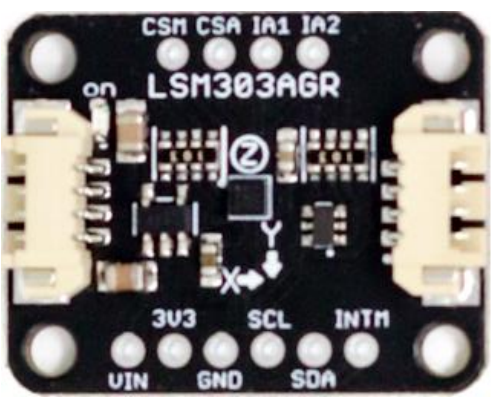
SmartElex LSM303AGR Accelerometer Magnetometer

The LSM303 breakout board combines a magnetometer/compass module with a triple-axis accelerometer to make a compact navigation subsystem. The I2C interface is compatible with both 3.3v and 5v processors and the two pins can be shared by other I2C devices. Combined with a 3-axis gyro such as the L3GD20, you have all the sensors you need for a complete IMU (Inertial Measurement Unit) for use in aerial, terrestrial or marine navigation.

In this tutorial we will show you how to connect the LSM303 to an Arduino and how to use it to measure orientation relative to the earth's magnetic field, and acceleration in three axis.



How it Works:

MEMS - Micro Electro-Mechanical Systems

The sensor consists of micro-machined structures on a silicon wafer. There are structures designed to measure acceleration and magnetic fields in the X, Y and Z axis

Acceleration Measurement

These structures are suspended by polysilicon springs which allow them to deflect when subject to acceleration in the X, Y and/or Z axis. Deflection causes a change in capacitance between fixed plates and plates attached to the suspended structure. This change in capacitance on each axis is converted to an output voltage proportional to the acceleration on that axis.

Magnetic Field Measurement

These structures are similar to the accelerometer structures, but are etched with microscopic coils. An excitation current is passed through the coils, and the Lorentz Force due to the magnetic field causes the structure to deflect. Once again the deflection is converted to an output voltage proportional to the strength of the magnetic field in that axis. **Note :** That while the LSM303AGR breakout has CS pins, it only supports 3-wire SPI which we were not able to get working.

Power Pins

- **Vin** - this is the power pin. Since the sensor chip uses 3.3 VDC, we have included a voltage regulator on board that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V microcontroller like Arduino, use 5V
- **3Vo** - this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this if you like
- **GND** - common ground for power and logic

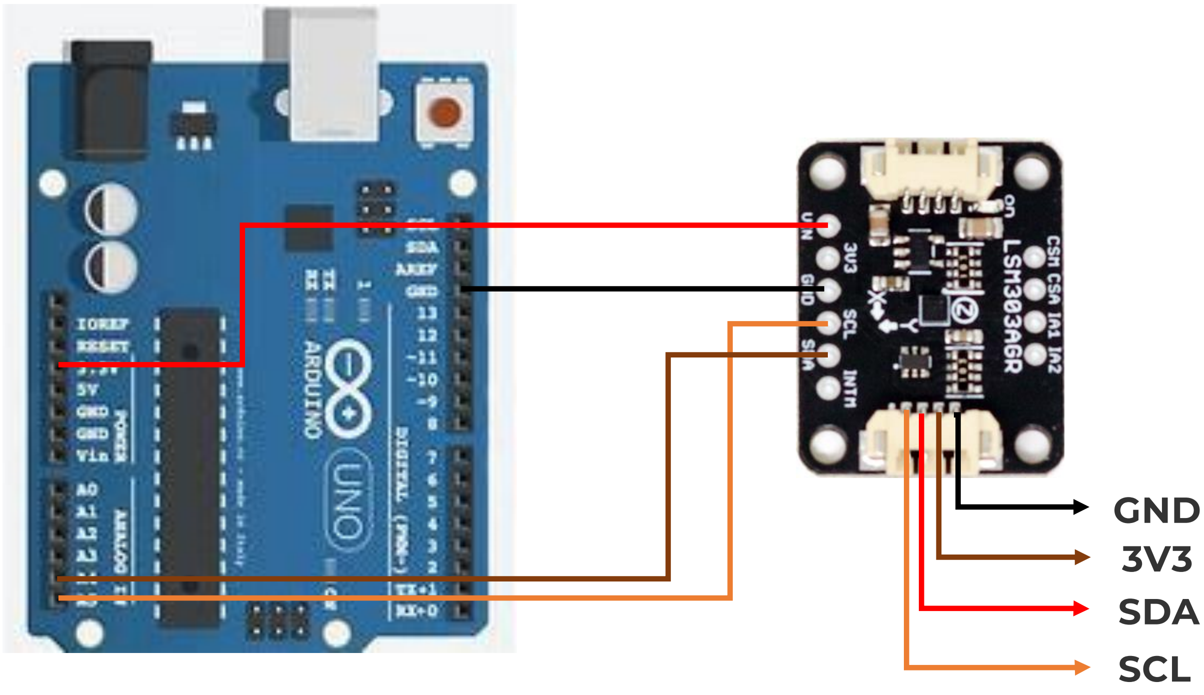
I2C Logic Pins

- **SCL** - I2C clock pin, connect to your microcontroller's I2C clock line. This pin is level shifted so you can use 3-5V logic, and there's a **10K pullup** on this pin.
- **SDA** -I2C data pin, connect to your microcontroller's I2C data line. This pin is level shifted so you can use 3-5V logic, and there's a **10K pullup** on this pin.

Other Common Pins

- **IA1** - The accelerometer's first interrupt pin. 3V logic only
- **IA2 -** The accelerometer's second interrupt pin. 3V logic only
- **INTM** - The interrupt pin for the magnetometer. 3V logic only
- **CSA -** The accelerometer's CS pin. 3V logic only
- **CSM -** The magnetometer's CS pin. 3V logic only

Wiring



Arduino	LSM303AGR
SCL(A5)	SCL
SDA(A4)	SDA
5v OR 3.3v	VIN
GND	GND

- Connect **board VCC** to **Arduino 5V** if you are running a **5V** board Arduino (Uno, etc.). If your board is **3V,** connect to that instead.
- Connect **board GND** to **Arduino GND**
- Connect **board SCL** to **Arduino SCL**
- Connect **board SDA** to **Arduino SDA**

Install the Libraries

To get started with the LSM303, you'll need to install the accelerometer library and the magnetometer library for your board. Additionally you will need the Adafruit_Sensor library that allows it to return data in a consistent way with other similar sensors, as well as the Adafruit_BusIO library. All of the libraries can be installed using the Library Manager in the Arduino IDE:

Install the Accelerometer Library

Click the **Manage Libraries** menu item, search for **Adafruit LSM303 Accel**, and select the **Adafruit_LSM303_Accel** library:

Install the Magnetometer Library

Next you'll need to install the library for the magnetometer in the LSM303. Make sure to **download the correct driver for your breakout board**.

LSM303AGR

Search the library manager for the **Adafruit_LIS2MDL** library:

Accelerometer Demo

This first demo will show you how to get readings of what an accelerometer does best: measure acceleration!

Open up **File -> Examples -> Adafruit LSM303 Accel-> accelsensor** and upload to your Arduino wired up to the sensor.

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at **115200 baud**. You should see the the values for the current configuration settings printed on startup, followed by acceleration readings for the X, Y, and Z axes similar to this:

```
Accelerometer Test

------------------------------------
Sensor:       LSM303
Driver Ver:   1
Unique ID:    54321
Max Value:    0.00 m/s^2
Min Value:    0.00 m/s^2
Resolution:   0.00 m/s^2
------------------------------------

Range set to: +- 4G
Mode set to: Normal
X: 0.31  Y: 0.00  Z: 9.89  m/s^2
X: 0.38  Y: -0.08  Z: 9.97  m/s^2
X: 0.38  Y: -0.08  Z: 9.82  m/s^2
```

The Adafruit_LSM303_Accel_Unified sensor class in the Adafruit_LSM303_Accel library reports X, Y and Z axis accelerometer readings directly in meters per second squared. The accelsensor example code in the library reads from the sensor and prints the acceleration readings to the Serial Monitor.

At rest, the sensor should report no acceleration except that due to gravity (about

9.8 meters/second squared). By calculating the angle of the gravity vector with respect to the X, Y and Z axis, the device can be used as an inclinometer.

Basic Magnetometer Readings

This first demo will show you how to get readings of what an accelerometer does best: measure acceleration!

LSM303AGR

Open up **File -> Examples -> Adafruit LIS2MDL-> magsensor** and upload to your Arduino wired up to the sensor.

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at **115200 baud**. You should see the the values for the current configuration settings printed on startup, followed by magnetic field readings for the X, Y, and Z axes similar to this:

```
Magnetometer Test

------------------------------------
Sensor:       LIS2MDL
Driver Ver:   1
Unique ID:    12345
Max Value:    0.00 uT
Min Value:    0.00 uT
Resolution:   0.00 uT
------------------------------------

X: 17.40  Y: 65.55  Z: -29.40  uT
X: 16.35  Y: 65.25  Z: -30.00  uT
X: 16.65  Y: 65.85  Z: -29.40  uT
```

The sensor class in the magnetometer library reports X, Y and Z axis magnetometer readings directly in micro-Teslas. The **magsensor** example code reads from the sensor and prints the micro-Tesla readings to the Serial Monitor.

In the absence of any strong local magnetic fields, the sensor readings should reflect the magnetic field of the earth (between 20 and 60 micro-Teslas). When the sensor is held level, by calculating the angle of the magnetic filed with respect to the X and Y axis, the device can be used as a compass.

Computing a Compass Heading

To convert the microTesla readings into a 0-360 degree compass heading, we can use the atan2() function to compute the angle of the vector defined by the Y and X axis readings. The result will be in radians, so we multiply by 180 degrees and divide by Pi to convert that to degrees.

LSM303AGR

Open up **File -> Examples -> Adafruit LIS2MDL-> compass** and upload to your Arduino wired up to the sensor.

Upload the sketch to your board and open up the Serial Monitor (**Tools -> Serial Monitor**) at **115200 baud**. You will see heading calculations printed out to the serial monitor. If you rotate the sensor as it runs you can see the heading change:

```
Magnetometer Test

Compass Heading: 91.14
Compass Heading: 87.49
Compass Heading: 81.48
```

Accelerometer Demo Code

```
#include <Adafruit_LSM303_Accel.h>

#include <Adafruit_Sensor.h>

#include <Wire.h>


/* Assign a unique ID to this sensor at the same time */

Adafruit_LSM303_Accel_Unified accel = Adafruit_LSM303_Accel_Unified(54321);


void displaySensorDetails(void) {

  sensor_t sensor;

  accel.getSensor(&sensor);

  Serial.println("----------------------------------------");

  Serial.print("Sensor:       ");

  Serial.println(sensor.name);

  Serial.print("Driver Ver:   ");

  Serial.println(sensor.version);

  Serial.print("Unique ID:    ");

  Serial.println(sensor.sensor_id);

  Serial.print("Max Value:    ");

  Serial.print(sensor.max_value);

  Serial.println(" m/s^2");

  Serial.print("Min Value:    ");
```

```
  Serial.print(sensor.min_value);
  Serial.println(" m/s^2");
  Serial.print("Resolution:   ");
  Serial.print(sensor.resolution);
  Serial.println(" m/s^2");
  Serial.println("------------------------------------");
  Serial.println("");
  delay(500);
}

void setup(void) {
#ifndef ESP8266
  while (!Serial)
    ; // will pause Zero, Leonardo, etc until serial console opens
#endif
  Serial.begin(9600);
  Serial.println("Accelerometer Test");
  Serial.println("");

  /* Initialise the sensor */
  if (!accel.begin()) {
    /* There was a problem detecting the ADXL345 ... check your connections */
    Serial.println("Ooops, no LSM303 detected ... Check your wiring!");
    while (1)
      ;
  }

  /* Display some basic information on this sensor */
  displaySensorDetails();
```

```
accel.setRange(LSM303_RANGE_4G);
Serial.print("Range set to: ");
lsm303_accel_range_t new_range = accel.getRange();
switch (new_range) {
case LSM303_RANGE_2G:
  Serial.println("+- 2G");
  break;
case LSM303_RANGE_4G:
  Serial.println("+- 4G");
  break;
case LSM303_RANGE_8G:
  Serial.println("+- 8G");
  break;
case LSM303_RANGE_16G:
  Serial.println("+- 16G");
  break;
}

accel.setMode(LSM303_MODE_NORMAL);
Serial.print("Mode set to: ");
lsm303_accel_mode_t new_mode = accel.getMode();
switch (new_mode) {
case LSM303_MODE_NORMAL:
  Serial.println("Normal");
  break;
case LSM303_MODE_LOW_POWER:
  Serial.println("Low Power");
  break;
case LSM303_MODE_HIGH_RESOLUTION:
  Serial.println("High Resolution");
```

```
    break;
  }
}

void loop(void) {
  /* Get a new sensor event */
  sensors_event_t event;
  accel.getEvent(&event);

  /* Display the results (acceleration is measured in m/s^2) */
  Serial.print("X: ");
  Serial.print(event.acceleration.x);
  Serial.print("  ");
  Serial.print("Y: ");
  Serial.print(event.acceleration.y);
  Serial.print("  ");
  Serial.print("Z: ");
  Serial.print(event.acceleration.z);
  Serial.print("  ");
  Serial.println("m/s^2");

  /* Delay before the next sample */
  delay(500);
}
```

Magnetometer and Compass Code

```
#include <Adafruit_LIS2MDL.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

/* Assign a unique ID to this sensor at the same time */
```

```
Adafruit_LIS2MDL lis2mdl = Adafruit_LIS2MDL(12345);
#define LIS2MDL_CLK 13
#define LIS2MDL_MISO 12
#define LIS2MDL_MOSI 11
#define LIS2MDL_CS 10

void setup(void) {
  Serial.begin(115200);
  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens

  Serial.println("LIS2MDL Magnetometer Test");
  Serial.println("");

  /* Enable auto-gain */
  lis2mdl.enableAutoRange(true);

  /* Initialise the sensor */
  if (!lis2mdl.begin()) {  // I2C mode
  //if (! lis2mdl.begin_SPI(LIS2MDL_CS)) {  // hardware SPI mode
  //if (! lis2mdl.begin_SPI(LIS2MDL_CS, LIS2MDL_CLK, LIS2MDL_MISO, LIS2MDL_MOSI)) { // soft SPI
    /* There was a problem detecting the LIS2MDL ... check your connections */
    Serial.println("Ooops, no LIS2MDL detected ... Check your wiring!");
    while (1) delay(10);
  }
  /* Display some basic information on this sensor */
  lis2mdl.printSensorDetails();
}
void loop(void) {
  /* Get a new sensor event */
```

```
  sensors_event_t event;
  lis2mdl.getEvent(&event);
  /* Display the results (magnetic vector values are in micro-Tesla (uT)) */
  Serial.print("X: ");
  Serial.print(event.magnetic.x);
  Serial.print("  ");
  Serial.print("Y: ");
  Serial.print(event.magnetic.y);
  Serial.print("  ");
  Serial.print("Z: ");
  Serial.print(event.magnetic.z);
  Serial.print("  ");
  Serial.println("uT");
  /* Note: You can also get the raw (non unified values) for */
  /* the last data sample as follows. The .getEvent call populates */
  /* the raw values used below. */
  // Serial.print("X Raw: "); Serial.print(lis2mdl.raw.x); Serial.print("  ");
  // Serial.print("Y Raw: "); Serial.print(lis2mdl.raw.y); Serial.print("  ");
  // Serial.print("Z Raw: "); Serial.print(lis2mdl.raw.z); Serial.println("");

  /* Delay before the next sample */
  delay(100);
}

#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_LIS2MDL.h>
Adafruit_LIS2MDL mag = Adafruit_LIS2MDL(12345);
void setup(void)
{
```

```
  Serial.begin(115200);
  Serial.println("Magnetometer Test"); Serial.println("");
  /* Initialise the sensor */
  if(!mag.begin())
  {
    /* There was a problem detecting the LIS2MDL ... check your connections */
    Serial.println("Ooops, no LIS2MDL detected ... Check your wiring!");
    while(1);
  }
}
void loop(void)
{
  /* Get a new sensor event */
  sensors_event_t event;
  mag.getEvent(&event);
  float Pi = 3.14159;
  // Calculate the angle of the vector y,x
  float heading = (atan2(event.magnetic.y,event.magnetic.x) * 180) / Pi;

  // Normalize to 0-360
  if (heading < 0)
  {
    heading = 360 + heading;
  }
  Serial.print("Compass Heading: ");
  Serial.println(heading);
  delay(500);
}
```