

Современные 32-разрядные ARM-микроконтроллеры серии STM32.

Подключение акселерометра LIS302DL к микроконтроллеру

Олег Вальпа (sandh@narod.ru)

В статье приведено описание акселерометра LIS302DL и его подключения к микроконтроллеру серии STM32 компании STMicroelectronics, а также представлены примеры программ для практического использования данного акселерометра.

ВВЕДЕНИЕ

В настоящее время широкое распространение получили миниатюрные акселерометры в виде интегральных микросхем, которые позволяют измерять разность между ускорением объекта и гравитационным ускорением земли. На практике данные устройства применяются для регистрации ударных нагрузок, вибрации объектов, а также для определения ориентации объекта в пространстве.

Как правило, акселерометр представляет собой чувствительный элемент, закреплённый в упругом подвесе. Отклонение элемента от его первоначального положения при наличии ускорения позволяет измерять величину данного ускорения. По конструктивному исполнению акселерометры подразделяются на однокомпонентные, двухкомпонентные и трёхкомпонентные устройства, измеряющие ускорение вдоль одной, двух или трёх осей соответственно. Некоторые акселерометры имеют встроенные системы сбора и обработки данных, что позволяет создавать на их основе системы для измерения ускорения и вибрации со всеми необходимыми элементами.

Компания STMicroelectronics [1] производит двухкомпонентный акселерометр LIS202DL и трёхкомпонентный LIS302DL.

ОПИСАНИЕ АКСЕЛЕРОМЕТРА

Рассмотрим структуру, способ подключения и пример использования акселерометра LIS302DL. Акселерометр LIS302DL обладает следующими характеристиками:

- напряжение питания от 2,16 до 3,6 В;
- совместимость с сигналами 1,8 В;
- потребляемая мощность менее 1 мВт;
- цифровые интерфейсы связи SPI и I²C;
- измерение ускорения по трём осям;
- два диапазона измерения: $\pm 2g$ и $\pm 8g$;
- программируемый генератор прерываний;
- встроенный режим самодиагностики;
- обнаружение постукиваний;
- распознавание одиночных и двойных постукиваний;
- встроенный фильтр высоких частот;
- стойкость к ударным воздействиям до 10 000g;
- размеры 3×5×0,9 мм;
- корпус LGA14.

На рисунке 1 показаны внешний вид акселерометра LIS302DL, направление его осей измерения ускорения и нумерация выводов. Структурная схема акселерометра LIS302DL приведена на рисунке 2. Назначение выводов акселерометра LIS302DL представлено в таблице 1. Схема подключения

микроконтроллера STM32 к акселерометру показана на рисунке 3.

Взаимодействие микроконтроллера с акселерометром осуществляется посредством регистров. На рисунке 4 приведена временная диаграмма обмена данными с акселерометром через интерфейс SPI. На этом рисунке представлены сигналы интерфейса SPI, а также разряды данных и адреса регистров акселерометра.

Разряд R/W служит для задания режима чтения и записи. Если этот разряд равен нулю, то байт данных DI7-DI0 будет записан в регистр по адресу AD5..AD0, в противном случае, при R/W, равном 1, байт данных DO7-DO0 будет прочитан из регистра по адресу AD5..AD0. Разряд M/S используется для блочного чтения и записи нескольких регистров подряд. Если этот разряд равен нулю, то после передачи адреса будет читаться или записываться один и тот же регистр для любого количества операций чтения и записи. Если этот разряд будет равен единице, то адрес будет автоматически увеличиваться на единицу после каждой операции чтения или записи.

ОПЕРАЦИИ ЧТЕНИЯ И ЗАПИСИ

Чтение одного регистра акселерометра, например с адресом 0x01, осуществляется следующим образом. Необходимо установить сигнал CS равным нулю, затем передать первый байт 0x81, который содержит адрес считываемого регистра, единичный бит R/W и нулевой бит M/S. Затем следует передать произвольный байт, для того

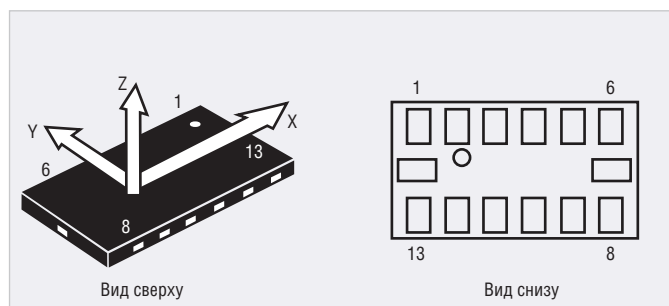


Рис. 1. Внешний вид акселерометра

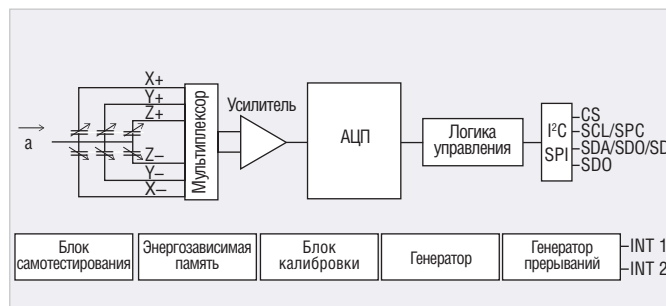


Рис. 2. Структурная схема акселерометра

Таблица 1. Назначение выводов акселерометра LIS302DL

Номер вывода	Имя	Назначение
1	Vdd_IO	Питание выводов I/O
2	GND	Общий вывод
3	Reserved	Резерв, подключается к Vdd
4	GND	Общий вывод
5	GND	Общий вывод
6	Vdd	Напряжение питания
7	CS	Выбор интерфейса (0 – SPI, 1 – I ² C)
8	INT1	Вывод прерывания 1
9	INT2	Вывод прерывания 2
10	GND	Общий вывод
11	Reserved	Резерв, подключается к GND
12	SDO	В режиме SPI – сигнал выходных данных; в режиме I ² C – младший бит адреса
13	SDI SDA SDO	В режиме SPI – сигнал входных данных; в режиме I ² C – сигнал данных; в режиме 3-проводного SPI – сигнал двустороннего обмена данными
14	SPC SCL	Вывод синхронизации SPI и I ² C

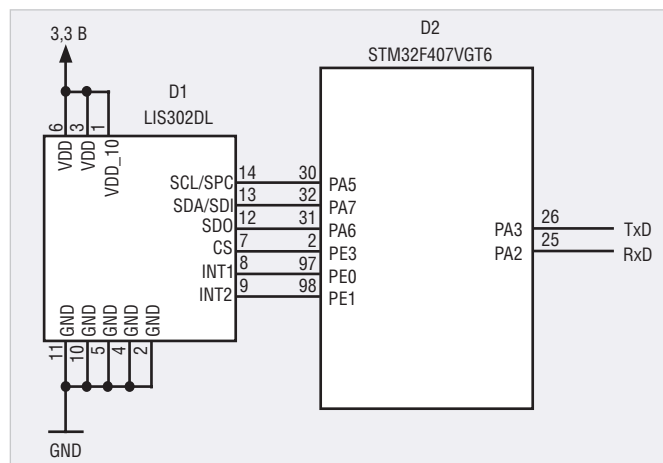


Рис. 3. Схема подключения микроконтроллера STM32 к акселерометру

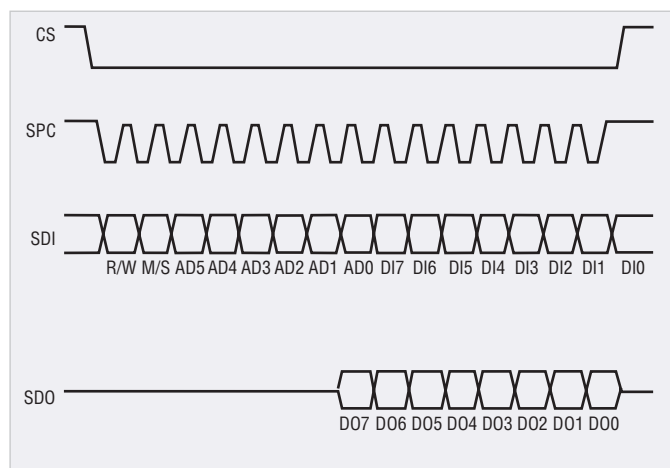


Рис. 4. Временная диаграмма

чтобы данные регистра от акселерометра поступили через интерфейс SPI в микроконтроллер, после чего необходимо сохранить принятые данные регистра и установить высокий логический уровень на выводе CS. Для регулярного чтения одного и того же регистра можно постоянно повторять операции получения данных, не изменяя сигнал CS.

Запись одного регистра, например с адресом 0x01, производится установкой сигнала CS равным нулю, затем передаются первый байт 0x01, который содержит адрес записываемого регистра, нулевой бит R/W и нулевой бит M/S, после чего необходимо передать новое значение регистра. Завершается операция установкой высокого логического уровня на выводе CS. Для регулярной записи данных в один и тот же регистр можно постоянно повторять операции записи, не изменяя сигнал CS.

Чтение нескольких регистров акселерометра, например начиная с адреса 0x01, выполняется в следующем порядке. Сигнала CS устанавлива-

ется в нулевое состояние, затем передаются первый байт 0xC1, который содержит адрес считываемого регистра, единичный бит R/W и единичный бит M/S. После этого необходимо последовательно передать произвольный байт, для того чтобы данные регистров от акселерометра поступали через интерфейс SPI в микроконтроллер. Адреса регистров при этом будут автоматически увеличиваться на единицу. При этом необходимо сохранять принимаемые данные регистров. Для завершения чтения следует установить высокий логический уровень на выводе CS.

Запись нескольких регистров производится аналогично чтению нескольких регистров, т.е. после каждой записи адрес автоматически увеличивается на единицу. Например, для записи трёх байт 0xD1, 0xD2 и 0xD3 в регистры с адресами 0x01, 0x02 и 0x03 следует установить сигнал CS равным нулю, затем передать первый байт 0x41, который содержит адрес записываемого регистра, нулевой бит R/W и единичный бит M/S. После этого

необходимо последовательно передать байты 0xD1, 0xD2 и 0xD3, которые автоматически запишутся в регистры с адресами 0x01, 0x02 и 0x03. Для завершения записи следует установить высокий логический уровень на выводе CS.

РЕГИСТРЫ АКСЕЛЕРОМЕТРА

Рассмотрим назначение основных регистров акселерометра, при помощи которых задаются режимы его работы и производится измерение ускорений. Регистр, содержащий идентификатор акселерометра LIS302DL, имеет название WhoAmI и адрес 0x0F. Его значение можно только прочитать. При чтении данный регистр возвращает код 0x3B, который идентифицирует акселерометр как LIS302DL.

Первый регистр настройки имеет имя CTRL_REG1 и адрес 0x20. Назначение его разрядов приведено в таблице 2. В таблице рядом с именем регистра в скобках указано значение регистра по умолчанию после подачи питания или сброса. Второй регистр настройки имеет имя CTRL_REG2 и адрес 0x21.

Таблица 2. Назначение разрядов регистра CTRL_REG1

Разряд	Имя	Назначение
7	DR (0)	Выбор частоты выборки: 0 – 100 Гц, 1 – 400 Гц
6	PD (0)	Управление питанием: 0 – питание отключено, 1 – питание включено
5	FS (0)	Выбор диапазона измерений: 0 – диапазон ±2g, 1 – диапазон ±8g
4	STP (0) STM (0)	Управление режимом самодиагностики: STP 0, STM 0 – самодиагностика выключена; STP 0, STM 1 – самодиагностика в режиме P; STP 1, STM 0 – самодиагностика в режиме M; STP 1, STM 1 – запрещённая комбинация
3		
2	Zen (1)	Разрешение генерации сигнала готовности данных для оси Z: 0 – запрещено, 1 – разрешено
1	Yen (1)	Разрешение генерации сигнала готовности данных для оси Y: 0 – запрещено, 1 – разрешено
0	Xen (1)	Разрешение генерации сигнала готовности данных для оси X: 0 – запрещено, 1 – разрешено

Таблица 3. Назначение разрядов регистра CTRL_REG3

Разряд	Имя	Назначение			
7	SIM (0)	Выбор режима SPI: 0 – 4-проводной режим, 1 – 3-проводной режим			
6	BOOT (0)	Сброс регистров в исходное состояние: 0 – работа, 1 – сброс			
5	–	Не используется			
4	FDS (0)	Выбор фильтрации данных: 0 – фильтрация притяжения земли отключена, 1 – фильтрация включена			
3	HP_FF_WU2 (0)	Управление фильтром верхних частот для модуля генерации прерывания FF_WU2: 0 – отключён, 1 – включён			
2	HP_FF_WU1 (0)	Управление фильтром верхних частот для модуля генерации прерывания FF_WU1: 0 – отключён, 1 – включён			
1 0	HP_coeff2 (0) HP_coeff1 (0)	Управление частотой среза фильтра верхних частот: Разрешение генерации сигнала готовности данных для оси X:			
		HP_coeff2	HP_coeff1	Скорость 100 Гц	Скорость 400 Гц
		0	0	2	8
		0	1	1	4
		1	0	0,5	2
		1	1	0,25	1

Таблица 5. Назначение разрядов регистра STATUS_REG

Разряд	Имя	Назначение
7	ZXYOR (0)	Данные в регистрах OUT_X, OUT_Y и OUT_Z обновлены новыми значениями, а их предыдущее содержимое не было прочитано
6	ZOR (0)	Данные в регистре OUT_Z обновлены до прочтения
5	YOR (0)	Данные в регистре OUT_Y обновлены до прочтения
4	XOR (0)	Данные в регистре OUT_X обновлены до прочтения
3	ZYXDA (0)	Доступны новые данные в регистрах OUT_X, OUT_Y и OUT_Z
2	ZDA (0)	Доступны новые данные в регистре OUT_Z
1	YDA (0)	Доступны новые данные в регистре OUT_Y
0	ZDA (0)	Доступны новые данные в регистре OUT_X

Таблица 7. Назначение разрядов регистров FF_WU_SRC_1 и FF_WU_SRC_2

Разряд	Имя	Назначение
7	–	Не используется
6	IA (0)	Активность прерывания: 0 – нет, 1 – одно из прерываний произошло
5	ZH (0)	Уровень по оси Z выше порога: 0 – нет, 1 – да
4	ZL (0)	Уровень по оси Z ниже порога: 0 – нет, 1 – да
3	YH (0)	Уровень по оси Y выше порога: 0 – нет, 1 – да
2	YL (0)	Уровень по оси Y ниже порога: 0 – нет, 1 – да
1	XH (0)	Уровень по оси X выше порога: 0 – нет, 1 – да
0	XL (0)	Уровень по оси X ниже порога: 0 – нет, 1 – да

Назначение его разрядов приведено в таблице 3.

Третий регистр акселерометра служит для настройки прерываний. Он имеет имя CTRL_REG3 и адрес 0x22. Назначение его разрядов приведено в таблице 4. Сигналы FF_WU_1 и FF_WU_2 формируются двумя независимыми блоками и предназначены для генерации прерываний, в случае если ускорение по одной или нескольким осям выходит за заданный порог.

Следующий регистр HP_FILTER_RESET по адресу 0x23 называется фиктивным. Чтение этого регистра обнуляет содержимое регистров OUT_X, OUT_Y и OUT_Z внутреннего высокочастотного фильтра, т.е. значения ускорений по всем трём осям одновременно принимают значения 0g.

Регистр статуса STATUS_REG с адресом 0x27 содержит флаги, сигнализирующие об определённых событиях. Назначение флагов приведено в таблице 5.

Регистры данных с именами OUT_X, OUT_Y, OUT_Z, размещённые по адресам 0x29, 0x2B, 0x2D, содержат текущие данные измерений ускорения по осям X, Y и Z соответственно.

Акселерометр имеет два независимых модуля, которые самостоятельно читают данные из регистров OUT_X, OUT_Y, OUT_Z, и на их основе при определённых условиях генерируются прерывания. Для настройки каждого модуля используются четыре регистра, которые описаны ниже. Регистр FF_WU_CFG_1 по адресу 0x30 и регистр FF_WU_CFG_2 по адресу 0x34 позво-

Таблица 4. Назначение разрядов регистра CTRL_REG3

Разряд	Имя	Назначение			
7	IHL (0)	Управление уровнем активного сигнала прерывания: 0 – высокий уровень, 1 – низкий уровень			
6	PP_OD (0)	Тип выводов INT1 и INT2: 0 – двухтактный выход, 1 – открытый коллектор			
5 4 3	I2CFG2 (0) I2CFG1 (0) I2CFG0 (0)	Управление выводом INT2:			
		I2CFG2	I2CFG1	I2CFG0	Источник сигнала
		0	0	0	Отключён и заземлён
		0	0	1	FF_WU_1
		0	1	0	FF_WU_2
		0	1	1	FF_WU_1 или FF_WU_2
1	0	0	Готовность данных		
1	1	1	Прерывание от щелчка		
2 1 0	I1CFG2 (0) I1CFG1 (0) I1CFG0 (0)	Управление выводом INT1:			
		I1CFG2	I1CFG1	I1CFG0	Источник сигнала
		0	0	0	Отключён и заземлён
		0	0	1	FF_WU_1
		0	1	0	FF_WU_2
		0	1	1	FF_WU_1 или FF_WU_2
		1	0	0	Готовность данных
		1	1	1	Прерывание от щелчка

Таблица 6. Назначение разрядов регистров FF_WU_CFG_1 и FF_WU_CFG_2

Разряд	Имя	Назначение
7	AOI (0)	Выбор логической комбинации ИЛИ/И между битами XHIE, YHIE, ZHIE, XLIE, YLIE и ZLIE для генерации прерывания: 0 – логическое ИЛИ, 1 – логическое И
6	LIR (0)	Защёлка запроса прерывания: 0 – запрос не фиксируется, 1 – запрос фиксируется
5	ZHIE (0)	Разрешение прерывания при ускорении выше порога по оси Z: 0 – разрешено, 1 – запрещено
4	ZLIE (0)	Разрешение прерывания при ускорении ниже порога по оси Z: 0 – разрешено, 1 – запрещено
3	YHIE (0)	Разрешение прерывания при ускорении выше порога по оси Y: 0 – разрешено, 1 – запрещено
2	YLIE (0)	Разрешение прерывания при ускорении ниже порога по оси Y: 0 – разрешено, 1 – запрещено
1	XHIE (0)	Разрешение прерывания при ускорении выше порога по оси X: 0 – разрешено, 1 – запрещено
0	XLIE (0)	Разрешение прерывания при ускорении ниже порога по оси X: 0 – разрешено, 1 – запрещено

Таблица 8. Назначение разрядов регистров FF_WU_THS_1 и FF_WU_THS_2

Разряд	Имя	Назначение
7	DCRM (0)	Режима сброса счетчика: 0 – счётчик сбрасывается, 1 – счётчик уменьшается
6	THS6 (0)	Порог ускорения
5	THS5 (0)	
4	THS4 (0)	
3	THS3 (0)	
2	THS2 (0)	
1	THS1 (0)	
0	THS0 (0)	

ляют настроить условия возникновения прерывания для первого и второго модуля соответственно. Назначение их разрядов приведено в таблице 6.

Регистр FF_WU_SRC_1 по адресу 0x31 и регистр FF_WU_SRC_2 по адресу 0x35 являются статусными и позволяют произвести анализ возникновения прерывания для первого и второго модуля соответственно. Назначение разрядов этих регистров приведено в таблице 7.

Регистр FF_WU_THS_1 по адресу 0x32 и регистр FF_WU_THS_2 по адресу 0x36 служат для задания порогового значения ускорения. Назначение разрядов этих регистров приведено в таблице 8.

Регистры FF_WU_DURATION_1 по адресу и FF_WU_DURATION_1 по адресу 0x37 задают минимальную продолжительность события. При частоте выбор-

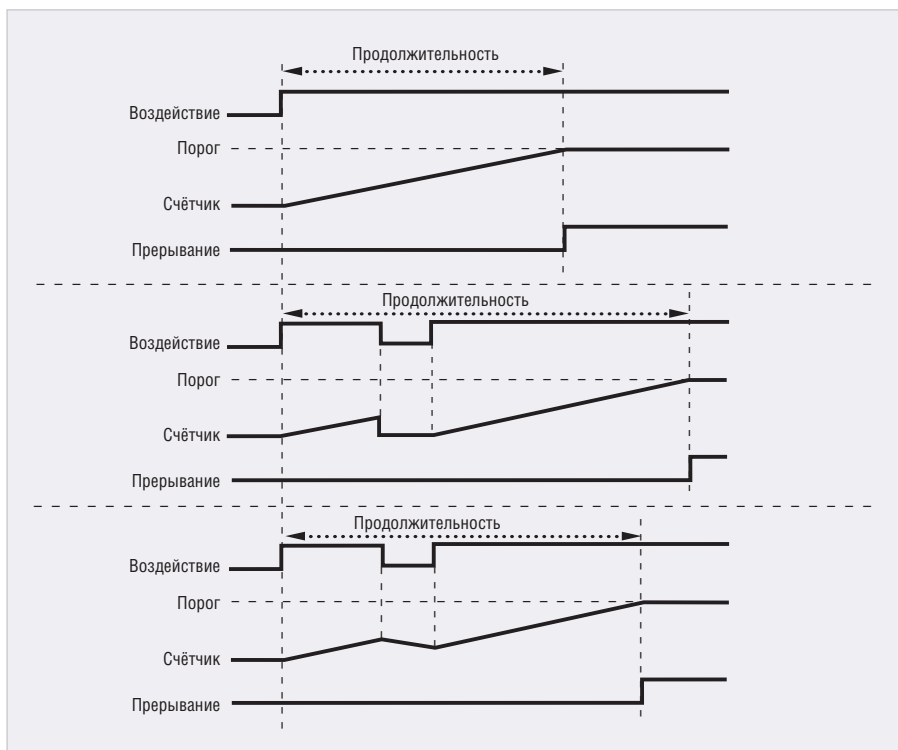


Рис. 5. Диаграммы формирования прерывания

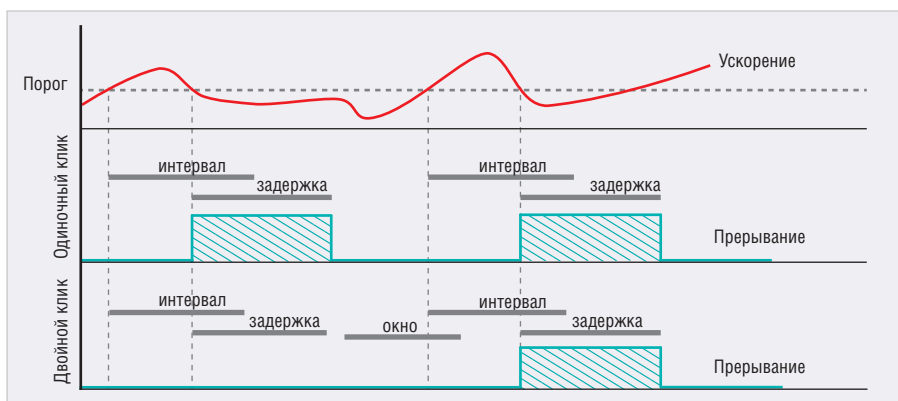


Рис. 7. Диаграммы формирования прерываний для одиночного и двойного кликов

ки 400 Гц регистры позволяют задавать значение от 0 до 637,5 мс с шагом 2,5 мс. При частоте выборки 100 Гц значение можно изменять в диапазоне от 0 до 2,55 с с шагом 10 мс. Назначение этих регистров проиллюстрировано временными диаграммами на рисунке 5.

При возникновении физического воздействия, например при ускорении по оси X больше порогового, в акселерометре запускается счётчик. Когда значение этого счётчика достигает значения регистра FF_WU_DURATION_1 или FF_WU_DURATION_2, формируется прерывание в виде смены логического уровня на выводах INT1 или INT2 соответственно.

В случае если воздействие на акселерометр, вызывающее прерывание, непродолжительно, например ускорение

уменьшилось и не достигло порогового, возможны два варианта развития событий. При сброшенном разряде DCRM счётчик обнулится и начнёт отсчёт с нуля при очередном превышении порога ускорения. Если разряд DCRM установлен в единичное состояние, то счётчик начнёт уменьшать своё значение. Таким способом осуществляется защита от ложных срабатываний. Если значение регистра FF_WU_DURATION_1 или FF_WU_DURATION_2 равно нулю, то прерывание формируется по событию без задержек.

Остальные регистры акселерометра предназначены для настройки блока, генерирующего прерывания при одиночном или двойном ударном воздействии на акселерометр, в результате которого ускорение по какой-либо оси превышает заданный порог CLICK_THS

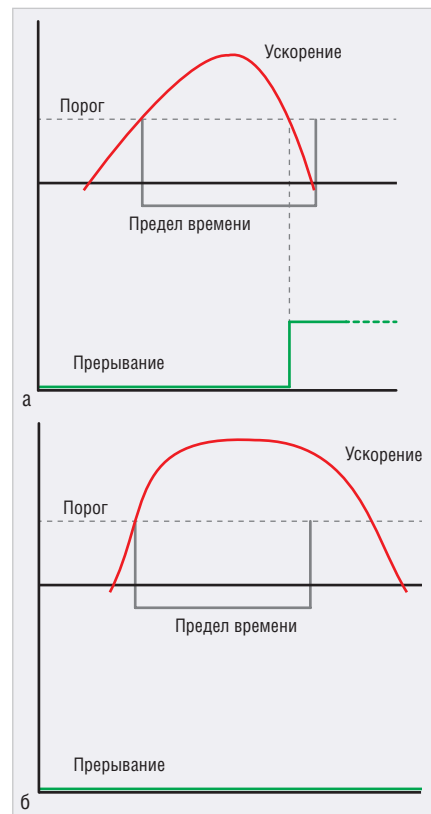


Рис. 6. Диаграммы формирования прерываний при различных воздействиях: а) прерывание происходит; б) прерывания не происходит

и опускается ниже этого порога раньше времени, заданном значением CLICK_TIMELIMIT. Такие воздействия называются кликами. На рисунке 6 приведены две диаграммы при различных воздействиях ускорения на акселерометр. На рисунке 6а видно, как ускорение плавно нарастает и превышает пороговый уровень, а затем так же плавно снижается. При этом ускорение пересекается с заданным порогом в отведённый предел времени. В таком случае формируется прерывание клика. На рисунке 6б изображён затянувшийся во времени клик: ускорение довольно медленно снижалось и пересеклось с порогом позже отпущенного времени. В данном случае прерывания клика не произошло.

Двойной клик представляет собой два одиночных клика, причём второй клик должен быть совершён до истечения временного окна CLICK_WINDOW с момента первого клика. На рисунке 7 показаны графики формирования прерываний для одиночного и двойного кликов при одинаковом изменении ускорения.

Регистр для выбора осей, по которым отслеживаются клики, называется CLICK_CFG и имеет адрес 0x38. Назначение его разрядов приведе-

Таблица 9. Назначение разрядов регистра CLICK_CFG

Разряд	Имя	Назначение
7	–	Не используется
6	LIR (0)	Защёлка запроса прерывания: 0 – запрос не фиксируется, 1 – запрос фиксируется
5	Double_Z (0)	Отслеживать двойные клики по оси Z: 0 – нет, 1 – да
4	Single_Z (0)	Отслеживать одиночные клики по оси Z: 0 – нет, 1 – да
3	Double_Y (0)	Отслеживать двойные клики по оси Y: 0 – нет, 1 – да
2	Single_Y (0)	Отслеживать одиночные клики по оси Y: 0 – нет, 1 – да
1	Double_X (0)	Отслеживать двойные клики по оси X: 0 – нет, 1 – да
0	Single_X (0)	Отслеживать одиночные клики по оси X: 0 – нет, 1 – да

Таблица 10. Назначение разрядов регистра CLICK_SRC

Разряд	Имя	Назначение
7	–	Не используется
6	IA (0)	Активность прерывания: 0 – нет, 1 – одно из прерываний произошло
5	Double_Z (0)	Произошло прерывание двойного клика по оси Z: 0 – нет, 1 – да
4	Single_Z (0)	Произошло прерывание одиночного клика по оси Z: 0 – нет, 1 – да
3	Double_Y (0)	Произошло прерывание двойного клика по оси Y: 0 – нет, 1 – да
2	Single_Y (0)	Произошло прерывание одиночного клика по оси Y: 0 – нет, 1 – да
1	Double_X (0)	Произошло прерывание двойного клика по оси X: 0 – нет, 1 – да
0	Single_X (0)	Произошло прерывание одиночного клика по оси X: 0 – нет, 1 – да

Листинг 1

```
#include <lib_func.c> // Подключить библиотеку функций
int main(void) { // Главный модуль программы
char str[4]; // Буфер строки
usart_init(); // Инициализация UART
spi_init(); // Инициализация SPI
setReg(LIS302DL_CTRL_REG1, (1<<PD_CTRL_REG1)); // Включить акселерометр
while(1) { // Циклическое чтение ускорения из трёх регистров и вывод в UART
delay(); // Задержка для очередного измерения
itoa((int8_t) getReg(LIS302DL_OUT_X), &str); send_str(&str); send_str(",");
itoa((int8_t) getReg(LIS302DL_OUT_Y), &str); send_str(&str); send_str(",");
itoa((int8_t) getReg(LIS302DL_OUT_Z), &str); send_str(&str); send_str("\n");
}
```

Листинг 2

```
#include <lib_func.c> // Подключить библиотеку функций
int main(void) { // Главный модуль программы
char str[4]; // Буфер строки
usart_init(); // Инициализация UART
spi_init(); // Инициализация SPI
enable_int(); // Разрешить прерывание по нарастающему фронту на выводе PEO
getReg(LIS302DL_FF_WU_SRC1_REG); // Читать регистр для его обнуления
// Включить акселерометр
setReg(LIS302DL_CTRL_REG1, (1<<PD_CTRL_REG1) | (1<<ZEN_CTRL_REG1) |
(1<<XEN_CTRL_REG1) | (1<<YEN_CTRL_REG1));
// Включить фильтр для исключения влияния ориентации акселерометра в
пространстве на измерения
setReg(LIS302DL_CTRL_REG2, (1<<FDS_CTRL_REG2));
// Разрешить прерывания от FF_WU_1 на выводе INT1
setReg(LIS302DL_CTRL_REG3, (1<<I1CFG0_CTRL_REG3));
setReg(LIS302DL_FF_WU_THS1_REG, 55); // Задать пороговое значение
// Разрешить прерывание при превышении ускорения по любой оси порогового
значения
setReg(LIS302DL_FF_WU_CFG1_REG, (1<<AOI_FF_WU_CFG_REG) | (1<<LIR_FF_WU_CFG_REG)
|
(1<<ZHIE_FF_WU_CFG_REG) | (1<<YHIE_FF_WU_CFG_REG) | (1<<XHIE_FF_WU_CFG_REG));
while(1){}; // Ждать прерывания
}
```

Листинг 3

```
#include <lib_func.c> // Подключить библиотеку функций
int main(void) { // Главный модуль программы
char str[4]; // Буфер строки
usart_init(); // Инициализация UART
spi_init(); // Инициализация SPI
enable_int(); // Разрешить прерывание по нарастающему фронту на выводе PEO
getReg(LIS302DL_FF_WU_SRC1_REG); // Читать регистр для его обнуления
// Включить акселерометр
setReg(LIS302DL_CTRL_REG1, (1<<PD_CTRL_REG1) | (1<<ZEN_CTRL_REG1) |
(1<<XEN_CTRL_REG1) | (1<<YEN_CTRL_REG1));
// Включить фильтр для исключения влияния ориентации акселерометра в
пространстве на измерения
setReg(LIS302DL_CTRL_REG2, (1<<FDS_CTRL_REG2));
// Разрешить прерывания от FF_WU_1 на выводе INT1
setReg(LIS302DL_CTRL_REG3, (1<<I1CFG0_CTRL_REG3) | (1<<I1CFG1_CTRL_REG3)
| (1<<I1CFG2_CTRL_REG3));
setReg(LIS302DL_CLICK_THSZ_REG, 0x0F); // Задать пороговое значение по оси Z
setReg(LIS302DL_CLICK_THSY_X_REG, 0xFF); // Задать пороговое значение по оси Y
и X
// Разрешить прерывание при двойном клике по оси Z
setReg(LIS302DL_CLICK_CFG_REG, (1<<LIR_CLICK_CFG_REG | (1<<Double_Z_CLICK_CFG_
REG) ));
// Заполнить регистры TIMELIMIT и LATENCY
setReg(LIS302DL_CLICK_TIMELIMIT_REG, 0x10);
setReg(LIS302DL_CLICK_LATENCY_REG, 0x10);
// Задать временное окно между кликами
setReg(LIS302DL_CLICK_WINDOW_REG, 0xFF);
while(1){}; // Ждать прерывания
}
```

но в таблице 9. Если разряд LIR установлен как 1, то при чтении регистра CLICK_SRC его содержимое обнулится, а логический уровень на выводах INT1 и INT2 будет сброшен в исходное состояние. Если разряд LIR установлен как 0, то логический уровень на выводах INT1 и INT2 сбросится автоматически через интервал времени, заданный регистром CLICK_LATENCY.

Регистр CLICK_SRC по адресу 0x39 является статусным. Назначение его разрядов приведено в таблице 10.

Регистр CLICK_THSY_X по адресу 0x3B задаёт пороговые значения ускорения для осей Y и X. Если ускорение превысит установленный порог, то акселерометр активирует режим обнаружения кликов. Младшие четыре бита регистра соответствуют порогу для оси X, старшие – для оси Y. Пороговое значение ускорения можно задавать для каждой оси в диапазоне от 0 до 15. Единица в младшем разряде регистра соответствует ускорению 0,5g. Таким образом, порог ускорения может быть задан в диапазоне от 0g до 7,5g. Аналогичную функцию выполняет регистр CLICK_THSZ по адресу 0x3C. В данном регистре для задания величины порога по оси Z используются только 4 младших разряда.

Регистр CLICK_TimeLimit по адресу 0x3D служит для задания временного интервала, в течение которого ускорение должно успеть снизиться ниже порогового, чтобы событие клика было обнаружено. Отсчёт времени начинается после того, как ускорение по выбранной оси превысило порог. Интервал можно задавать в диапазоне от 0 до 127,5 мс с шагом 0,5 мс.

Регистр CLICK_Latency по адресу 0x3E позволяет задать временной интервал, в течение которого клики не будут детектироваться. Отсчёт времени начинается после совершения одиночного или двойного клика. Интервал можно задавать в диапазоне от 0 до 255 мс с шагом в 1 мс.

Регистр CLICK_Window по адресу 0x3F определяет максимальный промежуток между двумя кликами в диапазоне от 0 до 255 мс. Чем меньше значение этого регистра, тем быстрее должен выполняться двойной клик для возникновения прерывания.

ПРИМЕРЫ ПРОГРАММ

Рассмотрим несколько примеров программ, иллюстрирующих использование акселерометра. В качестве среды разработки используется IDE CoCoX [2],

которая распространяется бесплатно и включает в себя удобный редактор кода и отладчик.

Все приведённые далее примеры используют готовые библиотеки STM, входящие в состав среды разработки. Назначение этих библиотек понятно из их названий. Например, библиотека `stm32f4xx_gpio.h` служит для настройки портов GPIO, а библиотека `stm32f4xx_rtc.h` – для настройки часов реального времени и т.д.

В первом примере (см. листинг 1) микроконтроллер читает данные измерения ускорения из акселерометра и транслирует их в последователь-

ный порт UART, который необходимо настроить на скорость 9600 бод с одним стоп-битом без проверки чётности.

В примерах, приведённых в листингах 2 и 3, последовательный порт UART можно не подключать, а для контроля работы использовать светодиодный индикатор, подключённый к порту PD15 микроконтроллера через ограничительный резистор 680 Ом. Пример программы в листинге 2 демонстрирует настройку прерывания от блока `FF_WU_1`. Программа, приведённая в листинге 3, позволяет с помощью акселерометра обнаруживать двойной клик и формировать соответствующее

прерывание. Содержимое файла `lib_func.c`, включающего в себя библиотеку функций, используемых в приведённых выше программах, доступно в дополнительных материалах к статье на сайте www.soel.ru.

Более подробное описание акселерометра LIS302DL приведено в документации производителя [3].

ЛИТЕРАТУРА

1. www.st.com
2. <http://www.coocox.org/software/coide.php>
3. <http://www.st.com/en/mems-and-sensors/lis302dl.html>



НОВОСТИ МИРА

Минкомсвязь России и Росприроднадзор подписали соглашения по использованию ИИТ

Минкомсвязь и Росприроднадзор подписали с компаниями «Еврохим» и «Фортум» соглашения по использованию промышленного Интернета вещей (IIoT) для автоматического контроля выбросов загрязняющих веществ в воздух. Соглашения подписаны в рамках пилотного проекта по автоматизации контрольно-надзорной деятельности, который проводят Минкомсвязь России и Росприроднадзор для реализации приоритетной программы «Реформа контрольной и надзорной деятельности».



Компании-участники пилотного проекта имеют высокий уровень технологического оснащения, в том числе приборами автоматического контроля физических параметров источников выбросов загрязняющих веществ в атмосферу. Росприроднадзор оснастит пост автоматического контроля выбросов на территории компании оборудованием, которое будет собирать данные о выбрасываемых загрязняющих веществах. Затем данные мониторинга будут переданы в реестр объектов, оказывающих негативное воздействие на окружающую среду. Мониторинг поступления информации о выбрасываемых загрязняющих веществах от источников выбросов обеспечит Минкомсвязь России.

По итогам пилотного проекта Минкомсвязь России выработает общие требования по использованию органами контроля и надзора данных, полученных с использованием IIoT. Это по-

зволит снизить количество выездных проверочных мероприятий и повысит качество оценки возникающих рисков нарушения требований.

Минкомсвязь России

Вышел новый каталог «Технологическое оборудование для микроэлектроники 2018»

Компания «Диполь» сообщает о публикации нового каталога «Технологическое оборудование для микроэлектроники 2018». В новой редакции представлены решения и оборудование по следующим направлениям:

- плазменные процессы;
- фотолитография;
- термические процессы;

- вакуумное напыление;
- шлифовка и полировка;
- дисковая резка и скрайбирование;
- микросварка и монтаж;
- аналитическое оборудование;
- оснастка для работы с подложками.

В каталоге подробно рассказывается о новинках оборудования для микроэлектроники, среди которых универсальные установки совмещения и экспонирования серии AG500, новые модели установок безмасковой литографии Miva, оборудование для уточнения, шлифовки и полировки материалов, большое количество аналитического оборудования, включая эллипсометры серии Semilab SE-1000, Semilab SE-2000 и профилометры модели MicroProf.

www.dipaul.ru

IF/RF & Microwave Design

Advantex

СВЧ-блоки

Синтезаторы частот до 20 ГГц, -140 дБн/Гц @ 1 ГГц, при отстройке 10 кГц, умножители частоты
Аттенюаторы, управляемые напряжением до 26 ГГц
Усилители до 20 ГГц, 0.5 Вт
I/Q модуляторы / демодуляторы, смесители, фильтры

Приборы

Генераторы сигналов, электронные аттенюаторы, I/Q-модуляторы

www.advantex.ru

Реклама