

**Образец выпадающего кадра**

Образец отбрасываемого кадра всегда имеет однобайтовую полезную нагрузку, определяемую через

Немного	7	6	5	4	3	2	1	0
Содержание	зарезервированный					aux_dro п	бронировать d	acc_ уронить

Кадр отбрасывания выборки будет вставлен после кадра Fifo\_Input\_Config в такте ODR, при котором выборка была отброшена, и только в том случае, если ни один другой датчик не предоставит действительную выборку в этом такте ODR. Если другой датчик предоставляет достоверные данные, данные этого датчика просто не включаются, и соответствующий бит заголовка кадра данных не устанавливается.

Кадры отбрасывания пробы будут вставлены только для переходных фаз после изменений конфигурации, но не для проб, отброшенных между включением датчика и первой действительной пробой. Подробное описание изменений конфигурации см. В разделе 4.5, подраздел «Изменения конфигурации».

**FIFO Чтение частичного кадра**

Когда кадр только частично считывается через регистр FIFO\_DATA, он будет полностью повторен при следующем доступе как в режиме без заголовка, так и в режиме заголовка. В режиме заголовка это включает заголовок. В случае переполнения FIFO между первым частичным чтением и второй попыткой чтения кадр может быть удален.

**FIFO перечитывает**

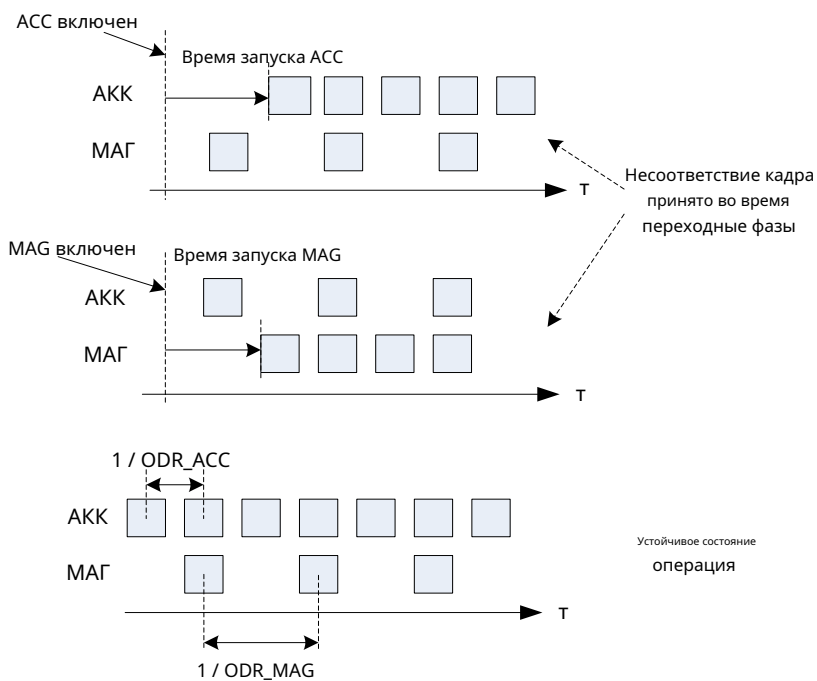
Когда из FIFO считывается больше данных, чем содержится допустимых данных, 0x8000 возвращается в режиме без заголовка. В режиме заголовка возвращается 0x0080, где 0x80 указывает на недопустимый кадр.

**4.6.3 Синхронизация данных FIFO**

Все данные сенсора дискретизируются по общей временной сетке ODR. Даже если для ускорения и магнитного датчика выбран другой ODR, данные остаются синхронизированными:

Если кадр содержит выборку сенсорного элемента с ODR  $x$ , тогда он должен также содержать образцы всех сенсорных элементов с ODR  $y \geq x$ . Это относится к работе в установившемся режиме. На переходных этапах важнее не потерять данные, поэтому возможны исключения, если сенсорные элементы с ODR  $y \geq x$  не имеют данных, например, из-за изменения конфигурации сенсора.

Схема синхронизации данных FIFO на следующем рисунке иллюстрирует установившееся состояние и переходные рабочие условия.



#### 4.6.4 Синхронизация FIFO с внешними прерываниями

Внешние прерывания могут быть синхронизированы с данными FIFO. Для этого режима работы необходимо включить FIFO\_CONFIG\_1.fifo\_tag\_int1\_en и / или FIFO\_CONFIG\_1.fifo\_tag\_int2\_en, а также INT1\_IO\_CTRL.input\_en и / или INT2\_IO\_CTRL.input\_en. Поле fh\_ext в заголовке FIFO будет затем установлено в соответствии с сигналом на входах INT1 / INT2.

#### 4.6.5 Прерывания FIFO

FIFO поддерживает два прерывания, полное прерывание FIFO и прерывание водяного знака:

Прерывание от заполнения FIFO выдается, когда уровень заполнения FIFO превышает полный порог. Полный порог достигается непосредственно перед тем, как последние два кадра будут сохранены в FIFO.

Водяной знак FIFO выдается, когда уровень заполнения FIFO равен или выше водяного знака, определенного в регистрах FIFO\_WTM\_0 и FIFO\_WTM\_1.

Чтобы включить / использовать прерывания FIFO full или watermark, сопоставьте их с желаемым выводом прерывания через INT\_MAP\_DATA.

Оба прерывания подавляются, когда продолжается операция чтения регистра FIFO\_DATA. Защищенные прерывания FIFO будут очищены только в том случае, если будет прочитан регистр состояния и уровень заполнения ниже соответствующего прерывания FIFO (полное или водяной знак).

#### 4.6.6 Сброс FIFO

Пользователь может вызвать сброс FIFO, записав команду fifo\_flash (0xB0) в CMD.

Автоматический сброс выполняется только в следующих случаях:

Датчик включен или отключен в режиме без заголовка

Произошел переход между режимом без заголовка и режимом заголовка или наоборот.

Размер данных вспомогательного датчика в кадре изменен в режиме заголовка или без заголовка

## 4,7 Набор интегрированных функций:

### 4.7.1 Обнаружение любого движения / отсутствия движения

#### Обнаружение любого движения:

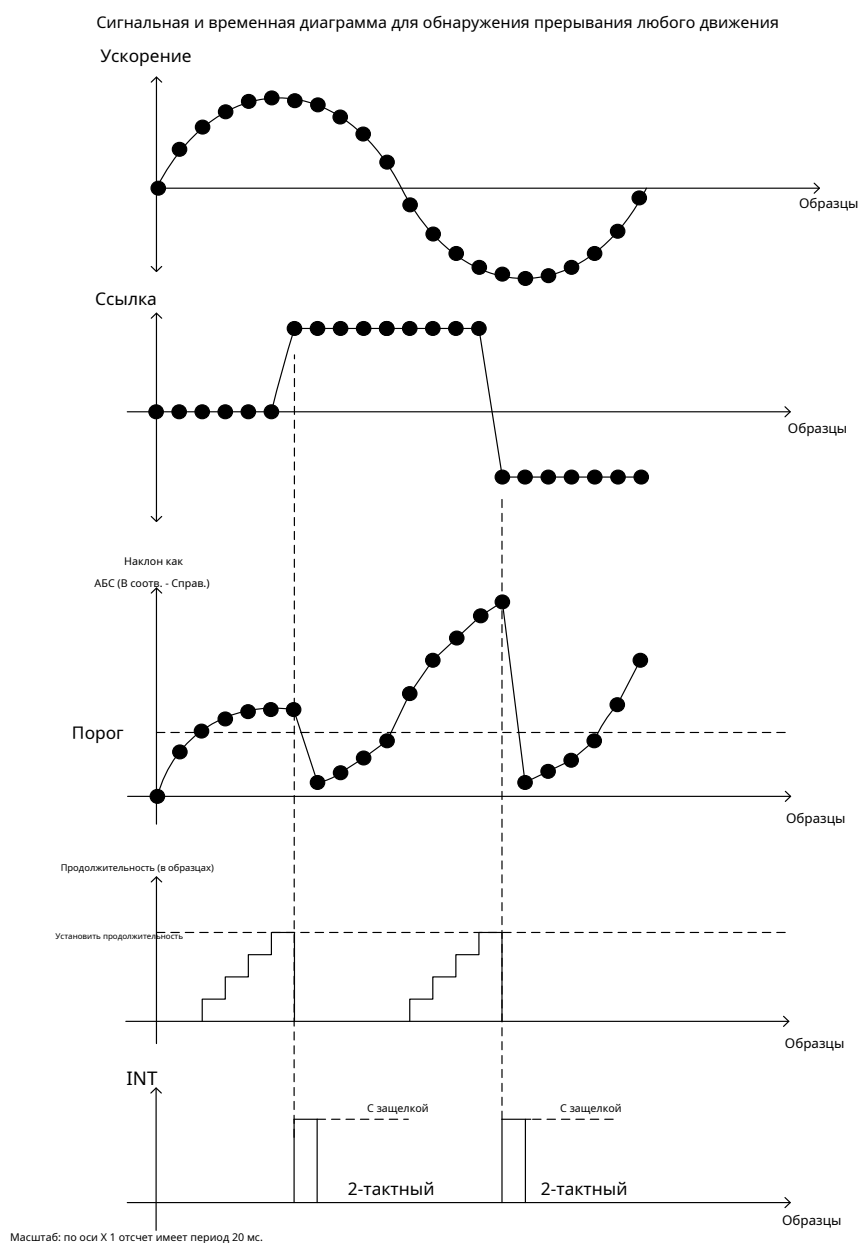
Обнаружение любого движения использует крутизну между текущими входными и эталонными отсчетами ускорения для определения состояния движения устройства. Функцию можно включить, установив хотя бы одно из следующих значений: FEATURES\_IN.any\_motion.settings\_2.x\_en, FEATURES\_IN.any\_motion.settings\_2.y\_en и FEATURES\_IN.any\_motion.settings\_2.z\_en соответственно для каждой оси.

Any-motion обеспечивает прерывание, когда абсолютное значение наклона превышает конфигурируемый FEATURES\_IN.any\_motion.settings\_1.threshold для последовательных

FEATURES\_IN.any\_motion.settings\_2. Выборки длительности по крайней мере для одной из включенных осей восприятия.

Эталонная выборка ускорения обновляется только при срабатывании прерывания любого движения. Состояние прерывания сбрасывается, как только крутизна падает ниже установленной.

FEATURES\_IN.any\_motion.setings\_1.threshold значение. Сигналы и тайминги, относящиеся к функции прерывания anymotion, показаны на рисунке ниже:



Параметры конфигурации:

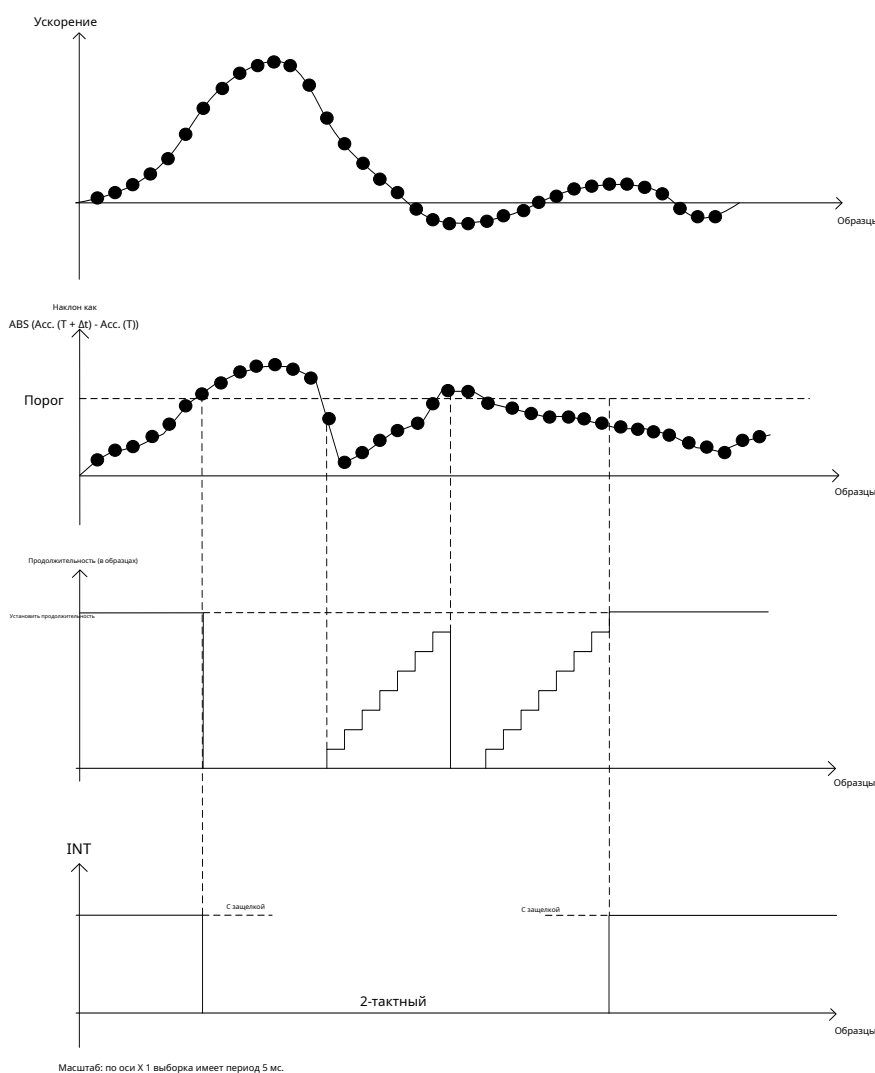
1. FEATURES\_IN.any\_motion.settings\_1.threshold - порог наклона.
2. FEATURES\_IN.any\_motion.settings\_2.duration - количество последовательных точек данных для которого должно соблюдаться пороговое условие для утверждения прерывания.
3. FEATURES\_IN.any\_motion.settings\_2.x\_en - указывает, включена ли эта функция для оси x
4. FEATURES\_IN.any\_motion.settings\_2.y\_en - указывает, включена ли эта функция для оси Y
5. FEATURES\_IN.any\_motion.settings\_2.z\_en - указывает, включена ли эта функция для оси z.

Нет обнаружения движения:

Обнаружение отсутствия движения использует наклон между двумя последовательными отсчетами сигнала ускорения для определения статического состояния устройства. Функцию можно включить, установив хотя бы один из следующих флагов: FEATURES\_IN.no\_motion.settings\_2.x\_en, FEATURES\_IN.no\_motion.settings\_2.y\_en и FEATURES\_IN.no\_motion.settings\_2.z\_en соответственно для каждой оси.

Прерывание отсутствия движения запускается, когда крутизна на всех включенных осях измерения остается меньше настраиваемого FEATURES\_IN.no\_motion.settings\_1.threshold в течение продолжительности, заданной параметром FEATURES\_IN.no\_motion.settings\_2.duration. Прерывание отсутствия движения сбрасывается, как только крутизна ускорения превышает установленный порог. Сигналы и тайминги, относящиеся к функции прерывания при отсутствии движения, показаны на рисунке ниже.

Сигнальная и временная диаграмма для обнаружения прерывания при отсутствии движения



Регистр FEATURES\_IN.no\_motion.settings\_2.duration определяет количество последовательных точек данных, для которых наклон разрешенной оси должен быть меньше порогового значения для прерывания, которое должно быть заявлено.

Параметры конфигурации:

1. FEATURES\_IN.no\_motion.settings\_1.threshold - порог наклона.
2. FEATURES\_IN.no\_motion.settings\_2.duration - количество последовательных точек данных для при котором должно соблюдаться пороговое условие для утверждения прерывания.
3. FEATURES\_IN.no\_motion.settings\_2.x\_en - указывает, включена ли эта функция для оси x
4. FEATURES\_IN.no\_motion.settings\_2.y\_en - указывает, включена ли эта функция для оси Y
5. FEATURES\_IN.no\_motion.settings\_2.z\_en - указывает, включена ли эта функция для оси z.

**Примечание:** Образ прошивки с любым движением и без набора функций движения, упомянутых в разделе выше, доступен для загрузки по следующей ссылке.

<https://github.com/BoschSensortec/BMA490L-Sensor-API>

## 4.8 Общая конфигурация контактов прерывания

### Поведение контактов электрического прерывания

Оба вывода прерывания INT1 и INT2 могут быть настроены для отображения желаемого электрического поведения.

Контакты прерывания могут быть включены в INT1\_IO\_CTRL.output\_en или INT2\_IO\_CTRL.output\_en. Характеристики выходного драйвера контактов прерывания могут быть сконфигурированы с помощью битов INT1\_IO\_CTRL.od и INT2\_IO\_CTRL.od. Установив эти биты в 0b1, выходной драйвер покажет характеристику открытого привода, установив биты конфигурации в 0b0, выходной драйвер покажет двухтактную характеристику.

Электрическое поведение выводов прерывания при срабатывании прерывания может быть сконфигурировано как «активный-высокий» или «активный-низкий» через INT1\_IO\_CTRL.lvl соответственно INT2\_IO\_CTRL.lvl.

Оба контакта прерывания могут быть настроены как входные контакты через INT1\_IO\_CTRL.input\_en или INT2\_IO\_CTRL.input\_en. Это необходимо, когда используется функция тега FIFO (см. Соответствующую главу FIFO). Если оба активированы, вход (например, маркировка FIFO) управляется выходом прерывания. BMA490L поддерживает входы прерываний, запускаемых по фронту и по уровню, это можно настроить с помощью INT1\_IO\_CTRL.edge\_ctrl или INT2\_IO\_CTRL.edge\_ctrl.

BMA490L поддерживает режимы прерывания без фиксации и с фиксацией для готовности данных, заполнения FIFO и водяного знака FIFO. Режим выбирается INT\_LATCH.int\_latch. Функциональные прерывания, описанные в главе Прерывания FIFO, поддерживают только фиксированный режим, описанный ниже.

В фиксированном режиме заявленное состояние прерывания в INT\_STATUS\_0 или INT\_STATUS\_1 и выбранный вывод очищаются, если считывается соответствующий регистр состояния. Если в фиксированном режиме используется более одного вывода прерывания, все прерывания в INT\_STATUS\_0 должны быть сопоставлены с одним выводом, а все прерывания в INT\_STATUS\_1 должны быть сопоставлены с другим выводом. Если используется только один вывод прерывания, все прерывания могут быть отображены на этот вывод. Если условие активации все еще сохраняется, когда оно сброшено, статус прерывания утверждается снова, когда условие прерывания снова сохраняется.

В 2-тактном режиме (только для данных готовности, FIFO full и FIFO watermark) бит состояния прерывания и выбранный вывод сбрасываются, как только условие активации больше не действует.

### Назначение контактов прерывания

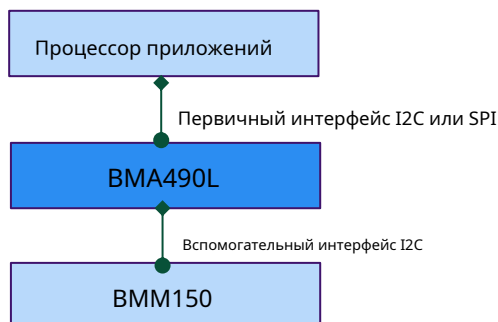
Чтобы Хост реагировал на выходные данные функций, они могут быть отображены на внешний вывод INT1 или вывод INT2, установив соответствующие биты из регистров INT1\_MAP, соответственно INT2\_MAP.

Чтобы отключить выходы функций на внешние контакты, те же соответствующие биты должны быть сброшены из регистров INT1\_MAP, соответственно INT2\_MAP.

Как только функция запускает выходной контакт, хост может считывать соответствующий бит из регистра, INT\_STATUS\_0 (прерывания функций) или INT\_STATUS\_1 (FIFO и данные готовы).

#### 4.9 Интерфейс дополнительного датчика

Вспомогательный интерфейс позволяет подключить один вспомогательный датчик (например, магнитометр) к специальному интерфейсу вспомогательного сенсора, как показано ниже.



Решение с 6 степенями свободы с BMA490L и BMM150

##### 4.9.1 Структура и концепция

BMA490L управляет сбором данных вспомогательным датчиком и передает данные процессору приложений через основной интерфейс I2C или SPI. Никакие другие ведущие или ведомые устройства I2C не должны подключаться к интерфейсу дополнительных датчиков.

BMA490L автономно считывает данные датчика с совместимого дополнительного датчика без вмешательства процессора приложений и сохраняет данные в своих регистрах данных и FIFO. Первоначальная настройка вспомогательного датчика после включения выполняется посредством косвенной адресации (в режиме настройки, как описано в следующем разделе).

Основные преимущества интерфейса дополнительных датчиков:

Синхронизация данных датчика вспомогательного датчика и акселерометра. Это приводит к улучшено качество слияния данных датчиков.

Использование BMA490L FIFO для данных дополнительных датчиков (BMM150 не имеет FIFO). Это важно для мониторинга приложений.

##### 4.9.2 Конфигурация интерфейса

Регистры конфигурации, которые управляют работой интерфейса вспомогательного датчика, влияют только на интерфейс вспомогательного датчика, но не на конфигурацию самого датчика акселерометра (это необходимо сделать в режиме настройки).

Существует три основных конфигурации / режима интерфейса дополнительного датчика:

Нет доступа к дополнительному датчику

Режим настройки: доступ к дополнительному датчику в ручном режиме

Режим данных: доступ к дополнительному датчику через аппаратный контур считывания.

Настройка самого дополнительного датчика должна выполняться через основной интерфейс с использованием косвенной адресации в режиме настройки. При сборе данных датчика BMA490L автономно запускает измерение дополнительного датчика, используя принудительный режим дополнительного датчика и считывание данных с дополнительного датчика (режим данных).



В режиме настройки вспомогательный датчик может быть настроен, и данные дифферента могут быть считаны с вспомогательного датчика. В режиме данных данные вспомогательного датчика непрерывно копируются в регистры BMA490L и могут быть считаны из BMA490L непосредственно через первичный интерфейс. Для магнитометра BMM150 это данные самого дополнительного датчика и сопротивление Холла, температура не требуется. В таблице ниже показано, как настроить эти три режима с помощью регистров PWR\_CONF, PWR\_CTRL и AUX\_IF\_CONF.aux\_manual\_en.

Режим	AUX_IF_CONF.aux_manual_en	PWR_CONF.adv_power_save	PWR_CTRL.aux_en
Нет дополнительного датчика доступ	1	1	0
Режим настройки	1	0	0
Режим данных	0	Икс	1

Режим интерфейса дополнительного датчика можно включить, установив бит IF\_CONF.if\_mode в соответствии с следующей таблицей.

IF_CONF.if_mode	Результат
0	Вторичный IF отключен (по умолчанию)
1	AuxIF включен

Интерфейс вспомогательного датчика работает на частоте 400 кГц. Это приводит к задержке считывания I2C около 250 мкс для 10 байтов данных.

Адрес ведомого устройства I2C дополнительного датчика определяется в AUX\_DEV\_ID. i2c\_device\_addr.

### 4.9.3 Режим настройки (AUX\_IF\_CONF.aux\_manual\_en = 0b1)

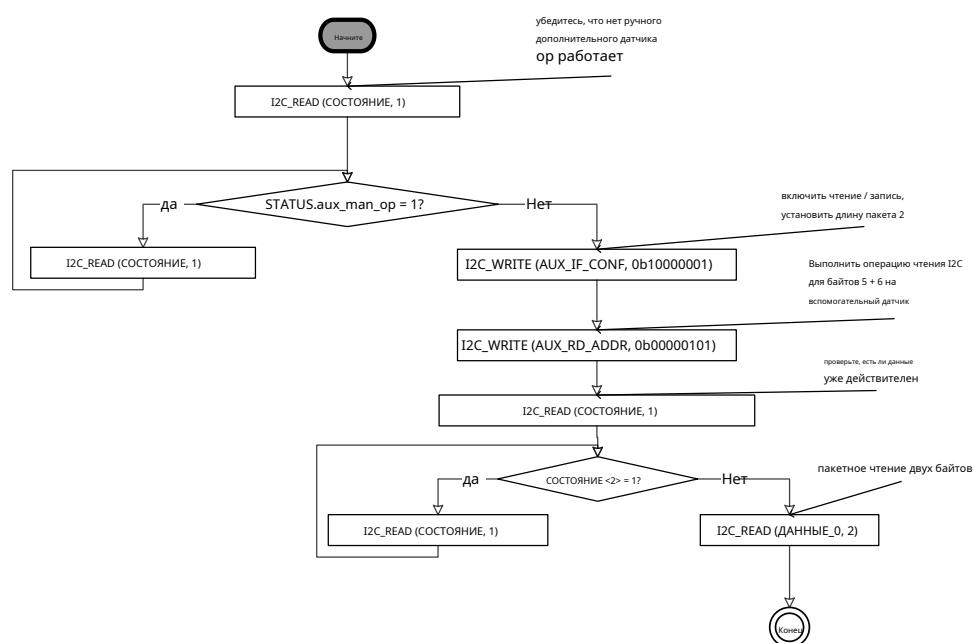
Доступ к дополнительному датчику через первичный интерфейс можно получить с помощью косвенной адресации через регистры AUX\_\*. AUX\_RD\_ADDR и AUX\_WR\_ADDR определяют адрес регистра для чтения / записи в карте регистров вспомогательного датчика и запускают саму операцию, когда интерфейс вспомогательного датчика включен через PWR\_CTRL.aux\_en.

Для чтения количество байтов данных, определенное в AUX\_IF\_CONF.aux\_rd\_burst, считывается из вспомогательного датчика и записывается в регистры BMA490L от DATA\_0 до DATA\_7. Для записи записываются только отдельные байты, независимо от настроек в AUX\_IF\_CONF.aux\_rd\_burst. Данные для записи I2C во вспомогательный датчик должны быть сохранены в AUX\_WR\_DATA до того, как адрес регистра вспомогательного датчика будет записан в AUX\_WR\_ADDR.

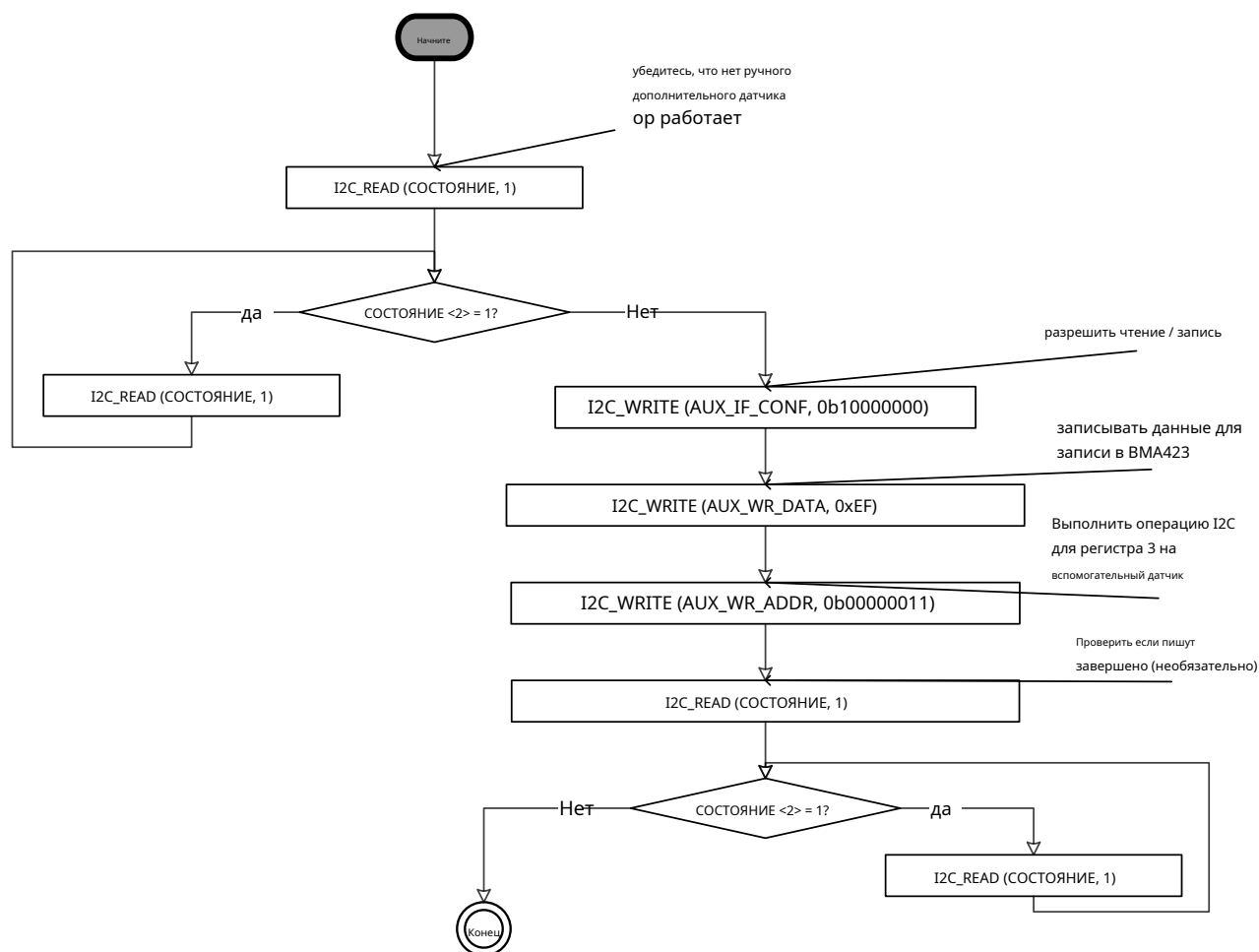
Когда операция чтения или записи запускается записью в AUX\_RD\_ADDR и AUX\_WR\_ADDR, устанавливается STATUS.aux\_man\_op и сбрасывается после завершения операции. Для операций чтения от DATA\_0 до DATA\_7 содержатся считанные данные, для записи AUX\_WR\_DATA может быть перезаписан снова.

Этап настройки дополнительного датчика.

Пример: считывание байтов 5 и 6 дополнительного датчика.



Пример: записать 0xEF в регистр 3 дополнительного датчика.



#### 4.9.4 Режим данных (AUX\_IF\_CONF.aux\_manual\_en = 0)

AUX\_RD\_ADDR.read\_addr определяет адрес регистра данных, из которого следует читать количество байтов данных, сконфигурированных в AUX\_IF\_CONF.aux\_rd\_burst из данных AUX\_0... AUX\_7 вспомогательного датчика. Эти данные хранятся в регистрах DATA\_0 до DATA\_7. Статус готовности данных устанавливается в STATUS.drdy\_aux, обычно он очищается путем чтения одного из регистров DATA\_0 - DATA\_7.

AUX\_WR\_ADDR.write\_addr определяет адрес регистра вспомогательного датчика для запуска измерения в принудительном режиме в карте регистров вспомогательного датчика. Задержка (временной сдвиг) между запуском измерения вспомогательного датчика и считыванием данных измерения указывается в AUX\_CONF.aux\_offset. Чтение данных выполняется за одну операцию чтения I2C с длиной пакета, указанной в AUX\_IF\_CONF.aux\_rd\_burst. Для BMM150 AUX\_IF\_CONF.aux\_rd\_burst должен быть установлен в 0b11, т.е. 8 байт. Если для AUX\_IF\_CONF.aux\_rd\_burst установлено значение меньше 8 байтов, оставшиеся данные вспомогательного датчика в регистрах с DATA\_0 по DATA\_7 и FIFO не определены.

Рекомендуется отключить интерфейс дополнительного датчика (IF\_CONF.if\_mode = 0b0) перед настройкой AUX\_RD\_ADDR.read\_addr и AUX\_WR\_ADDR.write\_addr для режима данных. Это не переводит сам вспомогательный датчик в режим ожидания, но позволяет избежать сбора нежелательных данных на этом этапе. После этого интерфейс дополнительного датчика можно снова включить (IF\_CONF.if\_mode = 0b1).

#### 4.9.5 Задержка (смещение по времени)

BMA490L поддерживает запуск измерения датчика на дополнительном интерфейсе датчика между 2,5 и 37,5 мс до обновления ДАННЫХ регистра. Это смещение определяется в AUX\_CONF.aux\_offset. Если установлено значение 0b0, измерение выполняется сразу после последнего обновления ДАННЫХ регистра, поэтому это измерение будет включено в следующее обновление ДАННЫХ регистра.

#### 4.10 Самопроверка сенсора

BMA490L имеет комплексную функцию самотестирования элемента MEMS путем приложения электростатических сил к сердечнику датчика вместо внешних ускорений. Фактически отклоняя сейсмическую массу, можно проверить весь путь прохождения сигнала датчика. Активация самотестирования приводит к статическому смещению данных ускорения; любое внешнее ускорение или гравитационная сила, приложенная к датчику во время активного самотестирования, будет наблюдаться на выходе как суперпозиция сигнала ускорения и самотестирования. Перед включением самотестирования диапазон  $g$  должен быть установлен на 8g.

Самотестирование активируется для всех осей записью `ACC_SELF_TEST.acc_self_test_en = 1b1`. Самопроверка отключается записью `ACC_SELF_TEST.acc_self_test_en = 1b0`. Можно контролировать направление отклонения через бит `ACC_SELF_TEST.acc_self_test_sign`. Возбуждение происходит в положительном (отрицательном) направлении, если `ACC_SELF_TEST.acc_self_test_sign = 1b1` ('b0). Амплитуда отклонения должна быть низкой, записав `ACC_SELF_TEST.acc_self_test_amp = 1b0`. После включения самотестирования пользователь должен подождать 50 мсек, прежде чем интерпретировать данные ускорения.

Чтобы обеспечить правильную интерпретацию сигнала самопроверки, рекомендуется выполнить самопроверку для обоих (положительного и отрицательного) направлений, а затем вычислить разницу полученных значений ускорения. В таблице ниже показаны минимальные различия для каждой оси, чтобы пройти самотестирование. Фактически измеренные различия сигналов могут быть значительно больше.

Самотестирование: Результат минимального разностного сигнала для BMA490L.

	сигнал оси x	сигнал оси y	сигнал оси z
BMA490L	1800 мг	1800 мг	1800 мг

Рекомендуется выполнить сброс устройства после выполнения самотестирования. Если сброс не может быть выполнен, необходимо соблюдать следующую последовательность, чтобы предотвратить создание нежелательного прерывания: отключить прерывания, изменить параметры прерываний, подождать не менее 50 мсек и разрешить требуемые прерывания.

Рекомендуемая процедура самотестирования следующая:

1. Включите акселерометр с регистром `PWR_CTRL.acc_en = 1b1`.
2. Установите диапазон  $\pm 8g$  в регистре `ACC_RANGE.acc_range`
3. Установите низкую амплитуду самопроверки, установив `ACC_SELF_TEST.acc_self_test_amp = 1b0`
4. Установите `ACC_CONF.acc_odr = 1600` Гц, режим непрерывной выборки, `ACC_CONF.acc_bwp = norm_avg4`, `ACC_CONF.acc_perf_mode = 1b1`.
5. Подождите > 2 мс
6. Включите самопроверку и установите положительный полярность самопроверки (`ACC_SELF_TEST.acc_self_test_sign = 1b1`)
7. Подождите > 50 мс.
8. Считайте и сохраните положительное значение ускорения каждой оси из регистров `DATA_8 - DATA_13`.
9. Включите самопроверку и установите отрицательный полярность самопроверки `ACC_SELF_TEST.acc_self_test_sign = 1b0`
10. Подождите > 50 мс.
11. Считайте и сохраните отрицательное значение ускорения каждой оси из регистров `DATA_8 - DATA_13`.
12. Рассчитайте разницу положительных и отрицательных значений ускорения и сравните с пороговые значения

## 4.11 Компенсация смещения

BMA490L предлагает ручную компенсацию, а также встроенную калибровку.

Компенсация смещения выполняется с предварительно отфильтрованными данными, а затем смещение применяется как к предварительно отфильтрованным, так и к отфильтрованным данным. При необходимости результат этого вычисления насыщается, чтобы предотвратить любые ошибки переполнения (устанавливается наименьшее или наибольшее возможное значение, в зависимости от знака).

Общедоступные регистры компенсации смещения от OFFSET\_0 до OFFSET\_2 являются образами соответствующих регистров в NVM. При каждом обновлении образа содержимое регистров NVM записывается в общедоступные регистры. Публичные регистры могут быть перезаписаны пользователем в любое время.

Регистры компенсации смещения имеют ширину 8 бит с использованием дополнения до двух. Разрешение смещения (LSB) составляет 3,9 мг, а диапазон смещения составляет + - 0,5 г. Оба не зависят от настройки диапазона. Компенсацию смещения необходимо включить через NV\_CONF.acc\_off\_en = 0b1

### 4.11.1 Ручная компенсация смещения

Содержимое регистра общественных компенсаций OFFSET\_0 - OFFSET\_2 может быть установлено вручную через цифровой интерфейс. После изменения регистра OFFSET\_0 на OFFSET\_2 следующая выборка данных недействительна.

Компенсацию смещения необходимо включить через NV\_CONF.acc\_off\_en.

### 4.11.2 Встроенная калибровка

Для некоторых приложений часто бывает желательно один раз откалибровать смещение и постоянно сохранять значения компенсации. Этого можно достичь, используя ручную компенсацию смещения для определения правильных значений компенсации, а затем постоянно сохраняя эти значения в NVM.

При каждом сбросе устройства значения компенсации загружаются из энергонезависимой памяти в регистры изображения и используются для компенсации смещения.

#### 4,12 Энергонезависимая память

Регистры NV\_CONF и OFFSET\_0 - OFFSET\_2 имеют резервную копию NVM, доступную для пользователя.

Содержимое NVM загружается в регистры изображений после сброса (либо POR, либо программный сброс). Пока идет обновление образа, STATUS.cmd\_rdy равен 0b0, в противном случае - 0b1.

Регистры изображений можно читать и записывать, как и любой другой регистр.

Запись в NVM - это четырехэтапная процедура:

1. Установите PWR\_CONF.adv\_power\_save = 0b0
2. Запишите новое содержимое в регистры изображений.
3. Запишите 0b1 в бит NVM\_CONF.nvm\_prog\_en, чтобы разблокировать NVM.
4. Запишите nvm\_prog в регистр CMD, чтобы запустить процесс записи.
5. Запишите 0b0 в бит NVM\_CONF.nvm\_prog\_en, чтобы заблокировать NVM, после завершения процесса записи.

Запись в NVM всегда обновляет все содержимое NVM. Состояние записи можно проверить, прочитав STATUS.cmd\_rdy. В то время как STATUS.cmd\_rdy знак равно 0b0, процесс записи все еще продолжается; когда STATUS.cmd\_rdy знак равно 0b1, запись завершена. Цикл записи NVM может быть инициирован, только если PWR\_CONF.adv\_power\_save = 0b0.

До завершения фазы загрузки (после POR или программного сброса) последовательный интерфейс не работает. К теневым регистрам NVM нельзя обращаться во время выполнения команды NVM (инициированной через регистр CMD).

Во всех остальных случаях регистр можно читать или писать.

Пока выполняется чтение NVM (во время загрузки датчика и программного сброса) или запись NVM, записи в регистры датчика сбрасываются, чтение возвращает статус регистра независимо от адреса чтения.

#### 4,13 Мягкий сброс

Программный сброс можно запустить в любой момент, написав команду *softreset* (0xB6) зарегистрировать CMD.

Программный сброс выполняет основной сброс устройства, который в значительной степени эквивалентен циклу включения питания. После задержки все пользовательские настройки конфигурации перезаписываются с их состоянием по умолчанию (настройки сохраняются в NVM), где это применимо. Эта команда работает во всех режимах работы, но не должна выполняться во время записи NVM.

5. Описание регистра

5.1 Основные пометки

Регистры могут быть прочитаны и записаны во всех конфигурациях питания, за исключением FEATURES\_IN и FIFO\_DATA, которым для PWR\_CONF.adv\_power\_save необходимо установить значение 0b0. Следующая глава содержит только из общей карты регистров регистры, относящиеся к функциям, исключены.

5.2 Зарегистрировать карту

читай пиши			только чтение			писать только			зарезервированный	
Я бы:										
регистр Адрес	регистр Имя	Дефолт Ценить	7	6	5	4	3	2	1	0
0x7E	<a href="#">CMD</a>	0x00	cmd							
0x7D	<a href="#">PWR_CTL</a>	0x00	зарезервированный					acc_en	зарезервированный	aux_en
0x7C	<a href="#">PWR_CONF</a>	0x03	зарезервированный						fifo_self_wakeup	adv_power_save
0x7B	-	-	зарезервированный							
...	-	-	зарезервированный							
0x74	-	-	зарезервированный							
0x73	<a href="#">КОМПЕНСИРОВАТЬ_2</a>	0x00	off_acc_z							
0x72	<a href="#">КОМПЕНСИРОВАТЬ_1</a>	0x00	off_acc_y							
0x71	<a href="#">КОМПЕНСИРОВАТЬ_0</a>	0x00	off_acc_x							
0x70	<a href="#">NV_CONF</a>	0x00	зарезервированный				acc_off_en	i2c_wdt_en	i2c_wdt_цел	spi_en
0x6F	-	-	зарезервированный							
0x6E	-	-	зарезервированный							
0x6D	<a href="#">ACCSelfTest</a>	0x00	зарезервированный				acc_self_test_amp	acc_self_test_sign	зарезервированный	acc_self_test_en
0x6C	-	-	зарезервированный							
0x6B	<a href="#">IF_CONF</a>	0x00	зарезервированный			if_mode	зарезервированный			spi3
0x6A	<a href="#">NVM_CONFIG</a>	0x00	зарезервированный						nvm_progen	зарезервированный
0x69	-	-	зарезервированный							
...	-	-	зарезервированный							
0x60	-	-	зарезервированный							
0x5F	<a href="#">INTERNAL_ERROR</a>	0x00	зарезервированный					int_err_2	int_err_1	зарезервированный
0x5E	<a href="#">ХАРАКТЕРНАЯ ЧЕРТА_S_IN</a>	0x00	features_in							
0x5D	-	-	зарезервированный							
...	-	-	зарезервированный							
0x5A	-	-	зарезервированный							



0x59	<a href="#">INIT_CTRL</a> <a href="#">L</a>	0x90	init_ctrl								
0x58	<a href="#">INT_MAP</a> <a href="#">ДАННЫЕ</a>	0x00	зарезервированный	int2_drdy	int2_fwM	int2_full	зарезервированный	int1_drdy	int1_fwM	int1_full	
0x57	<a href="#">INT2_MAP</a> <a href="#">L</a>	0x00	error_int_из	no_motion_out	any_motion_out	зарезервированный					
0x56	<a href="#">INT1_MAP</a> <a href="#">L</a>	0x00	error_int_из	no_motion_out	any_motion_out	зарезервированный					
0x55	<a href="#">INT_LATCH</a> <a href="#">ЧАС</a>	0x00	зарезервированный								int_latch
0x54	<a href="#">INT2_IO_CTRL</a>	0x00	зарезервированный			input_en	output_en	od	lvl	edge_ctrl	
0x53	<a href="#">INT1_IO_CTRL</a>	0x00	зарезервированный			input_en	output_en	od	lvl	edge_ctrl	
0x52	-	-	зарезервированный								
...	-	-	зарезервированный								
0x50	-	-	зарезервированный								
0x4F	<a href="#">AUX_WRITE_DATA</a>	0x02	write_data								
0x4E	<a href="#">AUX_WRITE_ADDR</a>	0x4C	write_addr								
0x4D	<a href="#">AUX_READ_ADDR</a>	0x42	read_addr								
0x4C	<a href="#">AUX_INTERRUPT_ENABLE</a>	0x83	aux_manual_en	зарезервированный					aux_rd_burst		
0x4B	<a href="#">AUX_DEVICE_ADDR</a> <a href="#">8 БЫТ</a>	0x20	i2c_device_addr								зарезервированный
0x4A	-	-	зарезервированный								
0x49	<a href="#">FIFO_CONFIG_1</a>	0x10	зарезервированный	fifo_acc_en	fifo_aux_en	fifo_header_en	fifo_tag_int1_en	fifo_tag_int2_en	зарезервированный		
0x48	<a href="#">FIFO_CONFIG_0</a>	0x02	зарезервированный							fifo_time_en	fifo_stop_on_full
0x47	<a href="#">FIFO_WATERMARK_1</a>	0x02	зарезервированный			fifo_water_mark_12_8					
0x46	<a href="#">FIFO_WATERMARK_0</a>	0x00	fifo_water_mark_7_0								
0x45	<a href="#">FIFO_DOWN_SAMPLES</a>	0x80	acc_fifo_filter_data	acc_fifo_downs			зарезервированный				
0x44	<a href="#">AUX_OFFSET</a>	0x46	aux_offset				aux_odr				
0x43	-	-	зарезервированный								
0x42	-	-	зарезервированный								
0x41	<a href="#">ACC_RANGE</a>	0x01	зарезервированный							acc_range	
0x40	<a href="#">ACC_CONFIG</a>	0xA8	acc_perf_Режим	acc_bwp			acc_odr				
0x3F	-	-	зарезервированный								

...	-	-	резервированный							
0x2B	-	-	резервированный							
0x2A	<a href="#">ИНТЕРНА L_STATU S</a>	0x00	резервированный	odr_5 0Hz_e ppop	axes_re map_err или	сообщение				
0x29	-	-	резервированный							
...	-	-	резервированный							
0x27	-	-	резервированный							
0x26	<a href="#">FIFO_DAT A</a>	0x00	fifo_data							
0x25	<a href="#">FIFO_LEN GTH 1</a>	0x00	резервированный		fifo_byte_counter_13_8					
0x24	<a href="#">FIFO_LEN GTH 0</a>	0x00	fifo_byte_counter_7_0							
0x23	-	-	резервированный							
0x22	<a href="#">ТЕМПЕРА ТЮРЕ</a>	0x00	температура							
0x21	-	-	резервированный							
...	-	-	резервированный							
0x1E	-	-	резервированный							
0x1D	<a href="#">INT_STAT US 1</a>	0x00	acc_drd y_int	резерв ред	aux_drd y_int	резервированный		fwm_int	ffull_int	
0x1C	<a href="#">INT_STAT US 0</a>	0x00	error_int _из	no_mo tion_o ут	any_mot ion_out	резервированный				
0x1B	<a href="#">МЕРОПРИЯТИЕ</a>	0x01	резервированный							por_dete cted
0x1A	<a href="#">СЕНСОРТ ИМЕ 2</a>	0x00	sensor_time_23_16							
0x19	<a href="#">СЕНСОРТ ИМЕ 1</a>	0x00	sensor_time_15_8							
0x18	<a href="#">СЕНСОРТ ИМЕ 0</a>	0x00	sensor_time_7_0							
0x17	<a href="#">DATA 13</a>	0x00	acc_z_15_8							
0x16	<a href="#">DATA 12</a>	0x00	acc_z_7_0							
0x15	<a href="#">DATA 11</a>	0x00	acc_y_15_8							
0x14	<a href="#">DATA 10</a>	0x00	acc_y_7_0							
0x13	<a href="#">ДАННЫЕ 9</a>	0x00	acc_x_15_8							
0x12	<a href="#">DATA 8</a>	0x00	acc_x_7_0							
0x11	<a href="#">DATA 7</a>	0x00	aux_r_15_8							
0x10	<a href="#">DATA 6</a>	0x00	aux_r_7_0							
0x0F	<a href="#">DATA 5</a>	0x00	aux_z_15_8							
0x0E	<a href="#">DATA 4</a>	0x00	aux_z_7_0							
0x0D	<a href="#">ДАННЫЕ 3</a>	0x00	aux_y_15_8							
0x0C	<a href="#">ДАННЫЕ 2</a>	0x00	aux_y_7_0							
0x0B	<a href="#">ДАННЫЕ 1</a>	0x00	aux_x_15_8							
0x0A	<a href="#">DATA 0</a>	0x00	aux_x_7_0							
0x09	-	-	резервированный							
...	-	-	резервированный							

0x04	-	-	зарезервированный							
0x03	<a href="#">ПОЛОЖЕНИЕ ДЕЛ</a>	0x10	drdy_ac с	резерв ред	drdy_au Икс	cmd_rdy	зарезервированный	aux_ma H_оп	зарезервированный	
0x02	<a href="#">ERR_REG</a>	0x00	aux_err	fifo_er р	зарезервированный	код ошибки			cmd_err	fatal_err
0x01	-	-	зарезервированный							
0x00	<a href="#">CHIP_ID</a>	0x1A	chip_id							

## ХАРАКТЕРНАЯ ЧЕРТА\_S\_IN

регистр Адрес	Registe r Имя	Дефолт Ценить	7	6	5	4	3	2	1	0
0x5E: 0x0B	<a href="#">Общее настройки. axes_re отображение [1]</a>	0x00	зарезервированный							map_z_ axis_sig п
0x5E: 0x0A	<a href="#">Общее настройки. axes_re отображение [0]</a>	0x88	map_z_axis	map_y_ axis_sig п		map_y_axis	map_x_ axis_sig п		map_x_axis	
0x5E: 0x09	<a href="#">Общее настройки. бронировать d [1]</a>	0x00	Зарезервированный							
0x5E: 0x08	<a href="#">Общее настройки. бронировать d [0]</a>	0x00	Зарезервированный							
0x5E: 0x07	<a href="#">no moti on.settin gs 2 [1]</a>	0x00	z_en	y_en	x_en	продолжительность				
0x5E: 0x06	<a href="#">no moti on.settin gs 2 [0]</a>	0x05	продолжительность							
0x5E: 0x05	<a href="#">no moti on.settin gs 1 [1]</a>	0x00	зарезервированный					порог		
0x5E: 0x04	<a href="#">no moti on.settin gs 1 [0]</a>	0xAA	порог							
0x5E: 0x03	<a href="#">any mot ion.setti ngs 2 [1]</a>	0x00	z_en	y_en	x_en	продолжительность				
0x5E: 0x02	<a href="#">any mot ion.setti ngs 2 [0]</a>	0x05	продолжительность							
0x5E: 0x01	<a href="#">any mot ion.setti ngs 1 [1]</a>	0x00	зарезервированный					порог		

0x5E: 0x00	<a href="#">any_mot</a> <a href="#">ion.setti</a> <a href="#">ngs_1 [0]</a>	0xAA	порог
---------------	---	------	-------

## 5.2.1 Регистр (0x00) CHIP\_ID

ОПИСАНИЕ: СБРОС идентификационного

кода чипа: 0x1A

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Зарегистрируйте (0x00) CHIP_ID			
Немного	7	6	5	4
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	1
Содержание	chip_id			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	1	0	1	0
Содержание	chip_id			

chip\_id: Идентификационный код чипа для BMA490L

## 5.2.2 Регистр (0x02) ERR\_REG

ОПИСАНИЕ: Сообщает о состояниях ошибки датчика

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x02) ERR_REG			
Немного	7	6 5		4
Читай пиши	p	R н / д		p
Сбросить значение	0	0 0		0
Содержание	aux_err	fifo_err зарезервировано		код ошибки
Немного	3	2 1		0
Читай пиши	p	RR		p
Сбросить значение	0	0 0		0
Содержание	код ошибки		cmd_err	fatal_err

fatal\_err: фатальная ошибка, микросхема не в рабочем состоянии (загрузка, система питания). Этот флаг будет сброшен только при включении питания или программном сбросе.

cmd\_err: сбой при выполнении команды.

error\_code: коды ошибок для постоянных ошибок

код ошибки		
0x00	Нет ошибки	об ошибках не сообщается об
0x01	acc_err	ошибке в регистре ACC_CONF

fifo\_err: обнаружена ошибка в FIFO: входные данные были отброшены в потоковом режиме. Этот флаг будет сброшен при чтении.

aux\_err: обнаружена ошибка в I2C-Master. Этот флаг будет сброшен при чтении.

### 5.2.3 Регистр (0x03) СОСТОЯНИЕ

ОПИСАНИЕ: Флаги состояния датчика

СБРОС: 0x10

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x03) СТАТУС			
Немного	7	6	5	4
Читай пиши	p	н / д	p	p
Сбросить значение	0	0	0	1
Содержание	drdy_acc	зарезервированный	drdy_aux	cmd_rdy
Немного	3	2	1	0
Читай пиши	н / д	p	н / д	н / д
Сбросить значение	0	0	0	0
Содержание	зарезервированный	aux_man_op	зарезервированный	

aux\_man\_op: '1' ('0') означает, что (нет) выполняется ручная операция вспомогательного интерфейса.

cmd\_rdy: статус декодера CMD: 0 -> Выполняется команда 1 -> Декодер команд готов принять новую команду

drdy\_aux: Данные для дополнительного датчика готовы. Он сбрасывается при считывании одного вспомогательного регистра

DATA. Drdy\_acc: Данные готовы для акселерометра. Сбрасывается при считывании одного регистра ДАННЫХ акселерометра.

### 5.2.4 Регистр (0x0A) DATA\_0

ОПИСАНИЕ: AUX\_X (LSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x0A) DATA_0			
Немного	7	6 5		4
Читай пиши	p	RR		p
Сбросить значение	0	0 0		0
Содержание	aux_x_7_0			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_x_7_0			

## 5.2.5 Регистр (0x0B) DATA\_1

ОПИСАНИЕ: AUX\_X (MSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Зарегистрируйте (0x0B) DATA_1			
Немного	7	6	5	4
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_x_15_8			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_x_15_8			

## 5.2.6 Регистр (0x0C) DATA\_2

ОПИСАНИЕ: AUX\_Y (LSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x0C) DATA_2			
Немного	7	6 5		4
Читай пиши	p	RR		p
Сбросить значение	0	0 0		0
Содержание	aux_y_7_0			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_y_7_0			

## 5.2.7 Регистр (0x0D) DATA\_3

ОПИСАНИЕ: AUX\_Y (MSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x0D) DATA_3			
Немного	7	6 5		4
Читай пиши	p	RR		p
Сбросить значение	0	0 0		0
Содержание	aux_y_15_8			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_y_15_8			

## 5.2.8 Регистр (0x0E) DATA\_4

ОПИСАНИЕ: AUX\_Z (LSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x0E) DATA_4			
Немного	7	6	5	4
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_z_7_0			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_z_7_0			

## 5.2.9 Регистр (0x0F) DATA\_5

ОПИСАНИЕ: AUX\_Z (MSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x0F) DATA_5			
Немного	7	6 5		4
Читай пиши	p	RR		p
Сбросить значение	0	0 0		0
Содержание	aux_z_15_8			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_z_15_8			

## 5.2.10 Регистр (0x10) DATA\_6

ОПИСАНИЕ: AUX\_R (LSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x10) DATA_6			
Немного	7	6 5		4
Читай пиши	p	RR		p
Сбросить значение	0	0 0		0
Содержание	aux_r_7_0			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_r_7_0			

### 5.2.11 Регистр (0x11) DATA\_7

ОПИСАНИЕ: AUX\_R (MSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x11) DATA_7			
Немного	7	6	5	4
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_r_15_8			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	aux_r_15_8			

### 5.2.12 Регистр (0x12) DATA\_8

ОПИСАНИЕ: ACC\_X (LSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x12) DATA_8			
Немного	7	6 5		4
Читай пиши	p	RR		p
Сбросить значение	0	0 0		0
Содержание	acc_x_7_0			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	acc_x_7_0			

### 5.2.13 Регистр (0x13) DATA\_9

ОПИСАНИЕ: ACC\_X (MSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x13) DATA_9			
Немного	7	6 5		4
Читай пиши	p	RR		p
Сбросить значение	0	0 0		0
Содержание	acc_x_15_8			
Немного	3	2	1	0
Читай пиши	p	p	p	p
Сбросить значение	0	0	0	0
Содержание	acc_x_15_8			



### 5.2.14 Регистр (0x14) DATA\_10

ОПИСАНИЕ: ACC\_Y (LSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x14) DATA_10			
Немного	7	6	5	4
Читай пиши	r	r	r	r
Сбросить значение	0	0	0	0
Содержание	acc_y_7_0			
Немного	3	2	1	0
Читай пиши	r	r	r	r
Сбросить значение	0	0	0	0
Содержание	acc_y_7_0			

### 5.2.15 Регистр (0x15) DATA\_11

ОПИСАНИЕ: ACC\_Y (MSB)

RESET: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x15) DATA_11			
Немного	7	6 5		4
Читай пиши	r	RR		r
Сбросить значение	0	0 0		0
Содержание	acc_y_15_8			
Немного	3	2	1	0
Читай пиши	r	r	r	r
Сбросить значение	0	0	0	0
Содержание	acc_y_15_8			

### 5.2.16 Регистр (0x16) DATA\_12

ОПИСАНИЕ: ACC\_Z (LSB)

СБРОС: 0x00

ОПРЕДЕЛЕНИЕ (Перейти к [зарегистрировать карту](#)):

Имя	Регистр (0x16) DATA_12			
Немного	7	6 5		4
Читай пиши	r	RR		r
Сбросить значение	0	0 0		0
Содержание	acc_z_7_0			
Немного	3	2	1	0
Читай пиши	r	r	r	r
Сбросить значение	0	0	0	0
Содержание	acc_z_7_0			