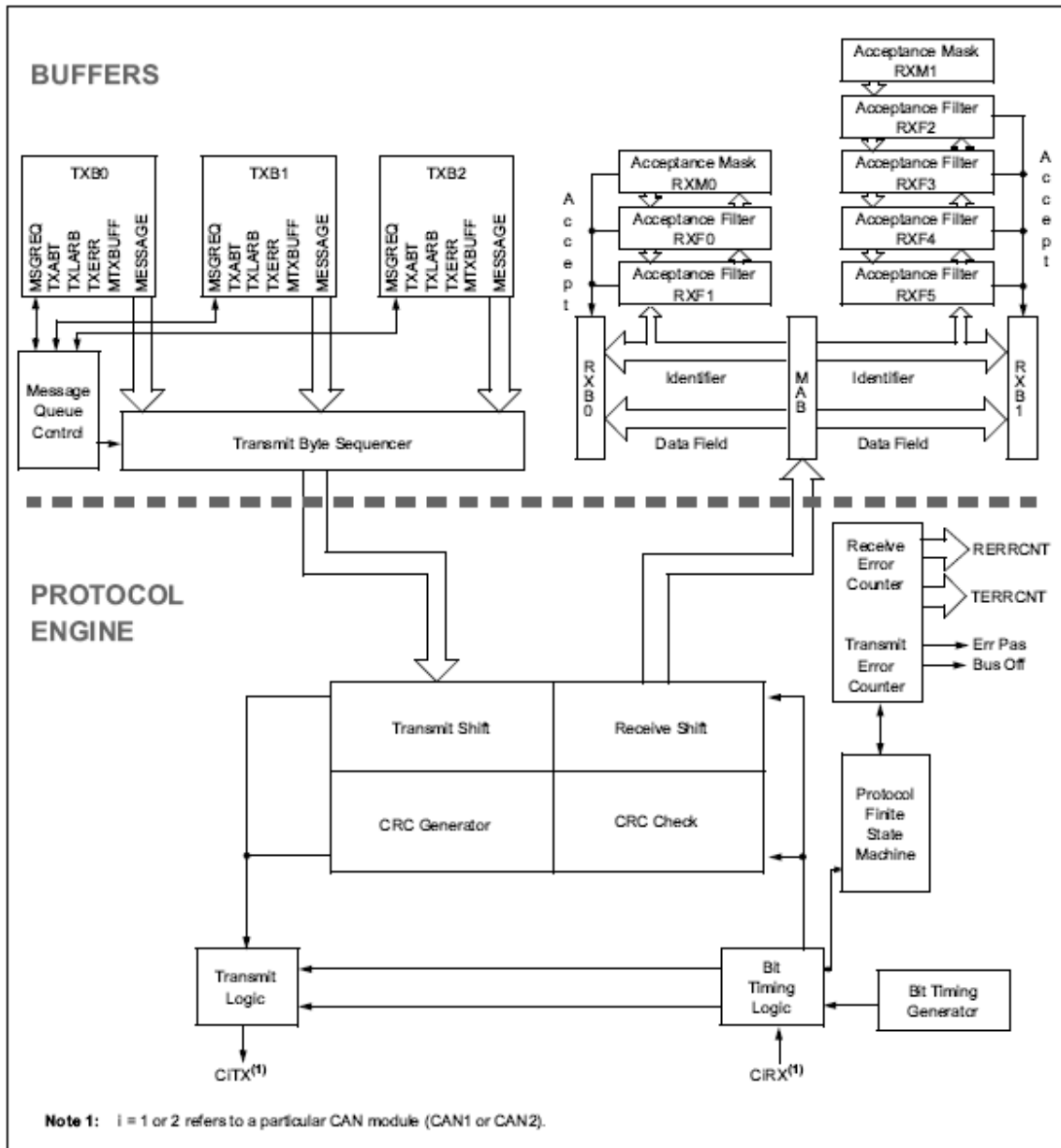


***Funcionamiento del  
controlador CAN en el  
dsPIC30F4013***

***José Manuel Peña  
Borja Miñana  
Carles Cortés  
Carlos Romero  
Alejandro Doménech***

**FIGURA 1: BUFFERS DEL BUS CAN Y DIAGRAMA DEL MOTOR DE PROTOCOLO**



## 1. **MODOS DE OPERACIÓN**

El módulo CAN puede funcionar en uno de varios modos de Operación elegido por el usuario. Estos modos incluyen:

- Modo de Inicialización
- Modo Inhabilitado
- Modo de Operación Normal
- Modo de Sólo de Escucha
- Modo de Loopback
- Modo del Reconocimiento del Error

Los modos son solicitados fijando los bits REQOP<2:0> (CiCTRL<10:8>), excepto el modo del reconocimiento del error el cual es requerido por los bits RXM<1:0> (CiRXnCON<6:5>), donde n=0 o 1 representa un detalle que termina en el buffer.

La entrada en un modo es conocida gracias a los bits POMODE<2:0> (CiCTRL<7:5>). El módulo no cambiará el modo y los bits OPMODE son aceptables hasta un cambio de modo, normalmente durante la marcha lenta del bus el cual está definido como por lo menos 11 bits recesivos consecutivos.

### 1.1 **Modo de inicialización**

En el modo de inicialización, como su propio nombre indica, es empezar parecido a un reset lo único que en ese instante se puede modificar como el usuario desee y en este instante el módulo no transmitirá ni recibirá. Los contadores de error son despejados y las banderas de interrupción permanecen sin cambiar. El programador tendrá acceso a los registros de configuración que son accesos restringidos en otros modos. El módulo protegerá al usuario de la violación accidental del protocolo CAN con errores de programación. Todos los registros que controlan la configuración de los módulos no pueden ser modificados mientras el módulo esté on-line. El módulo CAN no tendrá permitido entrar al modo de configuración mientras una transmisión se esté llevando a cabo. El modo de configuración sirve como una cerradura para proteger los registros siguientes:

- \* Todos los registros de control del módulo
- \* Registros de la configuración de la velocidad y de la interrupción.
- \* Registros de la sincronización del bus
- \* Registros de aceptación de filtro del identificador
- \* Registros de aceptación de máscara del identificador

## **1.2 Modo inhabilitado**

En el modo inhabilitado, el módulo no transmitirá ni recibirá. El módulo tiene la capacidad de fijar el bit del WAKIF debido a la actividad del bus, sin embargo, cualquier interrupción pendiente permanecerá y los contadores de error conservarán su valor.

Si los bits REQOP<2:0> (CiCTRL<10:8>)=001, el módulo entrará al modo de Módulo Inhabilitado. Si el módulo está activo, el módulo esperará por 11 recesivos bits en el CAN bus, detecta esta condición como marcha lenta del bus, entonces acepta el comando de módulo inhabilitado.

Cuando los bits OPMODE<2:0> (CiCTRL<7:5>)=001, eso indica si el módulo entró con éxito al modo de módulo inhabilitado. Los pins de I/O invertirán a la función normal de I/O cuando el módulo está en el modo módulo inhabilitado. El módulo se puede programar para aplicar una función low-pass del filtro a la línea de entrada de CiRX mientras el módulo o la cpu esté en modo de descanso.

## **1.3 Modo de operación normal**

El modo de operación normal es seleccionado cuando REQOP<2:0>=000. En este modo, el módulo esta activado y los pins I/O asumirán las funciones del CAN bus. El módulo transmitirá y recibirá mensajes del bus CAN vía los pins CxTX y CxRX.

## **1.4 Modo de escucha de todos los mensajes**

El módulo se puede fijar para no hacer caso de todos los errores y para recibir cualquier mensaje. El modo de escucha de todos los mensajes se activa fijando los bits REQOP<2:0> a "111". En este modo, el dato que está en el buffer intermediario del montaje del mensaje hasta el tiempo que un error ocurra, se copia en el buffer intermediario de la recepción y puede ser leído vía la interfaz de la CPU.

## **1.5 Modo de LoopBack**

En este modo se verifica si hay un buen funcionamiento en el módulo. Si está activado el modo Loopback, el módulo conectará la señal de transmisión interna a la señal de recepción interna, se genera una señal y luego se verifica que es la misma que se ha generado. La transmisión y recepción de los pins invierten la función de los puertos I/O. en el límite del módulo.

## **2. RECEPCIÓN DE MENSAJES**

### **2.1 Buffers de recepción**

El módulo del bus CAN tiene 3 buffers de recepción. Aunque, uno de los búffers siempre se encarga de monitorizar el bus de los mensajes entrantes. Este búffer es el (Message Assembly Buffer (MAB).

Aunque hay 2 buffers de recepción visibles, RXB0 y RXB1, que pueden recibir el mensaje instantaneamente desde el motor del protocolo. Todos los mensajes son montados por el MAB y transferidos a los almacenadores intermediarios de RXB solamente si se resuelve el criterio del filtro de la aceptación. Cuando se recibe un mensaje, la bandera de RxnIF (CiINTF<0> o CiINRF<1>) una interrupción será generada cuando se reciba un mensaje. RXF0 y los filtros RXF1 están asociados con RXB0. Los filtros RXF2, RXF3, RXF4 y RXF5 y la máscara RXM1 se asocian con RXB1.

### **2.2 Filtros de aceptación de mensajes**

El mensaje acepta los filtros y las máscaras se encargan de determinar si un mensaje en el almacenador hace de intermediario del buffer de recepción. Una vez que un mensaje válido se haya recibido en el almacenador intermediario del MAB, los campos del identificador del mensaje se comparan con los valores del filtro. Si hay una coincidencia, el mensaje será cargado en el buffer apropiado de recepción. El filtro de aceptación se muestra como un mensaje entrante para el bit RXID (CiRXnSID<0>) para determinar como se comparan los identificadores. Si el bit RXIDE está borrado, el mensaje es un marco estándar y solamente los filtros con el bit EXIDE se comparan. Si se fija el RXIDE, el mensaje es un marco extendido, y solamente los filtros con el sistema del EXIDE se comparan. Configuración del RXM<1:0> el bit de EXIDE puede ser sobrepasado.

### **2.3 Máscaras del filtro de la aceptación de mensajes**

Los bits de la máscara esencialmente determinan qué bit va a ser aplicado al filtro. Si cualquier bit de la máscara se fija a un cero, entonces ese bit será aceptado automáticamente sin importar el bit del filtro. Hay 2 máscaras programables asociadas a los almacenadores intermediarios del receptor, una del filtro de la aceptación para cada almacenador intermediario.

## 2.4 Latencia del receptor

Una condición del receptor es la latencia, ocurre cuando el almacenador intermediario del (MAB) ha montado un mensaje recibido válido, el mensaje se acepta a través de los filtros de la aceptación, y cuando el almacenador intermediario de la recepción asociado al filtro no se ha señalado claramente el anterior mensaje. Si se fija el bit de DBEN, la latencia RXB0 se maneja de diferente forma. Si un mensaje válido se recibe para RXB0 y RXFUL = 1 lo cual indica que RXB0 esta lleno; y RXFUL = 0 indica que RXB1 esta vacío, el mensaje para RXB0 será cargado en RXB1. RXB0 no generará error en la latencia. Si un mensaje es válido se recibe para RXB0 y RXFUL = 1, indicando que RXB0 y RXB1 estan llenos y el mensaje será perdido.

## 2.5 Errores en la recepción

El módulo CAN detectará los siguientes errores de recepción.

- Error de redundancia cíclica
- Error del bit sobrante
- Mensajes de error de recepción

Este error de recepción no utiliza ninguna interrupción. Sin embargo el error de recepción incrementa en uno los sucesos de errores. El bit de RXWAR (CiINTF<9>) indica que la recepción del error ha sido rechazada por la CPU debido al límite de interrupciones generadas (96).

## 2.6 Interrupciones de recepción

Las interrupciones de recepción se dividen en 3 grupos, cada uno incluye varias condiciones que generan interrupciones:

- Interrupción de recepción: El mensaje ha sido recibido y se ha cargado con éxito en uno de los buffers intermediarios de recepción. Esta interrupción se activa inmediatamente despues de recibir en el otro extremo del (EOF). La lectura de la flag RxnIF indicará que cuál recibe el buffer intermediario.
- Interrupción Wake-up: El módulo CAN se activa para inhabilitar el modo o el dispositivo que ha sido activado en el modo sleep.
- Interrupciones de error de recepción: Una interrupción de error de recepción estará indicada por el bit ERRIF. Este bit muestra que ha ocurrido un error de condición. El motivo del error puede ser determinado repasando los bits del registro de estado de interrupción del controlador CAN, CiINTF.

-Mensaje inválido recibido: Si ocurre un error durante la recepción del último mensaje, el error será mostrado mediante el bit IVRIF.

-El receptor sobrepasa el tiempo previsto: El bit RXnOVR indica que se ha producido un error de tiempo de espera.

-Alerta del receptor: El bit RXWAR indica que el contador de recepción de error (RERRCNT<7:0>) ha excedido el máximo permitido de 96.

-Error de receptor pasivo: El bit RXEP indica que el contador de recepción de errores ha excedido el límite de error pasivo de 127 y el módulo ha pasado al estado de error pasivo.

### **3. TRANSMISIÓN DE MENSAJES**

#### **3.1 Buffers de transmisión**

El módulo CAN tiene 3 módulos de transmisión. Cada uno de los buffers ocupa 14 bytes de datos. Ocho de los bytes son el máximo de 8 bytes del mensaje transmitido. Cinco bytes son los que guardan los identificadores estándar y extendido además de diversa información arbitraria del mensaje.

#### **3.2 Prioridad de transmisión de mensajes**

La prioridad de transmisión es una priorización de cada uno de los nodos de los mensajes pendientes de transmitir. Hay 4 niveles de prioridad de transmisión. Si TXPRI<1:0> (CiTXnCON<1:0> donde n= 0, 1 ó 2 representa el buffer transmitido) está configurado con "11" este buffer tiene la máxima prioridad. Si TXPRI <1:0> está configurado con "10" ó "01" este buffer tiene prioridad intermedia. Si TXPRI <1:0> tiene un "00" entonces el buffer tiene prioridad mínima.

### 3.3 Secuencia de transmisión

Para iniciar la transmisión del mensaje, el bit TXREQn debe ser configurado. El módulo del bus CAN resuelve cualquier conflicto de tiempo entre la configuración del bit TXREQ y el inicio de trama (SOF), asegurando que si la prioridad a cambiado, será resuelta correctamente antes de que ocurra el evento "Inicio de trama (SOF)". Cuando el bit TXREQ esta configurado, los bits de bandera TXABT(CiTXnCON<6>), TXLARB(CiTXnCON<5>) y TXERR (CiTXnCON<4>) son automáticamente borrados.

Al configurar el bit TXREQ simplemente se muestra un mensaje del buffer en cola para ser transmitido. Cuando el módulo detecta un bus disponible, este empieza a transmitir el mensaje al que se le ha dado la prioridad más alta.

Si la transmisión se completa satisfactoriamente al primer intento, entonces el contenido del bit TXREQ se borra automáticamente, y se genera una interrupción si el comando TXIE fue configurado anteriormente.

Si la transmisión del mensaje falla, una de las banderas de condición se confirma, y el bit TXREQ permanece configurado indicando que el mensaje está todavía pendiente de ser transmitido.

Si el mensaje contabiliza un error de condición durante el intento de transmisión, el bit TXERR se confirma y el error de condición puede que cause una interrupción. Si el mensaje pierde arbitrariedad durante el intento de transmisión el bit TXLARB se confirma. La pérdida de arbitrariedad no genera una interrupción de la señal.

### 3.4 Abortar la transmisión del mensaje.

El sistema también puede abortar un mensaje mediante el borrado del bit TXREQ asociado con cada buffer del mensaje. Mediante la configuración del bit ABAT (CiCTRL<12>) se produce el aborto de todos los mensajes pendientes.

Si el mensaje no se ha empezado a transmitir, o el mensaje se ha empezado a transmitir pero ha sido interrumpido ( ya sea por una pérdida de arbitrariedad o por un error) entonces se procesa el aborto del mensaje. El aborto del mensaje se indica mediante la configuración del bit TXABT mientras la bandera TXnIF no está configurada automáticamente.



### 3.5 Errores de transmisión

El módulo CAN detecta los siguientes errores de transmisión:

- Acknowledge Error
- From Error
- Bit Error

Estos errores de transmisión no producen necesariamente una interrupción pero son indicados por el contador de errores de transmisión.

Sin embargo, cada uno de estos errores hace que el controlador de errores de transmisión aumente en uno. Una vez que el contador supera los 96 errores los bits ERRIF (CiINTF<5>) y TXWAR (CiINTF<10>) se configuran. Una vez que se exceden los 96 errores en el contador de error se produce una interrupción y el bit TXWAR se configura en el registro de bandera de error.

### 3.6 Interrupciones de transmisión

- Interrupción de transmisión: Cuando uno de los 3 buffers de transmisión esta libre y se puede utilizar para transmitir un mensaje, se activa una bandera y una interrupción en el registro TXnIF que indica que buffer puede ser utilizado.
- Interrupción de error en transmisión: Se produce una interrupción cuando ocurre un error, la fuente del error se escribe en el registro ERRIF flag se compara con el CAN interrupt status register (CiINTF) para averiguar que error es.
- Interrupción de peligro en la transmisión: Cuando el número de errores en transmisión llega al límite de la CPU (96), El bit TXWAR se activa.
- Error de transmisión pasiva: El bit TXEP del registro (CiINTF<12>) indica que han ocurrido 127 y el módulo entra en estado de error pasivo.
- Bus en off: El bit TXB= (CiINTF<13>) indica que el contador de errores ha llegado a su máximo (255) se aísla del bus.

## 4. AJUSTES DE VELOCIDAD

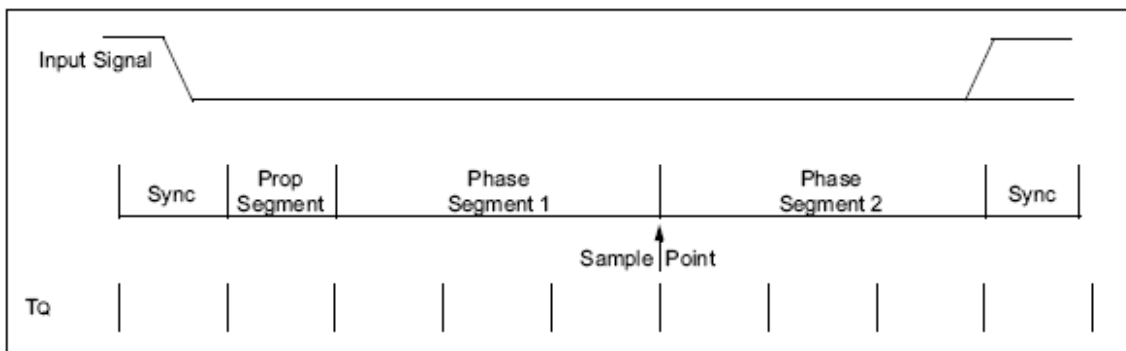
Todos los nodos del bus CAN deben tener el mismo bit rate (velocidad de transmisión) normalmente todos tienen la misma velocidad pero a veces diferentes controladores no tienen el mismo oscilador y hay que ajustar los siguientes parámetros para conseguir la misma velocidad:

- Synchronization jump width
- Baud rate prescaler
- Phase segments
- Length determination of phase segment 2
- Sample point

### 4.1 Bit timing

El tiempo de bit nominal, se divide en segmentos de tiempos más pequeños no solapados, para que se sincronicen todos los nodos, los segmentos de tiempos se llaman  $T_q$ , pueden haber un mínimo de  $8T_q$  hasta un máximo de  $25T_q$ . El tiempo de bit es de 1 seg siendo el máximo bit rate de un 1MHz.

FIGURA 2: TEMPORIZACIÓN DEL BIT EN EL CAN



### 4.2 Configuración de la preescala

Tenemos un valor de preescala programable de 1 hasta 64, a parte tenemos un valor fijo divide-por-dos para el reloj; el valor de tiempo es una unidad fija de tiempo resultante del valor del oscilador, y es dada por la ecuación 17-1, donde  $f_{CAN}$  es  $f_{cy}$ . (donde se da el valor de bit de los CANKS) o  $4f_{cy}$  (si los CANKS están libres)

Nota: Fcan no puede exceder 30Mhz. Si CANKS =0 entonces Fcy no deberá exceder 7,5 Mhz.

ECUACIÓN 17-1: Contador de tiempo para el generador del reloj

$$TQ = 2 (BRP<5:0>+1)/Fcan$$

### 4.3 Segmento de propagación

Esta parte del tiempo de bit es usada para compensar el retardo físico entre la red. Este tiempo de retardo consiste en la señal de propagación del tiempo en la línea de bus y en el retardo interno de los nodos. El Prop Seg puede ser programado desde 1 TQ hasta 8 TQ poniendo el PRSEG<2:0 bits (CiCFG2<2:0>).

### 4.4 Segmentos de fase

Los segmentos de fase son usados para óptimamente localizar el simple del bit recibido entre el tiempo de transmisión de bit. El Sampling point esta entre Phase1 seg y Phase2 seg. Estos segmentos són alargados o acortados por la resincronizacion. El final de la Phase1 Seg determina el sampling point entre el periodo de bit. El segmento es programable desde 1 TQ hasta 8 TQ. Phase2 Seg provee retardo hasta la siguiente transmisión de datos. El segmento es programable desde 1 TQ hasta 8 TQ o también puede ser definido para ser igual al mayor de Phase1 Seg o de la información del tiempo de procesado (2TQ). El Phase1 Seg es inicializado poniendo los bits SEG1PH<2:0> (CiCFG2<5:3>), Phase2 Seg es inicializado poniendo SEG2PH<2:0> (CiCFG2<10:8>).

Los siguientes requerimientos deben ser completados mientras se pone el tamaño del segmento de fase:

$$\text{Prop Seg} + \text{Phase1 Seg} \geq \text{Phase2 Seg}$$

### 4.5 Punto de ejemplo

El punto de ejemplo es el punto de tiempo donde el nivel del bus es leído e interpretado como el valor del respectivo bit. La localización es en el final de Phase1 Seg. Si el tiempo de bit es lento y contiene muchos TQ es posible especificar múltiples ejemplos de la línea de bus en el punto de ejemplo. El nivel determinado por el bus CAN corresponde a la mayoría de decisión de 3 valores. La mayoría de ejemplos son cogidos en los puntos de ejemplo y ambos con una distancia de TQ/2. El modulo CAN permite al usuario escoger entre analizar 4 tiempos en el mismo punto de ejemplo o uno en cada punto estableciendo o haciendo un borrado al SAM bit (CiCFG2<6>).

Típicamente el análisis del bit ocupa entre el 60 -70% del tiempo de bit, dependiendo de los parámetros del sistema.

## **4.6 Sincronización**

Para compensar los cambios de fase entre las frecuencias del oscilador de las diferentes estaciones del bus cada controlador CAN debe ser capaz de sincronizar al eje de la señal relevante de cada señal recibida. Cuando un eje en los datos transmitidos es detectado la lógica comparará la localización del eje en el tiempo esperado (segmento sincrónico). El circuito deberá entonces ajustar los valores de Phase1 Seg y Phase2 Seg entonces tenemos dos mecanismos usados para sincronizar.

### **4.6.1 Sincronización de seguridad**

La sincronización solo sucede cuando tenemos un eje “recesivo” a “dominante” durante el retardo del bus indicando el inicio de mensaje. Después de una sincronización dura los contadores de tiempo de bit son reseteados por el Sync Seg. La sincronización dura fuerza al eje que a causado la sincronización a mentir con el segmento de sincronización del tiempo de bit reseteado. Si la sincronización dura a terminado no tendremos una resincronización con ese tiempo de bit.

### **4.6.2 Resincronización**

Es el resultado de la resincronización, Phase1 Seg deberá ser agrandado o Phase2 Seg deberá ser acortado. El total del agrandado o acortado del buffer del segmento de fase tiene un límite superior conocido como el “synchronization jump width”, y está especificado en por el SJW<1:0> bits (CiCFG1<7:6>). El valor del “synchronization jump width” es añadido a Phase1 Seg o sustraída de Phase2 Seg. El “synchronization jump width” es programable entre 1 TQ y 4 TQ.

El siguiente requerimiento debe ser completado mientras se establece SJW<1:0> bits:

Phase2 Seg > synchronization jump width

TABLA 1: dsPIC30F4013 MAPA DE REGISTROS CAN 1

SFR Name	Addr.	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 State
C1RXFSID	0300	—	—	—	—	—	Receive Acceptance Filter 0 Standard Identifier <10:0>	—	—	—	—	—	—	—	—	—	EXIDE	0000 uuuu uuuu uuuu
C1RXF0EIDH	0302	—	—	—	—	—	Receive Acceptance Filter 0 Extended Identifier <17:6>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF0EIDL	0304	Receive Acceptance Filter 0 Extended Identifier <5:0>	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C1RXF1SID	0308	—	—	—	—	—	Receive Acceptance Filter 1 Standard Identifier <10:0>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF1EIDH	030A	—	—	—	—	—	Receive Acceptance Filter 1 Extended Identifier <17:6>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF1EIDL	030C	Receive Acceptance Filter 1 Extended Identifier <5:0>	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C1RXF2SID	0310	—	—	—	—	—	Receive Acceptance Filter 2 Standard Identifier <10:0>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF2EIDH	0312	—	—	—	—	—	Receive Acceptance Filter 2 Extended Identifier <17:6>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF2EIDL	0314	Receive Acceptance Filter 2 Extended Identifier <5:0>	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C1RXF3SID	0318	—	—	—	—	—	Receive Acceptance Filter 3 Standard Identifier <10:0>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF3EIDH	031A	—	—	—	—	—	Receive Acceptance Filter 3 Extended Identifier <17:6>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF3EIDL	031C	Receive Acceptance Filter 3 Extended Identifier <5:0>	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C1RXF4SID	0320	—	—	—	—	—	Receive Acceptance Filter 4 Standard Identifier <10:0>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF4EIDH	0322	—	—	—	—	—	Receive Acceptance Filter 4 Extended Identifier <17:6>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF4EIDL	0324	Receive Acceptance Filter 4 Extended Identifier <5:0>	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C1RXF5SID	0328	—	—	—	—	—	Receive Acceptance Filter 5 Standard Identifier <10:0>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF5EIDH	032A	—	—	—	—	—	Receive Acceptance Filter 5 Extended Identifier <17:6>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXF5EIDL	032C	Receive Acceptance Filter 5 Extended Identifier <5:0>	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C1RXM0SID	0330	—	—	—	—	—	Receive Acceptance Mask 0 Standard Identifier <10:0>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXM0EIDH	0332	—	—	—	—	—	Receive Acceptance Mask 0 Extended Identifier <17:6>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXM0EIDL	0334	Receive Acceptance Mask 0 Extended Identifier <5:0>	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C1RXM1SID	0338	—	—	—	—	—	Receive Acceptance Mask 1 Standard Identifier <10:0>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXM1EIDH	033A	—	—	—	—	—	Receive Acceptance Mask 1 Extended Identifier <17:6>	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C1RXM1EIDL	033C	Receive Acceptance Mask 1 Extended Identifier <5:0>	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C1TX2SID	0340	Transmit Buffer 2 Standard Identifier <10:6>	—	—	—	—	Transmit Buffer 2 Standard Identifier <5:0>	—	—	—	—	—	—	—	—	SRR	TXIDE	uuuu uu00 0000 0000
C1TX2EID	0342	Transmit Buffer 2 Extended Identifier <17:14>	—	—	—	—	Transmit Buffer 2 Extended Identifier <13:6>	—	—	—	—	—	—	—	—	—	—	uuuu 0000 uuuu uuuu
C1TX2DLC	0344	Transmit Buffer 2 Extended Identifier <5:0>	—	—	—	—	TXRTR	TXRB1	TXRBO	TXRBO	TXRBO	DLC<3:0>	—	—	—	—	—	uuuu uuuu uuuu uuuu
C1TX2B1	0346	Transmit Buffer 2 Byte 1	—	—	—	—	Transmit Buffer 2 Byte 0	—	—	—	—	—	—	—	—	—	—	uuuu uuuu uuuu uuuu
C1TX2B2	0348	Transmit Buffer 2 Byte 3	—	—	—	—	Transmit Buffer 2 Byte 2	—	—	—	—	—	—	—	—	—	—	uuuu uuuu uuuu uuuu
C1TX2B3	034A	Transmit Buffer 2 Byte 5	—	—	—	—	Transmit Buffer 2 Byte 4	—	—	—	—	—	—	—	—	—	—	uuuu uuuu uuuu uuuu
C1TX2B4	034C	Transmit Buffer 2 Byte 7	—	—	—	—	Transmit Buffer 2 Byte 6	—	—	—	—	—	—	—	—	—	—	uuuu uuuu uuuu uuuu
C1TX2CON	034E	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	—	TXPR<1:0>	—	—	—	—	0000 0000 0000 0000
C1TX1SID	0350	Transmit Buffer 1 Standard Identifier <10:6>	—	—	—	—	Transmit Buffer 1 Standard Identifier <5:0>	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 uuuu
C1TX1EID	0352	Transmit Buffer 1 Extended Identifier <17:14>	—	—	—	—	Transmit Buffer 1 Extended Identifier <13:6>	—	—	—	—	—	—	—	—	—	—	uuuu 0000 uuuu uuuu
C1TX1DLC	0354	Transmit Buffer 1 Extended Identifier <5:0>	—	—	—	—	TXRTR	TXRB1	TXRBO	TXRBO	DLC<3:0>	—	—	—	—	—	—	uuuu uuuu uuuu uuuu

Legend: u = uninitialized bit

TABLA 1: dsPIC30F4013 MAPA DE REGISTROS CAN 1  
(continuación)

SFR Name	Addr.	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 State
C-TX1B1	0356				Transmit Buffer 1 Byte 1									Transmit Buffer 1 Byte 0				uuuu uuuu uuuu uuuu
C-TX1B2	0358				Transmit Buffer 1 Byte 3									Transmit Buffer 1 Byte 2				uuuu uuuu uuuu uuuu
C-TX1B3	035A				Transmit Buffer 1 Byte 5									Transmit Buffer 1 Byte 4				uuuu uuuu uuuu uuuu
C-TX1B4	035C				Transmit Buffer 1 Byte 7									Transmit Buffer 1 Byte 6				uuuu uuuu uuuu uuuu
C-TX1CON	035E	—	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>		0000 0000 0000 0000
C-TX0SID	0360	Transmit Buffer 0 Standard Identifier <10:8>												Transmit Buffer 0 Standard Identifier <5:0>		SRR	TXIDE	0000 0000 0000 0000
C-TX0EID	0362	Transmit Buffer 0 Extended Identifier <17:14>												Transmit Buffer 0 Extended Identifier <13:6>				uuuu 0000 uuuu uuuu
C-TX0DLC	0364	Transmit Buffer 0 Extended Identifier <5:0>					TXRTR	TXRB1		TXRB0				DLC<3:0>				uuuu uuuu uuuu u000
C-TX0B1	0366				Transmit Buffer 0 Byte 1									Transmit Buffer 0 Byte 0				uuuu uuuu uuuu uuuu
C-TX0B2	0368				Transmit Buffer 0 Byte 3									Transmit Buffer 0 Byte 2				uuuu uuuu uuuu uuuu
C-TX0B3	036A				Transmit Buffer 0 Byte 5									Transmit Buffer 0 Byte 4				uuuu uuuu uuuu uuuu
C-TX0B4	036C				Transmit Buffer 0 Byte 7									Transmit Buffer 0 Byte 6				uuuu uuuu uuuu uuuu
C-TX0CON	036E	—	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>		0000 0000 0000 0000
C-RX1SID	0370	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRR	RXIDE	0000 uuuu uuuu uuuu
C-RX1EID	0372	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C-RX1DLC	0374	Receive Buffer 1 Extended Identifier <5:0>					RXRTR	RXRB1						Receive Buffer 1 Extended Identifier <17:6>				0000 uuuu uuuu uuuu
C-RX1B1	0376				Receive Buffer 1 Byte 1									Receive Buffer 1 Byte 0				uuuu uuuu uuuu uuuu
C-RX1B2	0378				Receive Buffer 1 Byte 3									Receive Buffer 1 Byte 2				uuuu uuuu uuuu uuuu
C-RX1B3	037A				Receive Buffer 1 Byte 5									Receive Buffer 1 Byte 4				uuuu uuuu uuuu uuuu
C-RX1B4	037C				Receive Buffer 1 Byte 7									Receive Buffer 1 Byte 6				uuuu uuuu uuuu uuuu
C-RX1CON	037E	—	—	—	—	—	—	—	—	RXFUL	—	—	—	RXRTRRO		FILHIT<2:0>		0000 0000 0000 0000
C-RX0SID	0380	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRR	RXIDE	0000 uuuu uuuu uuuu
C-RX0EID	0382	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 uuuu uuuu uuuu
C-RX0DLC	0384	Receive Buffer 0 Extended Identifier <5:0>					RXRTR	RXRB1						Receive Buffer 0 Extended Identifier <17:6>				uuuu uuuu uuuu uuuu
C-RX0B1	0386				Receive Buffer 0 Byte 1									Receive Buffer 0 Byte 0				uuuu uuuu uuuu uuuu
C-RX0B2	0388				Receive Buffer 0 Byte 3									Receive Buffer 0 Byte 2				uuuu uuuu uuuu uuuu
C-RX0B3	038A				Receive Buffer 0 Byte 5									Receive Buffer 0 Byte 4				uuuu uuuu uuuu uuuu
C-RX0B4	038C				Receive Buffer 0 Byte 7									Receive Buffer 0 Byte 6				uuuu uuuu uuuu uuuu
C-RX0CON	038E	—	—	—	—	—	—	—	—	RXFUL	—	—	—	RXRTRRO	DBEN	JTOFF	FILHIT0	0000 0000 0000 0000
C-CTRL	0390	CANCAP	—	CSIDLE	ABAT	CANCKS	REOOP<2:0>	—	—	OPMODE<2:0>	—	—	—	—	—	—	—	0000 0100 1000 0000
C-FCFG1	0392	—	—	—	—	—	—	—	—	SJM<1:0>	—	—	—	BRP<5:0>	—	—	—	0000 0000 0000 0000
C-FCFG2	0394	—	WAKFIL	—	—	—	SEG2PH<2:0>	SEG2PHTS	SAM	—	—	—	—	SEG1PH<2:0>	—	—	—	0u00 0uuu uuuu uuuu

Legend: u = uninitialized bit

TABLA 1: dsPIC30F4013 MAPA DE REGISTROS CAN 1  
(continuación II)

SFR Name	Addr.	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 State	
C1INTF	0396	RXDOVR	RX1OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN	VRIF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF	0000 0000 0000 0000	
C1INTE	0398	—	—	—	—	—	—	—	—	VRIE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE	0000 0000 0000 0000	
C1EC	038A	Transmit Error Count Register						Receive Error Count Register						0000 0000 0000 0000					

Legend: u = uninitialized bit

**Note:** Refer to *dsPIC30F Family Reference Manual* (DS70046) for descriptions of register bit fields.

