Свежие комментарии

- Семен к записи AVR Урок 15. Внутренняя энергонезависимая память EEPROM. Часть 4
- Стас к записи STM Урок 64. HAL. LTDC. Часть 2
- Семен к записи AVR Урок 15. Внутренняя энергонезависимая память EEPROM. Часть 4
- Семен к записи AVR Урок 15. Внутренняя энергонезависимая память EEPROM. Часть 4
- Boba к записи AVR Урок 16. Интерфейс TWI (I2C). Часть 2

Форум. Последние ответы

- alexander в Программирование МК STM32
  5 дн., 21 час. назад
- Narod Stream в Программирование МК STM32
  4 нед., 1 день назад
- Zandy в Программирование МК STM32
  1 месяц назад
- Narod Stream в Программирование МК STM32
  1 месяц, 1 неделя назад
- Narod Stream в Программирование МК STM32
  1 месяц, 1 неделя назад

Февраль 2018

Пн	Вт	Ср	Чт	Пт	Сб	Вс
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

« Янв

Архивы

- Февраль 2018
- Январь 2018

AVR УРОК 39. Акселерометр LSM6DS3. Часть 3

Posted on Январь 16, 2017 by Narod Stream
Опубликовано в Программирование AVR — 4 комментария ↓

Урок 39
Часть 3

Акселерометр LSM6DS3

Продолжаем работать с подключением акселерометра.

В **прошлой части занятия** мы уже написали некоторый код и считали идентификатор датчика, что даёт нам основание считать, что шину мы настроили правильно, и

Мета

- Регистрация
- Войти
- RSS записей
- RSS комментариев
- WordPress.org

искать здесь …
Фильтровать

Уроки по программированию МК

Программирование МК AVR

Заходите на канал Narod Stream

- Декабрь 2017
- Ноябрь 2017
- Октябрь 2017
- Сентябрь 2017
- Август 2017
- Июль 2017
- Июнь 2017
- Май 2017
- Март 2017
- Февраль 2017
- Январь 2017
- Декабрь 2016
- Ноябрь 2016

что мы общаемся именно с тем датчиком.

Сегодня мы продолжим писать код инициализаци.

Теперь начнем инициализировать регистры. Здесь нам уже придётся писать в шину I2C.

Но и здесь, зная, как работает шина, мы без труда напишем функцию для записи значения в регистр. Тем более, нам писать придется всегда только по одному байту

```
//---------------
static void
I2Cx_WriteData(unsigned
char Addr, unsigned char
Reg, unsigned char
Value)
{

I2C_StartCondition();//
Отправим условие START
  I2C_SendByte(Addr);
  I2C_SendByte(Reg);
  I2C_SendByte(Value);
  I2C_StopCondition();//
Отправим условие STOP
}
//---------------
unsigned char
Accel_ReadID(void)
```

Добавим функцию для инициализации регистров

```
//---------------
void AccInit(void)
{
}
//---------------
void Accel_Ini(void)
```

Вызовем её в главной функции инициализации

```
  if(read_buf[0]!=0x69)
Error();
  AccInit();
}
```

Начнем писать тело функии инициализации регистров, используя функции чтения и записи шины I2C и видоизменяя определённые регистры датчика таким же образом, как мы делали и в уроке по контроллеру STM

```
void AccInit(void)
{
  //автоувеличение
адреса регистра
```


Рубрики

- 1-WIRE (3)
- ADC (6)
- DAC (4)
- GPIO (26)
- I2C (19)
- SPI (13)
- USART (8)
- Программирование AVR (131)
- Программирование PIC (8)
- Программирование STM32 (217)
- Тесты устройств и аксессуаров (1)

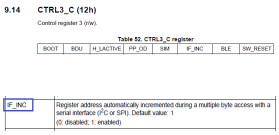
```
I2Cx_ReadData(0xD4,LSM6DS3_ACC_GYRO_CTRL3_C,1,read_buf);

read_buf[0]&=~LSM6DS3_ACC_GYRO_IF_INC_MASK;

read_buf[0]|=LSM6DS3_ACC_GYRO_IF_INC_ENABLED;

I2Cx_WriteData(0xD4,LSM6DS3_ACC_GYRO_CTRL3_C,read_buf[0]);
```

Данным кодом мы обращаемся к управляющему регистру 3 и включаем там бит автоувеличения адреса регистра



9.14 CTRL3_C (12h)

Control register 3 (rw).

Table 52. CTRL3_C register

BOOT	BDU	H_LACTIVE	PP_OD	SIM	IF_INC	BLE	SW_RESET

IF_INC	Register address automatically incremented during a multiple byte access with a serial interface (I²C or SPI). Default value: 1 (0: disabled; 1: enabled)

Продолжим дальше работать с регистрами управления

```
I2Cx_WriteData(0xD4,LSM6DS3_ACC_GYRO_CTRL3_C,read_buf[0]);
//установим бит BDU
I2Cx_ReadData(0xD4,LSM6DS3_ACC_GYRO_CTRL3_C,1,read_buf);
read_buf[0]&=~LSM6DS3_ACC_GYRO_BDU_MASK;
read_buf[0]|=LSM6DS3_ACC_GYRO_BDU_BLOCK_UPDATE;
I2Cx_WriteData(0xD4,LSM6DS3_ACC_GYRO_CTRL3_C,read_buf[0]);
```

Здесь мы включаем бит BDU, о назначении которого мы уже хорошо знаем из занятий по подобным датчикам с контроллером STM . Регистр используем тот же, бит включаем шестой

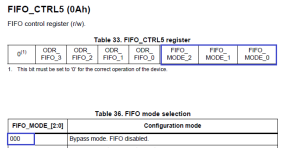
BDU	Block Data Update. Default value: 0 (0: continuous update; 1: output registers not updated until MSB and LSB have been read)

Идем дальше по регистрам

```
I2Cx_WriteData(0xD4,LSM6DS3_ACC_GYRO_CTRL3_C,read_buf[0]);
//выбор режима FIFO
I2Cx_ReadData(0xD4,LSM6DS3_ACC_GYRO_FIFO_CTRL5,1
```

```
,read_buf);
read_buf[0]&=~LSM6DS3_ACC_GYRO_FIFO_MODE_MASK;
read_buf[0]|=LSM6DS3_ACC_GYRO_FIFO_MODE_BYPASS;
I2Cx_WriteData(0xD4,LSM6DS3_ACC_GYRO_FIFO_CTRL5,read_buf[0]);
```

В данном коде мы выбираем режим ввода-вывода, работая уже с пятым регистром управления FIFO (Для FIFO существуют отдельные регистры). Режим мы установи без изменения (bypass)



FIFO_CTRL5 (0Ah)

FIFO control register (r/w).

Table 33. FIFO_CTRL5 register

0(1)	ODR_FIFO_3	ODR_FIFO_2	ODR_FIFO_1	ODR_FIFO_0	FIFO_MODE_2	FIFO_MODE_1	FIFO_MODE_0

1. This bit must be set to '0' for the correct operation of the device.

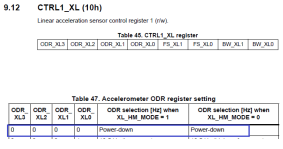
Table 36. FIFO mode selection

FIFO_MODE_[2:0]	Configuration mode
000	Bypass mode. FIFO disabled.

Продолжаем писать код настройки регистров управления

```
I2Cx_WriteData(0xD4,LSM6DS3_ACC_GYRO_FIFO_CTRL5,read_buf[0]);
//пока выключим датчик (ODR_XL = 0000)
I2Cx_ReadData(0xD4,LSM6DS3_ACC_GYRO_CTRL1_XL,1,read_buf);
read_buf[0]&=~LSM6DS3_ACC_GYRO_ODR_XL_MASK;
read_buf[0]|=LSM6DS3_ACC_GYRO_ODR_XL_POWER_DOWN;
I2Cx_WriteData(0xD4,LSM6DS3_ACC_GYRO_CTRL1_XL,read_buf[0]);
```

Ну тут всё также как и в уроках по STM. Пока отключим датчик. Используем первый управляющий регистр



9.12 CTRL1_XL (10h)

Linear acceleration sensor control register 1 (r/w).

Table 45. CTRL1_XL register

ODR_XL3	ODR_XL2	ODR_XL1	ODR_XL0	FS_XL1	FS_XL0	BW_XL1	BW_XL0

Table 47. Accelerometer ODR register setting

ODR_XL3	ODR_XL2	ODR_XL1	ODR_XL0	ODR selection [Hz] when XL_HM_MODE = 1	ODR selection [Hz] when XL_HM_MODE = 0
0	0	0	0	Power-down	Power-down

Дальше будет следующий код:

```
I2Cx_WriteData(0xD4,LSM6DS3_ACC_GYRO_CTRL1_XL,read_buf[0]);
```

```
//Full scale selection
2G
I2Cx_ReadData(0xD4,LSM6D
S3_ACC_GYRO_CTRL1_XL,1,r
ead_buf);
read_buf[0]&=~LSM6DS3_AC
C_GYRO_FS_XL_MASK;
read_buf[0]|=LSM6DS3_ACC
_GYRO_FS_XL_2g;
I2Cx_WriteData(0xD4,LSM6
DS3_ACC_GYRO_CTRL1_XL,re
ad_buf[0]);
```

Тут понятно. Включаем амплитуду измерения 2G. Регистр тот же.

Table 46. CTRL1_XL register description

ODR_XL [3:0]	Output data rate and power mode selection. Default value: 0000 (see Table 47)
FS_XL [1:0]	Accelerometer full-scale selection. Default value: 00. (00: ±2 g; 01: ±16 g; 10: ±4 g; 11: ±8 g)
BW_XL [1:0]	Anti-aliasing filter bandwidth selection. Default value: 00 (00: 400 Hz; 01: 200 Hz; 10: 100 Hz; 11: 50 Hz)

Пишем дальше:

```
I2Cx_WriteData(0xD4,LSM6
DS3_ACC_GYRO_CTRL1_XL,re
ad_buf[0]);
//Включим оси
I2Cx_ReadData(0xD4,LSM6D
S3_ACC_GYRO_CTRL9_XL,1,r
ead_buf);
read_buf[0]&=~
(LSM6DS3_ACC_GYRO_XEN_XL
_MASK|LSM6DS3_ACC_GYRO_Y
EN_XL_MASK|LSM6DS3_ACC_G
YRO_ZEN_XL_MASK);
read_buf[0]|=
(LSM6DS3_ACC_GYRO_XEN_XL
_ENABLED|LSM6DS3_ACC_GYR
O_YEN_XL_ENABLED|LSM6DS3
_ACC_GYRO_ZEN_XL_ENABLED
);
I2Cx_WriteData(0xD4,LSM6
DS3_ACC_GYRO_CTRL9_XL,re
ad_buf[0]);
```

Здесь мы уже включаем оси (с каких осей будут сниматься данные). Используем все три оси – X,Y и Z. Регистр используем девятый

9.20 CTRL9_XL (18h)

Linear acceleration sensor control register 9 (r/w).

Table 70. CTRL9_XL register

0(1)	0(1)	Zen_XL	Yen_XL	Xen_XL	SOFT_EN	0(1)	0(1)

Zen_XL	Accelerometer Z-axis output enable. Default value: 1 (0: Z-axis output disabled; 1: Z-axis output enabled)
Yen_XL	Accelerometer Y-axis output enable. Default value: 1 (0: Y-axis output disabled; 1: Y-axis output enabled)
Xen_XL	Accelerometer X-axis output enable. Default value: 1 (0: X-axis output disabled; 1: X-axis output enabled)
SOFT_EN	Enable soft-iron correction algorithm for magnetometer(1). Default value: 0 (0: soft-iron correction algorithm disabled; 1: soft-iron correction algorithm disabled)

Пишем дальше код функции

```
I2Cx_WriteData(0xD4,LSM6
DS3_ACC_GYRO_CTRL9_XL,re
ad_buf[0]);
```

```
  //Включим Data Rate
104 Гц

I2Cx_ReadData(0xD4,LSM6D
S3_ACC_GYRO_CTRL1_XL,1,r
ead_buf);

read_buf[0]&=~LSM6DS3_AC
C_GYRO_ODR_XL_MASK;

read_buf[0]|=LSM6DS3_ACC
_GYRO_ODR_XL_104Hz;

I2Cx_WriteData(0xD4,LSM6
DS3_ACC_GYRO_CTRL1_XL,re
ad_buf[0]);
}
```

По закрывающей фигурной скобки мы видим, что это окончание функции и также окончание инициализации акселерометра. Здесь мы, конечно, должны включить наш датчик. Частоту снятия данных будем использовать 104 кГц. Регистр первый. Мы его уже использовали выше

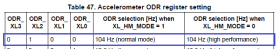
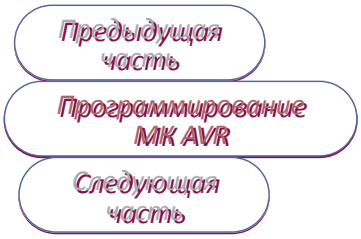


Table 47. Accelerometer ODR register setting

ODR_XL3	ODR_XL2	ODR_XL1	ODR_XL0	ODR selection [Hz] when XL_HM_MODE = 1	ODR selection [Hz] when XL_HM_MODE = 0
0	1	0	0	104 Hz (normal mode)	104 Hz (high performance)

Мы настроили все регистры, на этом, в принципе, инициализация закончена. В **следующей части** занятия мы попробуем считать показания с датчика и как-то их отмониторить, так сказать, отобразить.



Техническая документация:

Документация на датчик

Документация на оценочную плату

Приобрести плату Atmega 328p Pro Mini можно **здесь**.

Программатор (продавец надёжный) **USBASP USBISP 2.0**

Приобрести платы с датчиком LSM6DS3 можно

у следующих продавцов:

Надёжный продавец **LSM6DS33 STEVAL-MKI160V1**

Здесь дешевле **LSM6DS33 STEVAL-MKI160V1**

Здесь другая плата, намного дешевле, но от другого разработчика **LSM6DS33**

Смотреть ВИДЕОУРОК (нажмите на картинку)

‹ AVR УРОК 39. Акселерометр LSM6DS3. Часть 2

AVR УРОК 39. Акселерометр LSM6DS3. Часть 4 ›

4 комментария на “AVR УРОК 39. Акселерометр LSM6DS3. Часть 3”

Михаил:

Май 11, 2017 в 3:09 пп

не смог зайти на форрум,поэтому пишу здесь.Помогите очень нужна помощь по подключению датчика влажности DHT21 онже AM2301 к AVR atmega 8 на c+ буду признателен за помощ или видео урок ….

Ответить

admin:

Май 11, 2017 в 5:43 пп

Есть видеоурок по данному датчику на stm

Ответить

Михаил:

Май 11, 2017 в 8:36 пп

к сожалению ютo совершенно другой датчик DS3231 работает

по шине I2C. а датчик DHT21 или AM2301 по шине SPI.. **Вопрос открыт**, спасибо что откликнулись..

Ответить

Михаил:

Май 14, 2017 в 7:34 пп

здраствуйте участники форрума всёещё не могу решить задачю с подключением DHT21 онже AM2301
ВЫРУЧАЙТЕ!!!
или подскажите где посмотреть , куда обратиться.!!!!

Ответить

Добавить комментарий

Ваш e-mail не будет опубликован. Обязательные поля помечены *

Комментарий

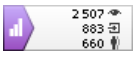
Имя *

E-mail *

Сайт

+ 5 = двенадцать

Отправить комментарий



Наверх