**Magnetômetro:**

**Como acessar**

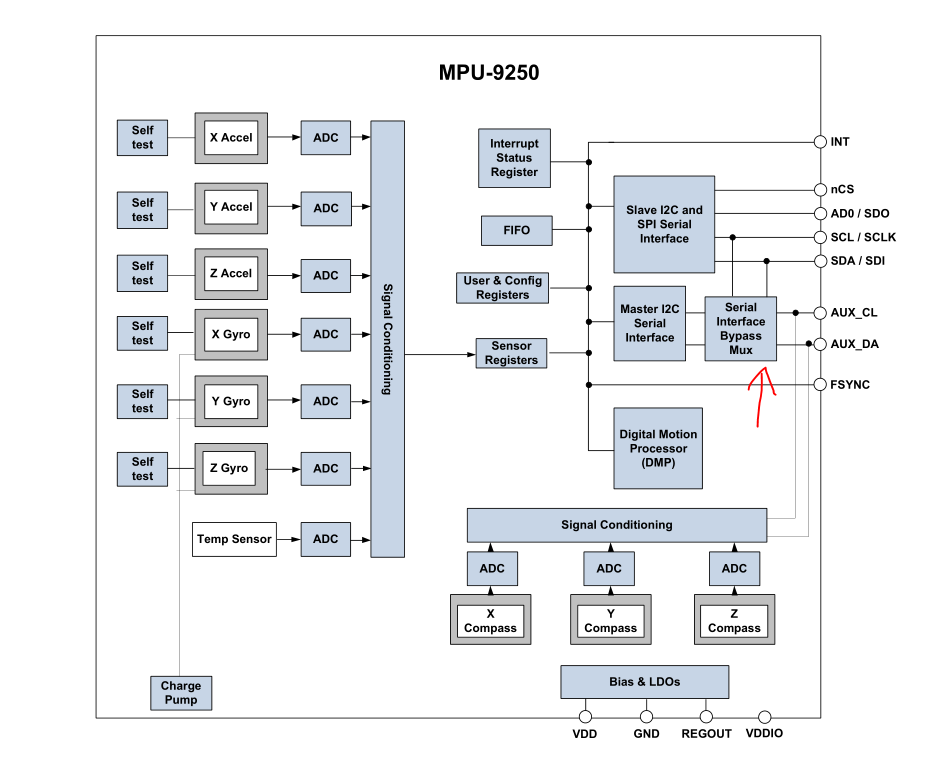
Acessado por I²C pelo endereço: 0x0C

addr de escrita: 0x0C << 1 = 0x18

addr de leitura: (0x0C << 1) + 1 = 0x19

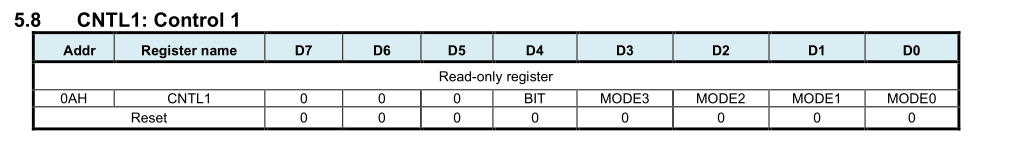
Existem duas formas de acessar o magnetômetro:

* configurando o i2c interno do MPU-9250 (difícil de debugar e usa muito mais registradores)
* desabilitando o i2c interno do MPU-9250 e habilitando o bypass mode para acessarmos o magnetômetro externamente com a nossa biblioteca de TWI pelo endereço 0x0C **(Forma implementada)**



**Modos de operação**

Possui 7 modos de operação definidos no registrador CNTL1 (0x0A):



* "0000": Power-down
* "0001": Única medida
* "0010": Medida contínua - modo 1 (sensor medindo a 8 Hz)
* "0110": Medida contínua - modo 2 (sensor medindo a 100 Hz)
* "0100": Modo de medida por trigger externo
* "1000": Modo Self-test
* "1111": Modo de acesso ao Fuse ROM
* Outros valores são proibidos

O BIT em D4 define se o output é de 14 (BIT = 0) ou 16 (BIT = 1) bits

Demora para trocar de modo, estou usando um delay de 100ms.

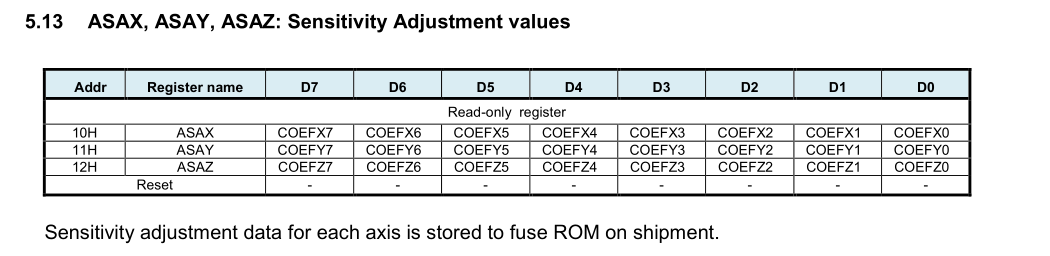
**Output**

O magnetômetro coleta dados em 3 eixos medindo num intervalo de ±4912µT.

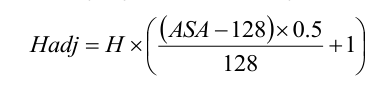
|  |  |  |
| --- | --- | --- |
| Nome do registrador | Endereço | Descrição |
| HXL | 0x03 | Dados do  eixo X |
| HXH | 0x04 |
| HYL | 0x05 | Dados do  eixo Y |
| HYH | 0x06 |
| HZL | 0x07 | Dados do  eixo Z |
| HZH | 0x08 |

Cuidado na hora de implementar. Diferente dos dados de aceleração e giro, o byte menos significativo é armazenado antes.

O output precisa ser corrigido com os *sensitivity adjustment values* que são únicos para cada sensor e são encontrados nos registradores ASAX, ASAY e ASAZ que podem ser acessados no modo FUSE ROM.



A leitura do sensor de cada eixo, “H”, precisa ser multiplicada seguindo a seguinte expressão:



Podemos converter para micro tesla multiplicado pela escala de 4912.0f / 32760.0f

O filtro de Madgwick usa a unidade Gauss, que pode ser obtida dividindo o valor em micro tesla por 100.

**Configuração:**

1. **Desabilitar o modo mestre do MPU-9250**

resetar bit 5 (I2C\_MST\_EN) do registrador 0x6A USER\_CTRL

Como não existe outra configuração, nesse registrador, escrevi 0x00

1. **Habilitar o bypass mode para acesso direto ao magnetômetro**

Habilitar bit 1 (BYPASS\_EN) do registrador 0x37 (INT\_PIN\_CFG)

Lembrar de não sobrescrevê-lo quando ativar o modo de interrupção

(fiz uma pequena alteração na função mpu\_int por conta disso trocando 0x80 para 0x82)

1. **Trocar para modo FUSE ROM (demora, usar um delay de 100ms depois)**

No registrador 0x0A (CNTL1), escolher o modo fuse rom (0x1F)

1. **Ler armazenar os registradores ASAX (0x10), ASAY (0x11) e ASAZ (0x12)**
2. **Trocar para modo POWER DOWN**

escrever 0x00 no registrador 0x0A (CNTL1)

esperar 100ms

1. **Trocar para modo contínuo de 100hz e output de 16 bits**

escrever 0x16 no registrador 0x0A (CNTL1)

esperar 100ms

**Código implementado:**

**defines para o magnetômetro em b-Defs.h:**

#define MAG\_CNTL\_1 0x0A //Controle 1

#define MAG\_CNTL\_2 0x0B //Controle 2

#define MAG\_XOUT\_L 0x03 //MAG XL seq:[XL XH YL YH ZL ZH]

#define MAG\_I2C\_ADDR 0x0C //endereço i2c do magnetometro

#define MAG\_I2C\_ADDR\_WR 0x18 //0x0c << 1

#define MAG\_I2C\_ADDR\_RD 0x19 //(0x0c << 1) + 1

#define MAG\_ASAX 0x10 //endereço do registrador ASAX do magnetometro

#define MAG\_ASAY 0x11 //endereço do registrador ASAY do magnetometro

#define MAG\_ASAZ 0x12 //endereço do registrador ASAZ do magnetometro

#define MAG\_ST1 0x02 //endereço do registrador ST1 do magnetometro que indica se o dado está pronto

**Variável global criada em b-Globs.h:**

volatile byte mag\_asa[3]; //local para armazenar os valores dos registradores asax, asay e asaz do magnetometro

**Testes realizados em w-Teste**

teste\_6: testa se a comunicação com o magnetometro funciona e envia por serial os dados do registrador ASA e as medidas de hx, hy e hz.

teste\_14: teste mpu junto com magnetometro e envia por serial os dados de x, y e z dos 3 sensores.

**funções criadas em c-MPU:**

**mpu\_mag\_config()**

configura o magnetômetro, executei ela ao final de mpu\_config, será que é melhor executar no setup em CXP?

**mpu\_rd\_mg\_reg(byte reg)**

Lê o registrador reg do magnetômetro usando nossa biblioteca de twi.

**mpu\_rd\_mg\_blk(byte reg, byte \*dado, byte qtd)**

Lê em bloco os registradores do magnetômetro a partir de reg

**mpu\_wr\_mg\_reg(byte reg, byte dado)**

escreve o byte dado no registrador reg do magnetômetro

**mag\_whoami()**

faz uma leitura do registrador 0x00 do magnetometro, que deve ser o id 0x48

**mpu\_rd\_mg\_out(word \*vetor)**

lê a saída do magnetometro. **vetor** deve ser um array com 3 palavras, pois serão armazenados em ordem [hx, hy, hz]

**Links interessantes:**

me ajudou a entender a configuração do magnetômetro:

[**https://longnight975551865.wordpress.com/2018/02/11/how-to-read-data-from-mpu9250/**](https://longnight975551865.wordpress.com/2018/02/11/how-to-read-data-from-mpu9250/)

calibração do magnetômetro:

[**https://appelsiini.net/2018/calibrate-magnetometer/**](https://appelsiini.net/2018/calibrate-magnetometer/)

[**https://github.com/kriswiner/MPU6050/wiki/Simple-and-Effective-Magnetometer-Calibration**](https://github.com/kriswiner/MPU6050/wiki/Simple-and-Effective-Magnetometer-Calibration)