



## **Руководство (v3.0)**

# **По работе с библиотекой модулей “mPCIe – TTCAN”, “PCIe – TTCAN”**

Интерфейс ISO-11898-4  
(TTCAN)

Для библиотек версии 3.x

**ОС WINDOWS**



**22.01.2021**

**ООО “НОВОМАР”**

## Оглавление

1. Назначение программы .....	4
2. Использование библиотеки.....	4
3. Список доступных функций .....	4
4. Описание методов библиотеки.....	6
4.1. TTCAN_enumerate .....	6
4.2. TTCAN_enumerate_free.....	6
4.3. TTCAN_open_device .....	6
4.4. TTCAN_d_enable .....	6
4.5. TTCAN_reset_device.....	7
4.6. TTCAN_reset_channel.....	7
4.7. TTCAN_read_bar.....	7
4.8. TTCAN_write_bar .....	7
4.9. TTCAN_read_can .....	8
4.10. TTCAN_write_can.....	8
4.11. TTCAN_modify_can.....	8
4.12. TTCAN_read_dma.....	9
4.13. TTCAN_device_version .....	9
4.14. TTCAN_driver_version .....	9
4.15. TTCAN_write_txbuf.....	10
4.16. TTCAN_send_data_now.....	10
4.17. TTCAN_check_tx .....	10
4.18. TTCAN_wait_tx.....	11
4.19. TTCAN_remove_txreq .....	11
4.20. TTCAN_abat .....	11
4.21. TTCAN_send_by_trigger.....	12
4.22. TTCAN_check_trigger.....	12
4.23. TTCAN_set_mode .....	13
4.24. TTCAN_get_mode.....	13
4.25. TTCAN_set_speed .....	13
4.26. TTCAN_set_speed_params.....	14
4.27. TTCAN_get_speed.....	14
4.28. TTCAN_dma_enable .....	15
4.29. TTCAN_dma_enable .....	15
4.30. TTCAN_set_oneshot.....	15

4.31. TTCAN_get_errors .....	15
4.32. TTCAN_set_masks .....	16
4.33. TTCAN_set_timer_trsh.....	17
4.34. TTCAN_set_timer_ceed .....	17
4.35. TTCAN_set_timer_free.....	18
4.36. TTCAN_set_timer_rst_rxb .....	18
4.37. TTCAN_stop_timer.....	18
4.38. TTCAN_get_timer .....	19
4.39. TTCAN_start_timer_int .....	19
4.40. TTCAN_stop_timer_int .....	19
4.41 TTCAN_wait_timer_int.....	20
4.42 TTCAN_set_timeouts .....	20
4.43 TTCAN_set_send_mode .....	21
4.44 TTCAN_get_fifo_count.....	21
4.45 TTCAN_write_data_to_fifo.....	21
4.46 TTACN_set_tx_pause.....	22
4.47 TTCAN_write_tg_fifo.....	22
4.48 TTCAN_decode_buf.....	22
5. Обновление руководства.....	23

## 1. Назначение программы

Программное обеспечение «Библиотека взаимодействия TTCAN» (далее – библиотека) обеспечивает вспомогательный сервисный функционал при взаимодействии с PCI-устройством TTCAN.

Библиотека обеспечивает выполнение следующих основных задач:

- поиск присутствующих в системе устройств
- реализация сервисных функций поканально

## 2. Использование библиотеки

Для получения доступа к функциям библиотеки в разрабатываемый проект необходимо подключить заголовочные файлы "*libnmttc.h*", "*ttcan.h*", "*stdafx.h*". Далее, в настройках сборщика указать путь к файлу библиотеки "*libnmttc.lib*"

## 3. Список доступных функций

TTCAN_enumerate	Получение списка устройств
TTCAN_enumerate_free	Освобождение памяти занятой списком устройств
TTCAN_open_device	Открытие устройства
TTCAN_read_bar	Запись в адресное пространство основного контроллера
TTCAN_write_bar	Чтение из адресного пространства основного контроллера
TTCAN_read_can	Запись в адресное пространство контроллера TTCAN
TTCAN_write_can	Чтение из адресного пространства контроллера TTCAN
TTCAN_modify_can	Модификация бит в адресном пространстве контроллера TTCAN
TTCAN_d_enable	Инициализация платы в отладочном режиме
TTCAN_device_version	Получение версии устройства
TTCAN_driver_version	Получение версии драйвера
TTCAN_read_dma	Чтение из буфера DMA
TTCAN_write_txbuf	Запись в буфер отправки
TTCAN_send_data_now	Отправка данных из буфера
TTCAN_check_tx	Проверка состояния отправки
TTCAN_wait_tx	Ожидание отправки
TTCAN_remove_txreq	Очистка буфера отправки
TTCAN_abat	Отмена всех ожидающих транзакций
TTCAN_send_by_trigger	Установка триггера отправки
TTCAN_check_trigger	Проверка триггера отправки
TTCAN_reset_device	Сброс основного контроллера

TTCAN_reset_channel	Сброс контроллера TTCAN
TTCAN_set_mode	Установка режима работы контроллера TTCAN
TTCAN_get_mode	Получение режима работы контроллера TTCAN
TTCAN_set_speed	Установка скорости работы
TTCAN_set_speed_params	Установка произвольной скорости работы
TTCAN_get_speed	Получение скорости работы
TTCAN_dma_enable	Включение DMA
TTCAN_dma_disable	Отключение DMA
TTCAN_set_oneshot	Установка режима однократной отправки
TTCAN_get_errors	Получение данных об ошибках
TTCAN_set_masks	Установка масок для получения
TTCAN_set_timer_trsh	Запуск таймера с указанным периодом.
TTCAN_set_timer_ceed	Установка режима и значений корректировки начального значения таймера.
TTCAN_set_timer_free	Установка таймера в режиме свободного счета
TTCAN_set_timer_rst_rxb	Установка/снятие режима сброса таймера по приёму сообщения в буфер CAN контроллера
TTCAN_stop_timer	Остановка таймера
TTCAN_get_timer	Получение значения таймера
TTCAN_start_timer_int	Установка таймера прерываний
TTCAN_stop_timer_int	Остановка таймера прерываний
TTCAN_wait_timer_int	Ожидание таймера прерываний
TTCAN_set_send_mode	Установка режима работы контроллера CAN (Native/FIFO)
TTCAN_get_fifo_count	Получение количества пакетов в буфере FIFO
TTCAN_write_data_to_fifo	Запись данных в буфер FIFO
TTCAN_set_tx_pause	Управление флагом TXPAUSE для режима FIFO
TTCAN_write_tg_fifo	Запись в FIFO триггеров
TTCAN_decode_buf	Декодирование пакета DMA

## 4. Описание методов библиотеки

Все методы, если не указано иное, возвращают код ошибки обращения к драйверу или 0 в случае успеха.

### 4.1. TTCAN\_enumerate

Получение списка устройств TTCAN

UINT8 TTCAN\_enumerate(TTCAN\_device\_info\*\* devices)

TTCAN\_device\_info\*\* devices - указатель, в который будет записан список устройств

Возвращает количество доступных устройств TTCAN

### 4.2. TTCAN\_enumerate\_free

Освобождение памяти занятой списком устройств

VOID TTCAN\_enumerate\_free(TTCAN\_device\_info\*\* devices)

TTCAN\_device\_info\*\* devices - указатель на список устройств

### 4.3. TTCAN\_open\_device

Открытие устройства

HANDLE TTCAN\_open\_device(TTCAN\_device\_info\* ttcn)

TTCAN\_device\_info\* ttcn - указатель на структуру данных об устройстве

Возвращает хэндл устройства или NULL

### 4.4. TTCAN\_d\_enable

Инициализация устройства в режиме отладки

VOID TTCAN\_d\_enable(HANDLE\* ttcn)

HANDLE\* ttcn - хэндл устройства

#### 4.5. TTCAN\_reset\_device

Сброс основного контроллера

UINT32 TTCAN\_reset\_device(HANDLE\* tcan)

HANDLE\* tcan - хэндл устройства

#### 4.6. TTCAN\_reset\_channel

Сброс одного из контроллеров TTCAN

UINT32 TTCAN\_reset\_channel(HANDLE\* tcan, UINT8 channel)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер канала

#### 4.7. TTCAN\_read\_bar

Чтение данных в адресном пространстве основного контроллера

UINT32 TTCAN\_read\_bar(HANDLE\* tcan, UINT32 bar\_addr, UINT32\* pbuf)

HANDLE\* tcan - хэндл устройства

UINT32 bar\_addr - адрес в памяти контроллера

UINT32\* pbuf - указатель по которому будет записано считанное значение

#### 4.8. TTCAN\_write\_bar

Запись данных в адресном пространстве основного контроллера

UINT32 TTCAN\_write\_bar(HANDLE\* tcan, UINT32 bar\_addr, UINT32\* pbuf)

HANDLE\* tcan - хэндл устройства

UINT32 bar\_addr - адрес в памяти контроллера

UINT32\* pbuf - указатель на значение для записи

#### 4.9. TTCAN\_read\_can

Чтение данных из адресного пространства контроллера TTCAN

UINT32 TTCAN\_read\_can(HANDLE\* tcan, UINT32 can\_addr, UINT8 channel, UINT8\* pbuf, UINT8 rsize)

HANDLE\* tcan - хэндл устройства

UINT32 can\_addr - адрес первого байта в памяти контроллера

UINT8 channel - номер контроллера

UINT8\* pbuf - указатель на область памяти, в которую будут прочитаны данные

UINT8 rsize - количество байт для чтения

Память под pbuf должна быть выделена заранее

#### 4.10. TTCAN\_write\_can

Запись данных в адресное пространство контроллера TTCAN

UINT32 TTCAN\_write\_can(HANDLE\* tcan, UINT32 can\_addr, UINT8 channel, UINT8\* pbuf, UINT8 rsize)

HANDLE\* tcan - хэндл устройства

UINT32 can\_addr - адрес первого байта в памяти контроллера

UINT8 channel - номер контроллера

UINT8\* pbuf - указатель на массив данных для записи

UINT8 rsize - количество байт для чтения

Память под pbuf должна быть выделена заранее

#### 4.11. TTCAN\_modify\_can

Изменение бит в памяти контроллера TTCAN

UINT32 TTCAN\_modify\_can(HANDLE\* tcan, UINT32 can\_addr, UINT8 channel, UINT8 data, UINT8 mask)

HANDLE\* tcan - хэндл устройства

UINT32 can\_addr - адрес в памяти контроллера

UINT8 channel - номер контроллера

UINT8 data - данные для записи

UINT8 mask - маска



## 4.12. TTCAN\_read\_dma

Чтение из буфера DMA

UINT32 TTCAN\_read\_dma(HANDLE\* tcan, UINT8 channel, UINT32 count, DMA\_SLOT\_TTCAN\* dmabuf, UINT32 timeout)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT32 count - количество блоков для чтения

DMA\_SLOT\_TTCAN\* dmabuf - указатель на структуру данных, в которую будут записаны блоки данных

UINT32 timeout - таймаут ожидания

Память под dmabuf должна быть выделена заранее

## 4.13. TTCAN\_device\_version

Получение версии устройства

UINT32 TTCAN\_device\_version(HANDLE\* tcan, TTCAN\_DEVINFO\* pdevinfo)

HANDLE\* tcan - хэндл устройства

TTCAN\_DEVINFO\* pdevinfo - структура данных с версией устройства

Память под pdevinfo должна быть выделена заранее

## 4.14. TTCAN\_driver\_version

Получение версии драйвера

UINT32 TTCAN\_driver\_version(HANDLE\* tcan, TTCAN\_DRVINFO\* pdrvinfo)

HANDLE\* tcan - хэндл устройства

TTCAN\_DRVINFO\* pdrvinfo - структура данных с версией драйвера

Память под pdrvinfo должна быть выделена заранее

#### 4.15. TTCAN\_write\_txbuf

Запись в буфер отправки

UINT32 TTCAN\_write\_txbuf(HANDLE\* tcan, UINT8 channel, UINT8 nbuf, UINT8 prio, UINT32 timeout, UINT32 sid, UINT32 eid, UINT8\* pdata, UINT32 datasize, BOOLEAN autorts)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 nbuf - номер буфера

UINT8 prio - приоритет

UINT32 timeout - таймаут отправки

UINT32 sid - SID

UINT32 eid - EID

UINT8\* pdata - указатель на буфер с данными для отправки

UINT32 datasize - размер буфера с данными

BOOLEAN autorts - автоматическая отправка после записи

#### 4.16. TTCAN\_send\_data\_now

Отправка записанных в буфер данных

UINT32 TTCAN\_send\_data\_now(HANDLE\* tcan, UINT8 channel, UINT8 nbuf)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 nbuf - номер буфера

#### 4.17. TTCAN\_check\_tx

Проверка состояния отправки

UINT32 TTCAN\_check\_tx(HANDLE\* tcan, UINT8 channel, UINT8 nbuf, UINT8 \*txbnctrl)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 nbuf - номер буфера

UINT8 \*txbnctrl - указатель на адрес куда будет записан регистр txbnctrl (см. документацию по программированию платы)

#### 4.18. TTCAN\_wait\_tx

Ожидание отправки данных

UINT32 TTCAN\_wait\_tx(HANDLE\* tcan, UINT8 channel, UINT8 nbuf, UINT32 timeout)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 nbuf - номер буфера

UINT32 timeout - таймаут ожидания

#### 4.19. TTCAN\_remove\_txreq

Отмена запроса на отправку

UINT32 TTCAN\_remove\_txreq(HANDLE\* tcan, UINT8 channel, UINT8 nbuf)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 nbuf - номер буфера

#### 4.20. TTCAN\_abat

Отмена всех транзакций

UINT32 TTCAN\_abat(HANDLE\* tcan, UINT8 channel)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

#### 4.21. TTCAN\_send\_by\_trigger

Программирование отправки по срабатыванию триггера

UINT32 TTCAN\_send\_by\_trigger(HANDLE\* tcan, UINT8 channel, UINT8 buffer, UINT32 epoch, UINT32 epoch\_val, UINT32 ntu, UINT32 div, BOOL loop)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 buffer - номер буфера

UINT32 epoch - флаг использования epoch

UINT32 epoch\_val - счётчик циклов

UINT32 ntu

UINT32 div

BOOLEAN loop - программирование многократной передачи

#### 4.22. TTCAN\_check\_trigger

Получение состояния триггера

UINT32 TTCAN\_check\_trigger(HANDLE\* tcan, UINT8 channel, UINT8 buffer)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 buffer - номер буфера

Функция читает регистр CANn\_TRIG\_CTRL и проверяет биты TX\_TRIGm\_EN.

Если биты равны 01, то триггер занят, функция вернёт значение 1.

Если биты равны 00, то триггер свободен и функция вернёт значение 0

#### 4.23. TTCAN\_set\_mode

Установка режима работы контроллера

UINT32 TTCAN\_set\_mode(HANDLE\* tcan, UINT8 channel, CAN\_MODE mode)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

CAN\_MODE mode - режим работы

#### 4.24. TTCAN\_get\_mode

Получение режима работы контроллера

UINT32 TTCAN\_get\_mode(HANDLE\* tcan, UINT8 channel, CAN\_MODE \*mode)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

CAN\_MODE \*mode - указатель для записи CAN\_MODE

Память под mode должна быть выделена заранее

#### 4.25. TTCAN\_set\_speed

Установка стандартной скорости работы контроллера TTCAN

UINT32 TTCAN\_set\_speed(HANDLE\* tcan, UINT8 channel, UINT32 speed)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT32 speed - скорость

#### 4.26. TTCAN\_set\_speed\_params

Установка произвольной скорости работы контроллера TTCAN

UINT32 TTCAN\_set\_speed\_params(HANDLE\* ttcан, UINT8 channel, UINT32 \*speed, UINT8 \*params)

HANDLE\* ttcан - хэндл устройства

UINT8 channel - номер контроллера

UINT32 \*speed - не используется

UINT8 \*params - параметры скорости контроллера (см. документацию по программированию модуля)

#### 4.27. TTCAN\_get\_speed

Получение скорости работы контроллера

UINT32 TTCAN\_get\_speed(HANDLE\* ttcан, UINT8 channel, UINT32 \*speed, UINT8 \*params)

HANDLE\* ttcан - хэндл устройства

UINT8 channel - номер контроллера

UINT32 \*speed - указатель для записи значения стандартной скорости

UINT8 \*params - указатель для записи значения параметров скорости

#### 4.28. TTCAN\_dma\_enable

Включение DMA

UINT32 TTCAN\_dma\_enable(HANDLE\* ttcан)

HANDLE\* ttcан - хэндл устройства

#### 4.29. TTCAN\_dma\_disable

Выключение DMA

UINT32 TTCAN\_dma\_disable (HANDLE\* ttcан)

HANDLE\* ttcан - хэндл устройства

#### 4.30. TTCAN\_set\_oneshot

Установка режима однократной передачи

UINT32 TTCAN\_set\_oneshot(HANDLE\* ttcан, UINT8 channel, UINT8 oneshot)

HANDLE\* ttcан - хэндл устройства

UINT8 channel - номер контроллера

UINT8 oneshot - 1 для включения или 0 для выключения режима однократной передачи

#### 4.31. TTCAN\_get\_errors

Чтение регистров ошибок TTCAN контроллера

UINT32 TTCAN\_get\_errors(HANDLE\* ttcан, UINT8 channel, CAN\_ERROR\_INFO \*errorinfo)

HANDLE\* ttcан - хэндл устройства

UINT8 channel - номер контроллера

CAN\_ERROR\_INFO \*errorinfo - указатель на структуру, в которую будут записаны регистры ошибок

Память под errorinfo должна быть выделена заранее

## 4.32. TTCAN\_set\_masks

Установка масок и фильтров приёма TTCAN контроллера.

UINT32 TTCAN\_set\_masks(HANDLE\* tcan, UINT8 channel, UINT8 filter, UINT8 ident, UINT32 id\_filter, UINT32 id\_mask, UINT8 rxb)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 filter -

UINT8 ident -

UINT32 id\_filter -

UINT32 id\_mask -

UINT8 rxb -

В зависимости от выбранного канала (**channel**) и номера фильтра (**filter**) функция записывает значение **rxb\_mode** в биты RXM регистра **RXBn\*CTRL\*\***.

\* n – номер буфера.

\*\* См. раздел 6.8.1 и 6.8.2 документа “Руководство по программированию модуля “mPCIe-TTCAN””.

Далее в зависимости от выбранного идентификатора (**ident**) и номера фильтра (**filter**) функция записывает значения масок и фильтров в соответствующие регистры\*.

\* См. раздел 6.9 документа “Руководство по программированию модуля “mPCIe-TTCAN””.



### 4.33. TTCAN\_set\_timer\_trsh

Запуск таймера с указанным периодом.

UINT32 TTCAN\_set\_timer\_trsh(HANDLE\* ttcан, UINT8 channel, UINT32 nValue, UINT8 bEpoch, UINT32 nEpoch)

HANDLE\* ttcан - хэндл устройства

UINT8 channel - номер контроллера

UINT32 nValue - значение таймера

UINT8 bEpoch - битовая маска

UINT32 nEpoch - не используется

См. раздел 5.2.2 документа “Руководство по программированию модуля “mPCIE-TTCAN”.

См. раздел 5.2.4 документа “Руководство по программированию модуля “mPCIE-TTCAN”.

### 4.34. TTCAN\_set\_timer\_ceed

Установка режима и значений корректировки начального значения таймера.

Значение поля **nValue** записывается в регистр **CANn\*\_TIMER\_TRSH\*\*** (значение записывается целиком с нулевого бита).

Биты RST, ENABLE и NTU\_MODE регистра **CANn\*\_TIMER\_CTRL\*\*\*** устанавливаются в единицу, а битовая маска EPOCH\_MASK устанавливается в соответствии с параметром **bEpoch**. Значения остальных битов данного регистра не изменяются.

UINT32 TTCAN\_set\_timer\_ceed(HANDLE\* ttcан, UINT8 channel, UINT32 nValue, UINT8 bEpoch, UINT32 nEpoch)

HANDLE\* ttcан - хэндл устройства

UINT8 channel - номер контроллера

UINT32 nValue

UINT8 bEpoch

UINT32 nEpoch

### 4.35. TTCAN\_set\_timer\_free

Запуск таймера в режиме свободного счёта.

UINT32 TTCAN\_set\_timer\_free(HANDLE\* tcan, UINT8 channel, UINT8 epochBits)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8epochBits - битовая маска

### 4.36. TTCAN\_set\_timer\_rst\_rxb

Установка/снятие режима сброса таймера по приёму сообщения в буфер CAN контроллера.

UINT32 TTCAN\_set\_timer\_rst\_rxb(HANDLE\* tcan, UINT8 channel, UINT32 nValue)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT32 nValue

В регистре **CANn\*\_TIMER\_CTRL\*\*** бит RST\_ON\_RXB0 устанавливается в соответствии со значением бита 0 параметра **nValue**, а бит RST\_ON\_RXB1 устанавливается в соответствии со значением бита 1 параметра **nValue**. Значения остальных битов данного регистра не изменяются.

\* n – номер канала.

\*\* См. раздел 5.2.4 документа “Руководство по программированию модуля “mPCIe-TTCAN”.

### 4.37. TTCAN\_stop\_timer

Остановка таймера

UINT32 TTCAN\_stop\_timer(HANDLE\* tcan, UINT8 channel)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

#### 4.38. TTCAN\_get\_timer

Получение значения таймера

UINT32 TTCAN\_get\_timer(HANDLE\* tcan, UINT8 channel, UINT32 \*value, UINT32 \*epoch)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT32 \*value - значение регистра CANn\_TIMER

UINT32 \*epoch - значение регистра CANn\_TIMER\_EPOCH

Память под value и epoch должна быть выделена заранее

#### 4.39. TTCAN\_start\_timer\_int

Запуск таймера прерываний

см. описание IOCTL **4.3.18 IOCTL\_START\_TIMER\_INT** в руководстве по работе с драйвером модулей TTCAN

UINT32 TTCAN\_start\_timer\_int(HANDLE\* tcan, UINT8 channel, UINT8 bEpoch, UINT32 nValue)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 bEpoch

UINT32 nValue

#### 4.40. TTCAN\_stop\_timer\_int

Остановка таймера прерываний

UINT32 TTCAN\_stop\_timer\_int(HANDLE\* tcan, UINT8 channel)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

#### 4.41 TTCAN\_wait\_timer\_int

Ожидание таймера прерываний

UINT32 TTCAN\_wait\_timer\_int(HANDLE\* tcan, UINT8 channel)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

#### 4.42 TTCAN\_set\_timeouts

Установка таймаутов

UINT32 TTCAN\_set\_timeouts(HANDLE\* tcan, UINT8 channel, UINT32 absolute, UINT32 interval)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT32 absolute

UINT32 interval

Поле **absolute** пишется в регистр **CANn\*\_TIMEOUT\_ABSOLUTE\*\***, а поле **interval** пишется в регистр **CANn\*\_TIMEOUT\_INTERVAL\*\*\***.

Значение обоих полей в микросекундах, допустимые значения - 0x0 .. 0x3FFFFFF

\* n – номер канала.

\*\* См. раздел 5.1.5 документа “Руководство по программированию модуля “mPCIE-TTCAN”.

\*\*\* См. раздел 5.1.6 документа “Руководство по программированию модуля “mPCIE-TTCAN”.

#### 4.43 TTCAN\_set\_send\_mode

Установка режима контроллера CAN

UINT32 TTCAN\_set\_send\_mode(HANDLE\* tcan, UINT8 channel, CTRL\_MODE mode)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

CTRL\_MODE mode - режим контроллера

Возможны 4 режима - CAN\_NATIVE (по умолчанию), CAN\_FIFO, CAN\_TIMEPLAN, CAN\_RSRV

В текущей ревизии драйвера поддерживаются режимы CAN\_NATIVE, CAN\_FIFO

#### 4.44 TTCAN\_get\_fifo\_count

Получение количества пакетов в буфере FIFO

UINT32 TTCAN\_get\_fifo\_count(HANDLE\* tcan, UINT8 channel, UINT8 isHPFIFO, UINT8 \*count)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 isHPFIFO - 1 - использовать буфер HPFIFO (иначе - FIFO)

UINT8\* count - указатель на переменную куда будет записано количество пакетов

#### 4.45 TTCAN\_write\_data\_to\_fifo

Запись данных в буфер FIFO

UINT32 TTCAN\_write\_data\_to\_fifo(HANDLE\* tcan, UINT8 channel, UINT32 sid, UINT32 eid, UINT8 dlc, UINT8 \*data, UINT32 dsize, UINT8 msgid, UINT8 isHPFIFO)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT32 sid - стандартный идентификатор

UINT32 eid - расширенный идентификатор

UINT8 \*data - указатель на массив с данными

UINT32 size - размер массива данных

UINT8 msgid - id сообщения

UINT8 isHPFIFO - 1 - использовать буфер HPFIFO (иначе - FIFO)

#### 4.46 TTCAN\_set\_tx\_pause

Управление флагом TXPAUSE для FIFO

UINT32 TTCAN\_set\_tx\_pause(HANDLE\* tcan, UINT8 channel, UINT32 txpause, UINT8 isHPFIFO)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT32 txpause - новое содержимое регистра

UINT8 isHPFIFO - 1 - использовать буфер HPFIFO (иначе - FIFO)

#### 4.47 TTCAN\_write\_tg\_fifo

Управление триггерами FIFO

UINT32 TTCAN\_write\_tg\_fifo(HANDLE\* tcan, UINT8 channel, UINT8 epoch, UINT32 div\_trig, UINT32 ntu\_trig)

HANDLE\* tcan - хэндл устройства

UINT8 channel - номер контроллера

UINT8 epoch

UINT32 div\_trig

UINT32 ntu\_trig

#### 4.48 TTCAN\_decode\_buf

Декодирование буфера с данными

void TTCAN\_decode\_buf(DMA\_SLOT\_TTCAN\* dma, UINT32\* nbuf, UINT8\* channel, UINT32\* tmr\_epoch, UINT32\* tmr\_ntu, UINT32\* tmr\_div, UINT32\* sid, UINT32\* eid, UINT8\* data, UINT8\* datasize)

## 5. Обновление руководства.

<b>Версия документа</b>	<b>Дата</b>	<b>Изменение</b>
1.0	18.09.2015	Документ создан
2.0	31.05.2018	Обновление до версии драйвера 2.0.
3.0	22.01.2021	Добавлен режим FIFO