

Сайт narodstream.ru

создан в поддержку
канала YouTube

NAROD STREAM



narod stream

Просмотреть как: Владелец ▾

[Главная](#) [Видео](#) [Плейлисты](#) [Каналы](#) [Обсуждение](#) [0 канале](#)

Цена и покупка станка гидро-
резки ▾

headwaterjet.ru



Яндекс.Директ

Свежие записи

- [STM32. Урок 90. Датчик освещённости VL6180X. Часть 1](#)
- [STM32. Урок 87. LAN. ENC28J60. HTTP Server. Подключаем карту SD](#)
- [AVR. Урок 52. LAN. ENC28J60. HTTP Server. Подключаем карту SD. Часть 2](#)
- [AVR. Урок 52. LAN. ENC28J60. HTTP Server. Подключаем карту SD. Часть 1](#)
- [STM32. Урок 88. SD. SPI. FATFS. Часть 4](#)

Нужно программир-е
контроллеров? ▾

festo.com/обучение

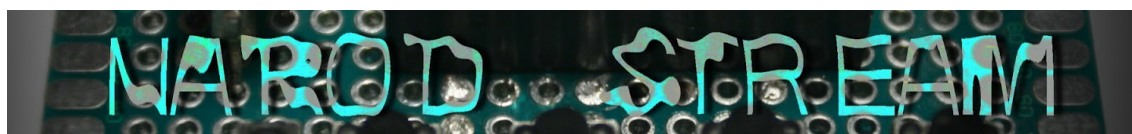




Яндекс.Директ

Рубрики

- [Программирование AVR](#)
- [Программирование STM32](#)

[Главная](#)[Новости](#)[Уроки по программированию МК](#)[Ссылки](#)[Форум](#)[Помощь](#)[Главная](#) › [STM Урок 42. Подключаем акселерометр LSM6DS0. Часть 2](#)

STM Урок 42. Подключаем

Мета

- [Регистрация](#)
- [Войти](#)
- [RSS записей](#)

акселерометр LSM6DS0. Часть 2

Урок 42

Часть 2

Подключаем акселерометр LSM6DS0

Предыдущая
часть

Программирование
МК STM32

Следующий
урок



Яндекс.Директ

Курсы программирования на Python

Курсы «Язык PYTHON для WEB-Программирования»
Оставьте заявку!

Как попасть в IT Трудоустройство Направление
Об Академии ШАГ

itstep.by Адрес и телефон

В **предыдущей части** нашего урока мы ознакомились с документацией на датчик, создали для него проект, добавили некоторые макросы и настроили библиотечные файлы, также написали функцию считывания идентификатора акселерометра и начали писать функцию инициализации датчика.

Продолжим писать код в функции AccInit

```
uint8_t value=0;
//установим бит BDU
value = Accel_IO_Read(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG8);
value &= ~LSM6DS0_ACC_GYRO_BDU_MASK;
value |= LSM6DS0_ACC_GYRO_BDU_ENABLE;
Accel_IO_Write(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG8, value); //пока
выключим датчик (ODR_XL = 000)
```

7.25 CTRL_REG8 (22h)

Control register 8.

Table 68. CTRL_REG8 register

BOOT	BDU	H_LACTIVE	PP_OD	SIM	IF_ADD_INC	BLE	SW_RESET
------	-----	-----------	-------	-----	------------	-----	----------

- [RSS](#)
- [комментариев](#)
- [WordPress.org](#)

Уроки по
программированию
МК

Программирован
ие МК AVR

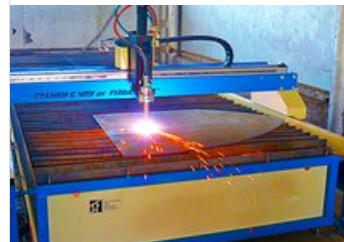
Программирование
МК STM32

искать здесь ...

Фильтровать

Станки для плаз-
менной резки с
ЧПУ! ✓

[amntech.ru/](#)
Плазменная-Резка



Яндекс.Директ

Заходите на
канал Narod
Stream



narod stream Просмотреть канал: Владимир

Главная Видео Плейлисты Каналы Обсуждение О канале

В данном регистре мы включим бит 6 – BDU, который отвечает за защиту старшего байта данных от изменения в том случае если еще не считан младший

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

Дальше напишем следующий код

```
Accel_IO_Write(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG8, value);
//пока выключим датчик (ODR_XL = 000)
value = Accel_IO_Read(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG6_XL);
value &= ~LSM6DS0_ACC_GYRO_ODR_XL_MASK;
value |=
LSM6DS0_ACC_GYRO_ODR_XL_POWER_DOWN;
Accel_IO_Write(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG6_XL, value);
```

7.23 CTRL_REG6_XL (20h)

Linear acceleration sensor control register 6.

Table 62. CTRL_REG6_XL register

ODR_XL2	ODR_XL1	ODR_XL0	FS_XL	FS0_XL	BW_SCAL_ODR	BW_XL1	BW_XL0
---------	---------	---------	-------	--------	-------------	--------	--------

В данном регистре мы сначала включим режим пониженного энергопотребления.

Table 64. ODR register setting (accelerometer only mode)

ODR_XL2	ODR_XL1	ODR_XL0	ODR selection [Hz]
0	0	0	Power-down

Продолжим работу с управляющими битами дальше

```
Accel_IO_Write(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG6_XL, value);
//Full scale selection 2G
value = Accel_IO_Read(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG6_XL);
value &= ~LSM6DS0_ACC_GYRO_FS_XL_MASK;
value |= LSM6DS0_ACC_GYRO_FS_XL_2g;
Accel_IO_Write(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG6_XL, value);
```

Работаем с тем же регистром 6, только настраиваем мы уже биты, отвечающие за максимальное ускорение. Выставим, как обычно 2g. Для этого нам не нужно включать 3 и 4 биты регистра.

FS_XL [1:0]	Accelerometer full-scale selection. Default value: 00 (00: ±2g; 01: ±16g; 10: ±4g; 11: ±8g)
-------------	--

Продолжим дальше

```
Accel_IO_Write(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG6_XL, value);
//Включим оси
value = Accel_IO_Read(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG5_XL);

value &= ~(LSM6DS0_ACC_GYRO_XEN_XL_MASK |
LSM6DS0_ACC_GYRO_YEN_XL_MASK |
```

Свежие комментарии

- Phoenix75 к записи [STM Урок 26. HAL. SPI. Драйвер индикатора MAX7219](#)
- Илья к записи [STM Урок 37. Дисплей TFT 240×320 8bit. Часть 4](#)
- admin к записи [STM Урок 18. HAL. ADC. Regular Channel. DMA](#)
- ано60 к записи [STM Урок 18. HAL. ADC. Regular Channel. DMA](#)

```
LSM6DS0_ACC_GYRO_Z
EN_XL_MASK);
value |= (LSM6DS0_ACC_GYRO_XEN_XL_ENABLE |
LSM6DS0_ACC_GYRO_Y
EN_XL_ENABLE |
LSM6DS0_ACC_GYRO_Z
EN_XL_ENABLE);
Accel_IO_Write(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG5_XL, value);
```

Работаем мы теперь с регистром 5

7.22 CTRL_REG5_XL (1Fh)

Linear acceleration sensor Control Register 5.

Table 60. CTRL_REG5_XL register

DEC_1	DEC_0	Zen_XL	Yen_XL	Xen_XL	g ⁽¹⁾	g ⁽¹⁾	g ⁽¹⁾
-------	-------	--------	--------	--------	------------------	------------------	------------------

Здесь мы включим использование всех трёх осей акселерометра

	11. update every 8 samples)
Zen_XL	Accelerometer's Z-axis output enable. Default value: 1 (0: Z-axis output disabled; 1: Z-axis output enabled)
Yen_XL	Accelerometer's Y-axis output enable. Default value: 1 (0: Y-axis output disabled; 1: Y-axis output enabled)
Xen_XL	Accelerometer's X-axis output enable. Default value: 1 (0: X-axis output disabled; 1: X-axis output enabled)

Продолжим дальше писать функцию инициализации

```
Accel_IO_Write(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG5_XL, value);
//Включим Data Rate 119 Гц
value = Accel_IO_Read(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG6_XL);
value &= ~LSM6DS0_ACC_GYRO_ODR_XL_MASK;
value |= LSM6DS0_ACC_GYRO_ODR_XL_119Hz;
Accel_IO_Write(0xD6,
LSM6DS0_ACC_GYRO_CTRL_REG6_XL, value);
}
```

Здесь мы снова возвращаемся к регистру 6 и включаем скорость (частоту измерений) 119 Гц

ODR_XL2	ODR_XL1	ODR_XL0	ODR selection [Hz]
0	1	1	119 Hz

Яндекс.Директ



Интересуетесь программированием?

Скачайте бесплатную книгу и станьте настоящим программистом. Получить сейчас: 18+

sheremetev.aoserver.ru

Функция инициализации закончена. Теперь в основной функции перед вызовом функции AccInit потушим

- admin к записи STM Урок 87. LAN. ENC28J60. TCP WEB Server. Передаём страницу побольше. Часть 1

Архивы

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

	65 154
31 ДЕНЬ	6 719
	15 134
07 ДЕНЬ	2 158
	2 508
24 ЧАСА	568
	300
СЕГОДНЯ	94
	42
НАПЛИНУ	16

светодиод, а после вызова зажжем, чтобы убедиться что у нас нигде ничего не подвисло.

```
else Error();
LD2_OFF;
AccInit(ctrl);
LD2_ON;
```

Соберем код, прошьём МК и посмотрим результат.

Теперь будем писать код в функцию считывания данных с датчика Accel_GetXYZ

```
uint8_t i=0;
buffer[0] =
Accel_IO_Read(0xD6,LSM6DS0_ACC_GYRO_OUT_X_L
_XL);
buffer[1] =
Accel_IO_Read(0xD6,LSM6DS0_ACC_GYRO_OUT_X_H
_XL);
buffer[2] =
Accel_IO_Read(0xD6,LSM6DS0_ACC_GYRO_OUT_Y_L
_XL);
buffer[3] =
Accel_IO_Read(0xD6,LSM6DS0_ACC_GYRO_OUT_Y_H
_XL);
buffer[4] =
Accel_IO_Read(0xD6,LSM6DS0_ACC_GYRO_OUT_Z_L
_XL);
buffer[5] =
Accel_IO_Read(0xD6,LSM6DS0_ACC_GYRO_OUT_Z_H
_XL);
```

Здесь мы считываем данные из регистров 0X28 — 0X2D, отвечающие за хранение считанных ускорений с осей — соответственно младших и старших байтов.

Затем в цикле занесем считанные байты уже двухбайтовыми величинами в соответствующий буфер

```
for(i=0;i<3;i++)
{
    pData[i] = ((int16_t)((uint16_t)buffer[2*i+1]
<<8)+buffer[2*i]);
}
```

Перейдем в функцию Accel_ReadAcc, которую мы затем вызовем в бесконечном цикле в main(). Закомментируем пока там строки, связанные с USART и зажжем светодиод в случае превышения значения по оси X

```
// sprintf(str1,"X:%06d Y:%06d Z:%06drn", xval, yval,
zval);
// HAL_UART_Transmit(&huart2,
(uint8_t*)str1,strlen(str1),0x1000);
// buf2[0]=0x12;
// buf2[1]=0x10;
// buf2[2]=(uint8_t)(xval>>8);
// buf2[3]=(uint8_t)xval;
// buf2[4]=0x10;
// buf2[5]=0x10;
// buf2[6]=(uint8_t)(zval>>8);
// buf2[7]=(uint8_t)zval;
// buf2[8]=0x13;
// HAL_UART_Transmit(&huart2,buf2,9,0x1000);
if(xval>1500)
{
    LD2_ON;
}
else
{
    LD2_OFF;
}
HAL_Delay(20);
```

Вызовем данную функцию в бесконечном цикле

```
/* USER CODE BEGIN 3 */
    Accel_ReadAcc();
}
```

Соберем код, прошьем контроллер и проверим работу программы. При определенном отклонении платы светодиод должен загораться и потухать.

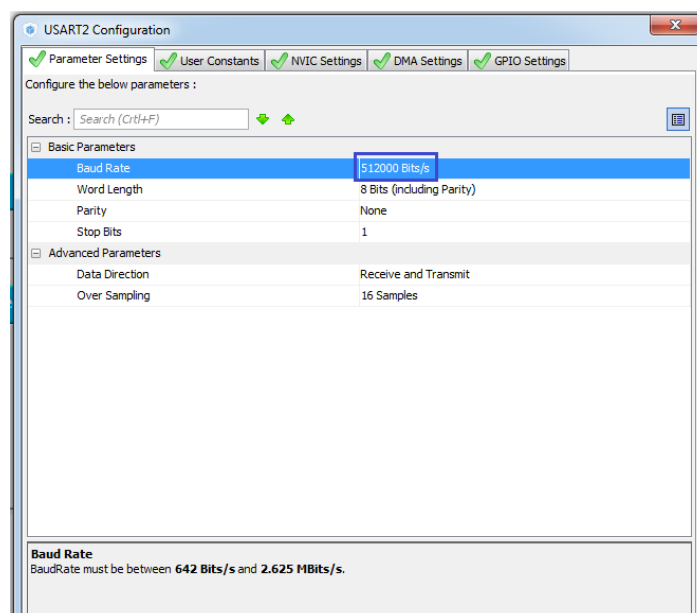
Если всё работает, то попытаемся считать уже точные данные через терминальную программу, раскомментировав код в функции Accel_ReadAcc, а также немного исправив, так как у нас подключен DMA.

```
sprintf(str1, "X:%06d Y:%06d Z:%06drn", xval, yval, zval);
HAL_UART_Transmit_DMA(&huart2,
(uint8_t*)str1, strlen(str1));
```

Соберем код, прошьем контроллер, запустим монитор порта из программы arduino-nightly и посмотрим результат

```
X:015672 Y:-03303 Z:-00180
X:016802 Y:-04873 Z:000504
X:016334 Y:-03887 Z:-00203
X:013644 Y:-02886 Z:-01492
X:014053 Y:-02992 Z:-01578
X:013476 Y:-02514 Z:-01092
X:014058 Y:-01396 Z:-01635
X:014175 Y:-01945 Z:-01378
X:015993 Y:-02922 Z:000132
X:015831 Y:-04247 Z:000295
X:016119 Y:-04552 Z:-01574
X:014568 Y:-05491 Z:-01139
X:014158 Y:-05606 Z:-01529
X:014234 Y:-05907 Z:-00781
X:016679 Y:-05665 Z:-00824
X:016520 Y:-05413 Z:-00080
X:015302 Y:-04926 Z:-01147
X:015678 Y:-05126 Z:-00773
X:015447 Y:-04991 Z:-00134
```

Зайдем в Cube MX и в настройках USART вместо 256000 включим 512000.

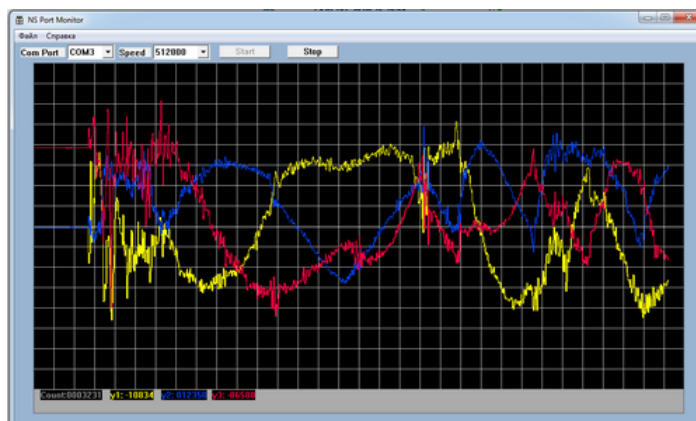


Перегенерируем проект, соберем его и внесем еще некоторые изменения в код функции Accel_ReadAcc, опять

закомментировав код вывода на терминал. Раскомментируем и внесем изменения в код вывода в программу визуализации. Программу будем использовать NS Port Monitor, поэтому код составим в соответствии с требованиями протокола данной программы (первые байты 0x11 и 0x55).

```
// sprintf(str1,"X:%06d Y:%06d Z:%06drn", xval, yval,
zval);
// HAL_UART_Transmit_DMA(&huart2,
(uint8_t*)str1,strlen(str1));
buf2[0]=0x11;
buf2[1]=0x55;
buf2[2] = (uint8_t) (xval >> 8);
buf2[3] = (uint8_t) xval;
buf2[4] = (uint8_t) (yval >> 8);
buf2[5] = (uint8_t) yval;
buf2[6] = (uint8_t) (zval >> 8);
buf2[7] = (uint8_t) zval;
HAL_UART_Transmit_DMA(&huart2,buf2,8);
if(xval>1500)
```

Соберем проект, прошьем контроллер, запустим программу и проверим результат на практике (нажмите на картинку для увеличения изображения)



Предыдущая
часть

Программирование
МК STM32

Следующий
урок

Исходный код

Техническая документация на датчик

**Техническая документация на плату
расширения**

Программа NS Port Monitor

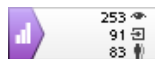
Смотреть ВИДЕОУРОК



[Главная](#) | [Новости](#) | [Уроки по программированию МК](#)
| [Программирование микроконтроллеров AVR](#)
| [Программирование микроконтроллеров STM32](#)
| [Программирование микроконтроллеров PIC](#)
| [Ссылки](#) | [Форум](#) | [Помощь](#)



Google



© 2017 Default copyright text

