



Руководство (v2.13)

По программированию модулей “mPCIe-TTCAN” “PCIe-TTCAN”

Интерфейс ISO 11898-4
(CAN Bus с дополнением TTCAN)

04.07.2023

ООО “НОВОМАР”

Оглавление

1	Область применения.....	4
2	Расшифровка наименования модуля.....	6
3	Термины определения и сокращения.....	7
4	Структурная схема модуля и таблица регистров.....	8
5	Управление режимами работы и функциями модуля.....	11
 5.1	Основные сервисные регистры модуля и регистры DMA.....	11
5.1.1	Регистр: DMA_DATA_BASE.....	11
5.1.2	Регистр: DMA _n _INDEX.....	12
5.1.3	Регистр: INTERRUPT	14
5.1.4	Регистр: INTERRUPT MASK.....	15
5.1.5	Регистр: CAN _x _TIMEOUT_ABSOLUTE	16
5.1.6	Регистр: CAN _x _TIMEOUT_INTERVAL	17
5.1.7	Регистр: DMA _n _RD_INDEX	17
 5.2	Таймеры локального времени.....	18
5.2.1	Регистр: CAN1_TIMER, CAN2_TIMER.....	19
5.2.2	Регистр: CAN1_TIMER_TRSH, CAN2_TIMER_TRSH	19
5.2.3	Регистр: CAN1_TIMER_CEED, CAN2_TIMER_CEED.....	20
5.2.4	Регистр: CAN1_TIMER_CTRL, CAN2_TIMER_CTRL	20
5.2.5	Регистр: CAN1_TIMER_EPOCH, CAN2_TIMER_EPOCH	21
5.2.6	Регистр: CAN1_EPOCH_CEED, CAN2_EPOCH_CEED	22
 5.3	Триггеры буферов отправки сообщений.....	23
5.3.1	Регистр: CAN1_TX1_TRIG, CAN1_TX2_TRIG, CAN1_TX3_TRIG	23
5.3.2	Регистр: CAN2_TX1_TRIG, CAN2_TX2_TRIG, CAN2_TX3_TRIG	23
5.3.3	Регистр: CAN1_TRIG_CTRL	24
5.3.4	Регистр: CAN2_TRIG_CTRL	24
5.3.5	Регистр: CAN1_INT_TRIG, CAN2_INT_TRIG.....	25
5.3.6	Регистр: CAN1_TXn_TRIG_EPOCH	25
5.3.7	Регистр: CAN2_TXn_TRIG_EPOCH	26
5.3.8	Регистр: CAN1_INT_TRIG_EPOCH, CAN2_INT_TRIG_EPOCH	26
 5.4	Временные метки прерываний.....	27
5.4.1	Регистр: CAN1_INT_TIMEMARK.....	28
5.4.2	Регистр: CAN2_INT_TIMEMARK.....	28
 5.5	FIFO передачи сообщений.....	28
5.5.1	Регистр: CAN _x _FIFO_CONSTAT.....	29
5.5.2	Регистр: CAN _x _FIFO_HP_CONSTAT.....	30
5.5.3	Регистр: CAN _x _HP_FIFO	30
5.5.4	Регистр: CAN _x _TG_FIFO	31
6	Работа с контроллерами шины CAN.....	32
6.1.1	Регистр: CAN1_CTRL.....	33
6.1.2	Регистр: CAN2_CTRL.....	34
6.1.3	Регистр: CAN1_ACS	35
6.1.4	Регистр: CAN1_BUF	36
6.1.5	Регистр: CAN2_ACS	36
6.1.6	Регистр: CAN2_BUF	36
 6.2	Регистры контроллера CAN.....	37
 6.3	Режимы работы контроллера CAN.....	38
6.3.1	Регистр: CAN_CTRL.....	38
6.3.2	Регистр: CAN_STAT	39
 6.4	Прерывания контроллера CAN.....	40
6.4.1	Регистр: CANINTE	40
6.4.2	Регистр: CANINTF	41

Руководство по программированию	mPCIe-TTCAN
6.4.3 Регистр: BFPCTRL	42
6.4.4 Регистр: TXRTSCTRL.....	43
6.5 Конфигурация скорости шины CAN.....	44
6.5.1 Регистр: CNF1.....	45
6.5.2 Регистр: CNF2.....	46
6.5.3 Регистр: CNF3.....	46
6.6 Ошибки CAN-шины.....	47
6.6.1 Регистр: TEC.....	47
6.6.2 Регистр: REC.....	48
6.6.3 Регистр: EFLG	48
6.7 Передача сообщений.....	49
6.7.1 Регистр: TXBnCTRL	51
6.7.2 Регистр: TXBnSIDH	51
6.7.3 Регистр: TXBnSIDL.....	52
6.7.4 Регистр: TXBnEID8.....	52
6.7.5 Регистр: TXBnEID0.....	52
6.7.6 Регистр: TXBnDLC	53
6.7.7 Регистр: TXBnDm	53
6.8 Приём сообщений.....	54
6.8.1 Регистр: RXB0CTRL.....	55
6.8.2 Регистр: RXB1CTRL.....	56
6.8.3 Регистр: RXBnSIDH.....	57
6.8.4 Регистр: RXBnSIDL	57
6.8.5 Регистр: RXBnEID8.....	57
6.8.6 Регистр: RXBnEID0.....	58
6.8.7 Регистр: RXBnDLC	58
6.8.8 Регистр: RXBnDm	58
6.9 Фильтрация принимаемых сообщений.....	59
6.9.1 Регистр: RXFnSIDH	60
6.9.2 Регистр: RXFnSIDL.....	60
6.9.3 Регистр: RXFnEID8	61
6.9.4 Регистр: RXFnEID0	61
6.9.5 Регистр: RXMnSIDH	61
6.9.6 Регистр: RXMnSIDL	62
6.9.7 Регистр: RXMnEID8.....	62
6.9.8 Регистр: RXMnEID0.....	62
6.10 Автоматическое зеркалирование регистров CAN.....	63
6.10.1 Регистр: CANx_MIRR_ERR	63
6.10.2 Регистр: CANx_MIRR_TXB	63
6.11 Статистика работы модуля.....	64
6.11.1 Регистр: CANx_STAT	64
Список исправлений и изменений.....	65

1 Область применения.

Интерфейсные модули "mPCIe-TTCAN", "PCIe-TTCAN" содержат два независимых CAN контроллера, поддерживающих спецификации CAN2.0A и CAN2.0B.

Модуль имеет в своём составе таймеры локального времени, для каждой шины свой, и ряд функций для реализации протоколов, детерминированных по времени, на основе стандарта ISO 11898-4: TTCAN, J1939, CANopen, DeviceNet и других;

CAN-шины гальванически изолированы от устройства и между собой.
Гальваническая развязка 2,5кВ.

Модули "mPCIe-TTCAN", "PCIe-TTCAN" полностью соответствуют всем требованиям, заявленным в стандарте CAN2.0B.

Модуль "mPCIe-TTCAN", "PCIe-TTCAN" обеспечивают приём и передачу кадров: стандартных (standard frame), расширенных (extended frame), кадров запроса данных (remote frame) .

Модули "**mPCIe-TTCAN**" выполнены в габаритах PCI Express Mini Card. Модули рассчитаны на применение в тяжелых условиях эксплуатации и расширенного температурного диапазона от минус 60°C до +85°C.

Модули "**PCIe-TTCAN**" выполнены в формате платы расширения для ПК, устанавливаемой в слот PCI Express v.1.1 x1 (и выше). Модули рассчитаны на применение в тяжелых условиях эксплуатации и расширенного температурного диапазона от минус 40°C до +70°C.

Идентификатор нашине PCI-Express

Поле	Значение		
	Firmware v.01	Firmware v.02	Firmware v.03
Vendor ID	0xA203	0xA203	0xA203
Device ID	0x9471	0x9471	0x9471
Revision ID	0x02	0x12	0x22
SubVendor ID	0xA203	0xA203	0xA203
SubDevice ID	0x9471	0x9471	0x9471
ClassCode	0x078000	0x078000	0x028000

В Firmware v.02 добавлены следующие функции:

- Двухканальный режим работы DMA;
- Таймеры прерываний: абсолютный и интервальный;
- Отображение регистров контроллеров CAN в адресное пространство PCIe;
- Развёрнутая трансляция прерываний контроллеров CAN в главный регистр прерывания;
- Буфер временных меток прерываний контроллера CAN.
- Ускоренная отправка сообщений;
- Аппаратный контроль переполнения буферов DMA.
- 8-битные счётчики циклов, удовлетворяющие TTCAN level 2 для построения полной временной матрицы.
- Таблица событий передачи «Timemark».

В Firmware v.03 добавлены следующие функции:

- Режим работы – «FIFO», с двумя буферами FIFO: обычным и высокоприоритетным, по 63 сообщения каждый;
- Таблица событий передачи «Timemark» расширена с 16 до 64 записей.
- Счётчики статистики работы модуля.

Настоящее руководство действительно для плат с Firmware v.03. Для обновления Firmware модуля обратитесь к производителю.

Модули с Firmware v.03 полностью обратно совместимы с драйверами, написанными для модулей с Firmware v.01 и Firmware v.02 за исключением адресов таблицы «Timemark».

Модули с Firmware v.02 полностью обратно совместимы с драйверами, написанными для модулей с Firmware v.01.

2 Расшифровка наименования модуля.

mPCIe-TTCAN-60		
1	2	3

1. Форм фактор модуля и интерфейс подключения к ПК:

- mPCIe - Mini PCI Express Card
- PCIe - PCI Express Card

2. Тип линии и протокол обмена:

- CAN – ISO11898 (CAN Bus).
- TTCAN – ISO11898 (CAN Bus) с расширением ISO11898-4 (TTCAN).

3. Температурный диапазон:

- пробел – -40..+85°C.
- 50 – -50..+85°C.
- 60 – -60..+85°C.

3 Термины определения и сокращения.

Список сокращений:

- ПК - персональный компьютер
- ПЛИС - программируемая логическая интегральная схема
- ИС - интегральная схема
- DMA - прямой доступ к памяти
- ПО - программное обеспечение
- Запись в виде: **CAN1_ACS.CA_LEN_CAN1** - означает ссылку на бит **CA_LEN_CAN1** в регистре **CAN1_ACS**

4 Структурная схема модуля и таблица регистров.

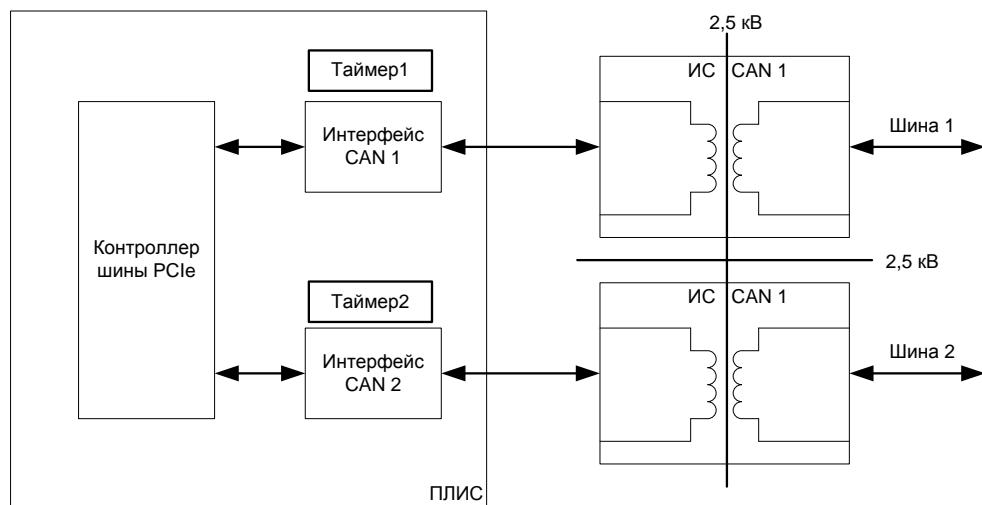


Рисунок 1. Структура модуля "mPCIe-TTCAN".

Модуль состоит из следующих основных компонентов:

- ПЛИС, в которой реализованы основные функции модуля, а также контроллер шины PCIe.
- Две ИС CAN шины.
- Вспомогательные компоненты, обеспечивающие работоспособность и сервисные функции модуля (на схеме не указаны).

ПЛИС содержит следующие структурные элементы:

- Контроллер шины PCIe с функцией DMA, , обеспечивающий управление и обмен данными с ПК, обработку прерываний, доступ к регистрам сервисных функций устройства.
- Интерфейсы шины CAN, реализующие функции контроллеров шины CAN, а также передачу в память ПК принимаемых данных через функцию DMA.
- Таймеры локального времени для привязки к сетевому времени и циклам обмена нашине.
- Набор триггеров для автоматического запуска передачи сообщений в заданные интервалы времени.

Таблица 1 описывает регистры модуля их название, адрес и возможные операции с регистром (запись, чтение).

Таблица 1

Адрес	Название	Read/ Write	Описание
1000h	DMA DATA BASE	R/W	Базовый адрес буфера данных в памяти ПК.
1008h	DMA1 INDEX	R/W	Указатель записи в буфере данных (CAN1 – SingleDMA, общий - DoubleDMA).
100Ch	INTERRUPT	R	Статус прерываний.
1010h	INTERRUPT MASK	R/W	Маска прерываний.
1028h	CAN1 TIMEOUT ABSOLUTE	RW	Регистр абсолютного таймера DMA CAN1.
102Ch	CAN2 TIMEOUT ABSOLUTE	RW	Регистр абсолютного таймера DMA CAN2.
1030h	DMA2 INDEX	R/W	Указатель записи CAN2 в буфере данных в режиме DoubleDMA.
1038h	CAN1 TIMEOUT INTERVAL	RW	Регистр интервального таймера DMA CAN1
103Ch	CAN2 TIMEOUT INTERVAL	RW	Регистр интервального таймера DMA CAN2
1040h	CAN1 CTRL	RW	Регистр управления и режима работы контроллера CAN1.
1044h	CAN1 ACS	W	Регистр управления/статуса доступа к контроллеру шины CAN1.
1048h	CAN1 MIRR ERR	R	Регистр ошибки, зеркальный из контроллера CAN1.
104Ch	CAN1 MIRR TXB	R	Регистр статуса буферов передачи, зеркальный из контроллера CAN1.
1050h	CAN2 CTRL	RW	Регистр управления и режима работы контроллера CAN2.
1054h	CAN2 ACS	W	Регистр управления/статуса доступа к контроллеру шины CAN2.
1058h	CAN2 MIRR ERR	R	Регистр ошибки, зеркальный из контроллера CAN2
105Ch	CAN2 MIRR TXB	R	Регистр статуса буферов передачи, зеркальный из контроллера CAN2
1060h	CAN1 TIMER	R	Текущее значение таймера шины CAN1.
1064h	CAN1 TIMER TRSH	R/W	Период счёта таймера шины CAN1.
1068h	CAN1 TIMER CEED	R/W	Начальное значение таймера шины CAN1 при сбросе.
106Ch	CAN1 TIMER CTRL	R/W	Регистр управления таймером шины CAN1.
1070h	CAN1 TX1 TRIG	R/W	Регистр порога срабатывания триггера отправки сообщения из буфера 1 CAN1.
1074h	CAN1 TX2 TRIG	R/W	Регистр порога срабатывания триггера отправки сообщения из буфера 2 CAN1.
1078h	CAN1 TX3 TRIG	R/W	Регистр порога срабатывания триггера отправки сообщения из буфера 3 CAN1.
107Ch	CAN1 TRIG CTRL	R/W	Регистр управления триггерами CAN1.
1080h	CAN2 TIMER	R	Текущее значение таймера шины CAN2.
1084h	CAN2 TIMER TRSH	R/W	Период счёта таймера шины CAN2.
1088h	CAN2 TIMER CEED	R/W	Начальное значение таймера шины CAN2 при сбросе.
108Ch	CAN2 TIMER CTRL	R/W	Регистр управления таймером шины CAN2.
1090h	CAN2 TX1 TRIG	R/W	Регистр порога срабатывания триггера отправки сообщения из буфера 1 CAN2.
1094h	CAN2 TX2 TRIG	R/W	Регистр порога срабатывания триггера отправки сообщения из буфера 2 CAN2.
1098h	CAN2 TX3 TRIG	R/W	Регистр порога срабатывания триггера отправки сообщения из буфера 3 CAN2.
109Ch	CAN2 TRIG CTRL	R/W	Регистр управления триггерами CAN2.
10A0h	CAN1 INT TRIG	R/W	Порог прерывания по таймеру CAN1.
10A4h	CAN2 INT TRIG	R/W	Порог прерывания по таймеру CAN2.
10A8h	DMA1 RD INDEX	R/W	Указатель чтения в буфере данных(CAN1 –

			SingleDMA, общий - DoubleDMA).
10ACh	<u>DMA2 RD INDEX</u>	R/W	Указатель чтения в буфере данных CAN2 в режиме DoubleDMA.
10B0h	<u>CAN1 TIMER EPOCH</u>	R/W	Текущее значение счётчика циклов CAN1.
10B4h	<u>CAN1 EPOCH CED</u>	R/W	Начальное значение счётчика циклов для триггера прерывания CAN1.
10B8h	<u>CAN2 TIMER EPOCH</u>	R/W	Текущее значение счётчика циклов CAN2.
10BCh	<u>CAN2 EPOCH CED</u>	R/W	Начальное значение счётчика циклов для триггера прерывания CAN2.
10C0h	<u>CAN1 TX1 TRIG EPOCH</u>	R/W	Значение счётчика циклов для триггера отправки сообщения из буфера 1 CAN1.
10C4h	<u>CAN1 TX2 TRIG EPOCH</u>	R/W	Значение счётчика циклов для триггера отправки сообщения из буфера 2 CAN1.
10C8h	<u>CAN1 TX3 TRIG EPOCH</u>	R/W	Значение счётчика циклов для триггера отправки сообщения из буфера 3 CAN1.
10CCh	<u>CAN1 INT TRIG EPOCH</u>	R/W	Значение счётчика циклов для триггера прерывания CAN1.
10D0h	<u>CAN2 TX1 TRIG EPOCH</u>	R/W	Значение счётчика циклов для триггера отправки сообщения из буфера 1 CAN2.
10D4h	<u>CAN2 TX2 TRIG EPOCH</u>	R/W	Значение счётчика циклов для триггера отправки сообщения из буфера 2 CAN2.
10D8h	<u>CAN2 TX3 TRIG EPOCH</u>	R/W	Значение счётчика циклов для триггера отправки сообщения из буфера 3 CAN2.
10DCh	<u>CAN2 INT TRIG EPOCH</u>	R/W	Значение счётчика циклов для триггера прерывания CAN2.
10F0h	<u>CAN1 FIFO CONSTAT</u>	R/W	Регистр управления и статуса FIFO CAN1.
10F4h	<u>CAN2 FIFO CONSTAT</u>	R/W	Регистр управления и статуса FIFO CAN2.
10F8h	<u>CAN1 FIFO HP CONSTAT</u>	R/W	Регистр управления и статуса высокоприоритетного FIFO CAN1.
10FCh	<u>CAN2 FIFO HP CONSTAT</u>	R/W	Регистр управления и статуса высокоприоритетного FIFO CAN2.
2000h-200Ch	<u>CAN1 BUF</u>	R/W	Буфер данных контроллера доступа к контроллеру шины CAN1
2010h-201Ch	<u>CAN2 BUF</u>	R/W	Буфер данных контроллера доступа к контроллеру шины CAN2
2020h-202Ch	<u>CAN1 HP FIFO</u>	W	Буфер высокоприоритетного FIFO CAN1
2030h-203Ch	<u>CAN2 HP FIFO</u>	W	Буфер высокоприоритетного FIFO CAN2
2040h-2044h	<u>CAN1 TG FIFO</u>	W	Буфер FIFO триггеров CAN1
2050h-2055h	<u>CAN2 TG FIFO</u>	W	Буфер высокоприоритетного FIFO CAN2
2100h-21Ch	<u>CAN1 STAT</u>	R/W	Блок регистров статистики CAN1
2200h-221Ch	<u>CAN2 STAT</u>	R/W	Блок регистров статистики CAN2
2400h-25FCh	<u>CAN1 INT TIMEMARK</u>	R	Буфер временных меток прерываний контроллера CAN1.
2600h-27FCh	<u>CAN2 INT TIMEMARK</u>	R	Буфер временных меток прерываний контроллера CAN2.

Адреса регистров указаны в виде смещения от базового адреса BAR0 устройства нашине PCI.

Значение адреса указано в байтах. Все регистры 32 битные за исключением DMA_DATA_BASE, соответственно инкремент адреса должен быть равен 4 (т.е. младшие биты адреса всегда = "00"). **Байтовые операции запрещены.**

5 Управление режимами работы и функциями модуля.

5.1 Основные сервисные регистры модуля и регистры DMA.

5.1.1 Регистр: DMA_DATA_BASE

Адрес: 1000h

Номер бита																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	R	R			
W	W	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	W	W	W		
Номер бита																																	
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32		
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
W	W	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	W	W	W	W	W

Описание: базовый адрес буфера данных в памяти ПК (64бит)

Номер бита	Обозначение	Описание
63-20	DATA_BAR	Адрес буфера данных.
19-3	RSRV	Не используются, должны быть "0".
2	HW_OVF_DET	Аппаратное детектирование переполнения буфера DMA: "0" – выключено "1" – включено
1	DMA_MODE	Режим работы DMA: "0" – один канал DMA для обоих контроллеров CAN "1" – два независимых канала DMA, свой для каждого контроллера CAN
0	DMA_EN	Разрешение работы DMA: "0" - выключено "1" - включено

Данные записываются в память ПК в кольцевой буфер размером 1Мб от младших адресов к старшим.

В режиме двух каналов DMA буфер разбивается на две половины: младшие 512Кб – CAN1, старшие 512Кб – CAN2.

Данные записываются блоками с выравниванием по 64 байт, из которых: первые 4 байта - служебные, содержат номер канала CAN шины - источника данных и временную метку, следующие 14 байт - данные пакета. Подробно структура принятых данных в памяти ПК описана в главе 6.

Для переключения режима работы DMA необходимо: выключить DMA, выдержать паузу 1 мкс, сбросить счётчики DMA_INDEX, включить DMA в новом режиме.

Работу DMA для каждого канала CAN отдельно можно включить или выключить в регистре CAN_CTRL.DMA_OFF.

По достижении половины буфера и 1/8 (512 Кб и 128Кб, соответственно) генерируются прерывания, отображаемые в регистре INTERRUPT. Эти прерывания работают только для одноканального режима DMA.

5.1.2 Регистр: DMA_n INDEX

Адрес: 1008h, 1030h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	R W	-	-	-	-	-	-													

Описание: указатель записи в буфере данных.

Номер бита	Обозначение	Описание
31-20	RSRV	Не используются, должны быть "0".
19-6	DATA_INDEX	Указатель записи в буфере данных.
5-0	RSRV	Не используются, "0".
При записи любого значения в регистр указатель сбрасывается в "0"		

Работа буфера данных производится по схеме FIFO.

Программа-драйвер должна иметь счетчик чтения данных. При старте системы оба счетчика равны нулю. При получении прерывания должен считываться указатель записи и сравниваться с указателем чтения. Если они различаются, то обрабатываются данные от указателя чтения (включительно) до указателя записи (исключая его) с учетом перехода через 0.



При записи любого значения в регистр указателя записи он сбрасывается в '0'.

Драйвер устройства должен отслеживать событие переполнения буфера данных в памяти ПК. Если в DMA DATA BASE.DMA HW OVF DET включен аппаратный контроль переполнения, после чтения данных из буфера драйвер должен обновлять указатель чтения DMA_n RD INDEX. В этом случае при достижении значения счётчиков (DMA_n_INDEX + 1) = DMA_n_RD_INDEX запись в память будет остановлена.

В режиме одного канала DMA используется только один счётчик DMA1_INDEX (1008h). Данные обоих каналов CAN записываются в один буфер.

В режиме двух каналов DMA принятые контроллерами CAN данные записываются в свой собственный буфер. Буфер в этом режиме разделяется на две половины: младшие 512Кб – CAN1, старшие 512Кб – CAN2. Бит 19 счётчиков DMA_n_INDEX не используется.

При необходимости, если ПК не успевает забирать данные, записанные платой, возможна временная остановка передачи данных в память ПК – сброс бита 0 регистра DMA DATA BASE. Для последующего запуска бит нужно снова установить в '1'.

Данные будут остановлены для обоих каналов. Убедитесь, что при возобновлении работы сохраняется тот же режим работы.

При раздельной работе с каждым каналом можно установить бит **CANx CTRL. DMA OFF** в ‘1’, а затем сбросить в ‘0’.

Если в момент приостановки DMA кадр уже находился в процессе записи в память ПК, он будет записан до конца. После чего указатель **DMA_n INDEX** увеличится на единицу. Если соответствующие прерывания не запрещены, они будут установлены.

Поэтому рекомендуется следующий порядок остановки DMA:

1. Сбросить регистры таймтаутов **CAN_n TIMEROUT ABSOLUTE** и **CAN_n TIMEROUT INTERVAL** чтобы устраниТЬ возможность отложенных прерываний.
2. Запретить работу DMA.
3. Выдержать паузу 1 мс.
4. Прочитать все полученные данные.

Возобновлять работу DMA следует в следующем порядке:

1. Установить регистры таймтаутов **CAN_n TIMEROUT ABSOLUTE** и **CAN_n TIMEROUT INTERVAL**.
2. Возобновить работу DMA.

Во время остановки DMA, намеренной или же при переполнении буфера и включенном аппаратном контроле может произойти потеря сообщений. В этом случае первое успешно записанное после возобновления работы сообщение будет содержать включенный флаг **RXB_nCTRL.OVF**.

5.1.3 Регистр: INTERRUPT

Адрес: 100Ch

Номер бита																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-	-	-	-	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	R	R

Описание: регистр прерывания.

Номер бита	Обозначение	Описание
31-28	RSRV	Не используются
27	INT_MERR_CAN2	Прерывание по ошибке приёма или передачи на шине CAN2. Если включено автоматическое зеркалирование, содержимое регистров EFLG, REC, TEC а также полей TXBnCTRL будет отображено в регистрах CAN2_MIRR_ERR и CAN2_MIRR_TXB .
26	INT_WAKE_CAN2	Прерывание выхода из состояния сна контроллера CAN2.
25	INT_ERR_CAN2	Прерывание порогов счетчиков ошибок контроллера CAN2. Если включено автоматическое зеркалирование, содержимое регистров EFLG, REC, TEC а также полей TXBnCTRL будет отображено в регистрах CAN2_MIRR_ERR и CAN2_MIRR_TXB .
24	INT_MSG_XMTD22	Прерывание по отправке сообщения из буфера 2 по шине CAN2.
23	INT_MSG_XMTD21	Прерывание по отправке сообщения из буфера 1 по шине CAN2.
22	INT_MSG_XMTD20	Прерывание по отправке сообщения из буфера 0 по шине CAN2.
21	INT_MERR_CAN1	Прерывание по ошибке приёма или передачи на шине CAN1. Если включено автоматическое зеркалирование, содержимое регистров EFLG, REC, TEC а также полей TXBnCTRL будет отображено в регистрах CAN1_MIRR_ERR и CAN1_MIRR_TXB .
20	INT_WAKE_CAN1	Прерывание выхода из состояния сна контроллера CAN1.
19	INT_ERR_CAN1	Прерывание порогов счетчиков ошибок контроллера CAN1. Если включено автоматическое зеркалирование, содержимое регистров EFLG, REC, TEC а также полей TXBnCTRL будет отображено в регистрах CAN1_MIRR_ERR и CAN1_MIRR_TXB .
18	INT_MSG_XMTD12	Прерывание по отправке сообщения из буфера 2 по шине CAN1.
17	INT_MSG_XMTD11	Прерывание по отправке сообщения из буфера 1 по шине CAN1.
16	INT_MSG_XMTD10	Прерывание по отправке сообщения из буфера 0 по шине CAN1.
15	INT_TIM_ITV2	Прерывание интервального таймера CAN2. Порог срабатывания указывается в регистре CAN2_TIMEOUT_INTERVAL .
14	INT_TIM_ITV1	Прерывание интервального таймера CAN1. Порог срабатывания указывается в регистре CAN1_TIMEOUT_INTERVAL .
13	INT_TIM_ABS2	Прерывание абсолютного таймера CAN2. Порог срабатывания указывается в регистре CAN2_TIMEOUT_ABSOLUTE .
12	INT_TIM_ABS1	Прерывание абсолютного таймера CAN1. Порог срабатывания указывается в регистре CAN1_TIMEOUT_ABSOLUTE .
11	INT_MSG_RCVD2	Прерывание по приёму сообщения по шине CAN2.
10	INT_MSG_RCVD1	Прерывание по приёму сообщения по шине CAN1.
9	INT_CAN_ACS2	Прерывание доступа к регистрам контроллера CAN2: CAN2_ACS.CA_BSY_CAN2 сбросился в '0', модуль готов.
8	INT_CAN_ACS1	Прерывание доступа к регистрам контроллера CAN1: CAN1_ACS.CA_BSY_CAN1 сбросился в '0', модуль готов.
7	INT_TIM_CAN2	Прерывание таймера CAN2. Порог срабатывания указывается в регистре CAN2_INT_TRIG .
6	INT_TIM_CAN1	Прерывание таймера CAN1. Порог срабатывания указывается в регистре CAN1_INT_TRIG .
5	INT_CAN2	Прерывание контроллера CAN2.
4	INT_CAN1	Прерывание контроллера CAN1.

3	INT_FLASH	Прерывание контроллера флэш-памяти
2	RSRV	Не используется, '0'.
1	INT_QDAT	Прерывание по заполнению 1/8 буфера данных.
0	INT_HDAT	Прерывание по заполнению половины буфера данных
При чтении регистра все установленные биты регистра сбрасываются в "0".		

5.1.4 Регистр: INTERRUPT MASK

Адрес: 1010h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Описание: регистр маски прерываний

Номер бита	Обозначение	Описание
31-28	RSRV	Не используются, "0".
27	INT_MERR_CAN1	Прерывание по ошибке приёма или передачи на шине CAN1.
26	INT_WAKE_CAN1	Прерывание выхода из состояния сна контроллера CAN1.
25	INT_ERR_CAN2	Разрешение прерывания по ошибке на шине CAN2.
24	INT_MSG_XMTD22	Разрешение прерывания по отправке сообщения из буфера 2 по шине CAN2.
23	INT_MSG_XMTD21	Разрешение прерывания по отправке сообщения из буфера 1 по шине CAN2.
22	INT_MSG_XMTD20	Разрешение прерывания по отправке сообщения из буфера 0 по шине CAN2.
21	INT_MERR_CAN1	Прерывание по ошибке приёма или передачи на шине CAN1.
20	INT_WAKE_CAN1	Прерывание выхода из состояния сна контроллера CAN1.
19	INT_ERR_CAN1	Разрешение прерывания по ошибке на шине CAN1.
18	INT_MSG_XMTD12	Разрешение прерывания по отправке сообщения из буфера 2 по шине CAN1.
17	INT_MSG_XMTD11	Разрешение прерывания по отправке сообщения из буфера 1 по шине CAN1.
16	INT_MSG_XMTD10	Разрешение прерывания по отправке сообщения из буфера 0 по шине CAN1.
15	INT_TIM_ITV2	Разрешение прерывания интервального таймера CAN2.
14	INT_TIM_ITV1	Разрешение прерывания интервального таймера CAN1.
13	INT_TIM_ABS2	Разрешение прерывания абсолютного таймера CAN2.
12	INT_TIM_ABS1	Разрешение прерывания абсолютного таймера CAN1.
11	INT_MSG_RCVD2	Разрешение прерывания по приёму сообщения по шине CAN2.
10	INT_MSG_RCVD1	Разрешение прерывания по приёму сообщения по шине CAN1.
9	INT_CAN_ACS2	Разрешение прерывания доступа к регистрам CAN2.
8	INT_CAN_ACS1	Разрешение прерывания доступа к регистрам CAN1.
7	INT_TIM_CAN2	Разрешение прерывания таймера контроллера CAN2.
6	INT_TIM_CAN1	Разрешение прерывания таймера контроллера CAN1.
5	INT_CAN2	Разрешение прерывания контроллера CAN2.
4	INT_CAN1	Разрешение прерывания контроллера CAN1.
3	INT_FLASH	Разрешение прерывания контроллера флэш-памяти
2	RSRV	не используется, '0'.
1	INT_QDAT	Разрешение прерывания по заполнению 1/8 буфера данных.
0	INT_HDAT	Разрешение прерывания по заполнению половины буфера данных
0 – прерывание запрещено, 1 – разрешено		

Генерация прерывания по каждому из событий 0-15 может быть запрещена сбросом (установкой в 0) или разрешена (установкой в 1) соответствующего бита в регистре маски прерываний **INTERRUPT MASK**. Однако, независимо от значения маски, события продолжают отображаться в регистре прерывания.

Запрет прерываний от каждого CAN в регистре CAN_CTRL

По умолчанию, после системного сброса все прерывания запрещены.

Прерывания по ошибке и передаче сообщений возникают при задействовании механизма автоматического зеркалирования регистров CAN в адресное пространство PCIE.

Зеркалирование регистров контроллера CAN позволяет максимально быстро читать регистры контроллеров CAN, минимизировать реакцию процессора на событии на CAN шинах и упростить взаимодействие процессора и CAN контроллеров.

Прерывания по приёму каждого сообщения позволяют максимально быстро реагировать на входящие сообщения. Прерывания по таймерам позволяют организовать работу приложений, не требующих немедленной реакции на каждое сообщение.

Прерывания INT_QDAT и INT_HDAT работают только в режиме одноканального DMA.

5.1.5 Регистр: CANx TIMEOUT ABSOLUTE

Адрес: 1028h, 102Ch

Номер бита																																
-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
										W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Описание: Регистр абсолютного таймера DMA CANx.

Номер бита	Обозначение	Описание
31-22	RSRV	Не используются, должны быть "0".
21-0	TIMER	Значение таймера в микросекундах.

Абсолютный таймер запускается по получении первого сообщения из соответствующего канала с момента последнего прерывания. После получения каждого последующего кадра счетчик не сбрасывается.

По достижении заданного значения в регистре **INTERRUPT** будет установлен бит 12 или 13 (CAN1/CAN2) и сгенерировано прерывание (если данное прерывание разрешено). Таймер останавливается без рестарта при достижении установленного значения. Таймер может использоваться для чтения всех принятых сообщений за определённый временной период.

5.1.6 Регистр: CANx TIMEOUT INTERVAL

Адрес: 1038h, 103Ch

Номер бита																																		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
-	-	-	-	-	-	-	-	-	-	R W																								

Описание: Регистр интервального таймера DMA CANx.

Номер бита	Обозначение	Описание
31-22	RSRV	Не используются, должны быть "0".
21-0	TIMER	Значение таймера в микросекундах.

Интервальный таймер запускается по получении каждого сообщения из соответствующего канала. То есть, после записи каждого последующего кадра счетчик будет сброшен и продолжит отсчет с нуля.

По достижении заданного значения в регистре **INTERRUPT** будет установлен бит 14 или 15 (CAN1/CAN2) и сгенерировано прерывание (если данное прерывание разрешено). Таймер останавливается без рестарта при достижении установленного значения. Таймер может быть использован для отслеживания редких сообщений в канале.

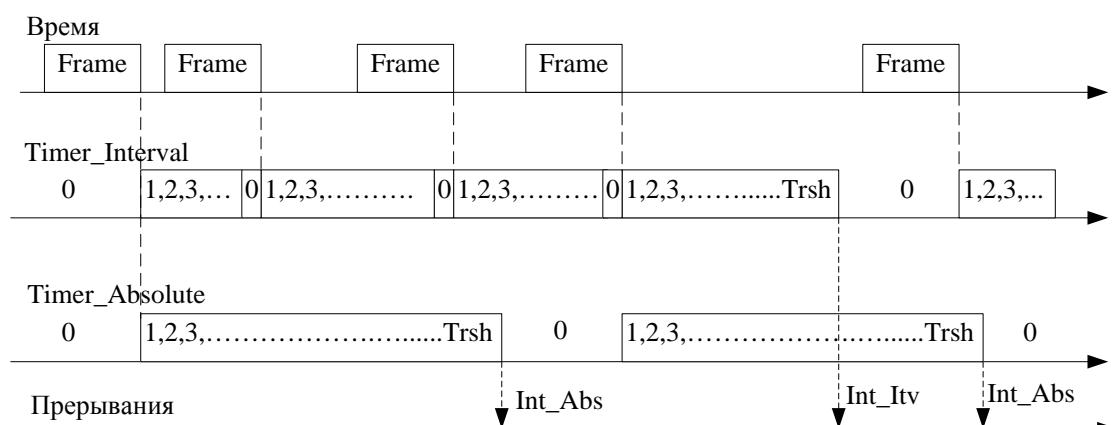


Рисунок 2. Работа интервального и абсолютного таймеров.

Единица счета абсолютного и интервального таймеров – 1 мкс.

5.1.7 Регистр: DMA_n RD INDEX

Адрес: 10A8h, 10ACh

Номер бита																																		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
-	-	-	-	-	-	-	-	-	-	-	-	R W																						

Описание: указатель чтения в буфере данных.

Номер бита	Обозначение	Описание
31-20	RSRV	Не используются, должны быть "0".
19-6	DATA_INDEX	Указатель чтения в буфере данных.
5-0	RSRV	Не используются, "0".

5.2 Таймеры локального времени.

Протокол TTCAN добавляет к CAN Bus временное разделение доступа к шине. Обмен узлов сети производится по определённому расписанию, известному как матрица циклов. Каждый узел передаёт сообщения только в определённый тайм-слот или временные окна.

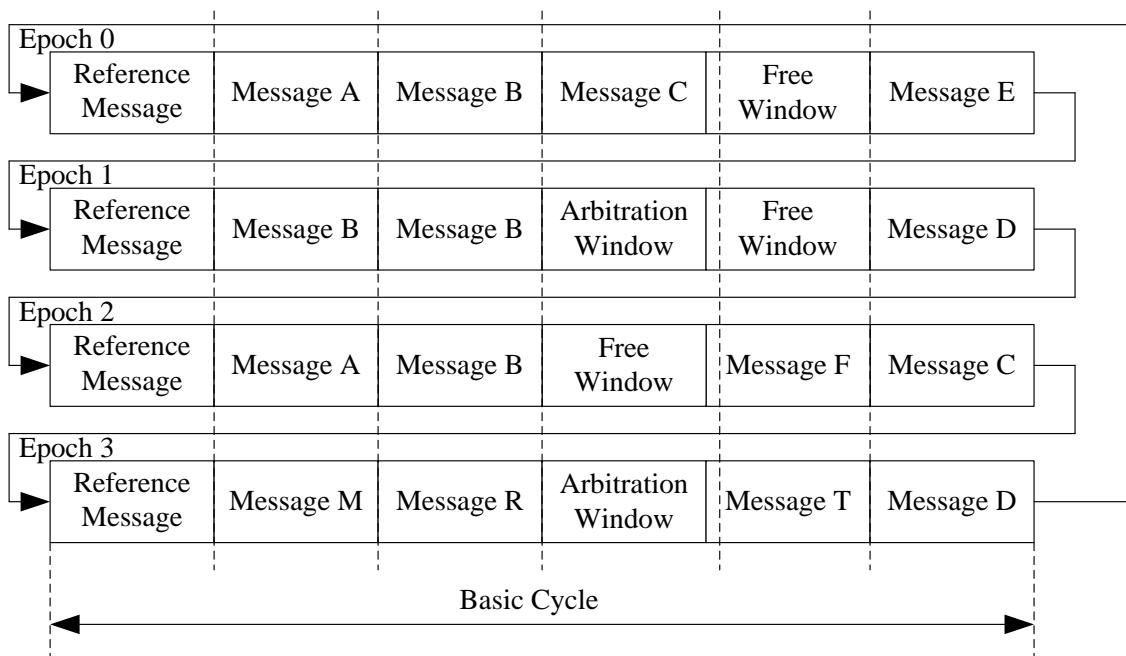


Рисунок 3. Матрица циклов – расписание временных окон.

Общая точка отсчёта определяется передачей специального референсного сообщения "reference message". Референсные сообщения передаёт активный Мастер Времени. Всего в сети может быть до восьми узлов - потенциальных Мастеров Времени для выполнения требований отказоустойчивости. Порядок замены активного Мастера Времени одним из резервных определяется используемым протоколом.

Референсное сообщение передаётся с определённым периодом. Этот период между двумя референсными сообщениями называется основной цикл "basic cycle".

Для каждой сети TTCAN составляется расписание - матрица циклов "matrix cycle", в котором столбцы - временные окна, а строки - циклы.

Время в сетях на основе TTCAN измеряется в единицах сетевого времени NTU (Network Time Unit). NTU - дробная часть секунды, определяется архитектором сети. Обычно совпадает с периодом битовой скорости. В каждом узле сети NTU образуется умножением периода собственной системной частоты на TUR (Time Unit Ratio). Для модуля mPCIe-CAN период системной частоты равен 50нс.

Пример конфигурации таймера:

Примем NTU равным 1мкс (Скорость передачи 1Мбит/с).

*Частота внутреннего опорного осциллятора - всегда 20МГц, период - 50нс.
TUR = 1мкс / 50нс = 20.*

При записи в регистр CANn_TIMER_TRSH в поле DIV_ от TUR
нужно отнять единицу и сдвинуть на два бита влево.*

20-1 = 19 = 0x0013 После сдвига DIV_TRSH (15...0) = 0x004C

Для 125Кбит/с NTU будет равен 8мкс.

DIV_TRSH нужно умножить на 8 = 0x0260

Теперь рассчитаем период таймера.

Пусть окно передачи будет 192мкс – на 30% больше времени передачи сообщения максимальной длины(~150 бит с учётом bit-stuffing). Основной цикл – 6 сообщений.

$192 * 6 = 1152$ мкс. В поле NTU_TRSH нужно записать $(1152-1) = 1151 = 0x047F$.

При изменении скорости передачи период таймера NTU_TRSH останется тем же самым.

5.2.1 Регистр: CAN1_TIMER, CAN2_TIMER

Адрес: 1060h, 1080h

Номер бита																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	-	-	-

Описание: текущее значение таймера локального времени

Номер бита	Обозначение	Описание
31-16	CUR_NTU	Счётчик в единицах сетевого времени.
15-2	CUR_DIV	Делитель единицы сетевого времени (TUR). Единица счёта - 50нс.
1-0	RSRV	Резерв, не используется.

Регистр **CANx_TIMER** отображает текущее значение локального таймера в единицах сетевого времени.

После системного сброса таймер выключен. Для нормальной работы его необходимо включить: **CANn_TIMER_CTRL.ENABLE = "1"**.

В режиме свободного счёта таймер считает 0 - 0xFFFFFFFF в 50нс отсчётах. Режим свободного счёта может быть полезен для начального определения периода основного цикла, а также работы протоколов CAN без временного разделения доступа к шине.

Чтобы таймер считал в режиме основного цикла, нужно установить период счёта в **CANn_TIMER_TRSH** и установить **NTU_MODE = "1"**.

5.2.2 Регистр: CAN1_TIMER_TRSH, CAN2_TIMER_TRSH

Адрес: 1064h, 1084h

Номер бита																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	-	-

Описание: период счёта таймера

Номер бита	Обозначение	Описание
31-16	NTU_TRSH	Счётчик в единицах сетевого времени.
15-2	DIV_TRSH	Делитель единицы сетевого времени. 0x0000 - запрещённое значение!
1-0	RSRV	Резерв, не используется.

Если установлен режим **CANn_TIMER.NTU_MODE = "1"** делитель считает от нуля до **CANn_TIMER_TRSH.DIV_TRSH**. По достижении делителя значения DIV_TRSH таймер сетевого времени увеличивается на единицу. Делитель сбрасывается и начинает счёт с нуля.

По достижении сетевого времени = NTU_TRSH таймер сетевого времени сбрасывается и начинает счёт с нуля.

5.2.3 Регистр: CAN1_TIMER_CEED, CAN2_TIMER_CEED

Адрес: 1068h, 1088h

Номер бита																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	-
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	

Описание: начальное значение таймера при сбросе

Номер бита	Обозначение	Описание
31-16	NTU_CEED	Начальное значение таймера в единицах сетевого времени.
15-2	DIV_CEED	Начальное значение делителя единицы сетевого времени, с которого будет вестись отсчёт. DIV_CEED не должен быть больше DIV_TRSH!
1-0	RSRV	Резерв, не используется.

CANn_TIMER_CEED и **CANn_EPOCH_CEED** предназначены для корректировки начального значения таймера. Корректировка может потребоваться:

- в начале работы для синхронизации с циклом шины чтобы выставить нулевую точку для удобства расчётов.

- при смещении начала цикла в ходе работы. Это может происходить вследствие разности частот узлов сети либо событий, нарушающих нормальное течение цикла: аварий, плановых пауз в работе.

Таймер будет сброшен в 0, а установлен значением:

CANn_TIMER.CUR_NTU будет загружен значением NTU_CEED,

CANn_TIMER.CUR_DIV будет загружен значением DIV_CEED,

CANn_TIMER_EPOCH.CUR_EPOCH будет загружен значением EPOCH_CEED (если включен бит **CANn_TIMER_CTRL.EPOCH_CEED_EN**).

CANn_TIMER_CTRL.CEED_EN оказывает действие в следующих случаях:

- по достижении **CANn_TIMER_TRSH**,
- при сбросе **CANn_TIMER_CTRL.RST**,
- при ресинхронизации **CANn_TIMER_CTRL.RST_ON_RXBn**.

5.2.4 Регистр: CAN1_TIMER_CTRL, CAN2_TIMER_CTRL

Адрес: 106Ch, 108Ch

Номер бита																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Описание: текущее значение таймера

Номер бита	Обозначение	Описание
31-16	RSRV	Резерв, не используется.
15-8	EPOCH_MASK	Маска счётчика циклов. Заполняется начиная с правого бита: 0x01 > 0x03 > 0x07 > 0x0F > 0x1F > 0x3F > 0x7F > 0xFF. 0x00 – счётчик циклов не используется.
7	RSRV	Резерв, не используется.
6	EPOCH_CEED_EN	Установка начала счёта счётчика циклов из регистра CANn_EPOCH_CEED .

5	RST_ON_RXB0	Сброс таймера в "0" по приёму сообщения в буфер RXB0. Может использоваться для автоматической корректировки начала цикла. При этом необходимо настроить буфер RXB0 на приём паттерна референсного сообщения. Период счёта таймера должен быть установлен равным основному циклу.
4	RST_ON_RXB1	Сброс таймера в "0" по приёму сообщения в буфер RXB1. Может использоваться для автоматической корректировки начала цикла. При этом необходимо настроить буфер RXB1 на приём паттерна референсного сообщения. Период счёта таймера должен быть установлен равным основному циклу.
3	NTU_MODE	Режим работы таймера. "0" - свободный счёт: 0-0xFFFFFFFF, "1" - счёт с периодом CANn_TIMER_TRSH .
2	CEED_EN	Установка начала счёта таймера из регистра CANn_TIMER_CEED .
1	ENABLE	Разрешение работы таймера. После сброса работа таймера запрещена. Необходимо записать "1" для начала работы.
0	RST	Немедленный сброс. Требуется однократная установка бита в "1". После сброса таймера данный бит примет значение "0".

В режиме RST_ON_RXBn по приёму референсного сообщения таймер локального времени будет установлен значением таймера синхронизации. Для корректной работы режима синхронизации по референсному сообщению необходима установка регистра **CANn_TIMER_TRSH**.

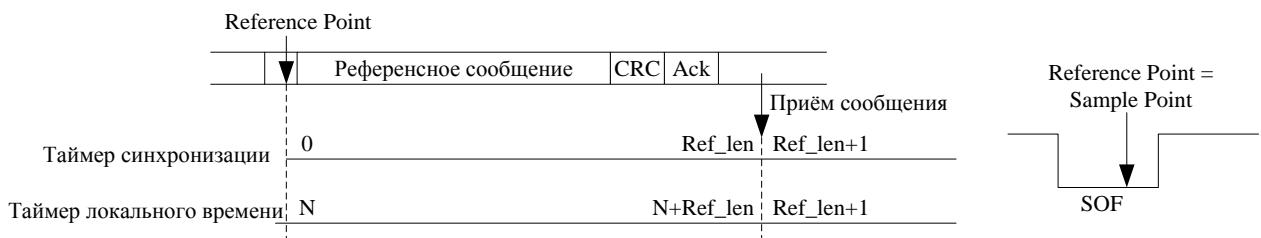


Рисунок 4. Синхронизация таймера по приёму референсного сообщения.

Таким образом, точка отсчёта - "0" таймера локального времени будет установлена в точку сэмплирования старта референсного сообщения, как требуется по стандарту ISO 11898-4.

Если в дополнении к битам RST_ON_RXB0 или RST_ON_RXB1 будет установлен ещё и бит CEED_EN, тогда значение таймера локального времени в момент синхронизации будет взято из регистра **CANn_TIMER_CEED**.

5.2.5 Регистр: [CAN1_TIMER_EPOCH, CAN2_TIMER_EPOCH](#)

Адрес: 10B0h, 10B8h

Номер бита		
31	30	29
-	-	-
28	27	26
-	-	-
25	24	23
-	-	-
22	21	20
-	-	-
19	18	17
-	-	-
16	15	14
-	-	-
13	12	11
-	-	-
10	9	8
-	-	-
7	6	5
R	R	R
W	W	W
4	3	2
R	R	R
W	W	W
1	0	

Описание: значение счетчика циклов

Номер бита	Обозначение	Описание
31-8	RSRV	Резерв, не используется.
7-0	CUR_EPOCH	Текущее значение счетчика циклов.

Регистр доступен на запись и чтение.

5.2.6 Регистр: CAN1_EPOCH_CEED, CAN2_EPOCH_CEED

Адрес: 10B4h, 10BCh

Номер бита																																						
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Описание: триггер отправки сообщения для соответствующего буфера

Номер бита	Обозначение	Описание
31-8	RSRV	Резерв, не используется.
7-0	EPOCH_CEED	Начальное значение счётчика циклов для триггера прерывания.

5.3 Триггеры буферов отправки сообщений.

Триггер отправки сообщения может быть установлен для каждого буфера передачи контроллера CAN. Триггер запускает отправку сообщения в буфере по достижении таймера локального времени определённого значения.

Триггер может работать однократно, либо в цикле. Циклическая отправка сообщений возможна, например, для выдачи референсных сообщений, если узел работает в режиме Мастера времени. ПО должно обеспечить своевременное обновление данных в буфере передачи, для которого включен циклический режим.

Отдельный триггер **CANn_INT_TRIG** может быть установлен для выдачи прерывания системе при заданном значении таймера локального времени.

В режиме FIFO пользоваться триггерами отправки нельзя. В этом случае триггеры передачи устанавливаются через **FIFO_TG**. За исключением регистра триггера прерывания **CANn_INT_TRIG**.

5.3.1 Регистр: **CAN1_TX1_TRIG, CAN1_TX2_TRIG, CAN1_TX3_TRIG**

Адрес: 1070h, 1074h, 1078h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	-
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	-	-

Описание: триггер отправки сообщения для соответствующего буфера

Номер бита	Обозначение	Описание
31-16	NTU_TRIG	Значение таймера в единицах сетевого времени.
15-2	DIV_TRIG	Значение делителя единицы сетевого времени.
1-0	RSRV	Резерв, не используется.

5.3.2 Регистр: **CAN2_TX1_TRIG, CAN2_TX2_TRIG, CAN2_TX3_TRIG**

Адрес: 1090h, 1094h, 1098h

Номер бита																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	-
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	-	-	

Описание: триггер отправки сообщения для соответствующего буфера

Номер бита	Обозначение	Описание
31-16	NTU_TRIG	Значение таймера в единицах сетевого времени.
15-2	DIV_TRIG	Значение делителя единицы сетевого времени.
1-0	RSRV	Резерв, не используется.

5.3.3 Регистр: CAN1 TRIG CTRL

Адрес: 107Ch

Номер бита																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R W																

Описание: Регистр управления триггерами CAN1.

Номер бита	Обозначение	Описание
31-16	RSRV	Резерв, не используется.
15	TX_TRIG3_EPOCH_EN	Учёт счётчика циклов для триггера буфера 3.
14	TX_TRIG2_EPOCH_EN	Учёт счётчика циклов для триггера буфера 2.
13	TX_TRIG1_EPOCH_EN	Учёт счётчика циклов для триггера буфера 1.
12	INT_TRIG_EPOCH_EN	Учёт счётчика циклов для триггера прерывания.
11	TX_TRIG3_RPT	Режим повтора триггера для буфера 3.
10	TX_TRIG2_RPT	Режим повтора триггера для буфера 2.
9	TX_TRIG1_RPT	Режим повтора триггера для буфера 1.
8	INT_TRIG_RPT	Режим повтора триггера прерывания.
7-6	TX_TRIG3_EN	Разрешение работы триггера для буфера 3.
5-4	TX_TRIG2_EN	Разрешение работы триггера для буфера 2.
3-2	TX_TRIG1_EN	Разрешение работы триггера для буфера 1.
1-0	INT_TRIG_EN	Разрешение работы триггера прерывания.

5.3.4 Регистр: CAN2 TRIG CTRL

Адрес: 109Ch

Номер бита																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R W																

Описание: Регистр управления триггерами CAN2.

Номер бита	Обозначение	Описание
31-16	RSRV	Резерв, не используется.
15	TX_TRIG3_EPOCH_EN	Учёт счётчика циклов для триггера буфера 3.
14	TX_TRIG2_EPOCH_EN	Учёт счётчика циклов для триггера буфера 2.
13	TX_TRIG1_EPOCH_EN	Учёт счётчика циклов для триггера буфера 1.
12	INT_TRIG_EPOCH_EN	Учёт счётчика циклов для триггера прерывания.
11	TX_TRIG3_RPT	Режим повтора триггера для буфера 3.
10	TX_TRIG2_RPT	Режим повтора триггера для буфера 2.
9	TX_TRIG1_RPT	Режим повтора триггера для буфера 1.
8	INT_TRIG_RPT	Режим повтора триггера прерывания.

7-6	TX_TRIG3_EN	Разрешение работы триггера для буфера 3.	<p>"01" - запуск триггера; "10" - сброс триггера; "00", "11" - не изменять состояние данного триггера.</p> <p>Если триггер запущен в режиме повтора или запущен в однократном режиме и ещё не сработал, результатом чтения из регистра будет "01". Если триггер уже сработал или выключен - "00".</p>
5-4	TX_TRIG2_EN	Разрешение работы триггера для буфера 2.	
3-2	TX_TRIG1_EN	Разрешение работы триггера для буфера 1.	
1-0	INT_TRIG_EN	Разрешение работы триггера прерывания.	

5.3.5 Регистр: CAN1_INT_TRIG, CAN2_INT_TRIG

Адрес: 10A0h, 10A4h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	-	-
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	-	-

Описание: значение триггера прерывания

Номер бита	Обозначение	Описание
31-16	NTU_TRIG	Значение таймера в единицах сетевого времени.
15-2	DIV_TRIG	Значение делителя единицы сетевого времени.
1-0	RSRV	Резерв, не используется.

5.3.6 Регистр: CAN1_TXn_TRIG_EPOCH

Адрес: 10C0h, 10C4h, 10C8h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	

Описание: триггер отправки сообщения для соответствующего буфера

Номер бита	Обозначение	Описание
31-8	RSRV	Резерв, не используется.
7-0	EPOCH_TRIG	Значение счётчика циклов для триггера прерывания.

5.3.7 Регистр: CAN2 TXn TRIG EPOCH

Адрес: 10D0h, 10D4h, 10D8h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	

Описание: триггер отправки сообщения для соответствующего буфера

Номер бита	Обозначение	Описание
31-8	RSRV	Резерв, не используется.
7-0	EPOCH_TRIG	Значение счётчика циклов для триггера прерывания.

5.3.8 Регистр: CAN1 INT TRIG EPOCH, CAN2 INT TRIG EPOCH

Адрес: 10CCh, 10DCh

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	

Описание: триггер отправки сообщения для соответствующего буфера

Номер бита	Обозначение	Описание
31-8	RSRV	Резерв, не используется.
7-0	EPOCH_TRIG	Значение счётчика циклов для триггера прерывания.

5.4 Временные метки прерываний.

История последних 64 прерываний контроллеров CAN сохраняется в виде записей (меток) в специальных буферах. Метки прерываний позволяют:

- получить точное время (значение таймера локального времени) на момент успешной отправки сообщений;
- получить точное время возникновения ошибок на шине;
- узнать о пропущенных событиях.

История прерываний сохраняется только при включенном автоматическом зеркалировании регистров контроллера.

Каждая запись в буфере содержит:

- идентификатор – кольцевой счётчик 0...255 для определения порядка событий;
- значение регистра прерывания контроллера CAN;
- значение регистра флагов ошибок шины;
- метку времени прерывания.

Буфер организован по принципу сдвигового регистра. 64 записей (64-бит каждая) сдвигаются от младших адресов к старшим по мере поступления прерываний.

Запись 0 – самая свежая. Запись F – самая старая.

Адрес (смещение в буфере)	Байт 3	Байт 2	Байт 1	Байт 0	Запись №
0x00	TM_NTU		TM_DIV	FLAG	0
0x04	TM_EPOCH	TM_EFLG	TM_CANINTF	TM_ID	
0x00	TM_NTU		TM_DIV	FLAG	1
0x04	TM_EPOCH	TM_EFLG	TM_CANINTF	TM_ID	
...					
0x78	TM_NTU		TM_DIV	FLAG	F
0x7C	TM_EPOCH	TM_EFLG	TM_CANINTF	TM_ID	

Значения TM_DIV, TM_NTU, TM_EPOCH указаны на момент возникновения прерывания. TM_DIV выровнен по старшему биту: 15...2.

После автоматического чтения регистра прерывания из контроллера CAN новое событие с номером TM_ID+1 сохраняется в запись 0. Все предыдущие записи в буфере сдвигаются на единицу (0>1, 1>2, 2>3,...E>F). Последняя запись (F) пропадает.

Идентификатор TM_ID сбрасывается в 0 при выключении автоматического зеркалирования регистров контроллера.

Для режима «FIFO» TM_ID копируется из MSG_ID соответствующего FIFO.

Для режима FIFO младшие биты 1...0 первого слова записи FLAG (чётные адреса) содержат флаги:

0 – флаг Msg_Abored_Timeout: сообщение прервано по таймауту;

1 – Резерв, всегда 0.

Важно! Чтение регистров должно производиться либо 64-битными операциями, либо 32-битными от младшего адреса к старшему в пределах одной записи. Для защиты от сдвига записей между 32-битными чтениями старший адрес буферизируется при чтении младшего.

5.4.1 Регистр: [CAN1_INT_TIMEMARK](#)

Адрес: 2400h – 25FCh

Описание: Буфер временных меток прерываний контроллера CAN1. 512 байт.
Только для чтения.5.4.2 Регистр: [CAN2_INT_TIMEMARK](#)

Адрес: 2600h – 27FCh

Описание: Буфер временных меток прерываний контроллера CAN2. 512 байт.
Только для чтения.**5.5 FIFO передачи сообщений.**

FIFO передачи позволяет существенно упростить процедуру отправки сообщений если не требуется жёсткий контроль ПО за ходом передачи и реакция на ошибки шины.

На каждое сообщение отводится 16 байт.

Сообщение имеет следующий вид (см. главу «6.7 Передача сообщений»):

Слово\байт	0	1	2	3
0h	TXBnSIDH	TXBnSIDL	TXBnEID8	TXBnEID0
4h	TXBnDLC	TXBnD0	TXBnD1	TXBnD2
8h	TXBnD3	TXBnD4	TXBnD5	TXBnD6
Ch	TXBnD7	MSG_ID	-	-

Запись сообщения в FIFO происходит в момент записи старшего 32-битного слова по смещению 0Ch.

Каждый канал имеет два FIFO: обычное и высокоприоритетное. Запись в обычное FIFO происходит через буфер CANx_BUФ, запись в высокоприоритетное FIFO происходит через CANx_HP_FIFO.

Обычное FIFO может использоваться в паре с FIFO триггеров [CANx_TG_FIFO](#). Запись в поля FIFO триггеров происходит перед записью в обычное FIFO. Если запись в CANx_TG_FIFO была, сообщения отправляются в указанное время. Иначе, сообщения отправляются в порядке очереди.

Если протокол не использует тайм-слоты с интервалами определённой длительности, а нужно просто начать передачу непрерывного блока сообщений в определённое время, запись в FIFO триггеров можно сделать только для первого сообщения. Последующие сообщения будут отправлены сразу за первым.

Высокоприоритетное FIFO позволяет передавать важные асинхронные сообщения в обход основной очереди. Передача из обычного FIFO будет приостановлена до опустошения высокоприоритетного FIFO. Если сообщение из обычного FIFO уже находится в буфере отправки, передача не будет прерываться.

Для обоих типов FIFO может быть установлен флаг приостановки передачи TX_PAUSE. Флаг позволяет предварительно записать данные в FIFO и начать передачу в нужное время одной быстрой записью в регистр управления FIFO. Для обычного FIFO флаг TX_PAUSE имеет приоритет над значением триггера отправки из CANx_TG_FIFO.

Для начала работы в режиме FIFO необходимо:

1. Проинициализировать контроллеры CAN и включить их в рабочий режим.
2. Включить режим «FIFO» в регистре CANX_CTRL.CAN_MODE = «01».

Внимание! Непосредственный доступ к регистрам контроллера в режиме «FIFO» закрыт. Флаг «занят» CAN_ACS.CA_BSY_CANx будет всегда установлен.

3. Установить таймаут передачи в регистре CANx_FIFO_CONSTAT.TIMEOUT, если это необходимо. Для корректной работы таймаут не должен иметь значение меньше полной длительности передачи сообщения. Для однократной передачи сообщений пользуйтесь флагом [CANCTRL.OSM](#).

4. Сформируйте сообщение. Указывайте каждому сообщению уникальный 8-битный ID, если порядок и статус передачи будет отслеживаться. Последний обработанный ID будет отображаться в регистре CANx_FIFO_CONSTAT.RECENT_ID. Подробную историю передачи сообщений и статусы можно наблюдать в таблице временных меток [TIMEMARK](#).

5. Перед записью сообщений проверьте заполненность FIFO в регистре CANx_FIFO_CONSTAT.COUNT: «0» - FIFO пустое, «0x7F» - полное. В каждое FIFO можно записать «63- CANx_FIFO_CONSTAT.COUNT» сообщений.

6. Запишите каждое сообщение в FIFO последовательно, 32-битными словами, от младшего к старшему. Для обычного FIFO запись необходимо начинать с записи в регистр CANx_TG_FIFO, если для данного сообщения устанавливается триггер отправки. Иначе запись в CANx_TG_FIFO делать не нужно. Далее нужно записать сообщение в FIFO целиком. Переход к следующей ячейке происходит при записи в старшее слово (смещение 0Ch).

7. Если была попытка записи в полностью заполненное FIFO, будет выставлен флаг CANx_FIFO_CONSTAT.OVF.

8. Очередь полностью пуста когда: CANx_FIFO_CONSTAT.COUNT равен нулю и CANx_FIFO_CONSTAT.RECENT_ID равен ID последнего записанного в FIFO сообщения.

9. Для останова всех передач необходимо сбросить оба FIFO, дождаться появления ID последнего записанного сообщения CANx_FIFO_CONSTAT.RECENT_ID.

10 Для выключения режима FIFO нужно переключить режим CANX_CTRL.CAN_MODE = «00» и дождаться сброса флага CAN_ACS.CA_BSY_CANx. Дождаться ID последнего записанного сообщения в CANx_FIFO_CONSTAT.RECENT_ID.

5.5.1 Регистр: [CANx_FIFO_CONSTAT](#)

Адрес: 10F0h, 10F4h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-	W	W	W	W	W	W	W	R	W

Описание: Регистр управления и статуса FIFO CAN1/CAN2.

При записи в регистр (32-битными словами):

Номер бита	Обозначение	Описание
31-16	TIMEOUT	Таймаут передачи сообщения в микросекундах. Счётчик таймаута запускается после запуска передачи сообщения. Если таймаут не используется, должно быть "0". В этом случае передача будет происходить бесконечно.
15-7	RSRV	Не используются, должны быть "0".
6	TX_PAUSE	Приостановка отправки сообщений из FIFO. После сброса "0". '1' – пауза, '0' – возобновление передачи.
5-1	RSRV	Не используются, должны быть "0".
0	RST	Сброс FIFO.

При чтении из регистра:

Номер бита	Обозначение	Описание
31-16	TIMEOUT	Таймаут передачи сообщения в микросекундах.
15-8	RECENT_ID	MSG_ID последнего завершённого сообщения. Успешно отправленного или прерванного по таймауту.
7	OVF	Флаг переполнения FIFO. Устанавливается в «1» если была попытка записи при полностью заполненном FIFO. Сбрасывается в «0» при следующей удачной записи.
6	TX_PAUSE	Приостановка отправки сообщений из FIFO.
5-0	COUNT	Количество сообщений в FIFO. Чтобы получить количество свободных блоков, значение этого поля нужно отнять от 63.

5.5.2 Регистр: CANx_FIFO_HP_CONSTAT

Адрес: 10F8h, 10FCh

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-	W	W	W	W	W	W	R	W	

Описание: Регистр управления и статуса высокоприоритетного FIFO CAN1/CAN2.

При записи в регистр (32-битными словами):

Номер бита	Обозначение	Описание
31-16	TIMEOUT	Таймаут передачи сообщения в микросекундах. Если таймаут не используется, должно быть "0".
15-7	RSRV	Не используются, должны быть "0".
6	TX_PAUSE	Приостановка отправки сообщений из FIFO. После сброса "0". '1' – пауза, '0' – возобновление передачи.
5-0	RSRV	Не используются, должны быть "0".
0	RST	Сброс FIFO.

При чтении из регистра:

Номер бита	Обозначение	Описание
31-16	TIMEOUT	Таймаут передачи сообщения в микросекундах.
15-8	RECENT_ID	MSG_ID последнего завершённого сообщения. Успешно отправленного или прерванного по таймауту.
7	OVF	Флаг переполнения FIFO. Устанавливается в «1» если была попытка записи при полностью заполненном FIFO. Сбрасывается в «0» при следующей удачной записи.
6	TX_PAUSE	Приостановка отправки сообщений из FIFO.
5-0	COUNT	Количество сообщений в FIFO. Чтобы получить количество свободных блоков, значение этого поля нужно отнять от 63.

5.5.3 Регистр: CANx_HP_FIFO

Адрес: 2020 – 202Ch, 2030 – 203Ch

Описание: Высокоприоритетное FIFO. Запись сообщения в FIFO происходит в момент записи старшего 32-битного слова по смещению 0Ch.

5.5.4 Регистр: CANx TG FIFO

Адрес: 2040 – 2044h, 2050 – 2054h

Описание: FIFO триггеров.

При записи в регистр (32-битными словами):

Номер бита	Обозначение	Описание
Слово 0h		
31-16	NTU_TRIG	Значение таймера в единицах сетевого времени.
15-2	DIV_TRIG	Значение делителя единицы сетевого времени.
1-0	RSRV	Резерв, не используется, должен быть “0”.
Слово 4h		
31-8	RSRV	Резерв, не используется, должен быть “0”.
7-0	EPOCH_TRIG	Значение счётчика циклов.

Для включения триггера, запись должна происходить хотя бы в слово 0h.

Если запись в слово 4h была, ЕПОСН будет учитываться для этого триггера. Иначе, нет.

6 Работа с контроллерами шины CAN.

Устройство имеет два независимых контроллера CAN.

Регистры обоих контроллеров CAN находятся в отдельных адресных пространствах, доступных посредством следующего механизма: в буфер 16 байт CANx _BUF (x - номер CAN контроллера) записывается значение регистра(ов) CAN, в регистр CANx ACS записывается количество записываемых/читываемых в/из контроллера байт, адрес (начальный - в случае обращения к нескольким регистрам) и команда - тип операции.

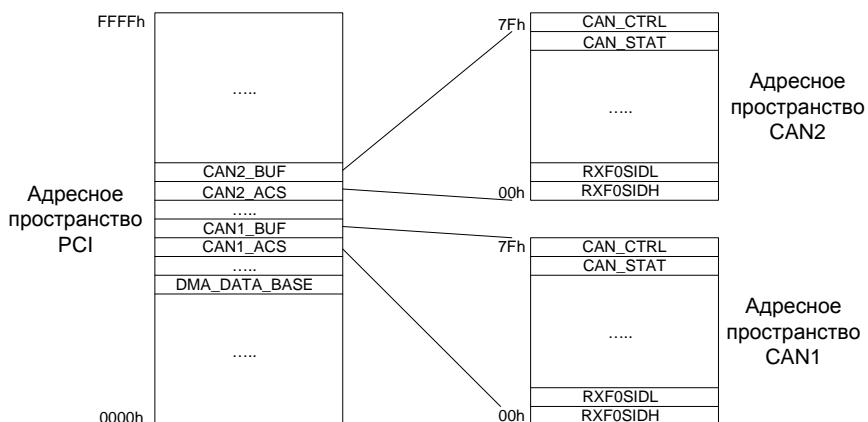


Рисунок 5. Адресное пространство контроллеров CAN.

Порядок обращения к регистрам CAN:

1. Драйвер должен убедиться, что предыдущая команда выполнена. Бит 0 регистра **CANx ACS** должен быть равен "0", что сигнализирует о завершении предыдущей операции.
2. В случае записи, либо побитной модификации регистров, записать новые значения регистров в буфер **CANx BUF**.
3. Записать в регистр **CANx ACS**: тип операции, (начальный) адрес регистра в контроллере, длину данных в байтах.
4. Убедиться, что операция выполнена, проверив бит 0 регистра **CANx ACS** или дождаться прерывания **INTERRUPT.INT CAN ACSx**, если оно разрешено.
5. В случае чтения, прочитать значения запрошенных регистров из буфера **CANx BUF**.

6.1.1 Регистр: [CAN1_CTRL](#)

Адрес: 1040h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R W	-	-	-	R W	-	R W	R W	R W								

Описание: Регистр управления и режима работы контроллера CAN1.

Номер бита	Обозначение	Описание
31	RST_CAN1	При установке бита 31 в "1" содержимое регистра не изменяется, осуществляется только сброс контроллера CAN1
30-15	RSRV	Резерв
15	INT_MERR_CAN1	Разрешение прерывания по ошибке приёма или передачи нашине CAN1.
14	INT_WAKE_CAN1	Разрешение прерывания по выходу из состояния сна контроллера CAN1.
13	INT_ERR_CAN1	Разрешение прерывания порогов счётчиков ошибок контроллера CAN1
12	INT_MSG_XMTD12	Разрешение прерывания по отправке сообщения избуфера 2 пошине CAN1.
11	INT_MSG_XMTD11	Разрешение прерывания по отправке сообщения избуфера 1 пошине CAN1.
10	INT_MSG_XMTD10	Разрешение прерывания по отправке сообщения избуфера 0 пошине CAN1.
9	INT_CAN_ACS1	Разрешение прерывания модуля доступа крегистрам CAN1.
8	INT_CAN1	Разрешение прерывания контроллера CAN1.
7	RSRV	Резерв
6-5	CAN_MODE	Режим работы контроллера CAN1 «00» - Native (по умолчанию) «01» - FIFO «10» - Timeplan (Резерв, не устанавливать) «11» - (Резерв, не устанавливать)
4	DMA_OFF	Выключение DMA для CAN1.
3	RSRV	Резерв
2	MIRR_TXB	Включение зеркалирования регистров TXBnCTRL контроллера CAN1 вадресное пространство PCIE.
1	MIRR_ERR	Включение зеркалирования регистров EFLG , REC и TEC контроллера CAN1 вадресное пространство PCIE.
0	MIRR_ON	Включение зеркалирования регистров контроллера CAN1 вадресное пространство PCIE.

Биты 8-15 разрешают прерывания CAN контроллера.

В отличие от [INTERRUPT_MASK](#), запрещённые вCANx_CTRL прерывания не будут видны врегистре [INTERRUPT](#).

6.1.2 Регистр: [CAN2_CTRL](#)

Адрес: 1050h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
w	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R W	-	-	-	R W	-	R W	R W	R W							

Описание: Регистр управления и режима работы контроллера CAN2.

Номер бита	Обозначение	Описание
31	RST_CAN2	При установке бита 31 в "1" содержимое регистра не изменяется, осуществляется только сброс контроллера CAN2
30-15	RSRV	Резерв
15	INT_MERR_CAN2	Разрешение прерывания по ошибке приёма или передачи нашине CAN2.
14	INT_WAKE_CAN2	Разрешение прерывания по выходу из состояния сна контроллера CAN2.
13	INT_ERR_CAN2	Разрешение прерывания порогов счётчиков ошибок контроллера CAN2.
12	INT_MSG_XMTD22	Разрешение прерывания по отправке сообщения избуфера 2 пошине CAN2.
11	INT_MSG_XMTD21	Разрешение прерывания по отправке сообщения избуфера 1 пошине CAN2.
10	INT_MSG_XMTD20	Разрешение прерывания по отправке сообщения избуфера 0 пошине CAN2.
9	INT_CAN_ACS2	Разрешение прерывания модуля доступа крегистрам CAN2.
8	INT_CAN2	Разрешение прерывания контроллера CAN2.
7	RSRV	Резерв
6-5	CAN_MODE	Режим работы контроллера CAN2 «00» - Native (по умолчанию) «01» - FIFO «10» - Timeplan (Резерв, не устанавливать) «11» - Резерв, не устанавливать
4	DMA_OFF	Выключение DMA для CAN2.
3	RSRV	Резерв
2	MIRR_TXB	Включение зеркалирования регистров TXBnCTRL контроллера CAN2 вадресное пространство PCIE.
1	MIRR_ERR	Включение зеркалирования регистров EFLG , REC и TEC контроллера CAN2 вадресное пространство PCIE.
0	MIRR_ON	Включение зеркалирования регистров контроллера CAN2 вадресное пространство PCIE.

Подробнее об автоматическом зеркалировании регистров читайте в [главе 6.10](#).

6.1.3 Регистр: CAN1_ACS

Адрес: 1044h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	W	W	W	W	W	W	W	R	W

Описание: Регистр управления доступом к контроллеру CAN1.

При записи в регистр:

Номер бита	Обозначение	Описание
31-24	CA_ADR_CAN1	Начальный адрес записи/чтения регистров контроллера CAN1.
23-16	CA_LEN_CAN1	Количество байт данных в операциях записи/чтения -1 . Например, для записи 3 байт необходимо указывать 2, для 4 - 3 и т.д.
15	CA_AUTO_RTS	Флаг включения автоматической отправки сообщения сразу после записи в буфер контроллера CAN1.
14	CA_ONLY_RTS	Флаг игнорирования команды CA_CMD_CAN1. Используется только совместно с CA_AUTO_RTS для немедленной установки TXREQ для CAN1. CA_BSY_CAN1 не изменится.
13-8	RSRV	Не используются, должны быть "0".
7-0	CA_CMD_CAN1	Команда для модуля доступа к регистрам контроллера CAN1. “02h” – Write. Запись регистров контроллера CAN1 “03h” – Read. Чтение регистров контроллера CAN1 “05h” – Bit Modify. Модификация отдельных битов регистра(только для одиночных регистров, количество байт должно быть 2). “C0h” – CAN Reset. Сброс контроллера CAN1.

При чтении из регистра:

Номер бита	Обозначение	Описание
31-1	RSRV	Не используются, "0".
0	CA_BSY_CAN1	Состояние модуля доступа к контроллеру CAN1: ‘0’ - модуль готов к выполнению новой команды, ‘1’ - модуль занят.

Отдельной командой, без учёта адреса и данных, является сброс контроллера CAN. Фактически, эта операция полностью аналогична записи "1" в 31-й бит регистра CAN1_CTRL, однако механизм сброса посредством регистра CAN2_CTRL является предпочтительным.

Команда Bit Modify используется для изменения отдельных бит в регистре. Например, для сброса отдельных бит, соответствующих обработанным прерываниям, чтобы не пропустить при этом новые прерывания от других источников. Команда применима только для одиночных регистров.

Для этой команды указываются параметры:

- адрес регистра,
- длина (всегда 1),
- два байта в буфере данных:
- первый байт - маска изменяемых бит ("1" - бит подлежит модификации, "0" - бит остаётся без изменения),
- второй байт - новое значение регистра.

6.1.4 Регистр: CAN1_BUF

Адрес: 2000 – 200Ch

Описание: Буфер данных модуля доступа к контроллеру CAN1 16 байт. Буфер сообщений для режима FIFO.

6.1.5 Регистр: CAN2_ACS

Адрес: 1054h

Номер бита																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	-	-	-	-	-	-	-	W	W	W	W	W	W	W	W

Описание: Регистр управления доступом к контроллеру CAN2.

При записи в регистр:

Номер бита	Обозначение	Описание
31-24	CA_ADR_CAN2	Начальный адрес записи/чтения регистров контроллера CAN2.
23-16	CA_LEN_CAN2	Количество байт данных в операциях записи/чтения -1 . Например, для записи 3 байт необходимо указывать 2, для 4 - 3 и т.д.
15	CA_AUTO_RTS	Флаг включения автоматической отправки сообщения сразу после записи в буфер контроллера CAN2.
14	CA_ONLY_RTS	Флаг игнорирования команды CA_CMD_CAN2. Используется только совместно с CA_AUTO_RTS для немедленной установки TXREQ для CAN2. CA_BSY_CAN2 не изменится.
13-8	RSRV	Не используются, должны быть "0".
7-0	CA_CMD_CAN2	Команда для модуля доступа к регистрам контроллера CAN2. “02h” – Write. Запись регистров контроллера CAN2 “03h” – Read. Чтение регистров контроллера CAN2 “05h” – Bit Modify. Модификация отдельных битов регистра (только для одиночных регистров, количество байт должно быть 2). “C0h” – CAN Reset. Сброс контроллера CAN2.

При чтении из регистра:

Номер бита	Обозначение	Описание
31-1	RSRV	Не используются, "0".
0	CA_BSY_CAN2	Состояние модуля доступа к контроллеру CAN2: ‘0’ - модуль готов к выполнению новой команды, ‘1’ - модуль занят.

6.1.6 Регистр: CAN2_BUF

Адрес: 2010 – 201Ch

Описание: Буфер данных модуля доступа к контроллеру CAN2 16 байт. Буфер сообщений для режима FIFO.

6.2 Регистры контроллера CAN.

Общая таблица регистров контроллера CAN.

Таблица 2

Младшие биты адреса	Старшие биты адреса							
	0000xxxx	0001xxxx	0010xxxx	0011xxxx	0100xxxx	0101xxxx	0110xxxx	0111xxxx
0000	RXF0SIDH	RXF3SIDH	RXM0SIDH	TXB0CTRL	TXB1CTRL	TXB2CTRL	RXB0CTRL	RXB1CTRL
0001	RXF0SIDL	RXF3SIDL	RXM0SIDL	TXB0SIDH	TXB1SIDH	TXB2SIDH	RXB0SIDH	RXB1SIDH
0010	RXF0EID8	RXF3EID8	RXM0EID8	TXB0SIDL	TXB1SIDL	TXB2SIDL	RXB0SIDL	RXB1SIDL
0011	RXF0EID0	RXF3EID0	RXM0EID0	TXB0EID8	TXB1EID8	TXB2EID8	RXB0EID8	RXB1EID8
0100	RXF1SIDH	RXF4SIDH	RXM1SIDH	TXB0EID0	TXB1EID0	TXB2EID0	RXB0EID0	RXB1EID0
0101	RXF1SIDL	RXF4SIDL	RXM1SIDL	TXB0DLC	TXB1DLC	TXB2DLC	RXB0DLC	RXB1DLC
0110	RXF1EID8	RXF4EID8	RXM1EID8	TXB0D0	TXB1D0	TXB2D0	RXB0D0	RXB1D0
0111	RXF1EID0	RXF4EID0	RXM1EID0	TXB0D1	TXB1D1	TXB2D1	RXB0D1	RXB1D1
1000	RXF2SIDH	RXF5SIDH	CNF3	TXB0D2	TXB1D2	TXB2D2	RXB0D2	RXB1D2
1001	RXF2SIDL	RXF5SIDL	CNF2	TXB0D3	TXB1D3	TXB2D3	RXB0D3	RXB1D3
1010	RXF2EID8	RXF5EID8	CNF1	TXB0D4	TXB1D4	TXB2D4	RXB0D4	RXB1D4
1011	RXF2EID0	RXF5EID0	CANINTE	TXB0D5	TXB1D5	TXB2D5	RXB0D5	RXB1D5
1100	BFPCTRL	TEC	CANINTF	TXB0D6	TXB1D6	TXB2D6	RXB0D6	RXB1D6
1101	TXRTSCTRL	REC	EFLG	TXB0D7	TXB1D7	TXB2D7	RXB0D7	RXB1D7
1110	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT	CANSTAT
1111	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL	CANCTRL

Выделенные регистры допускают манипуляцию с отдельными битами посредством команды Bit Modify.

Сводная таблица регистров управления.

Таблица 3

Регистр	Адрес	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0	Нач. знач.
<u>BFPCTRL</u>	0Ch	—	—	B1BFS	B0BFS	B1BFE	B0BFE	B1BFM	B0BFM	--00 0000
<u>TXRTSCTRL</u>	0Dh	—	—	B2RTS	B1RTS	B0RTS	B2RTSM	B1RTSM	B0RTSM	--xx x000
<u>CANSTAT</u>	x Eh	OPMOD2	OPMOD1	OPMOD0	—	ICOD2	ICOD1	ICOD0	—	100- 000-
<u>CANCTRL</u>	x Fh	REQOP2	REQOP1	REQOP0	ABAT	OSM	CLKEN	CLKPRE1	CLKPRE0	1110 0111
<u>TEC</u>	1Ch	Transmit Error Counter (TEC)							0000 0000	
<u>REC</u>	1Dh	Receive Error Counter (REC)							0000 0000	
<u>CNF3</u>	28h	SOF	WAKFIL	—	—	—	PHSEG22	PHSEG21	PHSEG20	00-- -000
<u>CNF2</u>	29h	BTLMODE	SAM	PHSEG12	PHSEG11	PHSEG10	PRSEG2	PRSEG1	PRSEGO	0000 0000
<u>CNF1</u>	2Ah	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	0000 0000
<u>CANINTE</u>	2Bh	MERRE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE	0000 0000
<u>CANINTF</u>	2Ch	MERRF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF	0000 0000
<u>EFLG</u>	2Dh	RX1OVR	RX0OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN	0000 0000
<u>TXB0CTRL</u>	30h	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0	-000 0-00
<u>TXB1CTRL</u>	40h	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0	-000 0-00
<u>TXB2CTRL</u>	50h	—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0	-000 0-00
<u>RXB0CTRL</u>	60h	—	RXM1	RXM0	—	RXRTR	BUKT	BUKT	FILHITO	-00- 0000
<u>RXB1CTRL</u>	70h	—	RXM1	RXM0	—	RXRTR	FILHIT2	FILHIT1	FILHITO	-00- 0000

Начальные значения - значения регистров после включения питания, либо выполнения сброса контроллера.

'1' - бит включен, '0' - бит выключен, 'x' - состояние неизвестно.

6.3 Режимы работы контроллера CAN.

6.3.1 Регистр: CAN_CTRL

Адрес: xFh

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
REQOP2	REQOP1	REQOP0	ABAT	OSM	-	-	-
RW - '1'	RW - '1'	RW - '0'	RW - '0'	RW - '0'	RW - '1'	RW - '1'	RW - '1'

Описание: Регистр управления контроллером CAN:

Номер бита	Обозначение	Описание
7-5	REQOP	Request Operation Mode. - Режим работы: 000 - рабочий режим; 001 - режим сна; 010 - режим автономной петли; 011 - режим монитора; 100 - режим конфигурации. Все остальные значения - резервные, устанавливать не рекомендуется. После включения питания значение регистра: "111".
4	ABAT	Abort All Pending Transmissions - Остановка всех активных передач: 1 - Запрос на остановку всех активных передач. 0 - Снятие запроса на остановку всех активных передач.
3	OSM	One Shot Mode - Однократный режим 1 - Включен. Будет сделана только одна попытка передачи сообщения. 0 - Выключен. Попытки передачи сообщения будут повторяться, если это необходимо.
2-0	RSV	Внимание! При записи в регистр, биты 2-0 всегда должны быть установлены значением "111".

Рабочий режим - нормальный режим работы контроллера. В этом режиме контроллер активно прослушивает все сообщения на шине, генерирует подтверждения, кадры ошибок. Также, это единственный режим, в котором контроллер способен передавать сообщения на шину.

Переключение из этого режима в другие происходит не мгновенно, а только после завершения всех активных передач.

Режим сна используется для уменьшения энергопотребления контроллера. В этом режиме тактовый генератор контроллера остановлен, а драйвер шины находится в рецессивном состоянии. Однако, контроллер способен производить мониторинг шины и выходить из режима сна по обнаружению активности на шине.

Чтобы войти в режим сна необходимо установить CAN_CTRL.REQOP = 001 и убедиться, что контроллер вошёл в режим сна, периодически читая регистр CAN_STAT.ORMODE.

В режиме сна всё так же доступно прерывание (если оно разрешено) Wake-up. В случае обнаружения активности на CAN шине контроллер выйдет из режима сна. Также, контроллер можно вывести из режима сна установкой бита CANINTF.WAKIF (CANINTE.WEKIE бит при этом должен быть установлен, разрешая генерацию события).

Контроллер выходит из режима сна в режим монитора в течение примерно 1.5 мкс после события, вызвавшего пробуждение. Кадр, вызвавший пробуждение контроллера,

равно как и все кадры, передача которых началась за время пробуждения, будут проигнорированы.

При отслеживании шины в режиме сна регистром [CNF3.WAKFIL](#) может быть включен ФНЧ для исключения случайных срабатываний.

Режим автономной петли может использоваться при разработке и тестировании системы. В этом режиме возможна передача сообщений самого на себя: из буфера передачи в буфер приёма. Драйвер шины при этом находится в рецессивном состоянии. Бит подтверждения (Ack) игнорируется, сообщения принимаются как будто от внешнего абонента. Фильтры и маски работают как обычно.

Режим монитора позволяет принимать все сообщения, включая сообщения с ошибками. Режим может использоваться для работы в режиме монитора шины, либо для определения скорости передачи на шине в случае горячего включения на шину с неизвестной скоростью.

Для определения скорости на шине необходимо, чтобы минимум два абонента общались между собой. Скорость определяется эмпирически, попытками подстановки различных значений параметров до тех пор пока сообщения не будут приниматься без ошибок.

Драйвер шины в этом режиме находится в рецессивном состоянии. Фильтры и маски работают как обычно. Счётчики ошибок сброшены и выключены.

Режим конфигурации - режим, в котором возможна инициализация контроллера основными параметрами. Режим автоматически выбирается сразу после включения питания, сброса либо может быть выбран из любого другого режима.

Это единственный режим, в котором могут быть изменены регистры: [CNF1](#), [CNF2](#), [CNF3](#), TXRTSCTRL, регистры фильтров и масок.

Счётчики ошибок в этом режиме сброшены и выключены.

6.3.2 Регистр: [CAN_STAT](#)

Адрес: xEh

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
OPMOD2	OPMOD 1	OPMOD 0	-	ICOD2	ICOD1	ICOD0	-
R - '1'	R - '0'	R - '0'	R - '0'	R - 'x'	R - 'x'	R - 'x'	R - '1'

Описание: Регистр управления контроллером CAN:

Номер бита	Обозначение	Описание
7-5	OPMODE	Request Operation Mode. - Текущий режим работы: 000 - рабочий режим; 001 - режим сна; 010 - режим автономной петли; 011 - режим монитора; 100 - режим конфигурации.
4		Резерв
3-1	ICOD	Код прерывания с максимальным приоритетом. Подробное описание см. 6.2.4
0		Резерв

6.4 Прерывания контроллера CAN.

Каждый CAN контроллер имеет 8 источников прерываний. Регистр CANINTE содержит биты разрешения прерывания от каждого источника. Флаги прерываний отображаются в регистре [CANINTF](#). Для сброса прерывания необходимо сбросить соответствующий бит регистра CANINTF. Флаг прерывания не может быть сброшен если сохраняется условие прерывания.

Для сброса прерываний рекомендуется пользоваться командой Bit Modify для сброса отдельных бит регистра. Иначе, при записи в регистр может быть сброшен флаг, который в момент записи только устанавливается в единицу.

6.4.1 Регистр: [CANINTE](#)

Адрес: 2Bh

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
MERRE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1IE	RX0IE
RW - '0'							

Описание: Регистр маски прерывания контроллера CAN:

Номер бита	Обозначение	Описание
7	MERRE	Message Error Interrupt Enable - Разрешение прерывания по ошибке приёма либо передачи сообщения.
6	WAKIE	Wakeup Interrupt Enable - Разрешение прерывания по событию выхода из режима сна.
5	ERRIE	Error Interrupt Enable - Разрешение прерывания по событию ошибки шины (источник - в регистре EFLG).
4	TX2IE	Transmit Buffer 2 Empty Interrupt Enable - Разрешение прерывания по отправке сообщения из буфера передачи 2
3	TX1IE	Transmit Buffer 1 Empty Interrupt Enable - Разрешение прерывания по отправке сообщения из буфера передачи 1
2	TX0IE	Transmit Buffer 0 Empty Interrupt Enable - Разрешение прерывания по отправке сообщения из буфера передачи 0
1	RX1IE	Receive Buffer 1 Full Interrupt Enable - Разрешение прерывания по получению сообщения в буфер приёма 1
0	RX0IE	Receive Buffer 0 Full Interrupt Enable - Разрешение прерывания по получению сообщения в буфер приёма 0
		'1' - прерывание разрешено, '0' - прерывание запрещено

Если разрешена функция DMA, биты RX1IE, RX0IE должны быть сброшены в 0. При этом также необходимо сконфигурировать регистр [BFPCTRL](#) и включить разрешение записи в регистре [DMA DATA BASE](#).

6.4.2 Регистр: [CANINTF](#)

Адрес: 2Ch

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
MERRF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF
RW - '0'							

Описание: Регистр прерывания контроллера CAN:

Номер бита	Обозначение	Описание
7	MERRF	Message Error Interrupt - Прерывание по ошибке приёма либо передачи сообщения.
6	WAKIF	Wakeup Interrupt - Прерывание по событию выхода из режима сна.
5	ERRIF	Error Interrupt - Прерывание по событию ошибки шины или переполнению буферов (конкретный источник - в регистре EFLG).
4	TX2IF	Transmit Buffer 2 Empty Interrupt - Прерывание по отправке сообщения из буфера передачи 2
3	TX1IF	Transmit Buffer 1 Empty Interrupt - Прерывание по отправке сообщения из буфера передачи 1
2	TX0IF	Transmit Buffer 0 Empty Interrupt - Прерывание по отправке сообщения из буфера передачи 0
1	RX1IF	Receive Buffer 1 Full Interrupt - Прерывание по получению сообщения в буфер приёма 1
0	RX0IF	Receive Buffer 0 Full Interrupt - Прерывание по получению сообщения в буфер приёма 0
		'1' - прерывание активно, '0' - прерывание не активно.

Прерывания RX0IF, RX1IF автоматически сбрасываются контроллером DMA сразу после вычитывания данных из буфера контроллера CAN и копирования их в память ПК.

Прерывания могут быть как сброшены, так и установлены программно. Например, флаг WAKIF для принудительного вывода контроллера из режима сна.

Также, активные прерывания отображаются в регистре [CANSTAT.ICOD](#). В поле ICOD отображается текущее прерывание с наивысшим приоритетом (соответствует меньшему номеру). Как только устраняется текущий источник прерывания, поле ICOD начинает отображать прерывание с меньшим приоритетом. В поле ICOD отображаются только прерывания, разрешённые регистром [CANINTE](#).

ICOD[2:0]	Логическое выражение
000	$\overline{ERR} \cdot \overline{WAK} \cdot \overline{TX0} \cdot \overline{TX1} \cdot \overline{TX2} \cdot \overline{RX0} \cdot \overline{RX1}$
001	\overline{ERR}
010	$\overline{ERR} \cdot \overline{WAK}$
011	$\overline{ERR} \cdot \overline{WAK} \cdot \overline{TX0}$
100	$\overline{ERR} \cdot \overline{WAK} \cdot \overline{TX0} \cdot \overline{TX1}$
101	$\overline{ERR} \cdot \overline{WAK} \cdot \overline{TX0} \cdot \overline{TX1} \cdot \overline{TX2}$
110	$\overline{ERR} \cdot \overline{WAK} \cdot \overline{TX0} \cdot \overline{TX1} \cdot \overline{TX2} \cdot \overline{RX0}$
111	$\overline{ERR} \cdot \overline{WAK} \cdot \overline{TX0} \cdot \overline{TX1} \cdot \overline{TX2} \cdot \overline{RX0} \cdot \overline{RX1}$

ERR = CANINTE.ERRIE

6.4.3 Регистр: **BFPCTRL**

Адрес: 0Ch

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
-	-	B1BFS	B0BFS	B1BFE	B0BFE	B0BFM	B0BFM
RW - '0'							

Описание: Регистр разрешения сигналов прерывания буферов приёма контроллера CAN:

Номер бита	Обозначение	Описание
7-6	-	Не используется.
5	B1BFS	Pin 1 State - Постоянный логический уровень линии прерывания, активен при B1FM = 0.
4	B0BFS	Pin 0 State - Постоянный логический уровень линии прерывания, активен при B0FM = 0.
3	B1BFE	Pin1 Function Enable - Разрешение линии прерывания буфера приёма RXB1.
2	B0BFE	Pin0 Function Enable - Разрешение линии прерывания буфера приёма RXB0.
1	B1BFM	Pin 1 Operation Mode - Функция линии прерывания 1: 1 - Выдча прерывания DMA по приходу сообщения в буфер приёма RXB1; 0 - Постоянный уровень, определяемый битом B1BFS.
0	B0BFM	Pin 0 Operation Mode - Функция линии прерывания 0: 1 - Выдча прерывания DMA по приходу сообщения в буфер приёма RXB0; 0 - Постоянный уровень, определяемый битом B0BFS.

Каждый контроллер CAN имеет два выхода прерывания на блок DMA, по одному от каждого буфера приёма сообщений. Для автоматической передачи принятых сообщений через механизм DMA необходимо активировать линии прерывания:

Биты BnBFE установить в 1.

Биты BnBFM установить в 1.

Состояние битов BnBFS не имеет значения.

Если же ПО не пользуется функцией DMA, линии прерывания должны быть деактивированы:

Биты BnBFE установить в 0.

Биты BnBFM установить в 0.

Состояние битов BnBFS не имеет значения.

Также возможна принудительная подача прерывания на блок DMA последовательной записью значений:

1 - BnBFE = 1, BnBFM =0, BnBFS=0;

2 - BnBFE = 1, BnBFM =0, BnBFS=1;

Предварительно в регистр должно быть записано значение:

BnBFE = 1, BnBFM =0, BnBFS=1;

6.4.4 Регистр: [TXRTSCTRL](#)

Адрес: 0Dh

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
-	-	B2RTS	B1RTS	B0RTS	B2RTSM	B1RTSM	B0RTSM
RW - '0'	RW - '0'	RW - 'X'	RW - 'X'	RW - 'X'	RW - '0'	RW - '0'	RW - '0'

Описание: Регистр разрешения сигналов прерывания буферов приёма контроллера CAN:

Номер бита	Обозначение	Описание
7-6	-	Не используется.
5	B2RTS	TX2RTS State - Текущее состояние линии запроса отправки сообщения из буфера 2, активен при B2RTSE = 1.
4	B1RTS	TX2RTS State - Текущее состояние линии запроса отправки сообщения из буфера 1, активен при B1RTSE = 1.
3	B0RTS	TX2RTS State - Текущее состояние линии запроса отправки сообщения из буфера 0, активен при B0RTSE = 1.
2	B2RTSE	TX2RTS Enable - Разрешение отправки сообщения из буфера 2 по триггеру: 1 - разрешено; 0 - запрещено.
1	B1RTSE	TX1RTS Enable - Разрешение отправки сообщения из буфера 1 по триггеру: 1 - разрешено; 0 - запрещено.
0	B0RTSE	TX0RTS Enable - Разрешение отправки сообщения из буфера 0 по триггеру: 1 - разрешено; 0 - запрещено.

Для работы в режиме TTCAN перед началом работы в режиме конфигурации необходимо разрешить отправку сообщений по триггерам таймера локального времени. Чтобы разрешить отправку сообщений из всех буферов, необходимо записать в регистр значение 07h.

6.5 Конфигурация скорости шины CAN.

Все абоненты CAN шины должны использовать одну и ту же номинальную частоту передачи. Однако, так как каждый абонент работает со своим осциллятором и времена задержки между абонентами в линии различаются, физический уровень CAN контроллера включает в себя ФАПЧ, синхронизирующуюся по фронтам битов данных. ФАПЧ разбивает время передачи одного бита на сегменты, образованные квантами времени (TQ - time quanta) единичной длины.

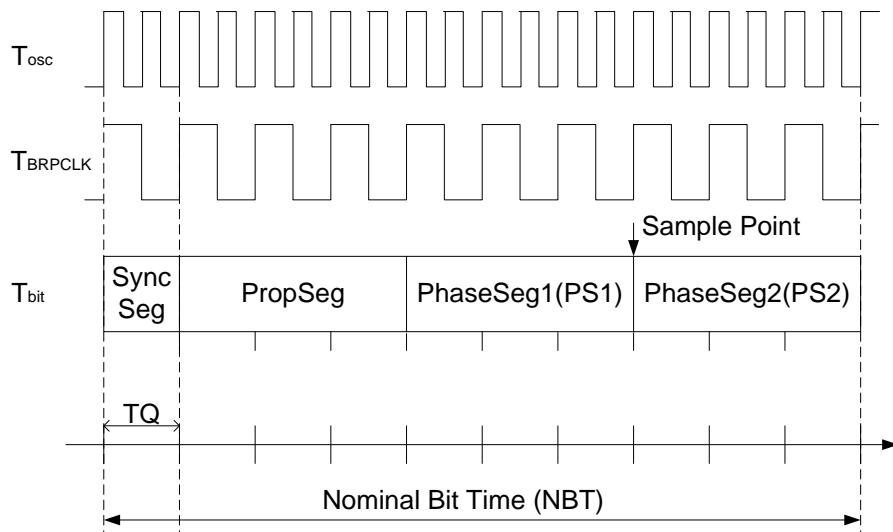


Рисунок 6 TQ и время бита

Сегменты состоят из целого числа квонтов времени. Длина квонта определяется частотой опорного осциллятора и коэффициентом деления (**BRP**). Частота осциллятора фиксирована и равна 20МГц (период 50нс).

$$TQ = \frac{2 * BRP}{F_{osc}} = BRP * 100\text{нс}$$

SyncSeg - Сегмент синхронизации. Предполагается, что все фронты переключения бит приходятся именно на этот сегмент. Длина сегмента фиксирована - 1TQ.

PropSeg - Сегмент распространения. Выдерживается для компенсации задержки распространения сигнала в линии между абонентами. Задержка должна быть равна удвоенной сумме времени распространения сигнала в линии, включая задержки на передатчиках. Длина сегмента: 1-8TQ.

PhaseSeg1 и **PhaseSeg2** - Фазовые сегменты служат для компенсации переходных процессов в линии. Длина PhaseSeg1: 1-8TQ. Длина PhaseSeg2: 2-8TQ. Длины фазовых сегментов ФАПЧ может менять в процессе ресинхронизации на значение SJW.

Sample Point - точка сэмплирования данных, определения логического уровня сигнала. Может быть как единичным, так и выполняться трижды в течение последнего TQ фазового сегмента 1. В последнем случае результат выбирается по мажоритарной схеме.

При обнаружении фронта сигнала вне сегмента синхронизации детектируется ошибка фазы. При этом происходит изменение длины фазовых сегментов на величину **SJW** (Synchronization Jump Width) - шаг перестройки синхронизации. SJW может быть задана от 1 до 4 TQ.

Пример конфигурации для частоты передачи 125кГц:

*Частота внутреннего опорного осциллятора - всегда 20МГц, период частоты - 50нс. При делителе BRP, установленном на 5 (CNF1.BRP[5:0]=00100) получаем TQ = 2*5*50нс = 500нс. Период частоты 125кГц ровно в 16 раз меньше, значит время бита должно состоять из 16 TQ.*

Точка сэмплирования данных обычно устанавливается на 60-70% времени бита. Время распространения сигнала обычно 1-2TQ.

SyncSeg всегда = 1TQ, PropSeg = 2TQ. Теперь, если установим PhaseSeg1 на 7, получим точку сэмплирования на 62,5% PhaseSeg2 будет равно 6.

Важно! Не смотря на то, что регистры CNF1, CNF2, CNF3 расположены по смежным адресам, производить в них запись возможно только отдельными операциями. То есть, нельзя записывать 3 байта последовательно начиная с адреса 28h. Только раздельно по байтам, по адресам: 28h, 29h и 2Ah.

6.5.1 Регистр: [CNF1](#)

Адрес: 2Ah

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
RW - '0'							

Описание: Регистр конфигурации 1:

Номер бита	Обозначение	Описание
7-6	SJW	Synchronization Jump Width - Шаг перестройки синхронизации. Значение параметра = значение регистра + 1: 11 - SJW = 4TQ; 10 - SJW = 3TQ; 01 - SJW = 2TQ; 00 - SJW = 1TQ.
5-0	BRP	Bit Rate Prescaler - коэффициент деления частоты опорного генератора. Значение параметра = значение регистра + 1: 000000 - BRP = 1 000001 - BRP = 2 000010 - BRP = 3 111111 - BRP = 64

6.5.2 Регистр: [CNF2](#)

Адрес: 29h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
BTLMODE	SAM	PHSEG12	PHSEG11	PHSEG10	PRSEG2	PRSEG1	PRSEG0
RW - '0'							

Описание: Регистр конфигурации 2:

Номер бита	Обозначение	Описание
7	BTLMODE	PS2 Bit Time Mode - выбор величины PS2. 1 - Длина PS2 определяется полем PHSEG2 регистра CNF3; 0 - Длина PS2=PS1, но минимум 2TQ.
6	SAM	Sample Point Configuration - Конфигурация точки сэмплирования. 1 - Сигнал сэмплируется три раза; 0 - Сигнал сэмплируется один раз.
5-3	PHSEG1	Phase Segment 1 - Длительность сегмента фазы 1. Значение параметра = значение регистра + 1;
2-0	PRSEG	Propagation Segment - Длительность сегмента фазы распространения. Значение параметра = значение регистра + 1;

6.5.3 Регистр: [CNF3](#)

Адрес: 28h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SOF	WAKFIL	-	-	-	PHSEG22	PHSEG21	PHSEG20
RW - '0'							

Описание: Регистр конфигурации 2:

Номер бита	Обозначение	Описание
7	SOF	Фиксировать время прихода кадра по старт-биту на шине.
6	WAKFIL	Wake-up Filter - ФНЧ шины для детектора активности шины в режиме сна. 1 - включен; 0 - выключен.
5-3		Не используется.
2-0	PHSEG2	Phase Segment 2 - Длительность сегмента фазы 2. Значение параметра = значение регистра + 1;

Важно! CNF3.SOF должен быть установлен в "1" для корректной установки временной метки в заголовке принятых сообщений в буфере DMA.

6.6 Ошибки CAN-шины.

Протокол CAN-шины имеет следующий механизм обработки ошибок.

Каждый абонент шины, продетектировавший ошибку, сообщает об этом событии остальным абонентам выдачей специального Error Frame - сообщения об ошибке. При этом передача текущего кадра прерывается и повторяется, как только это возможно.

Каждый абонент шины имеет два счётчика ошибок: REC (Receive Error Counter) - счётчик ошибок приёма и TEC (Transmit Error Counter) - счётчик ошибок передачи. В зависимости от их значения абонент может находиться в одном из трёх состояний:

Error Active - нормальное состояние абонента. Оба счётчика меньше 128. Он может передавать сообщения и выдавать активные сообщения об ошибке (доминантными битами) без ограничений.

Error Passive - в этом состоянии абонент может передавать сообщения и выдавать пассивные сообщения об ошибке (рецессивным битом). Для входа в этот состояния хотя бы один счётчик должен быть равен или больше 128.

Bus Off - в этом состоянии абонент не может принимать никакого участия в обмене на шине: приём и передача любых сообщений запрещены. В Bus-Off могут находиться только драйверы шины - активные абоненты, не мониторы. Для входа в это состояние счётчик TEC должен достигнуть значения 255.

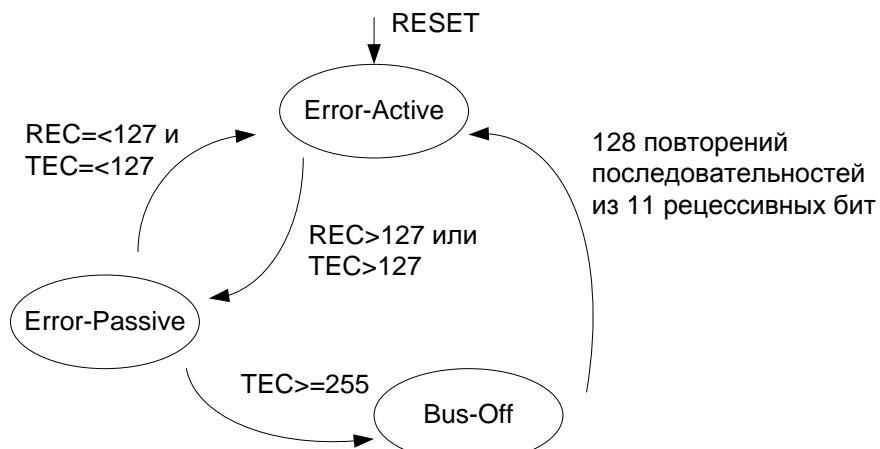


Рисунок 7 Диаграмма состояний CAN-шины

6.6.1 Регистр: TEC

Адрес: 1Ch

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
R - '0'							

Описание: Счётчик ошибок передачи:

Номер бита	Обозначение	Описание
7-0	TEC	Transmit Error Counter - счётчик ошибок передачи.

6.6.2 Регистр: [REC](#)

Адрес: 1Dh

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
R - '0'							

Описание: Счётчик ошибок приёма:

Номер бита	Обозначение	Описание
7-0	REC	Receive Error Counter - счётчик ошибок приёма.

6.6.3 Регистр: [EFLG](#)

Адрес: 2Dh

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
RX1OVR-	RX0OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN
RW - '0'	RW - '0'	R - '0'					

Описание: Флаги ошибок шины:

Номер бита	Обозначение	Описание
7	RX1OVR	Receiver Buffer 1 Overflow Flag - Переполнение буфера приёма 1.
6	RX0OVR	Receiver Buffer 0 Overflow Flag - Переполнение буфера приёма 0.
5	TXBO	Bus-Off Error Flag - TEC>=255, контроллер в режиме Bus Off
4	TXEP	Transmit Error-Passive Flag - TEC>127 контроллер в режиме Error-Passive.
3	RXEP	Receiver Error-Passive Flag - REC>127 контроллер в режиме Error-Passive.
2	TXWAR	Transmit Error-Warning Flag - TEC>=96
1	RXWAR	Receiver Error-Warning Flag - REC>=96.
0	EWARN	Error-Warning Flag - TEC или REC >=96.

6.7 Передача сообщений.

Способы отправки сообщений:

1. Запись в регистр [TXBnCTRL](#).

Чтобы начать передачу сообщения необходимо записать сообщение в буфер и установить бит [TXBnCTRL.TXREQ](#). Как только установлен бит [TXBnCTRL.TXREQ](#), автоматически сбрасываются биты [TXBnCTRL.ABTF](#), [TXBnCTRL.MLOA](#), [TXBnCTRL.TXERR](#) этого же буфера.

2. Автоматическая отправка после копирования в буфер.

Если сообщение нужно отправить немедленно, можно воспользоваться битом 15 [CANx_ACS.CA_AUTO_RTS](#). Сообщение будет отправлено сразу же после копирования в буфер TXBn контроллера CAN.

Если сообщение изменяется за две и больше записей (например, меняется поле SIDH, затем данные), бит следует установить во время последней записи.

3. Немедленная отправка сообщения в буфере.

Если сообщение уже находится в буфере, можно вызвать немедленную отправку сообщения дополнительной установкой бита 14 [CANx_ACS.CA_ONLY_RTS](#). Другие поля регистра [CANx_ACS](#) должны содержать параметры:

- CA_AUTO_RTS – установлен в ‘1’;
- CA_ADR_CAN – “30” – для TXB0, “40” – для TXB1, “50” – для TXB2;
- CA_CMD_CAN – “02” (запись);
- CA_LEN_CAN – не имеет значения.

В случае активного бита CA_ONLY_RTS бит занятости CA_BSY_CAN не будет установлен. Следующую команду можно подавать сразу же после команды с CA_ONLY_RTS.

Перед записью команды с CA_ONLY_RTS необходимо убедиться в то, что предыдущий доступ к контроллеру CAN завершён – CA_BSY_CAN сброшен в ‘0’.

4. Триггеры отправки сообщений.

Подробно отправка сообщений по триггерам описана в [главе 5.3](#).

5. Режим FIFO.

Подробно отправка сообщений с использованием FIFO описана в [главе 5.5](#).

Каждый CAN контроллер имеет три буфера для передачи сообщений. Каждый буфер занимает 14 байт в адресном пространстве контроллера.

Первый байт [TXBnCTRL](#) – регистр управления, связанный с данным буфером сообщения. Содержимое регистра определяет условия передачи сообщения и отображает статус передачи.

Следующие пять байт содержат стандартный либо расширенный идентификатор – [TXBnSIDH](#), [TXBnSIDL](#), [TXBnEID8](#), [TXBnEID0](#) а также флаги – [TXBnDLC](#).

Последние восемь байт содержат данные сообщения.

Как минимум три регистра должны быть загружены: [TXBnSIDH](#), [TXBnSIDL](#) и [TXBnDLC](#). Если сообщение содержит данные, то их следует записать в регистр [TXBnDm](#). Если сообщение имеет расширенный идентификатор, то он записывается в регистр [TXBnEIDm](#) и устанавливается бит [TXBnSIDL.EXIDE](#).

Прежде чем отправлять сообщение система должна проинициализировать бит [CANINTE.TXInE](#) чтобы разрешить либо запретить прерывание по окончанию передачи.

Важно! Перед началом записи сообщения в буфер бит запроса на отправку [TXBnCTRL.TXREQ](#) должен быть сброшен.

Каждому передаваемому сообщению может быть назначен приоритет отправки. Приоритет - внутренний для контроллера, не имеет отношения к протоколу CAN-шины.

Перед отправкой сообщения контроллер проверяет приоритет - [TXBnCTRL.TXP](#) всех буферов. Первым будет отправлено сообщение с наивысшим приоритетом. Например, если буфер 0 имеет приоритет больший, чем буфер 1, то первым будет отправлено сообщение из буфера 0.

Если два буфера имеют одинаковый приоритет, то первым будет отправлено сообщение из буфера с наибольшим порядковым номером. Например, если буфер 0 имеет приоритет равный буферу 1, то первым будет отправлено сообщение из буфера 1.

Важно! Установка флага [TXBnCTRL.TXREQ](#) не означает начало немедленной отправки сообщения. Она означает лишь готовность буфера к отправке. Действительно попытка отправки начнётся, как только CAN-шина будет свободна.

Как только сообщение будет передано, бит [TXBnCTRL.TXREQ](#) будет автоматически сброшен. Если был установлен бит [CANINTE.TXnIE](#), то будет выставлено прерывание и установлен бит [CANINTF.TXnIF](#).

Если попытка передачи сообщения не удалась, бит [TXBnCTRL.TXREQ](#) остаётся установленным. То есть, буфер данного сообщения всё также готов к отправке. Но при этом будут установлены следующие флаги:

- если передача сообщения началась, но в процессе передачи произошла ошибка нашине, будет установлен флаг [TXBnCTRL.TXERR](#); одновременно, если установлен бит [CANINTE.MERRE](#) будет выставлено прерывание и установлен бит [CANINTF.MERRF](#).
- если сообщение проиграло арбитраж нашине, то будет установлен бит [TXBnCTRL.MLOA](#).

Контроллер также имеет режим однократной передачи сообщений. Обычно, если сообщение проигрывает арбитраж либо в процессе передачи возникают ошибки, попытки его передачи повторяются. В режиме однократной передачи контроллер сделает лишь одну попытку передачи сообщения. Такой режим используется в ряде систем, детерминированных по времени на базе TTCAN.

Для включения данного режима необходимо установить бит [CANCTRL.OSM](#). После первой попытки передачи бит [TXBnCTRL.TXREQ](#) будет сброшен. При этом флаги статуса передачи будут выставлены точно так же, как и для обычного режима.

Запрос на передачу сообщения можно снять, обнулив соответствующий бит [TXBnCTRL.TXREQ](#).

Также, можно приостановить запросы на передачу всех сообщений, установив бит [CANCTRL.ABAT](#). В этом случае во всех буферах, запрашивавших передачу, будет установлен флаг [TXBnCTRL.ABTF](#). При обнулении бита [TXBnCTRL.TXREQ](#) флаг [TXBnCTRL.ABTF](#) не устанавливается.

Важно! Если в момент снятия запроса сообщение уже передавалось нашину, то передача будет продолжена. В случае неудачи, дальнейшие попытки передачи будут прерваны. В режиме однократной передачи при неудачной отправке будет установлен флаг [TXBnCTRL.ABTF](#).

6.7.1 Регистр: [TXBnCTRL](#)

Адрес: 30h, 40h, 50h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
-	ABTF	MLOA	TXERR	TXREQ	-	TXP1	TXP0
R - '0'	R - '0'	R - '0'	R - '0'	RW - '0'	R - '0'	RW - '0'	RW - '0'

Описание: Регистр управления буфера передачи n:

Номер бита	Обозначение	Описание
7	-	Не используется.
6	ABTF	Message Aborted Flag - флаг обрыва передачи 1 - передача сообщения была прервана; 0 - сообщение успешно передано.
5	MLOA	Message Lost Arbitration - флаг индикации арбитража 1 - сообщение проиграло арбитраж во время передачи; 0 - сообщение не проигрывало арбитраж.
4	TXERR	Transmission Error Detected флаг ошибки передачи. 1 - во время передачи произошла ошибка на шине; 0 - ошибок во время передачи не было.
3	TXREQ	Message Transmit Request - флаг запроса передачи. 1 - запрос передачи сообщения из данного буфера активен; 0 - данный буфер не требует передачи сообщения.
2	-	Не используется.
1-0	TXP	Transmit Buffer Priority - приоритет буфера передачи. 11 - высший приоритет; 00 - низший приоритет.

6.7.2 Регистр: [TXBnSIDH](#)

Адрес: 31h, 41h, 51h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
RW - 'x'							

Описание: Стандартный идентификатор, старшая часть, буфер передачи n:

Номер бита	Обозначение	Описание
7-0	SID	Стандартный идентификатор, биты с 10 по 3.

6.7.3 Регистр: [TXBnSIDL](#)

Адрес: 32h, 42h, 52h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SID2	SID1	SID0	-	EXIDE	-	EID17	EID16
RW - 'x'							

Описание: Стандартный идентификатор, младшая часть, буфер передачи n:

Номер бита	Обозначение	Описание
7-5	SID	Стандартный идентификатор, биты с 2 по 0.
4	-	Не используется.
3	EXIDE	Extended Identifier Enable - разрешение расширенного идентификатора 1 - сообщение будет передано с расширенным идентификатором; 0 - сообщение будет передано со стандартным идентификатором.
2	-	Не используется.
1-0	EID	Расширенный идентификатор, биты 17, 16.

6.7.4 Регистр: [TXBnEID8](#)

Адрес: 33h, 43h, 53h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
RW - 'x'							

Описание: Расширенный идентификатор, старшая часть, буфер передачи n:

Номер бита	Обозначение	Описание
7-0	EID	Расширенный идентификатор, биты 15 - 8.

6.7.5 Регистр: [TXBnEID0](#)

Адрес: 34h, 44h, 54h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
RW - 'x'							

Описание: Расширенный идентификатор, младшая часть, буфер передачи n:

Номер бита	Обозначение	Описание
7-0	EID	Расширенный идентификатор, биты 7 - 0.

6.7.6 Регистр: TXBnDLC

Адрес: 35h, 45h, 55h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
-	RTR	-	-	DLC3	DLC2	DLC1	DLC0
RW - 'x'							

Описание: Регистр длины данных, буфер передачи n:

Номер бита	Обозначение	Описание
7	-	Не используется.
6	RTR	Remote Transmission Request - внешний запрос транзакции 1 - передаваемое сообщение - удалённый запрос транзакции; 0 - передаваемое сообщение - данные.
5-4	-	Не используется.
3-0	DLC	Data Length Code длина передаваемых данных (0 - 8). Возможно установить значение и больше 8, но всё равно передано будет только 8 байт.

6.7.7 Регистр: TXBnDm

Адрес: 36h-3Dh, 46h-4Dh, 56-5Dh

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
TXBnDm 7	TXBnDm 6	TXBnDm 5	TXBnDm 4	TXBnDm 3	TXBnDm 2	TXBnDm 1	TXBnDm 0
RW - 'x'							

Описание: Буфер передачи n, байт данных m:

Номер бита	Обозначение	Описание
7-0	TXBnDm	Байты данных сообщения.

6.8 Приём сообщений.

Каждый CAN контроллер имеет два буфера для приёма сообщений: RXB0 и RXB1. Каждый буфер занимает в адресном пространстве контроллера 14 байт. Также имеется набор фильтров и масок для фильтрации входящих сообщений по различным критериям. Подробно механизм фильтрации сообщений описан [в следующей главе 6.9](#).

Если сообщение прошло фильтрацию и было записано в какой-либо буфер, функция DMA устройства немедленно копирует сообщение в память ПК в специально выделенную область памяти чтобы ускорить процедуру передачи полученного сообщения приложению.

Каждая запись в буфере ПК имеет длину 18 байт (с выравниванием по 64 байта) и состоит из: заголовка, где указан источник данных: номер CAN контроллера и буфера, временной метки в микросекундах, полной копии буфера сообщения.

Важно! Содержимое буфера приёма CAN-контроллера не обнуляется перед записью в него нового сообщения. Если длина данных сообщения либо идентификатора меньше длины той же области в предыдущем сообщении, незначащие области буфера могут содержать соответствующие поля предыдущих сообщений.

Запись имеет следующий вид:

Слово\байт	0	1	2	3
0h	MSG_HEADER			
4h	RXBnCTRL	RXBnSIDH	RXBnSIDL	RXBnEID8
8h	RXBnEID0	RXBnDLC	RXBnD0	RXBnD1
Ch	RXBnD2	RXBnD3	RXBnD4	RXBnD5
10h	RXBnD6	RXBnD7	-	EPOCH
14h-3C	-	-	-	-

Содержимое поля MSG_HEADER:

Номер бита	Обозначение	Описание
31-2	TIMESTAMP	Значение таймера локального времени на момент старта приёма сообщения. Биты 31-16 - в единицах сетевого времени; Биты 15-2 - в 50нс единицах
1	CANn	Номер CAN-контроллера: "0" - CAN1; "1" - CAN2.
0	RXBn	Номер буфера CAN-контроллера: "0" - RXB0; "1" - RXB1.

TIMESTAMP - метка времени получения сообщения. Значение таймера локального времени на момент начала приёма сообщения.

6.8.1 Регистр: [RXB0CTRL](#)

Адрес: 60h (смещение 4h буфере ПК)

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
OVF	RXM1	RXM0	-	RXRTR	-	-	FILHITO
R - '0'	RW - '0'	RW - '0'	R - '0'	R - '0'	RW - '0'	R - '0'	R - '0'

Описание: Регистр управления буфера приёма 0:

Номер бита	Обозначение	Описание
7	OVF	Overflow Flag – индикатор переполнения данных. Установленный в '1' бит сигнализирует, что предыдущие данные (одно или несколько сообщений) были утрачены из-за остановки DMA.
6-5	RXM	Receive Buffer Operating Mode - режим работы буфера: 11 -принимать все сообщения, маски/фильтры отключены; 10 - принимать все сообщения без ошибок с расширенными идентификаторами, которые прошли фильтрацию; 01 - принимать все сообщения без ошибок со стандартными идентификаторами, которые прошли фильтрацию; фильтры Extended ID регистров RXFnEID8:RXFnEID0 игнорируются; 00 - принимать все сообщения без ошибок с обоими типами идентификаторов, которые прошли фильтрацию; фильтры Extended ID регистров RXFnEID8:RXFnEID0 накладываются на первые два байта сообщений со стандартными идентификаторами.
4	-	Не используется.
3	RXRTR	Received Remote Transfer Request - удалённый запрос транзакции 1 - принятое сообщение - удалённый запрос транзакции; 0 - принятое сообщение - не является удалённым запросом транзакции.
2	-	Не используется.
1	-	Не используется.
0	FILHITO	Filter Hit - индикатор фильтра, пропустившего сообщение в буфер. 1 - Acceptance Filter 1 (RXF1); 0 - Acceptance Filter 0 (RXF0).

6.8.2 Регистр: [RXB1CTRL](#)

Адрес: 70h (смещение 4h буфере ПК)

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
OVF	RXM1	RXM0	-	RXRTR	FILHIT2	FILHIT1	FILHITO
R - '0'	RW - '0'	RW - '0'	R - '0'				

Описание: Регистр управления буфера приёма 1:

Номер бита	Обозначение	Описание
7	OVF	Overflow Flag – индикатор переполнения данных. Установленный в '1' бит сигнализирует, что предыдущие данные (одно или несколько сообщений) были утрачены из-за остановки DMA.
6-5	RXM	Receive Buffer Operating Mode - режим работы буфера: 11 -принимать все сообщения, маски/фильтры отключены; 10 - принимать все сообщения без ошибок с расширенными идентификаторами, которые прошли фильтрацию; 01 - принимать все сообщения без ошибок со стандартными идентификаторами, которые прошли фильтрацию; 00 - принимать все сообщения без ошибок с обоими типами идентификаторов, которые прошли фильтрацию.
4	-	Не используется.
3	RXRTR	Received Remote Transfer Request - удалённый запрос транзакции 1 - принятое сообщение - удалённый запрос транзакции; 0 - принятое сообщение - не является удалённым запросом транзакции.
2-0	FILHITO	Filter Hit - индикатор фильтра, пропустившего сообщение в буфер. 101 - Acceptance Filter 5 (RXF5); 100 - Acceptance Filter 4 (RXF4); 011 - Acceptance Filter 3 (RXF3); 010 - Acceptance Filter 2 (RXF2).

6.8.3 Регистр: [RXBnSIDH](#)

Адрес: 61h, 71h (смещение 5h буфере ПК)

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
R - 'x'							

Описание: Стандартный идентификатор, старшая часть, буфер приёма n:

Номер бита	Обозначение	Описание
7-0	SID	Стандартный идентификатор, биты с 10 по 3.

6.8.4 Регистр: [RXBnSIDL](#)

Адрес: 72h, 72h (смещение 6h буфере ПК)

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SID2	SID1	SID0	SRR	IDE	-	EID17	EID16
R - 'x'							

Описание: Стандартный идентификатор, младшая часть, буфер приёма n:

Номер бита	Обозначение	Описание
7-5	SID	Стандартный идентификатор, биты с 2 по 0.
4	SRR	Standard Frame Remote Transmit Request - флаг удалённого запроса транзакции для стандартного типа кадров; флаг достоверен только в случае, если бит IDE=0. 1 - стандартный кадр с удалённым запросом транзакции; 0 - стандартный кадр с данными.
3	IDE	Extended Identifier Flag -флаг расширенного идентификатора 1 - принятое сообщение имеет расширенный идентификатор; 0 - принятое сообщение имеет стандартный идентификатор.
2	-	Не используется.
1-0	EID	Расширенный идентификатор, биты 17, 16.

6.8.5 Регистр: [RXBnEID8](#)

Адрес: 63h, 73h (смещение 7h буфере ПК)

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
R - 'x'							

Описание: Расширенный идентификатор, старшая часть, буфер приёма n:

Номер бита	Обозначение	Описание
7-0	EID	Расширенный идентификатор, биты 15 - 8.

6.8.6 Регистр: [RXBnEID0](#)

Адрес: 64h, 74h (смещение 8h буфере ПК)

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
R - 'x'							

Описание: Расширенный идентификатор, младшая часть, буфер приёма n:

Номер бита	Обозначение	Описание
7-0	EID	Расширенный идентификатор, биты 7 - 0.

6.8.7 Регистр: [RXBnDLC](#)

Адрес: 65h, 75h (смещение 9h буфере ПК)

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
-	RTR	-	-	DLC3	DLC2	DLC1	DLC0
R - 'x'							

Описание: Регистр длины данных, буфер приёма n:

Номер бита	Обозначение	Описание
7	-	Не используется.
6	RTR	Extended Frame Remote Transmit Request - флаг удалённого запроса транзакции для расширенного типа кадров; флаг достоверен только в случае, если бит RXBnSIDL.IDE=1. 1 - расширенный кадр с удалённым запросом транзакции; 0 - расширенный кадр с данными.
5-4	-	Не используется, зарезервировано.
3-0	DLC	Data Length Code длина принятых данных (0 - 8).

6.8.8 Регистр: [RXBnDm](#)

Адрес: 66h-6Dh, 76h-7Dh (смещение 0Ah-11h буфере ПК)

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
RXBnDm 7	RXBnDm 6	RXBnDm 5	RXBnDm 4	RXBnDm 3	RXBnDm 2	RXBnDm 1	RXBnDm 0
R - 'x'							

Описание: Буфер приёма n, байт данных m:

Номер бита	Обозначение	Описание
7-0	RXBnDm	Байты данных сообщения.

6.9 Фильтрация принимаемых сообщений.

Все сообщения с CAN-шины принимаются во временный буфер. Маски и фильтры сообщений определяют, должно ли принятое сообщение попасть в регистры какого-либо буфера приёма, доступного на чтение. Сразу по приёму сообщения без ошибок поля идентификатора сравниваются с установленными фильтрами и принимается решение о копировании их в один из двух буферов приёма контроллера.

Фильтры RXF0, RXF1 и маска RXM0 используются для фильтрации сообщений в буфер RXB0. Фильтры RXF2, RXF3, RXF4, RXF5 и маска RXM1 используются для фильтрации сообщений в буфер RXB1.

Важно! При одновременном совпадении нескольких фильтров сообщение будет записано в буфер со статусом приёма по фильтру с наименьшим порядковым номером. При одновременном совпадении фильтров в обоих буферах сообщение будет записано в буфер RXB0.

Маска используется для указания, какие биты идентификатора сообщения следует сравнивать с фильтром. В таблице 6.1 приведена таблица истинности для каждого бита. Для приёма сообщения необходимо, чтобы все сравниваемые биты имели результат "Принять".

Бит маски	Бит фильтра	Бит ID сообщения	Результат
0	x	x	Принять
1	0	0	Принять
1	0	1	Не принимать
1	1	0	Не принимать
1	1	1	Принять

"x" - значение бита не важно

Таблица 6.1 Таблица истинности для фильтрации сообщений.

Для сообщений с расширенным идентификатором маска и фильтры применяются ко всем битам идентификатора. Для сообщений со стандартным идентификатором маска и фильтры применяются как к идентификатору, так и к первым двум байтам данных. Это может быть полезно при реализации некоторых систем на базе CAN шины, таких как DeviceNet.

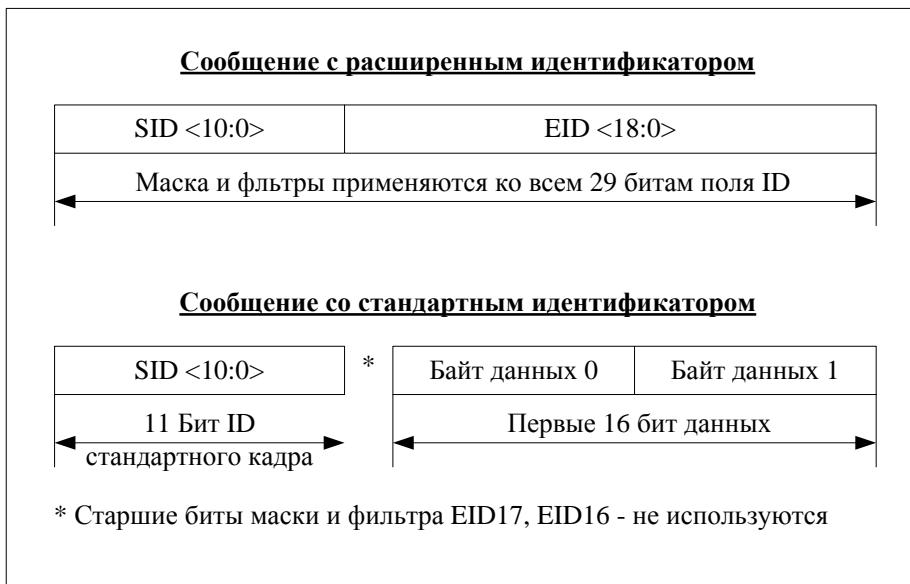


Рисунок 8. Применение фильтра и маски для сообщений.

Важно! Доступ к регистрам фильтров и масок возможен только в режиме конфигурации (см. главу "[Режимы работы контроллера CAN](#)"). В других режимах их изменение невозможно, регистры читаются как все "0".

6.9.1 Регистр: RXFnSIDH

Адрес: 00h, 04h, 08h, 10h, 14h, 18h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
R/W - 'x'							

Описание: Фильтр n стандартный идентификатор, старшая часть:

Номер бита	Обозначение	Описание
7-0	SID	Фильтр стандартного идентификатора, биты с 10 по 3.

6.9.2 Регистр: RXFnSIDL

Адрес: 01h, 05h, 09h, 11h, 15h, 19h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SID2	SID1	SID0	-	EXIDE	-	EID17	EID16
R/W - 'x'							

Описание: Фильтр n стандартный идентификатор, младшая часть:

Номер бита	Обозначение	Описание
7-5	SID	Фильтр стандартного идентификатора, биты с 2 по 0.
4	-	Не используется.
3	EXIDE	Extended Identifier Enable - режим расширенного идентификатора 1 - фильтр применяется только для сообщений с расширенным идентификатором; 0 - фильтр применяется только для сообщений со стандартным идентификатором.
2	-	Не используется.
1-0	EID	Фильтр расширенного идентификатора, биты 17, 16.

6.9.3 Регистр: [RXFnEID8](#)

Адрес: 02h, 06h, 0Ah, 12h, 16h, 1Ah

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
R/W - 'x'							

Описание: Фильтр n расширенного идентификатора, старшая часть:

Номер бита	Обозначение	Описание
7-0	EID	Фильтр расширенного идентификатора, биты 15 - 8. Фильтр применяется для бит 15:8 расширенного идентификатора или байту 0 принятых данных для сообщений со стандартным идентификатором (если соответствующие биты RXBnCTRL.RXM = 00 и RFXnSIDL.EXIDE = 0).

6.9.4 Регистр: [RXFnEID0](#)

Адрес: 03h, 07h, 0Bh, 13h, 17h, 1Bh

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
R/W - 'x'							

Описание: Фильтр n расширенного идентификатора, младшая часть:

Номер бита	Обозначение	Описание
7-0	EID	Фильтр расширенного идентификатора, биты 7 - 0. Фильтр применяется для бит 7:0 расширенного идентификатора или байту 1 принятых данных для сообщений со стандартным идентификатором (если соответствующие биты RXBnCTRL.RXM = 00 и RFXnSIDL.EXIDE = 0).

6.9.5 Регистр: [RXMnSIDH](#)

Адрес: 20h, 24h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
R/W - '0'							

Описание: Маска n стандартного идентификатора, старшая часть:

Номер бита	Обозначение	Описание
7-0	SID	Маска стандартного идентификатора, биты с 10 по 3.

6.9.6 Регистр: [RXMnSIDL](#)

Адрес: 21h, 25h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
SID2	SID1	SID0	-	EXIDE	-	EID17	EID16
R/W - '0'							

Описание: Маска п стандартного идентификатора, младшая часть:

Номер бита	Обозначение	Описание
7-5	SID	Маска стандартного идентификатора, биты с 2 по 0.
4-2	-	Не используется.
1-0	EID	Маска расширенного идентификатора, биты 17, 16.

6.9.7 Регистр: [RXMnEID8](#)

Адрес: 22h, 26h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
R/W - '0'							

Описание: Маска п расширенного идентификатора, старшая часть:

Номер бита	Обозначение	Описание
7-0	EID	Маска расширенного идентификатора, биты 15 - 8. Маска для фильтров применяется для бит 15:8 расширенного идентификатора или байту 0 принятых данных для сообщений со стандартным идентификатором (если соответствующие биты <u>RXBnCTRL.RXM</u> = 00 и <u>RFXnSIDL.EXIDE</u> = 0).

6.9.8 Регистр: [RXMnEID0](#)

Адрес: 23h, 27h

Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
R/W - '0'							

Описание: Маска п расширенного идентификатора, младшая часть:

Номер бита	Обозначение	Описание
7-0	EID	Маска расширенного идентификатора, биты 7 - 0. Маска для фильтров применяется для бит 7:0 расширенного идентификатора или байту 1 принятых данных для сообщений со стандартным идентификатором (если соответствующие биты <u>RXBnCTRL.RXM</u> = 00 и <u>RFXnSIDL.EXIDE</u> = 0).

6.10 Автоматическое зеркалирование регистров CAN.

Зеркалирование позволяет максимально быстро прочитать регистры статуса контроллеров CAN при возникновении прерывания, минимизировать реакцию процессора на событии на CAN шинах и упростить взаимодействие процессора и CAN контроллеров.

Для корректной работы зеркалирования необходимо:

1. Запретить прерывание контроллера INTERRUPT MASK.INTCANx.
2. Разрешить необходимые прерывания в регистре INTERRUPT MASK.
3. Разрешить автоматическое зеркалирование регистров в регистре CANx_CTRL.
4. Разрешить нужные прерывания контроллера CAN в регистре CANINTE.

Если включен бит CAN_CTRL.MIRR_ON, по прерыванию контроллера CAN будет прочитаны регистры CANINTE и EFLG, установлено прерывание в регистре INTERRUPT.

Если включен бит CAN_CTRL.MIRR_ERR или CAN_CTRL.MIRR_MERR, дополнительно будут прочитаны счётчики ошибок шины REC и TEC. Содержимое этих регистров будет отображено в регистре CANx_MIRR_ERR.

Если включен бит CAN_CTRL.MIRR_TXB, дополнительно будут прочитаны регистры TXBxCTRL и скопированы в регистр CANx_MIRR_TXB.

6.10.1 Регистр: CANx_MIRR_ERR

Адрес: 1048h, 1058h

Номер бита																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Описание: Регистр-копия регистров EFLG, REC, TEC контроллера CANx.

Номер бита	Обозначение	Описание
31-24	RSRV	Резерв, не используется.
23-16	TEC	Копия регистра TEC.
15-8	REC	Копия регистра REC.
7-0	EFLG	Копия регистра EFLG.

Данные регистра валидны только в случае, если установлены биты 0 и 1 регистра CANx_CTRL и установлены биты 16,18 (22,24) регистра INTERRUPT – копия бит регистра CANINTE.

6.10.2 Регистр: CANx_MIRR_TXB

Адрес: 104Ch, 105Ch

Номер бита																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-	-	-	-	-	-	-	-	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Описание: Регистр-копия регистров TXBnCTRL контроллера CANx.

Номер бита	Обозначение	Описание
31-24	RSRV	Резерв, не используется.
23-16	TXB2CTRL	Копия регистра TXB2CTRL.
15-8	TXB1CTRL	Копия регистра TXB1CTRL.
7-0	TXB0CTRL	Копия регистра TXB0CTRL.

Данные регистра валидны только в случае, если установлены биты 0 и 2 регистра CANx_CTRL

6.11 Статистика работы модуля.

Модуль имеет блок из 8 32-битных счётчиков статистики событий за всё время работы.

Сброс всех счётчиков производится записью в регистр CANx_STAT.CNT_TX.

6.11.1 Регистр: [CANx_STAT](#)

Адрес: 2100h, 2200h

Адрес	Обозначение	Описание
2100 2200	CNT_TX	Счётчик переданных сообщений.
2104 2204	CNT_RX	Счётчик принятых сообщений.
2108 2208	CNT_ERR_TX	Счётчик ошибок передачи на шине (количество инкрементов счётчика TEC).
210C 220C	CNT_ERR_RX	Счётчик ошибок приёма на шине (количество инкрементов счётчика REC).
2110 2210	CNT_ERR_OVF	Счётчик ошибок переполнения буфера передачи.
2114 2214	CNT_ERR_ABT	Счётчик ошибок передачи по таймауту.
2118 2218	CNT_ERR_BOFF	Счётчик переходов контроллера CAN-шины в состояние Bus Off..
211C 221C	CNT_ERR_EP	Счётчик переходов контроллера CAN-шины в состояние Error Passive.
2120 2220	CNT_ERR_OVF_RX	Счётчик ошибок переполнения буфера приёма.
2124 2224	CNT_ERR_WRN	Счётчик состояний предупреждения о высоком количестве ошибок линии, при котором REC или TEC становятся ≥ 96 .

Список исправлений и изменений.

Версия	Дата	Изменение
1.00	09.06.2015	1. Документ создан.
1.01	15.09.2015	2. Исправлено описание битов QDAT регистров INTERRUPT INTERRUPT_MASK . 3. Добавлено описание резервных битов регистра CAN_CTRL .
1.2	17.11.2016	4. Уточнен рабочий температурный диапазон в главе 1 .
1.3	20.03.2017	5. Переработан раздел 6.5 Конфигурация скорости шины CAN .
1.4	29.03.2018	6. Дополнено описание работы таймеров в главе 5.2 Таймеры локального времени и 5.3 Триггеры буферов отправки сообщений .
2.0	10.09.2018	7. Дополнено описанием функций Firmware v.02.
2.02	04.03.2020	8. Исправлены поля таблицы TIMEMARK .
2.10	20.05.2020	9. Дополнено описанием режима передачи «FIFO» . 10. Описание таблицы TIMEMARK дополнено битом <code>Msg_Abored_Timeout</code> .
2.11	27.05.2020	11. Добавлено описание FIFO триггеров . 12. Добавлено описание флага TX_PAUSE и бита сброса FIFO. 13. Добавлено описание счётчиков статистики .
2.12	17.07.2020	14. Незначительные корректировки.
2.13	04.07.2023	15. Исправлены ошибки в главе 6.5 Конфигурация скорости шины CAN .