



Главная | Новости | Уроки по программированию МК | Ссылки | Форум | Помощь

Главная > Программирование STM32 > STM Урок  
45. Подключаем гироскоп LSM6DS0. Часть 2

## Свежие комментарии

- Валерий к записи [AVR Урок 34. Дисплей TFT 240×320 8bit. Часть 5](#)
- [Narod Stream](#) к записи [STM Урок 88. SD. SPI. FATFS. Часть 3](#)
- [Anh](#) к записи [STM Урок 88. SD. SPI. FATFS. Часть 3](#)
- [Narod Stream](#) к записи [STM Урок 29. HAL. DAC. Triangle. DMA](#)
- [Narod Stream](#) к записи [STM Урок 29. HAL. DAC. Triangle. DMA](#)

## Форум. Последние ответы

- [imperor](#) в [Программирование МК STM32](#)  
1 час, 59 мин. назад
- [Mihail](#) в [Программирование МК STM32](#)  
1 день, 3 час. назад
- [imperor](#) в [Программирование МК STM32](#)  
6 дн., 6 час. назад
- [Rocket](#) в [Программирование МК STM32](#)  
1 неделя, 2 дн. назад
- [Narod Stream](#) в [Программирование МК STM32](#)  
1 неделя, 2 дн. назад

## Архивы

- [Ноябрь 2017](#)
- [Октябрь 2017](#)
- [Сентябрь 2017](#)
- [Август 2017](#)
- [Июль 2017](#)
- [Июнь 2017](#)
- [Май 2017](#)
- [Апрель 2017](#)
- [Март 2017](#)
- [Февраль 2017](#)
- [Январь 2017](#)
- [Декабрь 2016](#)
- [Ноябрь 2016](#)

# STM Урок 45. Подключаем гироскоп LSM6DS0. Часть 2

Posted on Ноябрь 23, 2016 by [Narod](#)

[Stream](#) Опубликовано в [Программирование STM32](#) — Нет комментариев ↓

Яндекс.Директ



**Программирование**  
Заходи к нам! Уникальный про  
языке C!  
[mcsi-c.ru](#) Адрес и телефон

Яндекс.Директ



## Нужно программ-е контроллеров?

Комплексное обучение. Доступные цены. Минимальные сроки. Большой опыт!  
О компании Услуги Продукция Преимущества  
[festo.com](#) Адрес и телефон

## Урок 45

### Часть 2

# Подключаем гироскоп LSM6DS0

В [предыдущей части](#) нашего урока мы написали все макросы, которые внесли удобства в чтение и написание кода, написали функцию инициализации и проверили её на практике.

Добавим ещё одну функцию Gyro\_GetXYZ, предназначенную для опроса осей гироскопа. Сделаем её полностью на основе подобной для акселерометра (Accel\_GetXYZ), скопировав в функцию также и весь код. Исправим код, используя в опросе регистры, предназначенные именно для чтения осей гироскопа

```
//-----
void Gyro_GetXYZ(int16_t* pData)
{
    uint8_t buffer[6];
    uint8_t i=0;
    buffer[0]=Accel_IO_Read(0xD6,L
SM6DS0_ACC_GYRO_OUT_X_L_G);
```

Яндекс.Директ



## Каретки 1853 в наличии.

Отгрузка со склада в СПб. Бесплатная нарезка в размер. Звоните!  
[ladogaprof.ru](#)  
Адрес и телефон



## Всё для предметной фотосъемки!

Профессиональное оборудование и поворотные столы для предметной фотосъемки!  
[360gear.ru](#)  
Адрес и телефон

**Заходите на канал  
Narod Stream**

```
buffer[1]=Accel_IO_Read(0xD6,L
SM6DS0_ACC_GYRO_OUT_X_H_G);
buffer[2]=Accel_IO_Read(0xD6,L
SM6DS0_ACC_GYRO_OUT_Y_L_G);
buffer[3]=Accel_IO_Read(0xD6,L
SM6DS0_ACC_GYRO_OUT_Y_H_G);
buffer[4]=Accel_IO_Read(0xD6,L
SM6DS0_ACC_GYRO_OUT_Z_L_G);
buffer[5]=Accel_IO_Read(0xD6,L
SM6DS0_ACC_GYRO_OUT_Z_H_G);
for(i=0;i<3;i++)
{
    pData[i] = ((int16_t)
((uint16_t)buffer[2*i+1]<<8)+buffer[2*i]);
}
}
```

Функцию Accel\_ReadAcc также для порядка переименуем в более универсальную – AccelGyro\_Read. То же самое сделаем и с прототипом данной функции

```
//-----
void AccelGyro_Read(void)
{
    int16_t buffer[3] = {0};
```

Исправим в ней строку

```
int16_t xval, yval, zval;
Gyro_GetXYZ(buffer);
```

Раскомментируем строки кода, отвечающие за вывод считанной с осей информации в текстовый вид, а в графический закомментируем

```
sprintf(str1,"X:%06d Y:%06d
Z:%06d", xval, yval, zval);
HAL_UART_Transmit_DMA(&hu
art2, (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(&h
uart2,buf2,8);
```

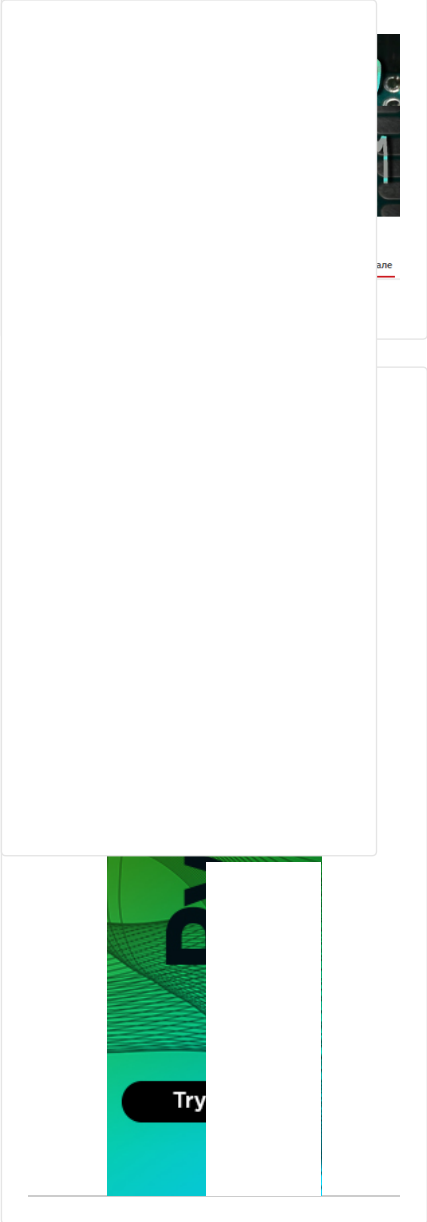
Здесь тоже немного исправим код

```
if(zval>500)
{
```

В бесконечном цикле в функции main() раскомментируем и исправим вызов функции

```
/* USER CODE BEGIN 3 */
AccelGyro_Read();
```

Соберем код, прошьем контроллер и посмотрим: при вращении платы против часовой стрелки относительно вертикальной оси должен загораться



Рубрики

- [Программирование AVR](#) (131)
- [Программирование PIC](#) (2)
- [Программирование STM32](#) (196)
- [Тесты устройств и аксессуаров](#) (1)

31.05.2017	
31.05.2017	108 025
30.05.2017	12 124
29.05.2017	26 264
28.05.2017	3 823
27.05.2017	4 350
26.05.2017	968
25.05.2017	3 714
24.05.2017	813
23.05.2017	158
22.05.2017	28

50D вспышка с...	ЖК-диспле
носки платье купи...	экран tft купи
2 284,73 руб.	630,21 ₴

зелёный светодиод. Затем проверим данные в мониторе порта Arduino. Должен быть вот такой результат.

```
X:001578 Y:008086 Z:-03865
X:000845 Y:008222 Z:-02895
X:-00323 Y:006354 Z:-01348
X:-00657 Y:-01131 Z:000323
X:-02944 Y:-10370 Z:003685
X:-07702 Y:-19699 Z:007720
X:-06648 Y:-24588 Z:009093
X:-03149 Y:-20381 Z:007676
X:-02230 Y:-14940 Z:006014
X:-02794 Y:-09631 Z:002280
X:-03097 Y:-07377 Z:-00142
X:-02752 Y:-06021 Z:000001
X:000189 Y:-01310 Z:-01925
X:009143 Y:012933 Z:-08242
X:010688 Y:020423 Z:-09788
X:007864 Y:021607 Z:-09634
X:003084 Y:009322 Z:-04361
X:002698 Y:000793 Z:-01602
X:-00537 Y:000170 Z:-00741
X:001125 Y:-02924 Z:001704
```

Попробуем немного скорректировать показания, т.к. если не крутить плату, то показания несколько будут отличаться от 0.

```
xval=buffer[0]-103;
yval=buffer[1]-47;
zval=buffer[2]-41;
```

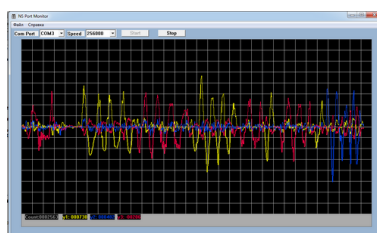
У меня получились такие цифры, у Вас, возможно будут другие. Пока другого способа калибровки датчика я не нашел.

Соберем проект и еще раз посмотрим показания.

Теперь наоборот, раскомментируем код для визуализации, а для текстового вывода закомментируем.

```
zval=buffer[2];
// sprintf(str1,"X:%06d Y:%06d
Z:%06dm", xval, yval, zval);
// HAL_UART_Transmit_DMA(&h
uart2, (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(&hu
art2,buf2,8);
```

Соберем код и прошьем контроллер. Запустим сначала программу NS Port Monitor. Покрутим плату. Результат должен быть таким (нажмите на картинку для увеличения изображения):



Теперь запустим ещё одну программу, NS Port Visual. Также покрутим плату.

Яндекс.Директ



**Магазин Чип и Дип в Минске**

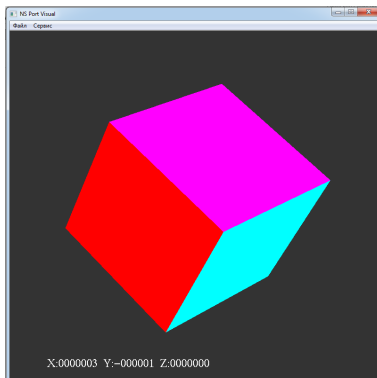
Все конденсаторы в нашем новом магазине в Минске.  
адрес магазина  
каталог  
компонентов  
каталог приборов  
наборы и модули  
[ru-chipdip.by](http://ru-chipdip.by)



**Завтра будут заморозки?**

Смотрите прогноз погоды на ноябрь.  
[yandex.by](http://yandex.by)

Результат должен быть таким:



Предыдущая  
часть

Программирование  
МК STM32

Следующий  
урок

[Исходный код](#)

[Техническая документация на датчик](#)

[Техническая документация на плату расширения](#)

[Программа Hyper Terminal](#)

[Программа NS Port Monitor](#)

[Программа NS Port Visual](#)

Отладочную плату можно приобрести  
здесь [Nucleo STM32F401RE](#)

Оценочную плату можно приобрести  
здесь [STM32 X-NUCLEO-IKS01A1](#)

[Смотреть ВИДЕОУРОК](#)



👁 Post Views: 27

◀ AVR УРОК 39. Акселерометр LSM6DS3.

Часть 4

STM Урок 22. HAL. I2C. I2C to LCD2004 ▶

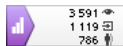
## Добавить комментарий

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

Комментарий

**Имя \*****E-mail \*****Сайт**три ×  = 12 

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



© 2017

[Наверх](#)