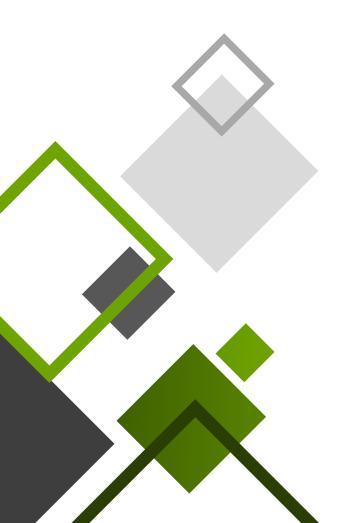
# Magnetometer

An **Unmanned aerial vehicle** (UAV) is a Unmanned Aerial Vehicle. UAVs include both autonomous (means they can do it alone) drones and remotely piloted vehicles (RPVs). A UAV is capable of controlled, sustained level flight and is powered by a jet, reciprocating, or electric engine.





## **CONTENTS** TITLE



- **O1 Magnetometer?**Magnetometer sensor에 대하여 알아보자.
- O2Magnetometer HMC5883LHMC5883L의 사용방법에 대하여 알아본다.
- O3Algorithm for HMC5883L측정 프로그램 알고리즘에 대하여 알아본다.
- 04
- 05



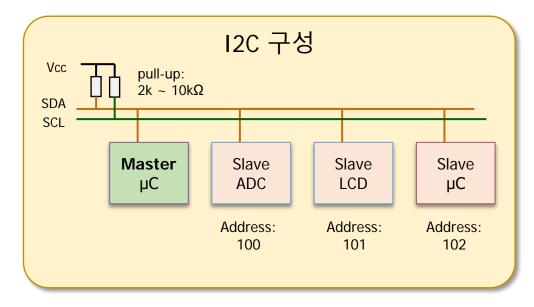
# I2C 통신



# I2C 통신

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- I2C는 1982년 Philips가 개발하였고 TWI라는 별칭도 있음.
  - 2선만으로 복수개의 장치와 1:n 통신이 가능하다는 것이 장점.
    - 마스터-슬레이브 방식
  - 초기에는 400kHz가 최대 속도였으나 최근에는 고속 모드도 지원.



### ■ 주요 특징

- 7비트 주소를 사용하여 최대 128대의 장치와 통신 가능
- SDA, SCL 라인을 2k~10kΩ 정도로 Pull-up 시켜 사용.
- master, slave 모두 Open drain 회로로 라인을 0으로 만들수 있음.



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 라이브러리 사용
  - #include <Wire.h>
- 통신의 초기화
  - Master/Slave 모드 결정, Slave 모드의 경우 주소 지정

## Wire.**begin(address)**;

• address : 생략 → Master mode, Slave 주소 → Slave mode



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 통신시작
  - 마스터모드의 경우 통신하고자하는 Slave주소 지정

## Wire.beginTransmission(address);

• address : 마스터가 통신하고자하는 Slave 주소 (0~127)



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 저장된 데이터 전송 종료
  - master mode에서만 사용한다.

## s = Wire.endTransmission(stop);

- stop: 정지신호 생성 여부, TRUE(생략 시 기본값): 생성, FALSE: 계속
- *s* : 전송성공여부: 0: 성공, 1:데이터 길이가 버퍼를 초과, 2: 해당 주소 대답 안함, 3, 데이터 전송 실패, 4: 그 외 오류



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

## ■ 데이터 전송 요청

• address의 slave에게 n바이트의 데이터 전송을 요청한다.

## b = Wire.requestFrom(address, quantity, stop);

- address : 통신하고자하는 Slave 주소 (0~127)
- quantity: 요청하는 데이터의 바이트 수
- stop: 정지신호 생성 여부 선택 TRUE (생략시 기본값): 생성, FALSE: 계속
- b : 요청 후 버퍼에 수신된 바이트 수



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

## ■ 데이터 전송

### Wire.write(value);

• value : 전송하고자 하는 byte형 데이터

## Wire.write(string);

• string : 전송하고자 하는 문자열 데이터

## Wire.write(dataArray, length);

- dataArray : 전송하고자 하는 byte형 배열
- length: dataArray의 길이 (바이트수)



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 데이터 읽기
  - Wire.requestFrom 으로 요청한 후 수신버퍼에서 1바이트 읽어 들인다.

byte=Wire.read();

- byte : 수신된 1바이트
- 수신 버퍼 확인
  - 수신 버퍼에 수신된 바이트 수 확인

nbye=Wire.available();

• nbyte : 버퍼에 수신된 바이트 수



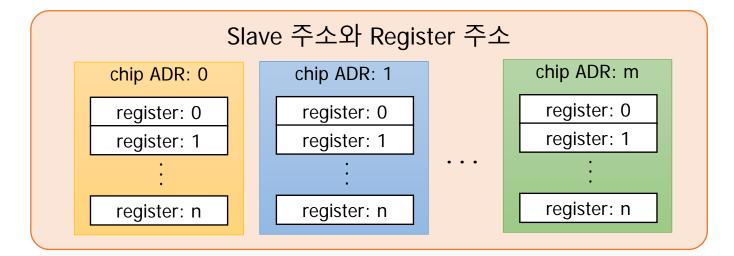
## 1바이트 쓰기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

## ■ 레지스터에 1바이트 데이터 쓰기 절차

- .beginTransmission(C\_ADR)
  - Slave 칩 주소 C\_ADR 지정
- .write(reg)
  - register 주소 reg전송
- .write(data)
  - 데이터 값 전송
- .endTransmission()

```
void writeRegister8(uint8_t reg, uint8_t value){
    Wire.beginTransmission(CHIP_ADDRESS);
    Wire.write(reg);
    Wire.write(value);
    Wire.endTransmission();
}
```



## 1바이트 읽기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

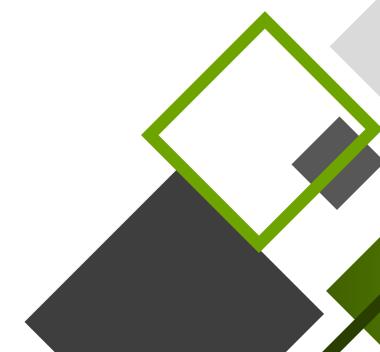
## ■ 레지스터에서 1바이트 읽기 절차

- .beginTransmission(C\_ADR)
  - Slave 칩 주소 C\_ADR 지정
- .write (reg)
  - 레지스터 주소 쓰기
- .endTransmission()
- .beginTransmission(ADR)
  - Slave 칩 주소 C\_ADR 지정
- .requestFrom(ADR,1)
  - 1 바이트 읽기 요청
- .available() == True 까지 대기
- read(data) → value
  - 1 바이트 읽기
- .endTransmission()

```
uint8_t readRegister8(uint8_t reg) {
    uint8_t value;
    Wire.beginTransmission(CHIP_ADDRESS);
    Wire.write(reg);
    Wire.endTransmission();
    Wire.beginTransmission(CHIP_ADDRESS);
    Wire.requestFrom(CHIP_ADDRESS, 1);
    while(!Wire.available()) {};
    value = Wire.read();
    Value.endTransmission();
    return value;
}
```



# Magnetometer



# AMR센서의 자기력 측정원리

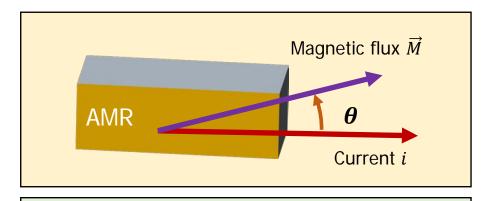
Dept. of Mechanical System Design, Seoul National University of Science and Technology.

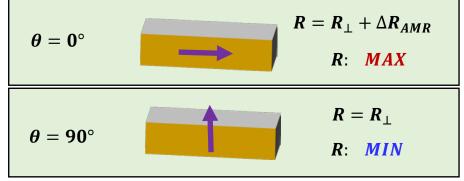
- AMR(Anisotropic Magneto Resistive) 센서
  - 가해지는 자기력방향에 따라 저항이 변화하는 소자.
- 동작원리

$$R = R_{\perp} + \Delta R_{AMR} cos^2 \theta$$

 $R_{\perp}$ : 자기장이 직각일 때의 저항

 $\Delta R_{AMR}$ : AMR상수, 민감도에 해당



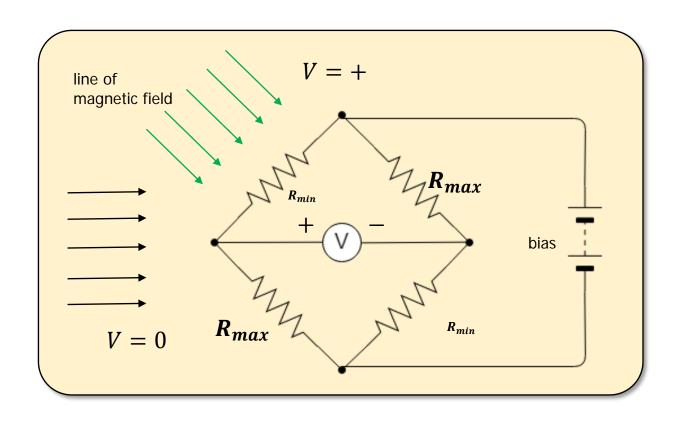




# AMR센서의 자기력 측정원리

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- AMR센서 측정 회로
  - AMR을 4개를 사용하여 Wheatstone bridge 를 구성함.
  - 이들을 조합하면 자기력선 의 방향을 알 수 있는 센서 를 만들 수 있음.



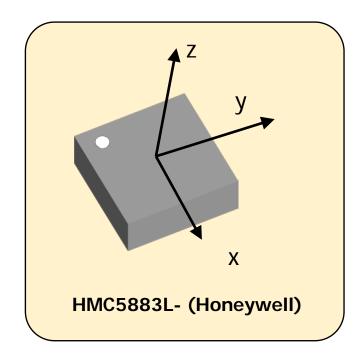


# 3-Axis Digital Compass HMC5883L

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

## ■ 주요 특징

- 3-Axis Magnetoresistive Sensors
- 12bit ADC
- 2 milli-gauss Field Resolution in ±8 Gauss Fields
  - Enables 1° to 2° Degree Compass Heading Accuracy
- I2C Digital Interface
  - Maximum Output Rate: 160 Hz
- Output:  $m_x$ ,  $m_y$ ,  $m_z$ ,

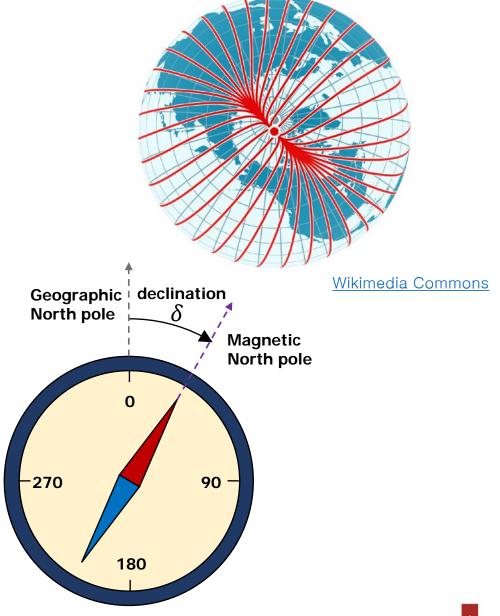




# **Magnetic declination**

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 진북 (Geographic North pole)
  - 지구의 회전축에 해당하는 지리학적 북극점
- 자북 (MagneticNorth pole)
  - 지구의 자기력선이 모이는 곳
- 자기 편향각(Magnetic declination)
  - 진북과 자북사이의 시계 방향으로의 각도

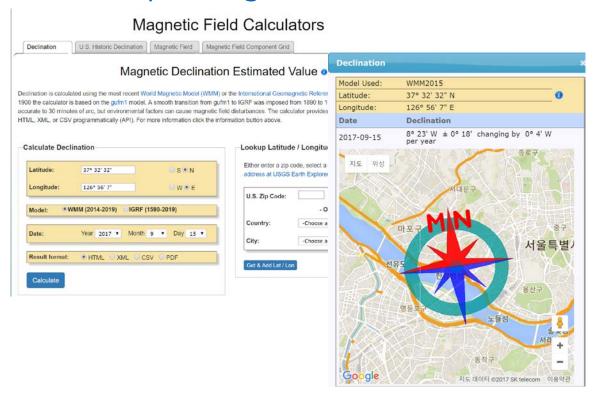


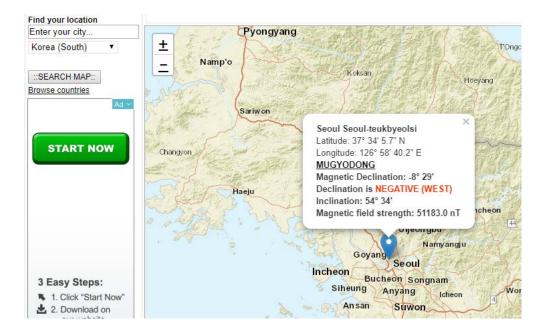


# **Magnetic declination**

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 자기편향각 찾기 사이트
  - https://www.ngdc.noaa.gov/geomag-web/
  - http://magnetic-declination.com/







# Redefining coordinate

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

## ■ 항공기에서는 주로 NED 좌표계로 제어를 실시함

- 변경내용
  - N: 전진 방향이 +x
  - E: 전진 방향의 오른쪽이 +y
  - D: 중력 방향이 +z
- 필요 내용
  - 다음과 같이 NED 좌표 변환이 필요함.
  - APM에 장착된 경우 : 전진 방향이 Y축으로 칩이 장착됨

 $x_{NED} = y_{APM}, y_{NED} = x_{APM}, z_{NED} = -z_{APM}$ 

• GPS에 장착된 경우: HMC5883L이 칩이 후면에 장착됨.

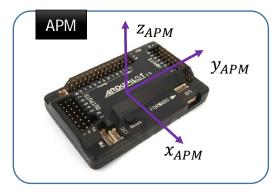
 $x_{NED} = y_{GPS}$ ,  $y_{NED} = -x_{GPS}$ ,  $z_{NED} = z_{GPS}$ 

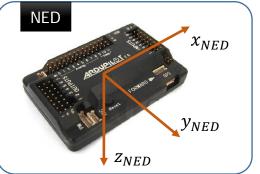
### NED 좌표계

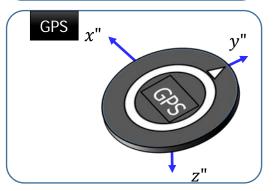
x: North

y: East

z: Down









# 지자기를 이용한 방위각 구하기

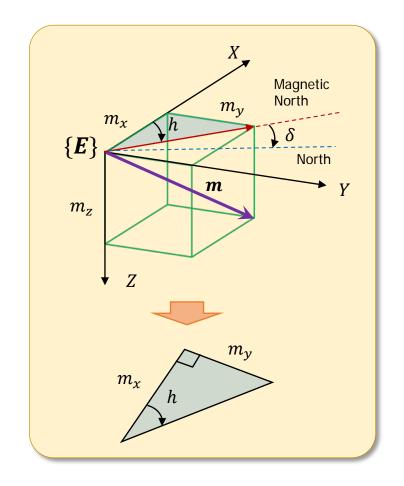
Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 기준 방위각
  - Heading, azimuth, bearing 등으로도 불림.
  - 진북 방향을 기준으로 정의
- {E} 좌표에서 yaw만 있는 경우
  - 지자기 센서로  $m_{\chi}$ ,  $m_{V}$ ,  $m_{Z}$  측정
  - 지자기 벡터 *m*을 *X,Y* 평면에 투영.
  - $\bullet$  직각 삼각형에서 h를 구함

$$h = atan2(m_y, m_x) (1)$$

• 자기 편향(declination)  $\delta$  을 반영:

$$h' = h + \delta \tag{2}$$





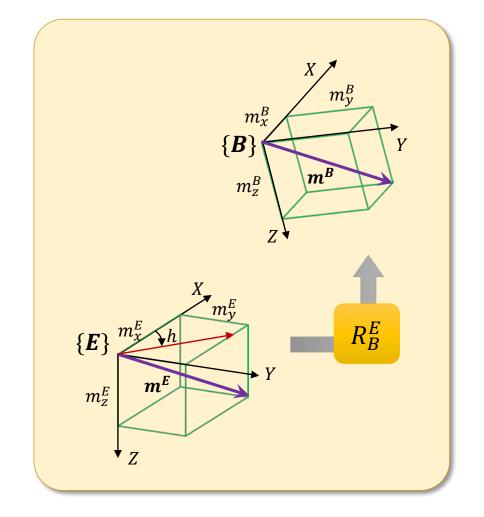
# 지자기를 이용한 방위각 구하기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- Vehicle에 roll, pitch가 있는 경우
  - $\{E\}$  에서 측정된 지자기 벡터를  $m^E$
  - $\{B\}$  에서 측정된 지자기 벡터를  $m^B$
  - $m^E$ ,  $m^B$  는 같은 벡터인데 다른 좌표계에서 측정함
  - {E} 에서 본 {B} 좌표계의 회전행렬  $R_B^E$

$$R_B^E = R(\theta, y)R(\phi, x) = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}$$
$$= \begin{bmatrix} c\theta & s\phi s\theta & c\phi s\theta \\ 0 & c\phi & -s\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

• 따라서,  $m^E = R_B^E m^B$ 





# 지자기를 이용한 방위각 구하기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

• 그러므로

$$\begin{bmatrix}
m_{x}^{E} \\
m_{y}^{E} \\
m_{z}^{E}
\end{bmatrix} = \begin{bmatrix}
c\theta & s\phi s\theta & c\phi s\theta \\
0 & c\phi & -s\phi \\
-s\theta & s\phi c\theta & c\phi c\theta
\end{bmatrix} \begin{bmatrix}
m_{x}^{B} \\
m_{y}^{B} \\
m_{z}^{B}
\end{bmatrix}$$
(3)

• 식 (3) 의 1,2 행에서

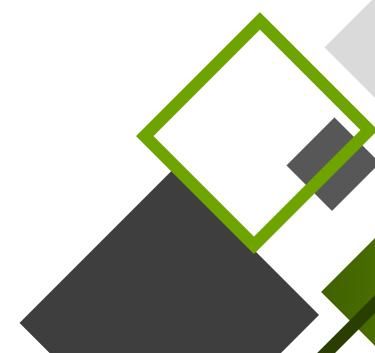
$$m_{\chi}^{E} = m_{\chi}^{B}c\theta + m_{y}^{B}s\phi s\theta + m_{z}^{B}c\phi s\theta \qquad (4)$$
  
$$m_{\gamma}^{E} = m_{\gamma}^{B}c\phi - m_{z}^{B}s\phi \qquad (5)$$

- $m_x^E$ ,  $m_y^E$  는 {E} 좌표계의 값이므로 (1) 식과 같이  $h = atan2(m_y^E, m_x^E)$
- 참고로 (4), (5) 에서  $\phi = 0$ ,  $\theta = 0$  이면  $m_x^E = m_x^B$ ,  $m_x^E = m_x^B$





# Registers of HMC5883L



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

## ■ 내부 레지스터

| Addr. | Name                       | Access | Descriptions   |
|-------|----------------------------|--------|--|
| 00    | Configuration Register A   | R/W    | <ol> <li>Select number of samples averaged,</li> <li>Data output rate, 3) Bias settings</li> </ol> |
| 01    | Configuration Register B   | R/W    | 1) Gain Configuration  |
| 02    | Mode Register              | R/W    | 1) Mode selection  |
| 03    | Data Output X MSB Register | R      | 16 bit V data  |
| 04    | Data Output X LSB Register | R      | - 16 bit X data  |
| 05    | Data Output Z MSB Register | R      | 1/ bit V data  |
| 06    | Data Output Z LSB Register | R      | - 16 bit X data  |
| 07    | Data Output Y MSB Register | R      | 1/ hit V data  |
| 08    | Data Output Y LSB Register | R      | - 16 bit X data  |
| 09    | Status Register            | R      | Status of chip   |
| 10    | Identification Register A  | R      |  |
| 11    | Identification Register B  | R      | Identify the chip (0x48, 0x34, 0x33)   |
| 12    | Identification Register C  | R      |  |



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

## Configuration Register A

 7
 6
 5
 4
 3
 2
 1
 0

 CRA7(0) MA1(1) MA0(1) DO2(1) DO1(0) DO0(0) MS1(0) MS0(0)

Number of samples averaged

| MA1 | MAO | # of samples |
|-----|-----|--------------|
| 0   | 0   | 1            |
| 0   | 1   | 2            |
| 1   | 0   | 4            |
| 1   | 1   | 8            |

Data output rate

| D02 | DO1 | DO0 | Data Rate (Hz) |
|-----|-----|-----|----------------|
| 0   | 0   | 0   | 0.75           |
| 0   | 0   | 1   | 1.5            |
| 0   | 1   | 0   | 3              |
| 0   | 1   | 1   | 7.5            |
| 1   | 0   | 0   | 15(default)    |
| 1   | 0   | 1   | 30             |
| 1   | 1   | 0   | 75             |
| 1   | 1   | 1   | NA             |

Bias setting

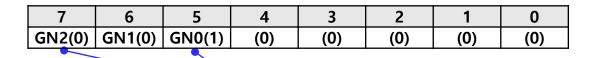
| MS1 | MS0 | Mode            |  |  |
|-----|-----|-----------------|--|--|
| 0   | 0   | Normal(default) |  |  |
| 0   | 1   | positive bias   |  |  |
| 1   | 0   | negative bias   |  |  |
| 1   | 1   | reserved        |  |  |



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

## Configuration Register B

**CRB** 



Gain

| GN2 | GN1 | GNO | Range(Ga)     | Gain (LSB/Ga) | Resolution(mGa/LSB) |
|-----|-----|-----|---------------|---------------|---------------------|
| 0   | 0   | 0   | <u>±</u> 0.88 | 1370          | 0.73                |
| 0   | 0   | 1   | <u>±</u> 1.3  | 1090(default) | 0.92                |
| 0   | 1   | 0   | <u>+</u> 1.9  | 820           | 1.22                |
| 0   | 1   | 1   | <u>±</u> 2.5  | 660           | 1.52                |
| 1   | 0   | 0   | <u>+</u> 4.0  | 440           | 2.27                |
| 1   | 0   | 1   | <u>+</u> 4.7  | 390           | 2.56                |
| 1   | 1   | 0   | ±5.6          | 330           | 3.03                |
| 1   | 1   | 1   | <u>+</u> 8.1  | 230           | 4.35                |



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

## Configuration Register B

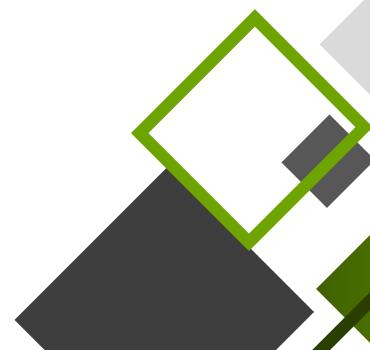
**MR** 

| 7   | 6   | 5   | 4   | 3   | 2   | 1      | 0      |
|-----|-----|-----|-----|-----|-----|--------|--------|
| (0) | (0) | (0) | (0) | (0) | (0) | MD1(0) | MD0(0) |

Operating mode

| MD1 | MD0 | Operating mode              |
|-----|-----|-----------------------------|
| 0   | 0   | Continuous Measurement Mode |
| 0   | 1   | Single Measurement Mode     |
| 1   | 0   | Idle Mode                   |
| 1   | 1   | Idle Mode                   |

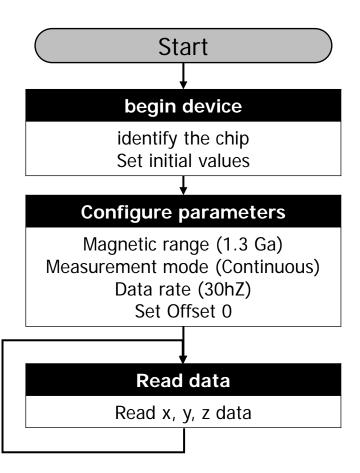




Dept. of Mechanical System Design, Seoul National University of Science and Technology.

### Structure:

- HMC5883L\_basic.ino
- HMC5883L.h
- HMC5883L.cpp



Dept. of Mechanical System Design, Seoul National University of Science and Technology.

### HMC5883L Class:

- Property
  - ▶ mgPerDigit: 비트 당 mGa
  - ▶ xOffset, yOffset: x,y offset 값
  - ▶ v : 벡터구조체
- Method
  - ▶ setRange(..): range 설정
  - ▶ setMeasurementMode(..): 측정 모드 설정
  - ▶ setDataRate(..): 데이터 측정 속도
  - ▶ setSamples(..): 데이터 평균 개수 설정
  - ▶ Vector readRaw(): 측정 데이터 반환
  - ▶ Vector readNormalize(): 정규화된 데이터 반환

### HMC5883L Class

property

float mgPerDigit int xOffset, yOffset Vector v

### public:

void begin()

setRange(range) setMeasurementMode(mode) setDataRate(dataRate) setSamples(samples) Vector readRaw() Vector readNormalize()

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

### ■ C 언어 팁:

- struct ASTRUCT{...};
  - ▶ 구조체 ASTRUCT를 정의함.
- ▶ enum AENUM{...} var;
  - ▶ 열거형 AENUM을 정의한다
- typedef old\_type new\_type;
  - ▶ new\_type 으로도 old\_type을 정의할 수 있게 함.
- typedef enum{...} new\_type;
  - ▶ new\_type으로 enum{...} 즉, 열거형을 정의할 수 있음.

```
struct Vector{
   float X;
   float Y;
   float Z;};
Vector v={1.0,-2.0,2.5};
enum COLOR{RED,
```

```
enum COLOR{RED,
BLUE,GREEN,YELLOW};
COLOR color=BLUE;
```

```
typedef enum{
  APPLE = 1,
  BANANA = 2,
  STRAWBERRY = 3,
} FRUIT;
FRUIT a = BANANA;
```

# 헤더 파일

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 주요내용
  - HMC5883L 의 레지스터 주소 지정

#### HMC5883L.h [1/3] #define HMC5883L\_ADDRESS (0x1E)#define HMC5883L\_REG\_CONFIG\_A (0x00)#define HMC5883L\_REG\_CONFIG\_B (0x01)#define HMC5883L REG MODE (0x02)#define HMC5883L\_REG\_OUT\_X\_M (0x03)#define HMC5883L\_REG\_OUT\_X\_L (0x04)#define HMC5883L\_REG\_OUT\_Z\_M (0x05)#define HMC5883L\_REG\_OUT\_Z\_L (0x06)#define HMC5883L REG OUT Y M (0x07)#define HMC5883L REG OUT Y L (0x08)#define HMC5883L\_REG\_STATUS (0x09)#define HMC5883L REG IDENT A $(0 \times 0 A)$

(0x0B)

(0x0C)

#define HMC5883L\_REG\_IDENT\_B

#define HMC5883L REG IDENT C

# 헤더 파일

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 주요 내용:
  - 열거형, 구조체 정의

```
HMC5883L.h [2/3]
typedef enum{
                          = 0b11,
   HMC5883L SAMPLES 8
   HMC5883L_SAMPLES_4
                         = 0b10
   HMC5883L_SAMPLES_2
                        = 0b01,
   HMC5883L SAMPLES 1
                         = 0b00
} hmc5883l_samples_t;
typedef enum{
                            = 0b110,
   HMC5883L DATARATE 75HZ
   HMC5883L_DATARATE_30HZ
                            = 0b101,
                           = 0b100
   HMC5883L DATARATE 15HZ
   HMC5883L_DATARATE_7_5HZ = 0b011,
                           = 0b010,
   HMC5883L DATARATE 3HZ
   HMC5883L_DATARATE_1_5HZ = 0b001
   HMC5883L_DATARATE_0_75_HZ = 0b000
} hmc5883l_dataRate_t;
```

```
typedef enum{
   HMC5883L_RANGE_8_1GA
                            = 0b111,
                            = 0b110
   HMC5883L RANGE 5 6GA
   HMC5883L RANGE 4 7GA
                            = 0b101.
   HMC5883L_RANGE_4GA
                            = 0b100
   HMC5883L_RANGE_2_5GA
                            = 0b011.
   HMC5883L_RANGE_1_9GA
                            = 0b010,
   HMC5883L RANGE 1 3GA
                            = 0b001.
                            = 0b000
    HMC5883L RANGE 0 88GA
} hmc5883l_range_t;
typedef enum {
                          = 0b10.
   HMC5883L IDLE
                          = 0b01.
   HMC5883L SINGLE
   HMC5883L CONTINOUS
                          = 0b00
} hmc5883l_mode_t;
struct Vector{ float XAxis; float YAxis;
              float ZAxis; };
```

# 헤더 파일

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 주요 내용
  - HMC5883L Class 정의

### HMC5883L.h [3/3]

```
class HMC5883L{
  public:
    Vector readRaw(void);
    Vector readNormalize(void);
    bool begin(void);
    bool init(void);
    void setOffset(int xo, int yo);
    void setRange(hmc5883l_range_t range);
    void setMeasurementMode(hmc5883l_mode_t mode);
    void setDataRate(hmc5883l_dataRate_t dataRate);
    void setSamples(hmc5883l samples t samples);
  private:
    float mgPerDigit;
    Vector v:
    int xOffset, yOffset;
    void writeRegister8(uint8_t reg, uint8_t value);
    uint8_t readRegister8(uint8_t reg);
    uint8_t fastRegister8(uint8_t reg);
    int16_t readRegister16(uint8_t reg);
};
```

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 주요 내용
  - begin 멤버 함수 구현
  - 각 축별 원시 데이터 읽기

#### HMC5883L.cpp [1/4]

```
#include <Wire.h>
#include "HMC5883L.h"
bool HMC5883L::begin() {
   Wire.begin();
    if ((readRegister8(HMC5883L_REG_IDENT_A) != 0x48)
        | | (readRegister8(HMC5883L REG IDENT B) != 0x34)
        || (readRegister8(HMC5883L_REG_IDENT_C) != 0x33))
          return false;
    setRange(HMC5883L_RANGE_1_3GA);
    setMeasurementMode(HMC5883L_CONTINOUS);
    setDataRate(HMC5883L DATARATE 15HZ);
    setSamples(HMC5883L SAMPLES 1);
    mgPerDigit = 0.92f;
    return true;
Vector HMC5883L::readRaw(void) {
    v.XAxis = readRegister16(HMC5883L_REG_OUT_X_M) - xOffset;
    v.YAxis = readRegister16(HMC5883L_REG_OUT_Y_M) - yOffset;
    v.ZAxis = readRegister16(HMC5883L REG OUT Z M);
    return v;
```

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

## ■ 측정 레인지 설정

#### HMC5883L.cpp [2/4]

```
void HMC5883L::setRange(hmc5883l_range_t range){
    switch(range) {
    case HMC5883L_RANGE_0_88GA:
        mgPerDigit = 0.073f;break;
    case HMC5883L_RANGE_1_3GA:
        mgPerDigit = 0.92f;break;//mGauss
    case HMC5883L_RANGE_1_9GA:
        mgPerDigit = 1.22f;break;
    case HMC5883L RANGE 2 5GA:
        mgPerDigit = 1.52f;break;
    case HMC5883L_RANGE_4GA:
        mgPerDigit = 2.27f;break;
    case HMC5883L RANGE 4 7GA:
        mgPerDigit = 2.56f;break;
    case HMC5883L_RANGE_5_6GA:
        mgPerDigit = 3.03f;break;
    case HMC5883L_RANGE_8_1GA:
        mgPerDigit = 4.35f;break;
   writeRegister8(HMC5883L_REG_CONFIG_B, range << 5);</pre>
```

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 주요 내용
  - 정규화
  - 옵셋 설정
  - 측정 모드 설정
    - 특정 비트의 선택적 변경

•

#### HMC5883L.cpp [3/4]

```
Vector HMC5883L::readNormalize(void) {
   v.XAxis=((float)readRegister16(HMC5883L_REG_OUT_X_M)
                              - xOffset)*mgPerDigit;
   v.YAxis=((float)readRegister16(HMC5883L_REG_OUT_Y_M)
                              - yOffset)*mgPerDigit;
   v.ZAxis=(float)readRegister16(HMC5883L REG OUT Z M)*mgPerDigit;
    return v;
void HMC5883L::setOffset(int xo, int yo) {
   x0ffset = xo; y0ffset = yo;
void HMC5883L::setMeasurementMode(hmc5883l_mode_t mode){
   uint8_t value;
   value = readRegister8(HMC5883L_REG_MODE);
   value &= 0b111111100;
                                 // Clear block
   value |= mode;
   writeRegister8(HMC5883L_REG_MODE, value);
```

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- 주요 내용
  - 데이터 출력 속도 지정
    - 특정 비트의 선택적 변경
  - 샘플수 지정

#### HMC5883L.cpp [4/4]

```
void HMC5883L::setDataRate(hmc5883l_dataRate_t dataRate) {
    uint8 t value;
    value = readRegister8(HMC5883L_REG_CONFIG_A);
    value &= 0b11100011;  // Clear block
    value |= (dataRate << 2);</pre>
    writeRegister8(HMC5883L_REG_CONFIG_A, value);
void HMC5883L::setSamples(hmc5883l_samples_t samples){
    uint8 t value;
    value = readRegister8(HMC5883L_REG_CONFIG_A);
    value &= 0b10011111;  // Clear block
    value |= (samples << 5);</pre>
    writeRegister8(HMC5883L_REG_CONFIG_A, value);
void HMC5883L::init(){
  setRange(HMC5883L_RANGE_1_3GA); // 1.3Ga/Lsb
  setMeasurementMode(HMC5883L_CONTINOUS);
  setDataRate(HMC5883L_DATARATE_30HZ);
  setSamples(HMC5883L_SAMPLES_8);
  setOffset(0, 0);
```

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

### ■ 코드

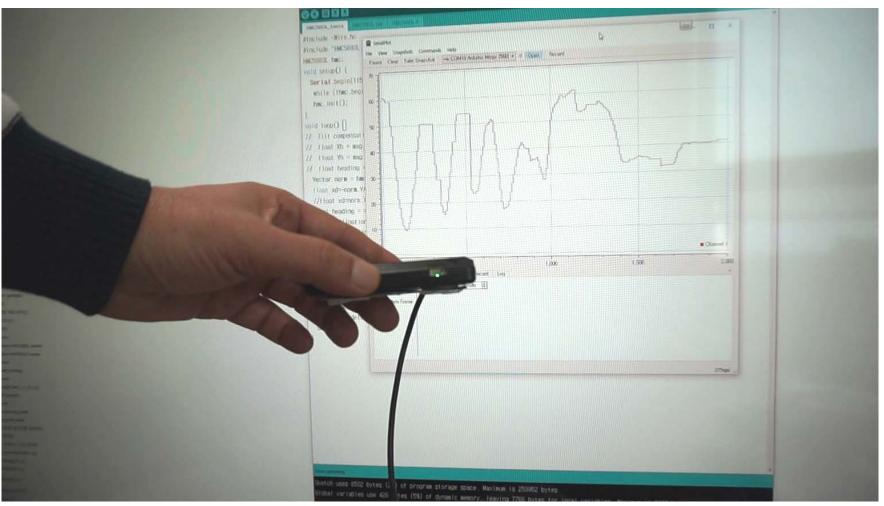
- HMC5883L hmc
  - hmc 객체 생성
- hmc.begin()
  - hmc 장치 활성화
- hmc.init()
  - hmc 설정
- hmc.readNormalize()
  - 데이터 정규화
- heading 각 계산
- 자기편향 보상
- 출력

### HMC5883L\_basic.ino

```
#include <Wire.h>
#include "HMC5883L.h"
HMC5883L hmc;
void setup() {
  Serial.begin(115200);
   while (!hmc.begin()) delay(500);
   hmc.init();
void loop() {
  Vector norm = hmc.readNormalize();
  float heading = atan2(norm.YAxis, norm.XAxis);
  //Seoul -8o29' NEGTIVE(WEST)
  float declinationAngle=-(8.0+ (29.0/60.0))/(180 /PI);
  heading += declinationAngle;
  float headingDegrees = heading * 180/PI;
  Serial.println(headingDegrees);
  delay(30);
```

## Results

Dept. of Mechanical System Design, Seoul National University of Science and Technology.







# **THANK** YOU

Powerpoint is a complete presentation graphic package it gives you everything you need to produce a professional-looking presentation

