

Изменено	09/03/2023 10:03:31	Создано	09/03/2023 10:02:22	http://avrproject.ru/publ/kak_podkljuchit/rabota_s_magnitometrom 1-0-150
----------	------------------------	---------	------------------------	--

Работа с магнитометром HMC5883L - Как подключить - AVR project.ru - Проекты на микроконтроллерах AVR

Работа с магнитометром HMC5883L - Как подключить - AVR project.ru

Сегодня будет описание работы одного очень интересного датчика - магнитометра HMC5883L. Как видно из названия этот датчик измеряет магнитное поле, причем делает это в трех осях. Это дает возможность получать трехмерную картину направленности магнитного поля и его величину. А если проще, то с помощью этого датчика можно собрать свой цифровой компас. Интересно как с ним работать? Тогда читаем дальше!
HMC5883L достаточно распространенный и недорогой датчик, для общения с микроконтроллером используется стандартный

I2C интерфейс.

Чип выпускается в крошечном 16-и выводном корпусе LPCC размерами 3x3 мм.

Обозначение выводов:

SCL - вход тактирования шины I2C

VDD - вход для подключения питания (кормится эта козявка напряжением в диапазоне 2,16-3,6 вольт)

не используется

S1 - дополнительное питание для портов ввода/вывода. Подключается напрямую к выводу VDDIO

не используется

не используется

не используется

SETP - первый вход для подключения керамического конденсатора на 0,22uF

GND - земля

C1 - еще один вход для подключения конденсатора. Электролитического или танталового на 4,7uF (другой конец конденсатора подключается к земле)

GND - земля

SETC - второй вход для подключения керамического конденсатора на 0,22uF

VDDIO - вход для подключения напряжения которое будет на портах ввода/вывода

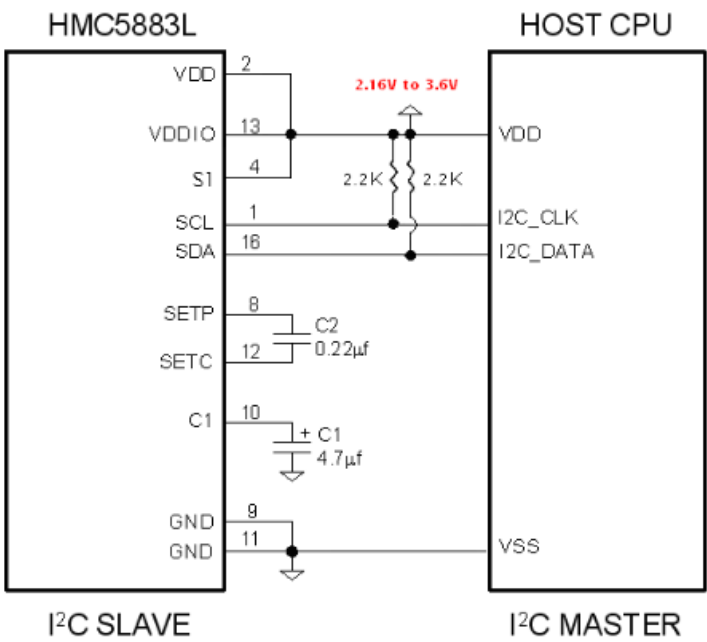
не используется

Выход прерывания, когда данные готовы на этом выводе появляется логическая 1

SDA - линия данных интерфейса I2C

В общем обвязка у сенсора минимальная, всего два конденсатора. Не считая двух pullup резисторов, необходимых для шины I2C. В простейшем случае, когда порты ввода/вывода

запитаны напряжением питания схемы и не используется вывод прерывания, схема подключения будет выглядеть так:



Работа по шине I2C

Как и любое устройство, работающее по интерфейсу I2C, датчик имеет свой собственный уникальный адрес для работы. HMC5883L присвоен адрес 0x1E. С восьмым битом записи/чтения адрес будет 0x3D для чтения и 0x3C для записи.

Карта регистров датчика представлена в таблице ниже:

Как видно, для настройки датчика доступны регистры (это те куда мы можем что-то записать) находящиеся на адресах 00 (configuration register A), 01 (configuration register B) и 02 (mode register).

Configuration Register A (CRA)

CRA7	CRA6	CRA5	CRA4	CRA3	CRA2	CRA1	CRA0
(0)	MA1(0)	MA0(0)	DO2 (1)	DO1 (0)	DO0 (0)	MS1 (0)	MS0 (0)

Первый из доступных, для настройки датчика, регистров. Каждый бит имеет свое имя, для того чтобы было видно с каким еще битом в регистре он связан (например, биты CRA6 и CRA5 имеют общее имя MA1 и MA0 соответственно, так как отвечают за один и тот же параметр)

CRA7 - зарезервированный бит и пока не используется

CRA6|CRA5 - устанавливают число замеров (семплов) перед выдачей результата измерения.

эти биты могут принимать значения: 00 = 1(Default); 01 = 2; 10 = 4; 11 = 8

Пример: чтобы установить 4 семпла записываем в CRA6 единицу, а в CRA5 ноль. CRA4|CRA3|CRA2 - устанавливают с какой скоростью будут записываться данные в регистры считывания данных (Data Output Registers).

000 - 0,75 Гц

001 - 1,5 Гц

010 - 3 Гц
 011 - 7,5 Гц
 100 - 15 Гц (дефолтное значение)
 101 - 30 Гц
 110 - 75 Гц
 111 - зарезервировано

Пример: для того чтобы увеличить скорость записи данных до 75 Гц записываем в CRA4 единицу, в CRA3 единицу, в CRA2 ноль.

CRA1|CRA0 - настраивают тип измерения. Тут какие-то технические заморочки с импедансом в магниторезистивной нагрузке датчика в каждой из осей. Не стал разбираться в деталях, оставил значение регистров по умолчанию (00 - нормальный режим измерения) .

Пример: настроим полностью регистр A, установим количество семплов равным 8 , скорость записи данных 3 Гц, нормальный тип измерения. Для этого мы должны отправить в регистр 8 бит данных, в двоичном представлении это будет: &b01101000.

Configuration Register B (CRB)

CRB7	CRB6	CRB5	CRB4	CRB3	CRB2	CRB1	CRB0
GN2 (0)	GN1 (0)	GN0 (1)	(0)	(0)	(0)	(0)	(0)

В регистре B настраивается чувствительность датчика к магнитному полю. За эти установки отвечают биты CRB7|CRB6|CRB5. Остальные 5 бит не используются, но для корректной работы они должны быть забиты нулями.

Зависимость чувствительности (Lsb/Gauss) датчика от установленных битов CRB7|CRB6|CRB5

000 -1370
 001 -1090 (дефолтное значение)
 010 - 820
 011 - 660
 100 - 440
 101 - 390
 110 - 330
 111 - 230

Пример: установим максимальную чувствительность датчика, для этого отправим в регистр B бинарное значение &b00000000

Mode Register (MR)

В последнем оставшемся в нашем распоряжении регистре, настраиваются такие параметры как скорость работы на шине I2C и режим работы датчика.

Высокоскоростной режим работы I2C (3400 кГц) устанавливается записью в бит MR7 единицы.

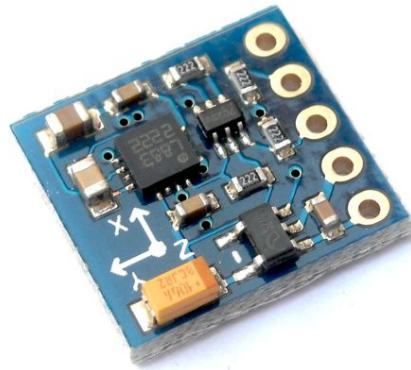
MD1|MD0 устанавливают режим работы датчика:

00 - режим непрерывного измерения.
 01 - режим единичного измерения (в этом режиме датчик работает умолчанию). После измерения и записи результата в регистр вывода данных датчик уходит в режим ожидания (Idle)
 10 - погружает датчик в режим ожидания. Потребление в этом режиме падает до 2 мкА
 11 - также погружает датчик в режим ожидания.

Пример: установим высокоскоростное соединение датчика с микроконтроллером на скорости 3400кГц и запустим режим непрерывного измерения. Для этого запишем в регистр бинарное значение &b10000000.

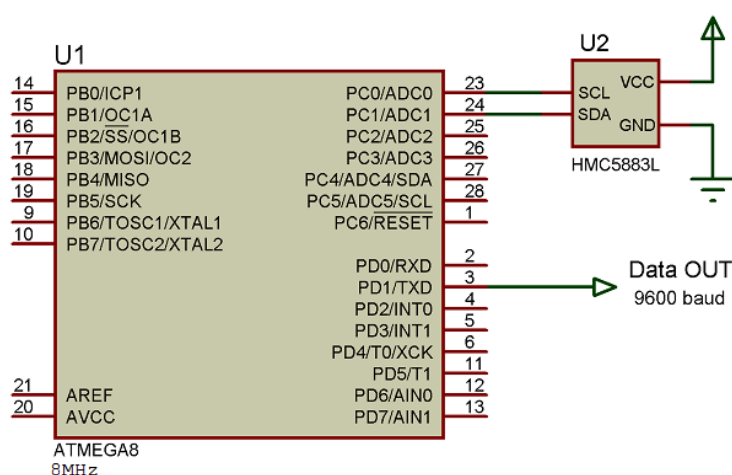
Измеренные по трем осям данные забираются из соответствующих регистров, для каждой из осей данные состоят из двух байт - старшего и младшего. Для получения результата их необходимо сложить, но это уже очевидные вещи поэтому приступим непосредственно к коду и железу.

Для того чтобы самостоятельно не паять эту букашку купил на ебай вот такую готовую платку:



Обошлась она мне в 5\$ и помимо минимально необходимой обвязки содержит стабилизатор питания и схему преобразователя уровней. Поэтому эту платку можно подключать к 5-и вольтовой схеме без опаски спалить сенсор.

Подключается плата с датчиком напрямую к микроконтроллеру, подтягивающие резисторы ставить не нужно, они уже имеются на плате:



Исходник в Bascom-AVR для работы с датчиком HMC5883L. В программе данные полученные с датчика (значение магнитного поля на каждой из осей и рассчитанный угол относительно севера) отправляются в терминал.

```
$regfile = "m8def.dat"
```

```
$crystal = 8000000
```

```
$hwstack = 64
```

\$swstack = 64

\$framesize = 150

\$baud = 9600

'переменные и константы

Dim Hmc_x As Integer

Dim Hmc_xl As Byte At Hmc_x + 0 Overlay

Dim Hmc_xh As Byte At Hmc_x + 1 Overlay

Dim Hmc_y As Integer

Dim Hmc_yl As Byte At Hmc_y + 0 Overlay

Dim Hmc_yh As Byte At Hmc_y + 1 Overlay

Dim Hmc_z As Integer

Dim Hmc_zl As Byte At Hmc_z + 0 Overlay

Dim Hmc_zh As Byte At Hmc_z + 1 Overlay

Dim X As Single

Dim Y As Single

Dim Z As Single

Dim Angle As Single

Dim Angle_str As String * 6

Const Hmc_w = &H3C 'адрес датчика + бит записи

Const Hmc_r = &H3D 'адрес датчика + бит чтения

'конфигурация I2C

Config Scl = Portc.0 'линия тактовых импульсов

Config Sda = Portc.1 'линия данных

I2cinit

Gosub Hmc_setup

'ГЛАВНЫЙ ЦИКЛ

Do

Gosub Hmc_read

X = Hmc_x

Y = Hmc_y

Z = Hmc_z

Angle = Atn2(y, X)

Angle = Angle * 57.2957795 ' (180 / 3.141592)

Angle = Angle + 180

Angle_str = Fusing(angle, "#.#")

Angle_str = Angle_str + "°"

'выводим данные в терминал

Print "x: " ; X ; " " ;

Print "y: " ; Y ; " " ;

Print "z: " ; Z ; " " ;

Print "Angle: " ; Angle_str

Print

Waitms 500

Loop

End

'инициализация магнитометра и его настройка

Hmc_setup:

```

I2cstart
I2cwbyte Hmc_w
I2cwbyte &H00          'записываем с нулевого адреса
I2cwbyte &B01011000    'регистр A. 4 сэмпла, 75Гц, нормальный режим
I2cwbyte &H00100000    'регистр B. усиление 1090 (по умолчанию)
I2cwbyte &H00000000    'устанавливаем режим постоянного измерения
I2cstop
Return
'чтение данных
Hmc_read:
I2cstart
I2cwbyte Hmc_w
I2cwbyte &H03          'начинаем читать данные с регистра хранения данных
I2cstart
I2cwbyte Hmc_r
I2crbyte Hmc_xh , Ack  'читаем старший байт x
I2crbyte Hmc_xl , Ack  'читаем младший байт x
I2crbyte Hmc_zh , Ack  'читаем старший байт z
I2crbyte Hmc_zl , Ack  'читаем младший байт z
I2crbyte Hmc_yh , Ack  'читаем старший байт y
I2crbyte Hmc_yl , Nack  'читаем младший байт y
I2cstop
Return

```

А так выглядят данные отправленные в терминал

