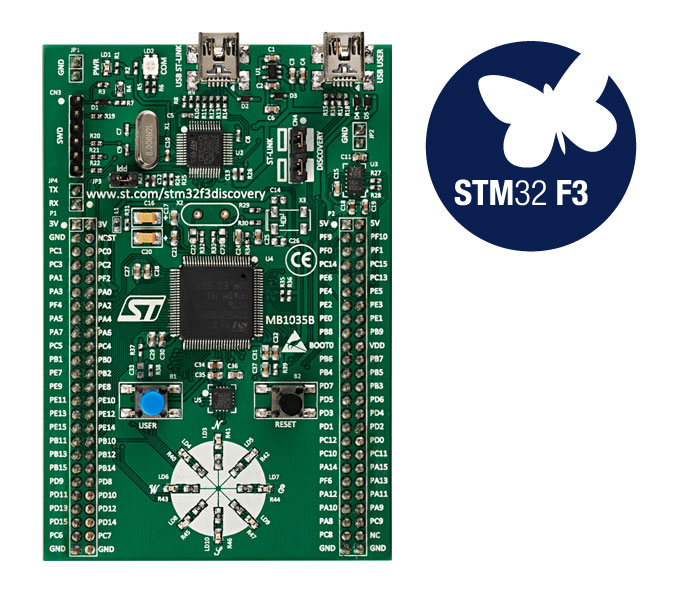
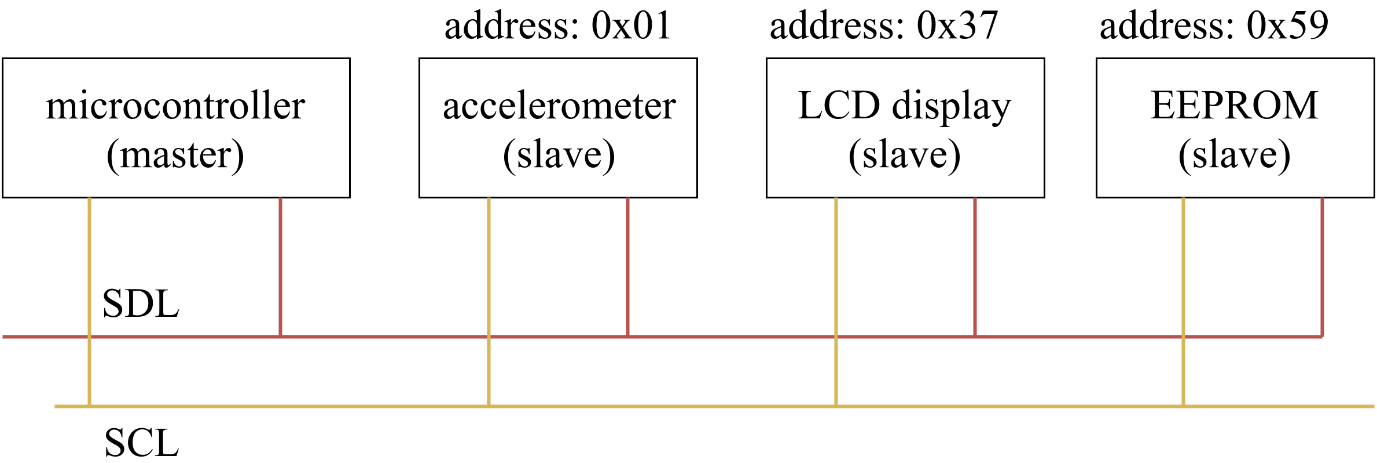
Основы работы с I2C в STM32F3



# Основные сведенье об I2C

Кроме работы со встроенной периферией, зачастую необходим работать с внешними модулями и датчиками (гироскопы, радиомодули и т.п.). Наиболее распространенными интерфейсами для подобной периферии являются UART, SPI, I2C.

I2C (Inter-Integrated Circuit, произносится I-squared-C) представляет собой последовательную шину из 2-х линий не включая землю. Одна линия – это линия тактового сигнала (SCL), другая – линия передачи (SDL). К шине обычно подключается одно master устройство и несколько slave:

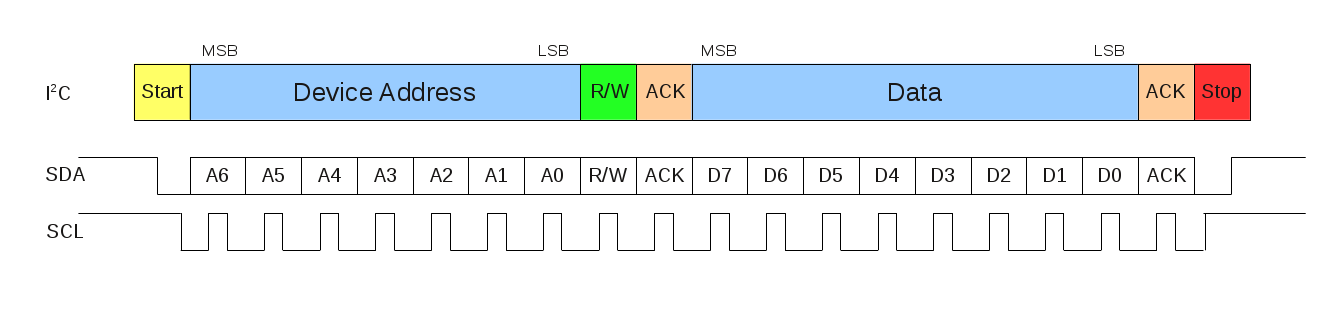


Главным здесь является master устройство. Именно оно инициирует общение со slave устройствами и управляет линией с тактовым сигналом. slave устройства имеют 7-ми или 10-ти битные адреса (зависит от режима I2C). Частота шины может быть: 100 КГц, 400 КГц, 1 мГц , 3.4 МГц или 5 МГц. Выбранную частоту на I2C шине должны поддерживать все устройства.

Протокол I2C работает следующим образом:

1) master передает на шину адрес slave устройства с меткой что они хочет получить или передать один или несколько байт;

2) slave устройством с указанными адресом передает или получает данные.



Таким образом с программной точки зрения у master имеется 2 возможности:

1) передать n байт на slave устройство;

2) получить n байт со slave устройство.

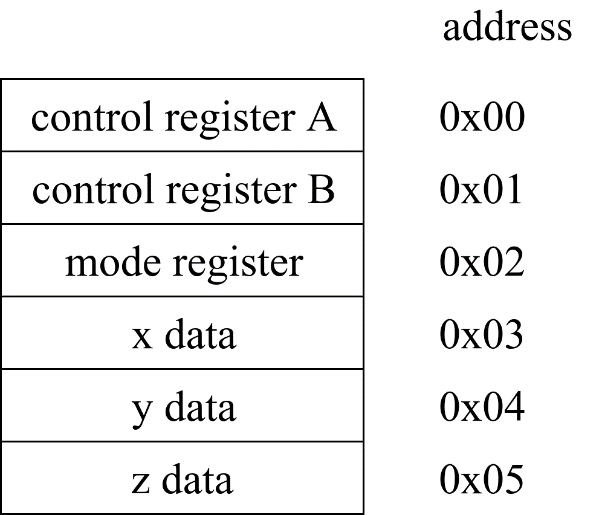
Из особенностей прокола выделим:

1) адреса slave устройств никак не регулируются I2C протоколам, и зачастую просто зашиты в конкретные устройства без возможности их изменения;

2) протокол взаимодействия со slave устройствами (когда читать/писать данные, сколько писать/читать байтов) должен быть согласован между master и slave заранее.

## Взаимодействие с периферией по I2C

Как правило, сенсоры для взаимодействия с микроконтроллером имеют набор регистров, запись в которые задает режим работы сенсора, а чтение – получение результатов измерений. Каждый регистр имеет адрес.



Для того чтобы прочитать регистр с помощью I2C, обычно используется следующий сценарий:

1. master посылает на I2C шину адрес slave с требованием записи;
2. master посылает на I2C шину адрес регистра, из которого он хочет получить данные. На этом шаг передачи данных закончен;
3. master посылает на I2C шину адрес slave с требованием чтения;
4. master читает с I2C шины от slave один или несколько байт. На этом шаг чтения данных закончен.

Для записи в регистр обычно используется следующий сценарий:

1. master посылает на I2C шину адрес slave с требованием записи;
2. master посылает на I2C шину адрес регистра, в который нужно записать данные;
3. master передает данные, которые нужно записать в регистр.

Сценарий записи данных в регистр требует меньше шагов, так как нет необходимости переключения из режима записи в режим чтения.

# Инициализация I2C

Для примера рассмотрим инициализацию I2C, подключенного к LSM303DLHC в STM32F3Discovery. Блок регистров имеет название I2C1, и подключен к пинам PB6 и PB7.

1. сконфигурируйте пины для I2C:
   1. включите тактирующий сигнал для пинов портов (регистр RCC -> AHBENR)
   2. **включите** подтягивающий резистор (**PULL-UP** режим) (регистр GPIOx->PUPDR)
   3. установить **open drain** режим (регистр GPIOx-> OTYPER)
   4. выбрать альтернативную функцию на работу с I2C (регистр GPIOx->AFR[0])
   5. переключить пин на альтернативную функцию (регистр GPIOx->MODER)
2. сконфигурируйте I2C:
   1. включите тактирующий сигнал I2C (регистр RCC-> APB1ENR)
   2. сконфигурируйте тайминги (регистр I2Cx -> TIMINGR). Если у вас стандартная частота процессора 48 МГц и частота шины 100 КГц, то просто запишите в регистр значение: 0x2000090E
   3. включите I2C (регистр I2Cx->CR1, бит «Peripheral enable»)

# Передача байтов на устройство

1. дождитесь конца предыдущей передачи (регистр I2Cx->ISR, бит «Bus busy»)
2. заполните регистр I2Cx->CR2 для начала передачи. Значение в данный регистр необходимо записать за одни раз. В регистре необходимо задать: адрес устройства на шине, количество байт для передачи, бит «Start generation» для формирования старт-сигнала, бит «Automatic end mode» для автоматической генерации стоп сигнала, бит «Transfer direction» с указанием, что мы передаем (пишем) данные.
3. передайте сами байты, повторив следующую процедуру для каждого байта:
   1. проверьте, что регистр TXDR готов к приему новых данных (регистр I2Cx->ISR, бит «Transmit data register empty»)
   2. запишите данные в регистр I2Cx->TXDR
4. дождитесь окончания передачи (регистр I2Cx->ISR, бит «Stop detection flag»)
5. очистите флаг stop бита (регистр I2Cx->IСR, бит « Stop detection flag clear»)

# Получение байтов с устройство

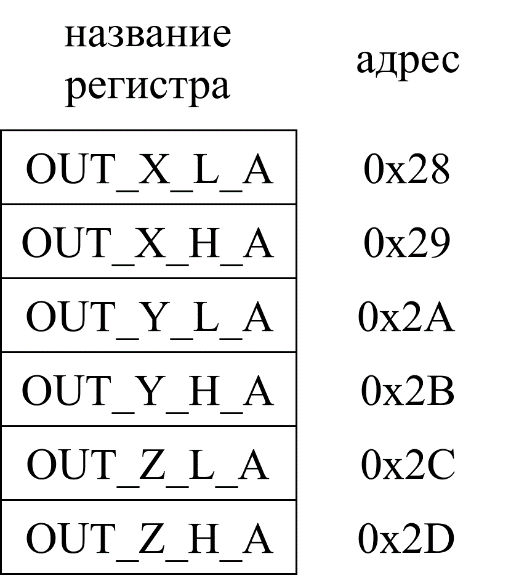
1. дождитесь конца предыдущей передачи (регистр I2Cx->ISR, бит «Bus busy»)
2. заполните регистр I2Cx->CR2 для начала передачи. Значение в данный регистр необходимо записать за одни раз. В регистре необходимо задать: адрес устройства на шине, количество байт для передачи, бит «Start generation» для формирования старт-сигнала, бит «Automatic end mode» для автоматической генерации стоп сигнала, бит «Transfer direction» с указанием, что мы получаем (читаем) данные.
3. получите байты, повторив следующую процедуру для каждого байта:
   1. проверьте, что данные пришли. (регистр I2Cx->ISR, бит « Receive data register not empty »)
   2. прочтите данные из регистра I2Cx->RXDR
4. дождитесь окончания передачи (регистр I2Cx->ISR, бит «Stop detection flag»)
5. очистите флаг stop бита (регистр I2Cx->IСR, бит « Stop detection flag clear»)

# Работа с акселерометром LSM303DLHC

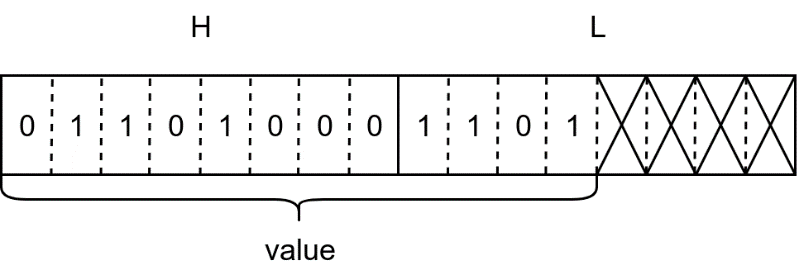
Акселерометр имеет фиксированный адрес на I2C шине: 0x32.

Как и микроконтроллер, он требует инициализации. Для этого необходимо записать в его регистр по адресу 0x20 значение 0x57. Это установить частоту обновления данных в 100 Hz, и включит все 3 оси (X, Y и Z).

Датчик является 12-ти битным, и данные об ускорения хранятся в 6-ти регистрах:



При этом 8 бит хранится в старшем (H) регистре, и 4 бита в младшем (L):



Чтобы считать эти данные нужно:

1. передать акселерометру адрес первого регистра с установленным старшим битом (0xA8). Старший бит нужен, чтобы считать несколько байт сразу;
2. считать 6 байт разом.
3. соединить старшие байты с младшими для получения результатов

Код для этого выглядит примерно следующим образом:

uint8\_t data[6];

// input params:

// - register address

// - output array

// - number of bytes to read

LSM303DLHC\_ACC\_ReadBytes(0x28 | 0x80, data, 6);

int16\_t x = data[0] + (data[1] << 8);

int16\_t y = data[2] + (data[3] << 8);

int16\_t z = data[4] + (data[5] << 8);

x >>= 4;

y >>= 4;

z >>= 4;