

**Ошибки в микросхемах MDR1206FI, MDR1206AFI,  
MDR1215LGI, MDR1219LGI**

Настоящий документ содержит описание всех ошибок, выявленных в микросхемах на момент создания данной версии документа.

## **Статус документа**

Настоящий документ является НЕКОНФИДЕНЦИАЛЬНЫМ.

## **Адрес в сети Интернет**

<http://www.milandr.ru>

## **Обратная связь по продукту**

Если у Вас есть какие-либо комментарии или предложения по данному продукту, свяжитесь с Вашим поставщиком, указав:

- название продукта;
- комментарии, либо краткое описание Ваших предложений;
- предпочтительный способ связи с Вами и контакты (организация, электронная почта, номер телефона).

## **Обратная связь по документу**

Если у Вас есть какие-либо комментарии или предложения по данному документу, пожалуйста, пришлите их на электронную почту [support@milandr.ru](mailto:support@milandr.ru), указав:

- название документа;
- номер и/или дата документа;
- номер страницы;
- комментарии, либо краткое описание Ваших предложений;
- предпочтительный способ связи с Вами и контакты (организация, электронная почта, номер телефона).

## Оглавление

Обзор.....	4
Категории ошибок .....	4
Сводная таблица ошибок .....	5
Ошибки категории 3 .....	7
0001 Пропуск канала SAR АЦП при последовательном преобразовании нескольких каналов после выключения АЦП.....	7
0002 Не запуск вспомогательного LDO батарейного домена.....	8
0003 Прекращение обработки запросов прерываний в режиме отладки при исполнении по шагам и наличии активного векторного прерывания .....	9
0004 Команда dret не декодируется как illegal instruction при исполнении не в Debug-режиме .....	10
0005 Некорректная работа аппаратной точки останова на коде в Debug ROM .....	10
0006 Ошибка в tinfo-регистре модуля отладки .....	11
0007 Команда s.lwsp (если rd == x0) не декодируется как illegal instruction.....	11
0008 Чтение CSR-регистров командами csrrs, csrrsi и csrrci вызывает запись .....	12
0009 Некорректное значение по сбросу поля sbaccess регистра sbcs .....	13
0010 Остановка ядра при обработке не векторных прерываний по уровню в случае, если уровень изменяется в неактивный до завершения взятия прерывания.....	14
0011 Остановка счета IWDG при отсутствии частоты PCLK в процессе перезагрузки значения таймера или обновления значения предделителя .....	15
0012 Состояние вывода JTAG TDO (PA7) не определено после подачи питания .....	16
0013 Пропуск импульса SEC_CLK при использовании калибровки RTC_CLK .....	17
0014 "Залипание" флага HSE_RDY после пропадания частоты на OCS_IN и сброса не по питанию.....	19
0015 Контроллер интерфейса I2C не отслеживает clock stretching при формировании START и DIV=1-2 .....	20
Лист регистрации изменений .....	21

## Обзор

Настоящий документ содержит описание ошибок в продукте с указанием категории критичности. Каждое описание содержит:

- уникальный идентификатор ошибки;
- текущий статус ошибки;
- где существует отклонение от спецификации и условия, при которых возникает ошибка;
- последствия возникновения ошибки в типичных применениях;
- ограничения, рекомендации и способы обхода ошибки, где это возможно.

## Категории ошибок

Ошибки разделены на три категории критичности:

Категория 1.

Ошибочное поведение, которое невозможно обойти. Ошибки данной категории серьезно ограничивают использование продукта во всех или в большинстве приложений, что делает устройство непригодным для использования.

Категория 2.

Ошибочное поведение, которое противоречит требуемому поведению. Ошибки данной категории могут ограничивать или серьезно ухудшать целевое использование указанных функций, но не делают продукт непригодным для использования во всех или в большинстве приложений.

Категория 3.

Ошибочное поведение, которое не было изначально определено, но не вызывает проблем в приложениях при соблюдении рекомендаций.

## Сводная таблица ошибок

В таблице указано, в каких версиях микросхем присутствует ошибка. Наличие ошибки обозначено символом «X».

Версия микросхем определяется датой изготовления, указанной на корпусе микросхемы в формате ГГНН, где ГГ – год изготовления, НН – неделя изготовления.

ID	Описание	Микросхемы, изготавливаемые с даты							
		MDR1206FI		MDR1206AFI		MDR1215LGI		MDR1219LGI	
		2345 (рев. 1)	2441 (рев. 2)	2335 (рев. 1)		2345 (рев. 1)	2523 (рев. 2)	2607 (рев. 2)	
Категория 1									
Категория 2									
Категория 3									
0001	Пропуск канала SAR АЦП при последовательном преобразовании нескольких каналов после выключения АЦП	X	X	X			X	X	X
0002	Не запуск вспомогательного LDO батарейного домена	X	X	X					
0003	Прекращение обработки запросов прерываний в режиме отладки при выполнении по шагам и наличии активного векторного прерывания	X		X			X		
0004	Команда dret не декодируется как illegal instruction при выполнении не в Debug-режиме	X	X	X			X	X	X
0005	Некорректная работа аппаратной точки останова на коде в Debug ROM	X	X	X			X	X	X
0006	Ошибка в tinfo регистре модуля отладки	X	X	X			X	X	X
0007	Команда c.lwsp (если rd == x0) не декодируется как illegal instruction	X	X	X			X	X	X
0008	Чтение CSR-регистров командами csrrc, csrrsi и csrrci вызывает запись	X	X	X			X	X	X
0009	Некорректное значение по сбросу поля sbaccess регистра sbcs	X	X	X			X	X	X

ID	Описание	Микросхемы, изготавливаемые с даты							
		MDR1206FI		MDR1206AFI		MDR1215LGI		MDR1219LGI	
		2345 (рев. 1)	2441 (рев. 2)	2335 (рев. 1)		2345 (рев. 1)	2523 (рев. 2)	2607 (рев. 2)	
0010	Остановка ядра при обработке не векторных прерываний по уровню в случае, если уровень изменяется в неактивный до завершения взятия прерывания	X	X	X		X	X	X	
0011	Остановка счета IWDG при отсутствии частоты PCLK в процессе перезагрузки значения таймера или обновления значения предделителя	X	X	X		X	X	X	
0012	Состояние вывода JTAG TDO (PA7) не определено после подачи питания	X	X	X		X	X	X	
0013	Пропуск импульса SEC_CLK при использовании калибровки RTC_CLK	X	X	X		X	X	X	
0014	"Залипание" флага HSE_RDY после пропадания частоты на OCS_IN и сброса не по питанию	X	X	X		X	X	X	
0015	Контроллер интерфейса I2C не отслеживает clock stretching при формировании START и DIV=1-2	X	X	X		X	X	X	

## Ошибки категории 3

### *0001 Пропуск канала SAR АЦП при последовательном преобразовании нескольких каналов после выключения АЦП*

#### Статус

Исследование.

#### Описание

В режиме последовательного преобразования нескольких каналов после отключения АЦП (бит Gfg\_REG\_ADON) и при последующем его включении однократно пропускается канал, на котором остановилось преобразование в момент отключения АЦП. Происходит преобразование следующего канала, участвующего в последовательном преобразовании.

#### Условия

Включение АЦП после его отключения в режиме последовательного преобразования нескольких каналов.

#### Последствия

Пропуск преобразования канала АЦП, на котором остановилось преобразование при отключении.

#### Рекомендации и способы обхода

После отключения АЦП при использовании последовательного преобразования нескольких каналов:

- 1 Отключить переключение каналов (бит Cfg\_REG\_CHCH).
- 2 Включить переключение каналов только для канала, на котором остановилось преобразование при отключении, и участвующих в преобразовании каналов с большими номерами (регистр ADC1\_CHSEL).

После включения АЦП:

- 3 Включить переключение для всех требуемых каналов.

## ***0002 Не запуск вспомогательного LDO батарейного домена***

### **Статус**

Исследование.

### **Описание**

Вспомогательный LDO батарейного домена может не запуститься при несоответствии фронта питания  $V_{CCB}$  требованиям спецификации.

### **Условия**

Фронт питания  $V_{CCB}$  более 1 мс или менее 10 мкс.

### **Последствия**

Батарейный домен будет работать, используя основное питание и основной LDO. Но при отключении основного питания ( $V_{CCA}$  и  $V_{CC}$ ) информация из батарейного домена может исчезнуть, несмотря на наличие батарейного питания.

### **Рекомендации и способы обхода**

После подачи напряжения питания на вывод  $V_{CCB}$  необходимо однократно прописать в бит  $BLDO\_BOOST$  регистра  $BKP\_LDO$  значение «1» и через задержку не менее 20 мкс – значение «0».

## ***0003 Прекращение обработки запросов прерываний в режиме отладки при исполнении по шагам и наличии активного векторного прерывания***

### **Статус**

Исправлено в ревизии 2.

### **Описание**

В режиме отладки при исполнении по шагам (команда `stepi`) наличие активного векторного прерывания вызывает некорректное изменение контрольно-статусных регистров CSR (`xstatus`, `херс`, `хcause`, `хtval`, `хintstatus`). Некорректное изменение `xstatus` приводит к тому, что при дальнейшем исполнении программы запросы прерываний не обрабатываются ядром.

### **Условия**

Наличие активного векторного прерывания в момент выполнения шага (команда `stepi`) в режиме отладки.

### **Последствия**

Некорректное изменение контрольно-статусных регистров CSR (`xstatus`, `херс`, `хcause`, `хtval`, `хintstatus`), которое приводит к невозможности взятия прерывания после `continue` (команда режима отладки).

### **Рекомендации и способы обхода**

В версии программы, предназначенной для отладки, реализовать один из способов обхода ниже.

Способ 1. Установить для всех используемых прерываний невекторный режим работы: в регистре `clicintattr[i]` установить бит `shv` равным «0».

Способ 2. Установить для всех используемых векторных прерываний, кроме прерываний от DMA и `EXT_INTx`, режим захвата запросов прерываний по уровню: в регистре `clicintattr[i]` установить бит `trig[0]` равным «0». Для прерываний от DMA и `EXT_INTx` (при их использовании) установить невекторный режим работы: в регистре `clicintattr[i]` установить бит `shv` равным «0». Включить в OpenOCD маскирование прерываний на время выполнения процедуры `step` с помощью команды `set_maskisr`. Это можно сделать одним из способов:

- добавить в файл конфигурации OpenOCD: `riscv set_maskisr steponly`;
- выполнить в GDB команду: `monitor riscv set_maskisr steponly`.

В этом случае OpenOCD перед командой `stepi` сохраняет текущее значение `mstatus`, после чего сбрасывает в `mstatus` биты глобальных разрешений прерываний `xIE`. После выполнения команды `stepi` OpenOCD восстанавливает ранее сохраненное значение `mstatus`. Данный способ обхода помогает в случае, если команда `stepi` выполняется над кодом, который не изменяет `mstatus` прямо или косвенно.

Версия OpenOCD с поддержкой команды `set_maskisr` доступна на сайте технической поддержки ЦП ИС АО «ПКК Миландр»: <https://support.milandr.ru/soft/>

***0004 Команда dret не декодируется как illegal instruction при исполнении не в Debug-режиме***

**Статус**

Исследование.

**Описание**

Неверное декодирование инструкции (команды) dret не в Debug-режиме.

**Условия**

Исполнение ассемблерной инструкции (команды) dret не в режиме Debug-режиме.

**Последствия**

Не декодирование как illegal instruction.

**Рекомендации и способы обхода**

Не использовать dret в пользовательском коде программы.

***0005 Некорректная работа аппаратной точки останова на коде в Debug ROM***

**Статус**

Исследование.

**Описание**

Останов на коде в Debug ROM не выполняется.

**Условия**

Установка точки останова на коде в Debug ROM.

**Последствия**

Возникновение ошибки.

**Рекомендации и способы обхода**

Не использовать отладку в Debug ROM.

**0006 Ошибка в tinfo-регистре модуля отладки****Статус**

Исследование.

**Описание**

Ошибка в tinfo-регистре, приводящая к проблеме механизма Discovery.

**Условия**

Применение OpenOCD с включённым механизмом Discovery.

**Последствия**

Возникновение ошибки.

**Рекомендации и способы обхода**

- I. Использовать OpenOCD из основной ветки <https://github.com/openocd-org/openocd> до версии 0.12.0 включительно.
- II. 1. Использовать OpenOCD для RISCv <https://github.com/riscv/riscv-openocd> до версии 2018.12.0;  
2. Использовать patch\* для OpenOCD для RISCv версии 0.12.0+ dev-03629-g87331a82a. Patch заменяет значение, прочитанное из регистра tinfo, на верное.

**0007 Команда c.lwsp (если rd == x0) не декодируется как illegal instruction****Статус**

Исследование.

**Описание**

Команда (инструкция) c.lwsp должна декодироваться как illegal instruction.

**Условия**

Исполнение команды (инструкции) c.lwsp (если rd == x0).

**Последствия**

Не декодирование illegal instruction.

**Рекомендации и способы обхода**

Компилятор “C” такие команды не генерирует, поэтому не использовать ассемблерные вставки в “C”-код с такими командами.

---

\* <https://support.milandr.ru/upload/iblock/366/1bla1mied7u7vkvie2nuenks9d440kw0/tinfo-patch.zip>

**0008 Чтение CSR-регистров командами csrrs, csrrsi и csrrci вызывает запись****Статус**

Исследование.

**Описание**

Команды (инструкции) csrrs, csrrc, csrrsi и csrrci могут быть использованы для выполнения операции чтения CSR-регистров без выполнения операции записи при rs1 = x0 (csrrs, csrrc) или uimm = 0 (csrrsi, csrrci):

- csrrs rd, csr, x0;
- csrrc rd, csr, x0;
- csrrsi rd, csr, 0;
- csrrci rd, csr, 0.

Однако команды csrrc, csrrsi и csrrci при выполнении операции чтения CSR-регистров также ошибочно выполняют операцию записи, что вызывает побочные эффекты записи (например, для регистра xNXTI) или вызывает исключение illegal-instruction при обращении к CSR-регистру, доступному только для чтения.

**Условия**

Выполнение чтения CSR-регистра командой csrrc, csrrsi и csrrci при rs1 = x0 (csrrc) и uimm = 0 (csrrsi, csrrci).

**Последствия**

Помимо операции чтения также выполняется операция записи, которая не приводит к изменению содержимого регистра (т.к. маска установки/сброса бит равна 0), но вызывает побочные эффекты записи (например, для регистра xNXTI) или вызывает исключение illegal-instruction при обращении к CSR-регистру, доступному только для чтения.

**Рекомендации и способы обхода**

Для выполнения операции чтения CSR-регистров без выполнения операции записи использовать команду csrrs с rs1 = x0 или псевдокоманду csrr (кодируется как csrrs с rs1 = x0):

- csrrs rd, csr, x0;
- csrr rd, csr.

## ***0009 Некорректное значение по сбросу поля sbaccess регистра sbcs***

### **Статус**

Исследование.

### **Описание**

Согласно спецификации RISC-V External Debug Support Version 0.13 значение поля sbaccess, которое задает разрядность данных для доступа к памяти с помощью System Bus Access, должно быть установлено по сбросу в значение 2, что соответствует размеру транзакции в 32 бита. Фактически значение по сбросу поля sbaccess равно 0, что соответствует размеру транзакции в 8 бит.

### **Условия**

Всегда.

### **Последствия**

Доступ к памяти в отладочном режиме с помощью System Bus Access будет осуществляться к данным с размером, отличным от ожидаемого.

### **Рекомендации и способы обхода**

Устанавливать значение поля sbaccess в требуемое значение перед осуществлением любых операций с помощью System Bus Access.

***0010 Остановка ядра при обработке невекторных прерываний по уровню в случае, если уровень изменяется в неактивный до завершения взятия прерывания***

**Статус**

Исследование.

**Описание**

Зависание (счетчик PC останавливается, на шине АНВ нет полноценного обмена) процессорного ядра может произойти в случае обработки невекторных прерываний, настроенных на взятие по уровню. Такое событие может произойти, если после начала обработки прерывания его уровень изменяется в неактивный до завершения взятия прерывания.

**Условия**

- 1 Прерывание настроено на срабатывание по уровню.
- 2 Прерывание невекторное.
- 3 Уровень запроса прерывания изменяется в неактивный до завершения взятия прерывания.

**Последствия**

Остановка ядра микросхемы.

**Рекомендации и способы обхода**

В контроллере CLIC настраивать используемые прерывания в один из описанных ниже режимов:

- 1 В векторный режим работы на срабатывание по уровню или фронту;
- 2 В невекторный режим работы на срабатывание по фронту.

## ***0011 Остановка счета IWDG при отсутствии частоты PCLK в процессе перезагрузки значения таймера или обновления значения предделителя***

### **Статус**

Исследование.

### **Описание**

В процессе работы IWDG остановка частоты PCLK блока IWDG после вызова запроса перезагрузки значения таймера через регистр IWDG\_KEY (записью значения 0xAAAA) и перед фактическим обновлением значения таймера или после вызова запроса обновления предделителя (запись в регистр IWDG\_PR) и перед фактическим обновлением значение предделителя блокирует сигнал запроса на обновление значения таймера в активном состоянии. В результате чего до повторного появления частоты PCLK или любого сброса сторожевой таймер IWDG не осуществляет счет и не формирует сброс. Так как в случае запроса перезагрузки IWDG непрерывно перезагружается значением перезагрузки, а в случае запроса обновления предделителя – непрерывно осуществляет обновление предделителя. После появления частоты PCLK и фактического обновления соответствующих значений или после любого сброса IWDG восстанавливает процесс счета.

### **Условия**

Остановка частоты PCLK блока IWDG:

- 1 После вызова запроса перезагрузки значения таймера через регистр IWDG\_KEY (записью значения 0xAAAA) и перед фактическим обновлением значения таймера в процессе работы IWDG;
- 2 После вызова запроса обновления предделителя таймера через регистр IWDG\_PR и перед фактическим обновлением предделителя в процессе работы IWDG.

### **Последствия**

Отсутствие счета таймера IWDG и формирования им сброса до появления частоты PCLK или любого сброса.

### **Рекомендации и способы обхода**

Реализовать в разрабатываемой системе один или несколько из предложенных способов:

- 1 Перед вызовом запроса перезагрузки значения таймера и на время сброса флага RVU, а также перед вызовом запроса обновления предделителя и на время сброса флага PVU, переводить тактирование PCLK (соответствует частоте HCLK) на частоту, которая гарантируется в разрабатываемой системе – например, LSI (поскольку наличие частоты LSI обязательно для работы IWDG, в случае её отсутствия IWDG не сбросит систему при любых обстоятельствах);
- 2 Использовать сторожевые таймеры IWDG и WWDG совместно, при этом время между их перезагрузкой должно быть не менее одного периода частоты LSI и WWDG должен быть запущен до настройки IWDG;
- 3 Использовать внешний сторожевой таймер.

**0012 Состояние вывода JTAG TDO (PA7) не определено после подачи питания****Статус**

Исследование.

**Описание**

Отладочный TAP-контроллер интерфейса JTAG при подаче питания сбрасывается по сигналу POR (сброс по питанию) и переходит в состояние Test-Logic-Reset. Если работа отладочного интерфейса JTAG разрешена, то выводы PA[9:6] переходят под управление отладочного TAP-контроллера.

Согласно стандарту JTAG IEEE 1149.1, TAP-контроллер в состоянии Test-Logic-Reset должен перевести вывод TDO в высокоимпедансное состояние. Однако из-за ошибки состояние вывода TDO (PA7) после сброса по питанию не определено – вывод может находиться в одном из следующих состояний: высокоимпедансное, высокий логический уровень, низкий логический уровень. Гарантированный переход вывода TDO (PA7) в высокоимпедансное состояние после сброса по питанию выполняется по спадающему фронту сигнала на входе TCK (PA6).

**Условия**

Отладочный TAP-контроллер интерфейса JTAG после сброса по питанию, работа отладочного интерфейса JTAG разрешена.

**Последствия**

Состояние вывода TDO (PA7) не определено: высокоимпедансное, высокий логический уровень, низкий логический уровень.

**Рекомендации и способы обхода**

Учитывать при разработке.

Для управления выводом TDO (PA7) с помощью контроллера портов ввода-вывода необходимо запретить работу интерфейса JTAG одним из следующих способов:

- сброс бита JTAG\_ON в регистре ВКР\_LDO;
- установка защиты адресного пространства Flash-памяти;
- блокировка интерфейса JTAG в регистре CHIP\_ID\_CTRL путем программирования памяти OTP.

## ***0013 Пропуск импульса SEC\_CLK при использовании калибровки RTC\_CLK***

### **Статус**

Исследование.

### **Описание**

В блоке RTC для формирования частоты SEC\_CLK из частоты RTC\_CLK используется делитель, выполненный на счетчике ВКР\_RTC\_PREDIV\_S с основанием счета ВКР\_RTC\_PRL. Для калибровки (замедления) RTC\_CLK используется счетчик ВКР\_RTC\_PREDIV\_A, который на время  $\text{ВКР\_RTC\_PREDIV\_A} < \text{RTC\_CAL}$  останавливает счет ВКР\_RTC\_PREDIV\_S.

Во время работы RTC с калибровкой частоты RTC\_CLK (поле  $\text{RTC\_CAL}[7:0] \neq 0$  в регистре ВКР\_RTC\_CR) при определенных значениях RTC\_CAL и ВКР\_RTC\_PRL периодически происходит одновременное выполнение событий  $\text{ВКР\_RTC\_PREDIV\_S} == \text{ВКР\_RTC\_PRL}$  и  $\text{ВКР\_RTC\_PREDIV\_A} == 0$ , которое приводит к ошибочному сбросу счетчика ВКР\_RTC\_PREDIV\_S и пропуску импульса SEC\_CLK.

При пропуске импульса SEC\_CLK счетчик/календарь (регистры ВКР\_RTC\_TR/DR) и сторожевой таймер ВКР\_RTC\_WUT (при тактировании от SEC\_CLK: бит  $\text{WUCK\_SEL}[2] == 1$  в регистре ВКР\_RTC\_CR) не изменяются и начинают отставать на 1 секунду. Соответствующие флаги в регистре ВКР\_RTC\_CS при пропуске импульса SEC\_CLK также не устанавливаются: SECF, ALRAF, ALRBF, WUTF и OWF.

### **Условия**

Установлена калибровка частоты RTC\_CLK (поле  $\text{RTC\_CAL}[7:0] \neq 0$ ) и заданы определенные значения RTC\_CAL и ВКР\_RTC\_PRL, при которых периодически происходит одновременное выполнение событий  $\text{ВКР\_RTC\_PREDIV\_S} == \text{ВКР\_RTC\_PRL}$  и  $\text{ВКР\_RTC\_PREDIV\_A} == 0$ .

### **Последствия**

Счетчик ВКР\_RTC\_PREDIV\_S ошибочно сбрасывается, при этом импульс SEC\_CLK не формируется. При пропуске импульса SEC\_CLK счетчик/календарь и сторожевой таймер ВКР\_RTC\_WUT (при тактировании от SEC\_CLK) не изменяются, соответствующие флаги также не устанавливаются: SECF, ALRAF, ALRBF, WUTF и OWF.

### **Рекомендации и способы обхода**

1 Температурная компенсация частоты SEC\_CLK используется.

Не использовать калибровку частоты RTC\_CLK (поле  $\text{RTC\_CAL}[7:0] == 0$ ), подстройку хода RTC выполнять с помощью настройки CONSTx и ВКР\_RTC\_PRL с учетом следующих особенностей (значения для кварцевого резонатора с номинальной частотой 32768 Гц):

- RTC\_CAL выполняет замедление RTC с шагом  $\sim 0,95$  ppm;
- CONSTx выполняет ускорение RTC с шагом  $\sim 7,63$  ppm;
- ВКР\_RTC\_PRL выполняет замедление/ускорение RTC с шагом  $\sim 30,52$  ppm.

При подстройке хода RTC необходимо выполнить замедление/ускорение с помощью ВКР\_RTC\_PRL (грубая подстройка), после чего выполнить ускорение с помощью CONSTx (точная подстройка), увеличив значения всех CONSTx на N, где N – значение от 0 до 3, дающее оптимальный результат.

Алгоритм расчета  $BKP\_RTC\_PRL$  и  $CONSTx$  при подстройке хода RTC на  $C$  ppm с использованием кварцевого резонатора с номинальной частотой 32768 Гц:

- настройка  $BKP\_RTC\_PRL$ :
  - $K = C \text{ ppm} / 30,52 \text{ ppm}$ ;
  - округление  $K$  к меньшему целому;
  - $BKP\_RTC\_PRL = BKP\_RTC\_PRL(\text{nominal}) - K$ ;
- настройка  $CONSTx$ :
  - $N = (C - K \cdot 30,52) \text{ ppm} / 7,63 \text{ ppm}$ ;
  - округление  $N$  к ближайшему целому;
  - $CONSTx = CONSTx(\text{nominal}) + N$ ;
- ошибка подстройки в ppm:
  - $E = K \cdot 30,52 \text{ ppm} + N \cdot 7,63 \text{ ppm} - C \text{ ppm}$ .

Пример замедления хода RTC (номинальная частота 32768 Гц):

- исходные данные:
  - исходная частота  $RTC\_CLK$ : 32773,3 Гц, т.е.  $\sim 32768 \text{ Гц} + 161,74 \text{ ppm}$ ;
  - требуемое изменение хода RTC:  $-161,74 \text{ ppm}$  (замедление);
- настройка  $BKP\_RTC\_PRL$ :
  - $K = -161,74 / 30,52 \approx -5,3 \rightarrow -6$ ;
  - $BKP\_RTC\_PRL = 32767 - (-6) = 32773$ ;
- настройка  $CONSTx$ :
  - $N = (-161,74 - (-6) \cdot 30,52) / 7,63 = 21,38 / 7,63 \approx 2,8 \rightarrow 3$ ;
  - $CONSTx = CONSTx(\text{nominal}) + 3$ ;
- ошибка подстройки:
  - $E = -6 \cdot 30,52 + 3 \cdot 7,63 - (-161,74) = 1,51 \text{ ppm}$ .

## 2 Температурная компенсация частоты $SEC\_CLK$ не используется.

Для калибровки частоты  $RTC\_CLK$  использовать только ограниченный набор значений  $RTC\_CAL$ , полученный с помощью скрипта\* для заданного значения  $BKP\_RTC\_PRL$ . Перед настройкой блок RTC должен быть выключен и сброшен, при настройке регистр  $BKP\_RTC\_PREDIV\_S$  не должен записываться значением, отличным от 0. После настройки и включения RTC значения  $RTC\_CAL[7:0]$ ,  $BKP\_RTC\_PREDIV\_S$  и  $BKP\_RTC\_PRL$  не должны перезаписываться. В этом случае ошибка проявляться не будет.

Пример инициализации RTC:

- включить источник тактирования RTC и выбрать его в поле  $RTC\_SEL[1:0]$  регистра  $BKP\_RTC\_CR$ ;
- дождаться формирования валидной частоты  $RTC\_CLK$  согласно спецификации;
- выключить RTC путем сброса бита  $RTC\_EN$  в регистре  $BKP\_RTC\_CR$ ;
- выполнить сброс RTC путем последовательной установки и сброса бита  $RTC\_RESET$  в регистре  $BKP\_RTC\_CR$ ;
- записать требуемое значение в регистр  $BKP\_RTC\_PRL$ ;
- записать допустимое значение в поле  $RTC\_CAL[7:0]$  регистра  $BKP\_RTC\_CR$ ;
- записать 0 в регистр  $BKP\_RTC\_PREDIV\_S$ ;
- дождаться окончания записи с помощью бита  $WEC$  в регистре  $BKP\_RTC\_CS$ ;
- выполнить другие настройки RTC;
- включить RTC путем установки бита  $RTC\_EN$  в регистре  $BKP\_RTC\_CR$ .

\* [https://support.milandr.ru/upload/iblock/285/9bliqywbkpfcoptft5dctv327ycqh9s/bkp\\_rtc\\_cal.py](https://support.milandr.ru/upload/iblock/285/9bliqywbkpfcoptft5dctv327ycqh9s/bkp_rtc_cal.py)

## ***0014 "Залипание" флага HSE\_RDY после пропадания частоты на OSC\_IN и сброса не по питанию***

### **Статус**

Исследование.

### **Описание**

После включения HSE и его выхода в рабочий режим ( $HSE\_RDY = 1$ ), пропадания частоты HSE на OSC\_IN и последующего сброса не по питанию (например, сброс от IWDG или nRESET), флаг HSE\_RDY выставляется в "1" сразу после включения HSE ( $HSE\_ON = 1$ ) не смотря на отсутствие частоты на выводе OSC\_IN.

### **Условия**

Пропадание частоты HSE с последующим сбросом сброс не по питанию.

### **Последствия**

Флаг HSE\_RDY "залипает" в "1", нельзя использовать для определения готовности HSE.

### **Рекомендации и способы обхода**

Использовать следующий алгоритм:

1. Стандартно включить HSE и дождаться готовности ( $HSE\_RDY = 1$ ).
2. Настроить PLL с тактированием от HSI ( $CPU\_C1 = HSI$ , при этом HSI должен быть активен и готов), включить PLL и дождаться готовности  $PLL\_RDY = 1$ , после чего выключить PLL.
3. Настроить PLL с тактированием от HSE, включить PLL и ждать готовности  $PLL\_RDY = 1$ .
  - Если  $PLL\_RDY = 0$ , то на выводе OSC\_IN нет частоты, использовать HSE нельзя.
  - Если  $PLL\_RDY = 1$ , то на выводе OSC\_IN есть частота, HSE работает и готов для использования. Если использование PLL не требуется, PLL можно отключить.

## ***0015 Контроллер интерфейса I2C не отслеживает clock stretching при формировании START и DIV=1-2***

### **Статус**

Исследование.

### **Описание**

Контроллер интерфейса I2C при задании делителя частоты DIV равного 1 или 2 не отслеживает растяжение тактовой частоты SCL (clock stretching) ведомым устройством при формировании сигнала START. В связи с этим отправка сигнала START (установка битов START и RD/WR в регистре CMD) при удерживании ведомым устройством линии SCL в состоянии низкого уровня будет выполнена без ожидания поднятия SCL в состояние высокого уровня, что приведет к некорректному формированию сигнала START на шине I2C (данная ситуация может наблюдаться при передаче повторного сигнала START).

### **Условия**

Внутренний делитель частоты DIV равен 1 или 2, ведомое устройство удерживает линию SCL в состоянии низкого уровня, контроллер интерфейса I2C отправляет сигнал START (установка битов START и RD/WR в регистре CMD).

### **Последствия**

Отправка сигнала START выполняется без ожидания поднятия SCL в состояние высокого уровня, что приводит к некорректному формированию сигнала START на шине I2C.

### **Рекомендации и способы обхода**

Если ведомое устройство применяет растяжение тактовой частоты SCL (clock stretching), то перед формированием сигнала START необходимо выдержать время, за которое ведомое устройство гарантированно отпустит линию SCL, либо использовать делитель частоты DIV равный 3 и более.

**Лист регистрации изменений**

Дата	Страница	Статус	ID	Категория	Описание
25.01.24					Документ создан
26.01.24	5, 8		0003	3	Добавлено описание ошибки
08.02.24					Добавлено исполнение MDR1206AFI
13.02.24	5, 9 – 11		0004 – 0008		Добавлено описание ошибок
19.03.24	8		0003		Дополнено описание ошибки
28.03.24	5, 8		0003		Уточнено название ошибки
15.04.24	7		0002		Уточнены рекомендации и способы обхода
19.06.24					Добавлено исполнение MDR1215LGI
01.08.24	11 – 13		0009 – 0011	3	Добавлено описание ошибок
12.12.24	5, 9		0003	3	Добавлена ревизия 2 для MDR1206FI Уточнен статус ошибки
17.03.24	10 12 16 5		0004 0008 0012		Исправлено название ошибки Обновлено описание ошибки Добавлено описание ошибки Сводная таблица ошибок – исправлена дата производства для MDR1206AFI
24.06.25	8 17		0002 0013	3 3	Исправлена опечатка Добавлено описание ошибки
06.08.25	18		0013	3	Уточнены рекомендации и способы обхода, вариант 2
12.11.25	19 20		0014 0015	3 3	Добавлено описание ошибки Добавлено описание ошибки
25.03.26					Добавлено исполнение MDR1219LGI