—	—	—	xxxx
C1TX0B1	366	Transmit Buffer 0 Byte 1								Transmit Buffer 0 Byte 0								xxxx
C1TX0B2	368	Transmit Buffer 0 Byte 3								Transmit Buffer 0 Byte 2								xxxx
C1TX0B3	36A	Transmit Buffer 0 Byte 5								Transmit Buffer 0 Byte 4								xxxx
C1TX0B4	36C	Transmit Buffer 0 Byte 7								Transmit Buffer 0 Byte 6								xxxx
C1TX0CON	36E	—	—	—	—	—	—	—	—	—	TX ABT	TX LARB	TX ERR	TX REQ	—	TXPRI<1:0>		0000

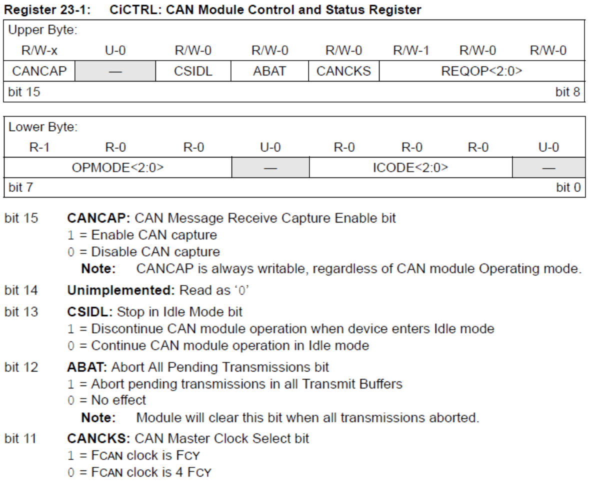
Legend: x = Unknown

Table 23-1: CAN1 Register Map (Continued)

File Name	ADR	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset
C1RX1SID	370	—	—	—	SID<10:6>					SID<5:0>						SRR	RX IDE	xxxx
C1RX1EID	372	—	—	—	—	EID<17:14>				EID<13:6>								xxxx
C1RX1DLC	374	EID<0:5>						RX RTR	RX RB1	—	—	—	RX RB0	DLC<3:0>				xxxx
C1RX1B1	376	Receive Buffer 1 Byte 1								Receive Buffer 1 Byte 0								xxxx
C1RX1B2	378	Receive Buffer 1 Byte 3								Receive Buffer 1 Byte 2								xxxx
C1RX1B3	37A	Receive Buffer 1 Byte 5								Receive Buffer 1 Byte 4								xxxx
C1RX1B4	37C	Receive Buffer 1 Byte 7								Receive Buffer 1 Byte 6								xxxx
C1RX1CON	37E	—	—	—	—	—	—	—	—	RX FUL	—	—	RX ERR	RX RTR R0	FILHIT<2:0>			0000
C1RX1SID	380	—	—	—	SID<10:6>					SID<5:0>						SRR	RX IDE	xxxx
C1RX1EID	382	—	—	—	—	EID<17:14>				EID<13:6>								xxxx
C1RX1DLC	384	EID<0:5>						RX RTR	RX RB1	—	—	—	RX RB0	DLC<3:0>				xxxx
C1RX0B1	386	Receive Buffer 0 Byte 1								Receive Buffer 0 Byte 0								xxxx
C1RX0B2	388	Receive Buffer 0 Byte 3								Receive Buffer 0 Byte 2								xxxx
C1RX0B3	38A	Receive Buffer 0 Byte 5								Receive Buffer 0 Byte 4								xxxx
C1RX0B4	38C	Receive Buffer 0 Byte 7								Receive Buffer 0 Byte 6								xxxx
C1RX0CON	38E	—	—	—	—	—	—	—	—	RX FUL	—	—	RX ERR	RX RTR R0	RXB0 DBEN	JTOFF	FIL HIT 0	0000
C1CTRL	390	CAN CAP	—	C SIDL	ABAT	CAN CKS	REQOP<2:0>			OPMODE<2:0>			—	ICODE<2:0>			—	0480
C1CFG1	392	—	—	—	—	—	—	—	—	SJW<1:0>S		BRP<5:0>						0000
C1CFG2	394	—	WAK FIL	—	—	—	SEG2PH<2:0>			SEG2 PHTS	SAM	SEG1PH<2:0>			PRSEG<2:0>			0000
C1INTF	396	RXB0 OVR	RXB1 OVR	TXBO	TXBP	RXBP	TX WARN	RX WARN	E WARN	IVR IF	WAK IF	ERR IF	TXB2 IF	TXB1 IF	TXB0 IF	RXB1 IF	RXB0 IF	0000
C1INTE	398	—	—	—	—	—	—	—	—	IVR IE	WAK IE	ERR IE	TXB2 IE	TXB1 IE	TXB0 IE	RXB1 IE	RXB0 IE	0000
C1EC	39A	Transmit Error Counter								Receive Error Counter								0000
Reserved	39C 3FE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx

Legend: x = Unknown

Registre de contrôle et d’état :



Register 23-1: CiCTRL: CAN Module Control and Status Register

Upper Byte:							
R/W-x	U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
CANCAP	—	CSIDL	ABAT	CANCKS	REQOP<2:0>		
bit 15							bit 8

Lower Byte:							
R-1	R-0	R-0	U-0	R-0	R-0	R-0	U-0
OPMODE<2:0>			—	ICODE<2:0>			—
bit 7							bit 0

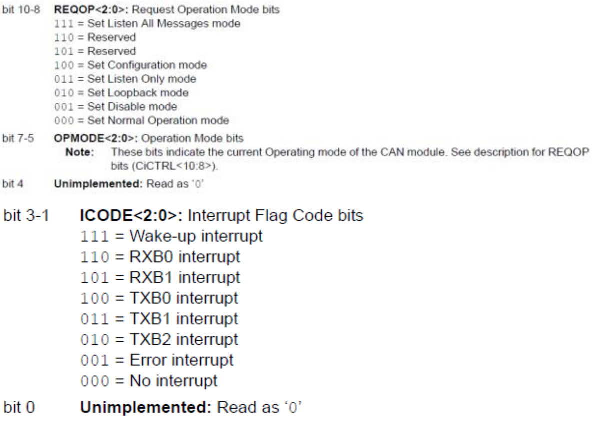
bit 15 **CANCAP:** CAN Message Receive Capture Enable bit
1 = Enable CAN capture
0 = Disable CAN capture
Note: CANCAP is always writable, regardless of CAN module Operating mode.

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **CSIDL:** Stop in Idle Mode bit
1 = Discontinue CAN module operation when device enters Idle mode
0 = Continue CAN module operation in Idle mode

bit 12 **ABAT:** Abort All Pending Transmissions bit
1 = Abort pending transmissions in all Transmit Buffers
0 = No effect
Note: Module will clear this bit when all transmissions aborted.

bit 11 **CANCKS:** CAN Master Clock Select bit
1 = FCAN clock is FCY
0 = FCAN clock is 4 FCY



bit 10-8 **REQOP<2:0>:** Request Operation Mode bits
111 = Set Listen All Messages mode
110 = Reserved
101 = Reserved
100 = Set Configuration mode
011 = Set Listen Only mode
010 = Set Loopback mode
001 = Set Disable mode
000 = Set Normal Operation mode

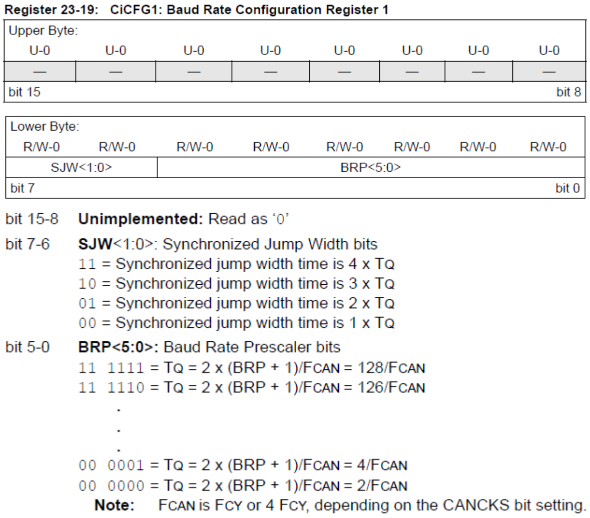
bit 7-5 **OPMODE<2:0>:** Operation Mode bits
Note: These bits indicate the current Operating mode of the CAN module. See description for REQOP bits (CiCTRL<10:8>).

bit 4 **Unimplemented:** Read as ‘0’

bit 3-1 **ICODE<2:0>:** Interrupt Flag Code bits
111 = Wake-up interrupt
110 = RXB0 interrupt
101 = RXB1 interrupt
100 = TXB0 interrupt
011 = TXB1 interrupt
010 = TXB2 interrupt
001 = Error interrupt
000 = No interrupt

bit 0 **Unimplemented:** Read as ‘0’

Configuration du Bit Timing (Baud rate) :



Register 23-19: CiCFG1: Baud Rate Configuration Register 1

Upper Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

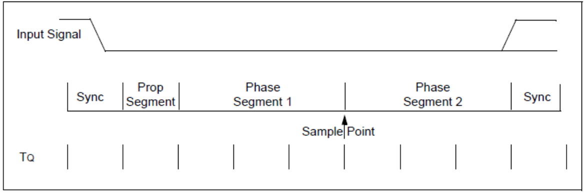
Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SJW<1:0>		BRP<5:0>					
bit 7							bit 0

bit 15-8 **Unimplemented:** Read as ‘0’

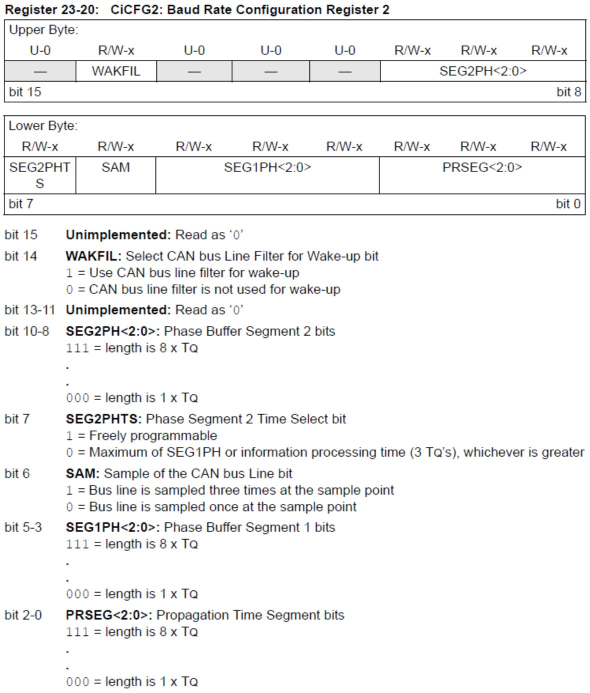
bit 7-6 **SJW**<1:0>: Synchronized Jump Width bits
11 = Synchronized jump width time is 4 x TQ
10 = Synchronized jump width time is 3 x TQ
01 = Synchronized jump width time is 2 x TQ
00 = Synchronized jump width time is 1 x TQ

bit 5-0 **BRP<5:0>:** Baud Rate Prescaler bits
11 1111 = TQ = 2 x (BRP + 1)/FCAN = 128/FCAN
11 1110 = TQ = 2 x (BRP + 1)/FCAN = 126/FCAN
.
.
.
00 0001 = TQ = 2 x (BRP + 1)/FCAN = 4/FCAN
00 0000 = TQ = 2 x (BRP + 1)/FCAN = 2/FCAN
Note: FCAN is FCY or 4 FCY, depending on the CANCKS bit setting.

Figure 23-20: CAN Bit Timing



Avec Fcan ≤ 30 MHz



Register 23-20: CiCFG2: Baud Rate Configuration Register 2

Upper Byte:							
U-0	R/W-x	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	WAKFIL	—	—	—	SEG2PH<2:0>		
bit 15							bit 8

Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SEG2PHTS	SAM	SEG1PH<2:0>			PRSEG<2:0>		
bit 7							bit 0

bit 15 **Unimplemented:** Read as ‘0’

bit 14 **WAKFIL:** Select CAN bus Line Filter for Wake-up bit
1 = Use CAN bus line filter for wake-up
0 = CAN bus line filter is not used for wake-up

bit 13-11 **Unimplemented:** Read as ‘0’

bit 10-8 **SEG2PH<2:0>:** Phase Buffer Segment 2 bits
111 = length is 8 x TQ
.
.
000 = length is 1 x TQ

bit 7 **SEG2PHTS:** Phase Segment 2 Time Select bit
1 = Freely programmable
0 = Maximum of SEG1PH or information processing time (3 TQ’s), whichever is greater

bit 6 **SAM:** Sample of the CAN bus Line bit
1 = Bus line is sampled three times at the sample point
0 = Bus line is sampled once at the sample point

bit 5-3 **SEG1PH<2:0>:** Phase Buffer Segment 1 bits
111 = length is 8 x TQ
.
.
000 = length is 1 x TQ

bit 2-0 **PRSEG<2:0>:** Propagation Time Segment bits
111 = length is 8 x TQ
.
.
000 = length is 1 x TQ

Figure 23-21: Lengthening a Bit Period

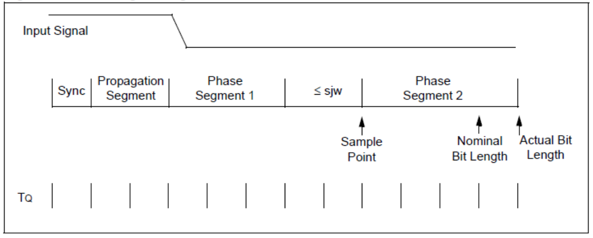
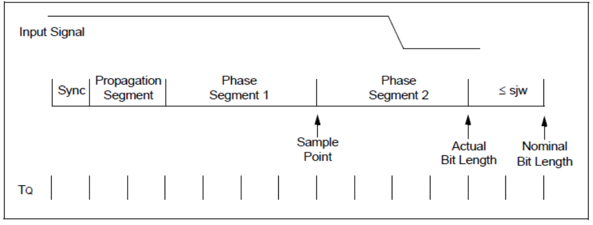
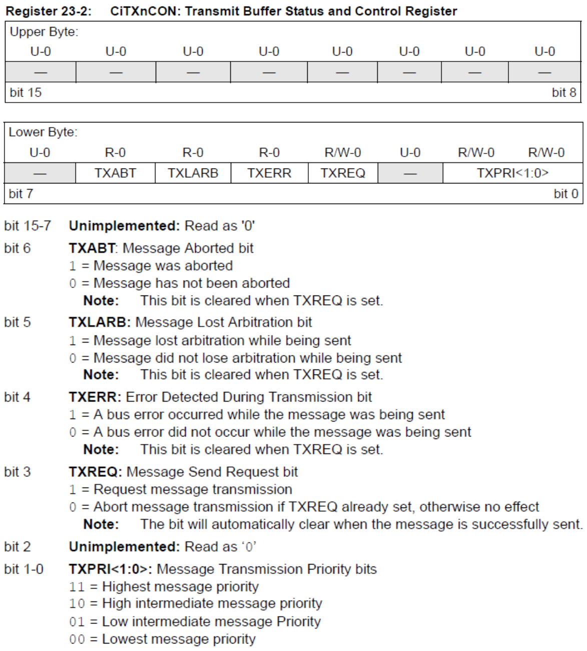


Figure 23-22: Shortening a Bit Period



Registres associés à la transmission :



Register 23-2: CiTXnCON: Transmit Buffer Status and Control Register

Upper Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
U-0	R-0	R-0	R-0	R/W-0	U-0	R/W-0	R/W-0
—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>	
bit 7							bit 0

bit 15-7 **Unimplemented**: Read as '0'

bit 6 **TXABT**: Message Aborted bit
1 = Message was aborted
0 = Message has not been aborted
Note: This bit is cleared when TXREQ is set.

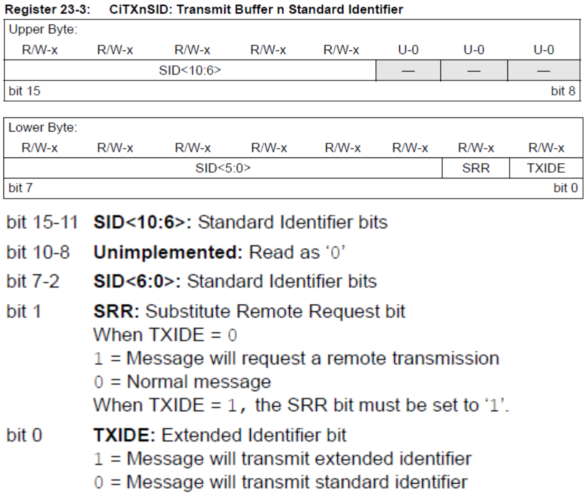
bit 5 **TXLARB**: Message Lost Arbitration bit
1 = Message lost arbitration while being sent
0 = Message did not lose arbitration while being sent
Note: This bit is cleared when TXREQ is set.

bit 4 **TXERR**: Error Detected During Transmission bit
1 = A bus error occurred while the message was being sent
0 = A bus error did not occur while the message was being sent
Note: This bit is cleared when TXREQ is set.

bit 3 **TXREQ**: Message Send Request bit
1 = Request message transmission
0 = Abort message transmission if TXREQ already set, otherwise no effect
Note: The bit will automatically clear when the message is successfully sent.

bit 2 **Unimplemented**: Read as '0'

bit 1-0 **TXPRI<1:0>**: Message Transmission Priority bits
11 = Highest message priority
10 = High intermediate message priority
01 = Low intermediate message Priority
00 = Lowest message priority



Register 23-3: CiTXnSID: Transmit Buffer n Standard Identifier

Upper Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	U-0	U-0	U-0
SID<10:6>					—	—	—
bit 15							bit 8

Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID<5:0>						SRR	TXIDE
bit 7							bit 0

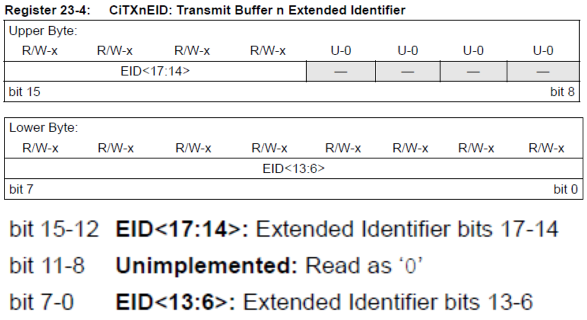
bit 15-11 **SID<10:6>**: Standard Identifier bits

bit 10-8 **Unimplemented**: Read as '0'

bit 7-2 **SID<6:0>**: Standard Identifier bits

bit 1 **SRR**: Substitute Remote Request bit
When TXIDE = 0
1 = Message will request a remote transmission
0 = Normal message
When TXIDE = 1, the SRR bit must be set to '1'.

bit 0 **TXIDE**: Extended Identifier bit
1 = Message will transmit extended identifier
0 = Message will transmit standard identifier



Register 23-4: CiTXnEID: Transmit Buffer n Extended Identifier

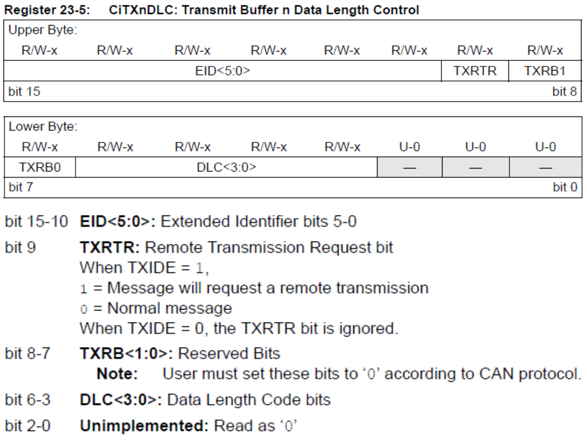
Upper Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	U-0	U-0	U-0	U-0
EID<17:14>				—	—	—	—
bit 15							bit 8

Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID<13:6>							
bit 7							bit 0

bit 15-12 **EID<17:14>**: Extended Identifier bits 17-14

bit 11-8 **Unimplemented:** Read as '0'

bit 7-0 **EID<13:6>**: Extended Identifier bits 13-6



Register 23-5: CiTXnDLC: Transmit Buffer n Data Length Control

Upper Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID<5:0>						TXRTR	TXRB1
bit 15							bit 8

Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	U-0	U-0	U-0
TXRB0	DLC<3:0>				—	—	—
bit 7							bit 0

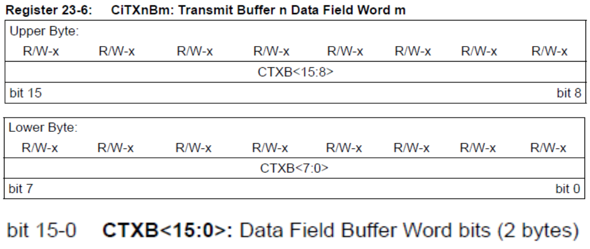
bit 15-10 **EID<5:0>**: Extended Identifier bits 5-0

bit 9 **TXRTR**: Remote Transmission Request bit
When TXIDE = 1,
1 = Message will request a remote transmission
0 = Normal message
When TXIDE = 0, the TXRTR bit is ignored.

bit 8-7 **TXRB<1:0>**: Reserved Bits
Note: User must set these bits to '0' according to CAN protocol.

bit 6-3 **DLC<3:0>**: Data Length Code bits

bit 2-0 **Unimplemented**: Read as '0'



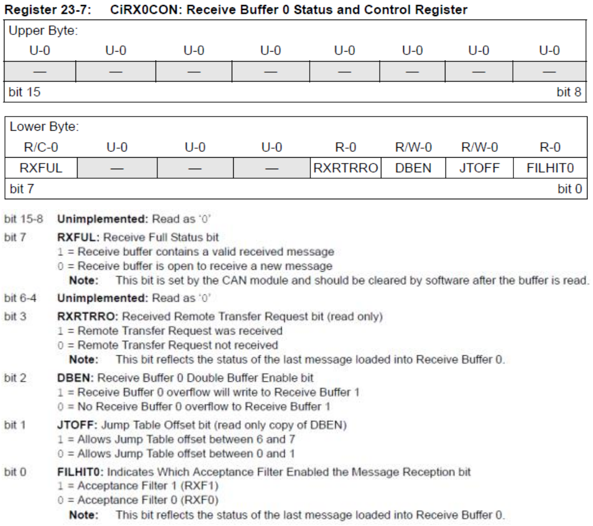
Register 23-6: CiTXnBm: Transmit Buffer n Data Field Word m

Upper Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CTXB<15:8>							
bit 15							bit 8

Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CTXB<7:0>							
bit 7							bit 0

bit 15-0 **CTXB<15:0>**: Data Field Buffer Word bits (2 bytes)

Registres associés à la réception :



Register 23-7: CiRX0CON: Receive Buffer 0 Status and Control Register

Upper Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
R/C-0	U-0	U-0	U-0	R-0	R/W-0	R/W-0	R-0
RXFUL	—	—	—	RXRTRRO	DBEN	JTOFF	FILHIT0
bit 7							bit 0

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7 **RXFUL:** Receive Full Status bit
1 = Receive buffer contains a valid received message
0 = Receive buffer is open to receive a new message
Note: This bit is set by the CAN module and should be cleared by software after the buffer is read.

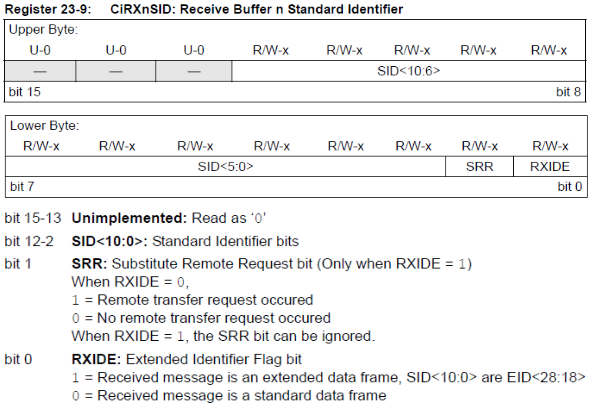
bit 6-4 **Unimplemented:** Read as ‘0’

bit 3 **RXRTRRO:** Received Remote Transfer Request bit (read only)
1 = Remote Transfer Request was received
0 = Remote Transfer Request not received
Note: This bit reflects the status of the last message loaded into Receive Buffer 0.

bit 2 **DBEN:** Receive Buffer 0 Double Buffer Enable bit
1 = Receive Buffer 0 overflow will write to Receive Buffer 1
0 = No Receive Buffer 0 overflow to Receive Buffer 1

bit 1 **JTOFF:** Jump Table Offset bit (read only copy of DBEN)
1 = Allows Jump Table offset between 6 and 7
0 = Allows Jump Table offset between 0 and 1

bit 0 **FILHIT0:** Indicates Which Acceptance Filter Enabled the Message Reception bit
1 = Acceptance Filter 1 (RXF1)
0 = Acceptance Filter 0 (RXF0)
Note: This bit reflects the status of the last message loaded into Receive Buffer 0.



Register 23-9: CiRXnSID: Receive Buffer n Standard Identifier

Upper Byte:							
U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	SID<10:6>				
bit 15							bit 8

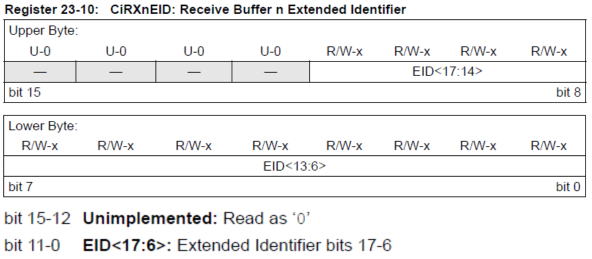
Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID<5:0>						SRR	RXIDE
bit 7							bit 0

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-2 **SID<10:0>:** Standard Identifier bits

bit 1 **SRR:** Substitute Remote Request bit (Only when RXIDE = 1)
When RXIDE = 0,
1 = Remote transfer request occured
0 = No remote transfer request occured
When RXIDE = 1, the SRR bit can be ignored.

bit 0 **RXIDE:** Extended Identifier Flag bit
1 = Received message is an extended data frame, SID<10:0> are EID<28:18>
0 = Received message is a standard data frame



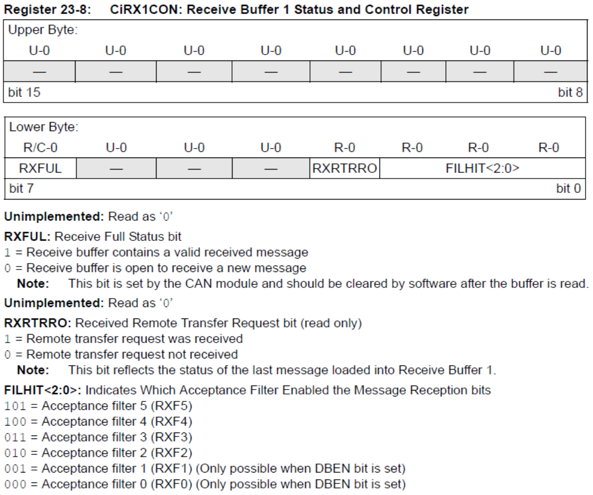
Register 23-10: CiRXnEID: Receive Buffer n Extended Identifier

Upper Byte:							
U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	EID<17:14>			
bit 15							bit 8

Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID<13:6>							
bit 7							bit 0

bit 15-12 **Unimplemented:** Read as ‘0’

bit 11-0 **EID<17:6>:** Extended Identifier bits 17-6



Register 23-8: CiRX1CON: Receive Buffer 1 Status and Control Register

Upper Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
R/C-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
RXFUL	—	—	—	RXRTRRO	FILHIT<2:0>		
bit 7							bit 0

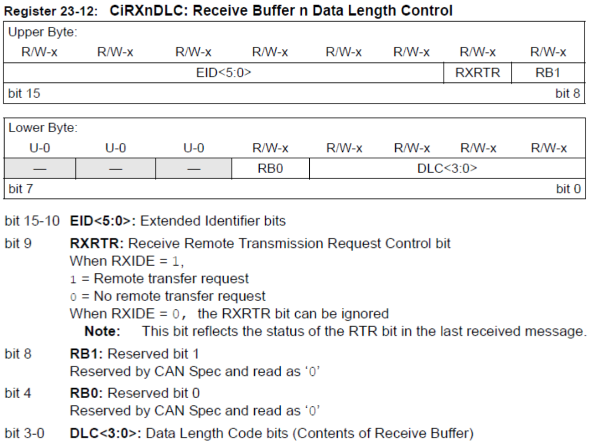
Unimplemented: Read as ‘0’

RXFUL: Receive Full Status bit
1 = Receive buffer contains a valid received message
0 = Receive buffer is open to receive a new message
Note: This bit is set by the CAN module and should be cleared by software after the buffer is read.

Unimplemented: Read as ‘0’

RXRTRRO: Received Remote Transfer Request bit (read only)
1 = Remote transfer request was received
0 = Remote transfer request not received
Note: This bit reflects the status of the last message loaded into Receive Buffer 1.

FILHIT<2:0>: Indicates Which Acceptance Filter Enabled the Message Reception bits
101 = Acceptance filter 5 (RXF5)
100 = Acceptance filter 4 (RXF4)
011 = Acceptance filter 3 (RXF3)
010 = Acceptance filter 2 (RXF2)
001 = Acceptance filter 1 (RXF1) (Only possible when DBEN bit is set)
000 = Acceptance filter 0 (RXF0) (Only possible when DBEN bit is set)



Register 23-12: CiRXnDLC: Receive Buffer n Data Length Control

Upper Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID<5:0>						RXRTR	RB1
bit 15							bit 8

Lower Byte:							
U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	RB0	DLC<3:0>			
bit 7							bit 0

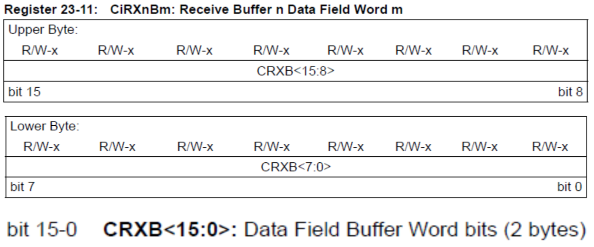
bit 15-10 **EID<5:0>:** Extended Identifier bits

bit 9 **RXRTR:** Receive Remote Transmission Request Control bit
When RXIDE = 1,
1 = Remote transfer request
0 = No remote transfer request
When RXIDE = 0, the RXRTR bit can be ignored
Note: This bit reflects the status of the RTR bit in the last received message.

bit 8 **RB1:** Reserved bit 1
Reserved by CAN Spec and read as ‘0’

bit 4 **RB0:** Reserved bit 0
Reserved by CAN Spec and read as ‘0’

bit 3-0 **DLC<3:0>:** Data Length Code bits (Contents of Receive Buffer)



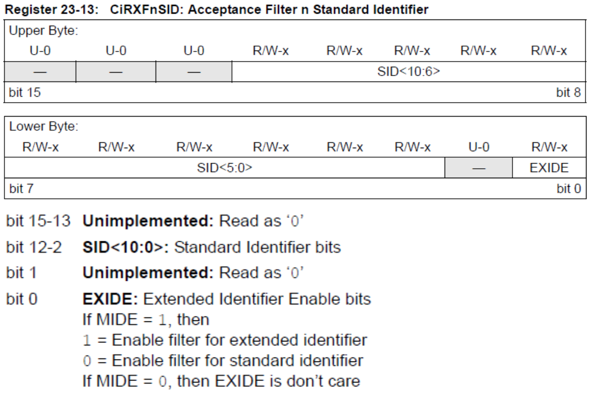
Register 23-11: CiRXnBm: Receive Buffer n Data Field Word m

Upper Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CRXB<15:8>							
bit 15							bit 8

Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CRXB<7:0>							
bit 7							bit 0

bit 15-0 **CRXB<15:0>:** Data Field Buffer Word bits (2 bytes)

Filtres des messages et masques de filtrage :



Register 23-13: CiRXFnSID: Acceptance Filter n Standard Identifier

Upper Byte:							
U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	SID<10:6>				
bit 15							bit 8

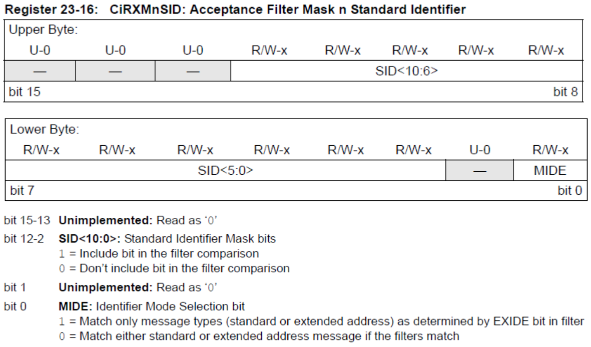
Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	U-0	R/W-x
SID<5:0>						—	EXIDE
bit 7							bit 0

bit 15-13 **Unimplemented:** Read as '0'

bit 12-2 **SID<10:0>:** Standard Identifier bits

bit 1 **Unimplemented:** Read as '0'

bit 0 **EXIDE:** Extended Identifier Enable bits
If MIDE = 1, then
1 = Enable filter for extended identifier
0 = Enable filter for standard identifier
If MIDE = 0, then EXIDE is don't care



Register 23-16: CiRXMnSID: Acceptance Filter Mask n Standard Identifier

Upper Byte:							
U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	SID<10:6>				
bit 15							bit 8

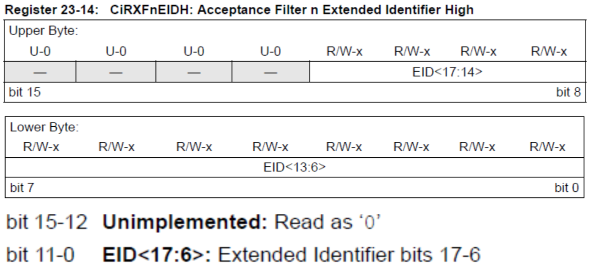
Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	U-0	R/W-x
SID<5:0>						—	MIDE
bit 7							bit 0

bit 15-13 **Unimplemented:** Read as '0'

bit 12-2 **SID<10:0>:** Standard Identifier Mask bits
1 = Include bit in the filter comparison
0 = Don't include bit in the filter comparison

bit 1 **Unimplemented:** Read as '0'

bit 0 **MIDE:** Identifier Mode Selection bit
1 = Match only message types (standard or extended address) as determined by EXIDE bit in filter
0 = Match either standard or extended address message if the filters match



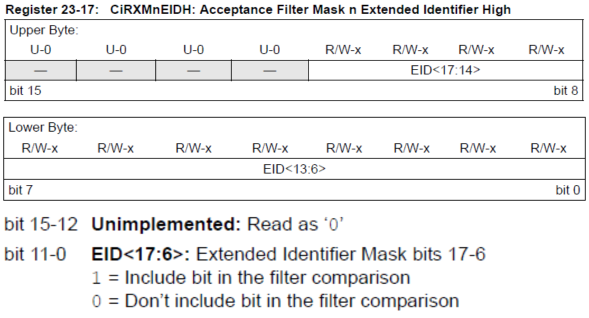
Register 23-14: CiRXFnEIDH: Acceptance Filter n Extended Identifier High

Upper Byte:							
U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	EID<17:14>			
bit 15							bit 8

Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID<13:6>							
bit 7							bit 0

bit 15-12 **Unimplemented:** Read as '0'

bit 11-0 **EID<17:6>:** Extended Identifier bits 17-6



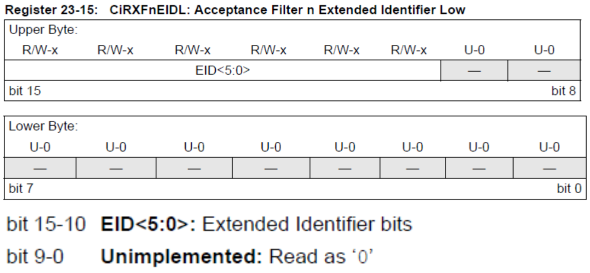
Register 23-17: CiRXMnEIDH: Acceptance Filter Mask n Extended Identifier High

Upper Byte:							
U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	EID<17:14>			
bit 15							bit 8

Lower Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID<13:6>							
bit 7							bit 0

bit 15-12 **Unimplemented:** Read as '0'

bit 11-0 **EID<17:6>:** Extended Identifier Mask bits 17-6
1 = Include bit in the filter comparison
0 = Don't include bit in the filter comparison



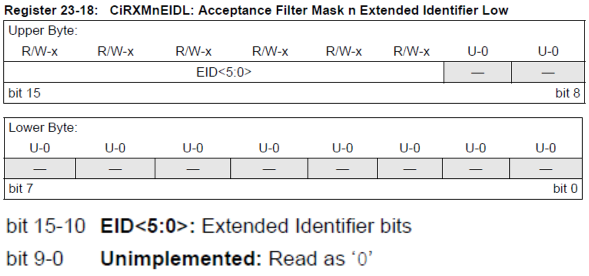
Register 23-15: CiRXFnEIDL: Acceptance Filter n Extended Identifier Low

Upper Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	U-0	U-0
EID<5:0>						—	—
bit 15							bit 8

Lower Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

bit 15-10 **EID<5:0>:** Extended Identifier bits

bit 9-0 **Unimplemented:** Read as '0'



Register 23-18: CiRXMnEIDL: Acceptance Filter Mask n Extended Identifier Low

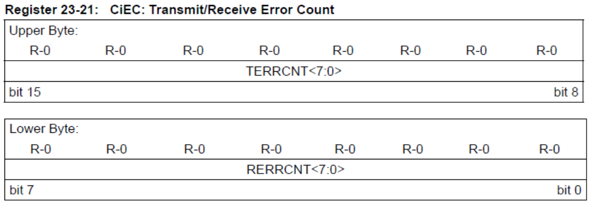
Upper Byte:							
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	U-0	U-0
EID<5:0>						—	—
bit 15							bit 8

Lower Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

bit 15-10 **EID<5:0>:** Extended Identifier bits

bit 9-0 **Unimplemented:** Read as '0'

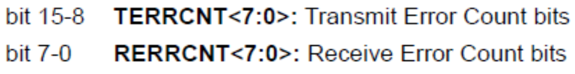
Compteurs d'erreurs :



Register 23-21: CiEC: Transmit/Receive Error Count

Upper Byte:							
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TERRCNT<7:0>							
bit 15							bit 8

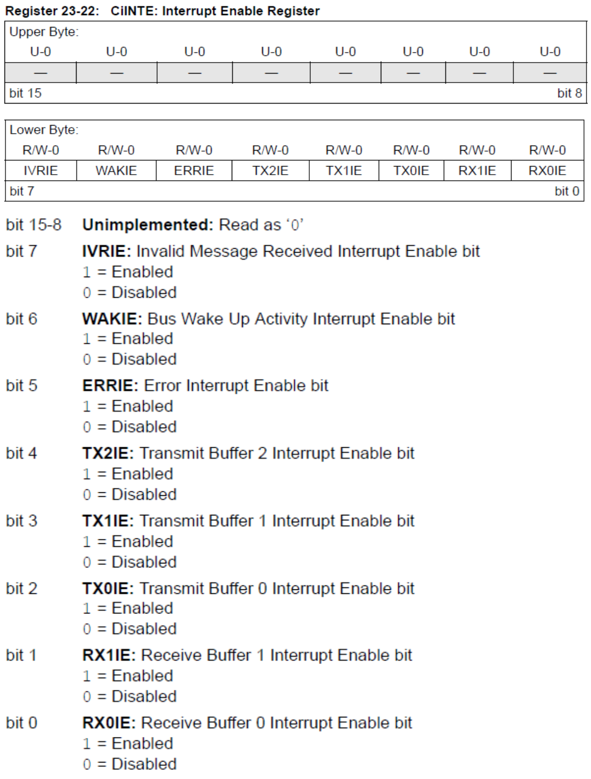
Lower Byte:							
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RERRCNT<7:0>							
bit 7							bit 0



bit 15-8 **TERRCNT<7:0>:** Transmit Error Count bits

bit 7-0 **RERRCNT<7:0>:** Receive Error Count bits

Registres de gestion des interruptions :



Register 23-22: CiINTE: Interrupt Enable Register

Upper Byte:

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

Lower Byte:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IVRIE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE
bit 7							bit 0

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **IVRIE:** Invalid Message Received Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **WAKIE:** Bus Wake Up Activity Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **ERRIE:** Error Interrupt Enable bit
1 = Enabled
0 = Disabled

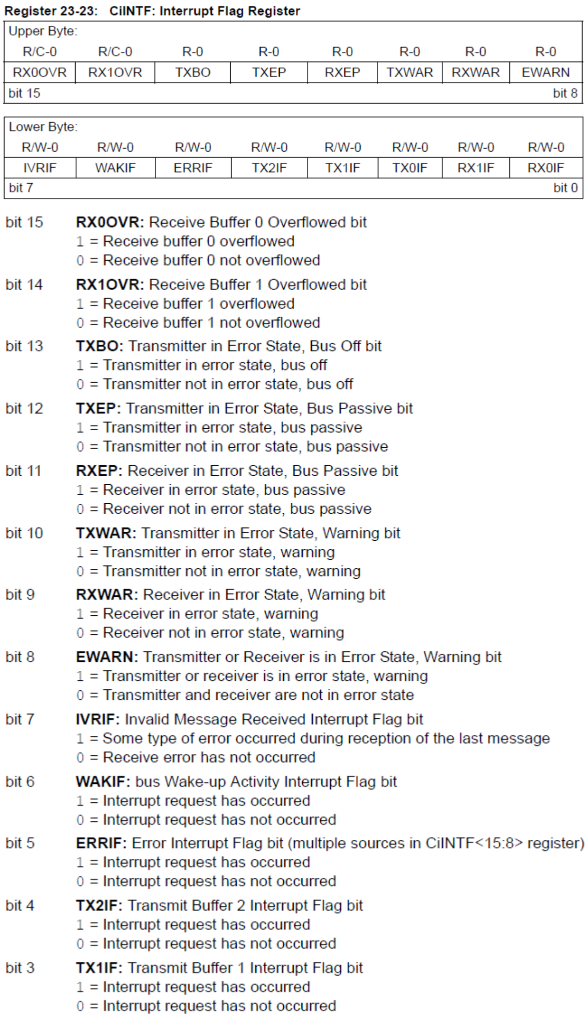
bit 4 **TX2IE:** Transmit Buffer 2 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 3 **TX1IE:** Transmit Buffer 1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **TX0IE:** Transmit Buffer 0 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **RX1IE:** Receive Buffer 1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **RX0IE:** Receive Buffer 0 Interrupt Enable bit
1 = Enabled
0 = Disabled



Register 23-23: CiINTF: Interrupt Flag Register

Upper Byte:

R/C-0	R/C-0	R-0	R-0	R-0	R-0	R-0	R-0
RX0OVR	RX1OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN
bit 15							bit 8

Lower Byte:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IVRIF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF
bit 7							bit 0

bit 15 **RX0OVR:** Receive Buffer 0 Overflowed bit
1 = Receive buffer 0 overflowed
0 = Receive buffer 0 not overflowed

bit 14 **RX1OVR:** Receive Buffer 1 Overflowed bit
1 = Receive buffer 1 overflowed
0 = Receive buffer 1 not overflowed

bit 13 **TXBO:** Transmitter in Error State, Bus Off bit
1 = Transmitter in error state, bus off
0 = Transmitter not in error state, bus off

bit 12 **TXEP:** Transmitter in Error State, Bus Passive bit
1 = Transmitter in error state, bus passive
0 = Transmitter not in error state, bus passive

bit 11 **RXEP:** Receiver in Error State, Bus Passive bit
1 = Receiver in error state, bus passive
0 = Receiver not in error state, bus passive

bit 10 **TXWAR:** Transmitter in Error State, Warning bit
1 = Transmitter in error state, warning
0 = Transmitter not in error state, warning

bit 9 **RXWAR:** Receiver in Error State, Warning bit
1 = Receiver in error state, warning
0 = Receiver not in error state, warning

bit 8 **EWARN:** Transmitter or Receiver is in Error State, Warning bit
1 = Transmitter or receiver is in error state, warning
0 = Transmitter and receiver are not in error state

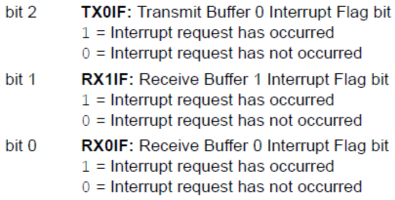
bit 7 **IVRIF:** Invalid Message Received Interrupt Flag bit
1 = Some type of error occurred during reception of the last message
0 = Receive error has not occurred

bit 6 **WAKIF:** bus Wake-up Activity Interrupt Flag bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **ERRIF:** Error Interrupt Flag bit (multiple sources in CiINTF<15:8> register)
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **TX2IF:** Transmit Buffer 2 Interrupt Flag bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **TX1IF:** Transmit Buffer 1 Interrupt Flag bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred



bit 2 **TX0IF:** Transmit Buffer 0 Interrupt Flag bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **RX1IF:** Receive Buffer 1 Interrupt Flag bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **RX0IF:** Receive Buffer 0 Interrupt Flag bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred