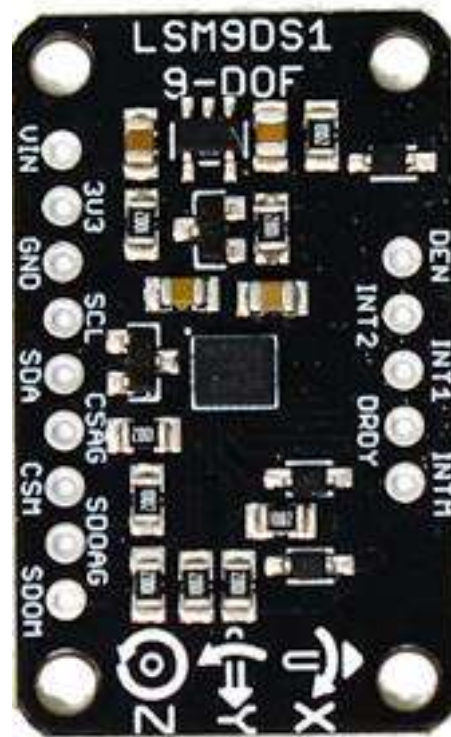




## SmartElex 9-DOF Accel/Mag/Gyro+Temp Breakout Board - LSM9DS1



Add motion, direction and orientation sensing to your Arduino project with this all-in-one 9-DOF sensor. Inside the chip are three sensors, one is a classic 3-axis accelerometer, which can tell you which direction is down towards the Earth (by measuring gravity) or how fast the board is accelerating in 3D space. The other is a 3-axis magnetometer that can sense where the strongest magnetic force is coming from, generally used to detect magnetic north. The third is a 3-axis gyroscope that can measure spin and twist.

The breakout board version of this sensor has both I2C and SPI interfaces. Attaching it to the Arduino is simple, power Vin and GND with 3-5VDC, and wire up I2C data on SCL and SDA. More advanced users can use SPI, the library has support for both.

## Power Pins

The sensor on the breakout requires 3V power. Since many customers have 5V microcontrollers like Arduino, we tossed a 3.3V regulator on the board. Its ultra-low dropout so you can power it from 3.3V-5V just fine.

- **Vin** - this is the power pin. Since the chip uses 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 micro like Arduino, use 5V
- **3V3** - 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 Pins

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

## SPI Pins

If you're interested in using SPI to interface with the LSM9DS1, you can!

- **SCL** - this is also the SPI clock pin, it's level shifted so you can use 3-5V logic input
- **SDA** - this is also the SPI MOSI pin, it's level shifted so you can use 3-5V logic input
- **CSAG** - this is the Accelerometer+Gyro subchip Chip Select, it's level shifted so you can use 3-5V logic input
- **CSM** - this is the Magnetometer subchip Select, it's level shifted so you can use 3-5V logic input
- **SDOAG** - this is the Accelerometer+Gyro subchip MISO pin - it's 3V logic out, but can be read properly by 5V logic chips.
- **SDOM/DOM** - this is the Magnetometer subchip MISO pin - it's 3V logic out, but can be read properly by 5V logic chips.

## Interrupt & Misc Pins

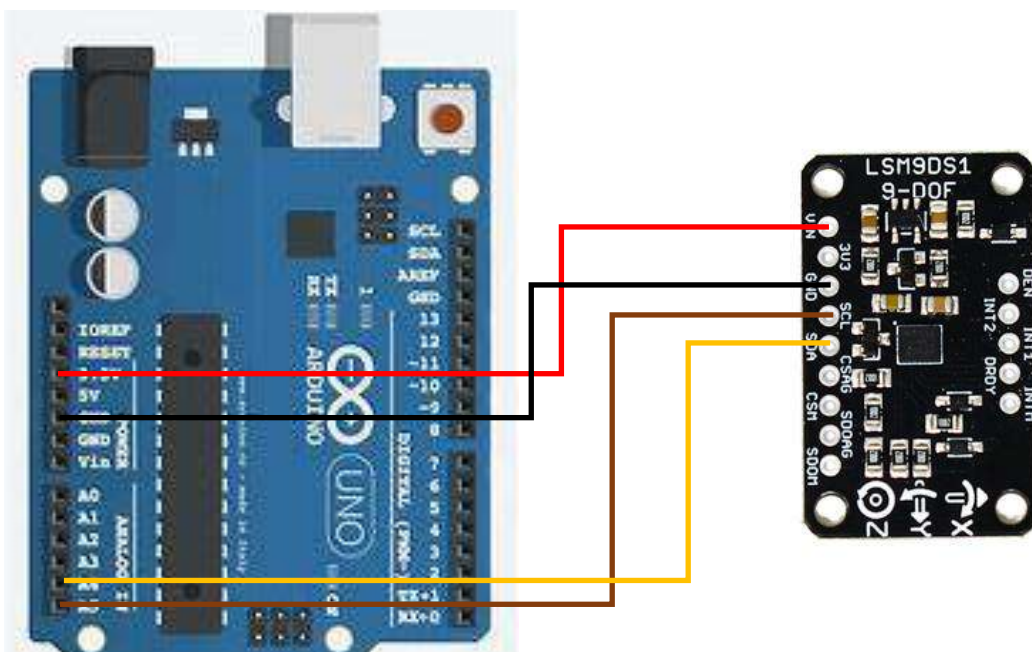
Since there's so many sensors in the LSM9DS1, there's quite a number of interrupt outputs.

- **DEN** - this is a pin that supposedly could be used to dynamically enable/disable the Gyro. There's actually no documentation on it but we break it out for you anyways.
- **INT1 & INT2** - These are interrupts from the accelerometer/gyro subchip. We don't have specific library support for these so check the datasheet for what you can make these indicate. They are 3V-logic outputs
- **DRDY** - this is the accelerometer/gyro subchip data ready output. We don't have specific library support for these so check the datasheet for how you can set the registers to enable this pin. It is a 3V-logic output.
- **INTM** - This is the interrupt from the magnetometer subchip. We don't have specific library support for it so check the datasheet for what you can make it indicate. It is a 3V-logic output.

## Wiring

You can easily wire this breakout to any microcontroller, we'll be using a Arduino UNO. For another kind of microcontroller, just make sure it has I2C or SPI, then port the code.

Let's start with just I2C interfacing since it requires the fewest number of wires. Here is how you would wire up the board.



Arduino	LSM9DS1
SCL(A5)	SCL
SDA(A4)	SDA
5v OR 3.3v	VIN
GND	GND

- Connect **Vin** to the power supply, 3-5V is fine. Use the same voltage that the microcontroller logic is based off of. For most Arduinos, that is 5V
- Connect **GND** to common power/data ground
- Connect the **SCL** pin to the I2C clock **SCL** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A5**, on a Mega it is also known as **digital 21** and on a Leonardo/Micro, **digital 3**
- Connect the **SDA** pin to the I2C data **SDA** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A4**, on a Mega it is also known as **digital 20** and on a Leonardo/Micro, **digital 2**

## Download Arduino libraries

To begin reading sensor data, you will need to download the **Adafruit\_LSM9DS1** library and the **Adafruit\_Sensor** library from the Arduino library manager. Open up the Arduino library manager. Search for the **Adafruit LSM9DS1** library and install it.

Load Demo Sketch

Now you can open up **File->Examples->Adafruit\_LSM9DS1->lsm9ds1** and upload to your Arduino wired up to the sensor

## Example Code

```
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_LSM9DS1.h>
#include <Adafruit_Sensor.h> // not used in this demo but required!
// i2c
Adafruit_LSM9DS1 lsm = Adafruit_LSM9DS1();
#define LSM9DS1_SCK A5
#define LSM9DS1_MISO 12
#define LSM9DS1_MOSI A4
#define LSM9DS1_XGCS 6
#define LSM9DS1_MCS 5
// You can also use software SPI
```

```

//Adafruit_LSM9DS1 lsm = Adafruit_LSM9DS1(LSM9DS1_SCK, LSM9DS1_MISO,
LSM9DS1_MOSI, LSM9DS1_XGCS, LSM9DS1_MCS);
// Or hardware SPI! In this case, only CS pins are passed in
//Adafruit_LSM9DS1 lsm = Adafruit_LSM9DS1(LSM9DS1_XGCS, LSM9DS1_MCS);

void setupSensor()
{
  // 1.) Set the accelerometer range
  lsm.setupAccel(lsm.LSM9DS1_ACCEL_RANGE_2G);
  //lsm.setupAccel(lsm.LSM9DS1_ACCEL_RANGE_4G);
  //lsm.setupAccel(lsm.LSM9DS1_ACCEL_RANGE_8G);
  //lsm.setupAccel(lsm.LSM9DS1_ACCEL_RANGE_16G);

  // 2.) Set the magnetometer sensitivity
  lsm.setupMag(lsm.LSM9DS1_MAG_GAIN_4GAUSS);
  //lsm.setupMag(lsm.LSM9DS1_MAG_GAIN_8GAUSS);
  //lsm.setupMag(lsm.LSM9DS1_MAG_GAIN_12GAUSS);
  //lsm.setupMag(lsm.LSM9DS1_MAG_GAIN_16GAUSS);

  // 3.) Setup the gyroscope
  lsm.setupGyro(lsm.LSM9DS1_GYRO_SCALE_245DPS);
  //lsm.setupGyro(lsm.LSM9DS1_GYRO_SCALE_500DPS);
  //lsm.setupGyro(lsm.LSM9DS1_GYRO_SCALE_2000DPS);
}
void setup()
{
  Serial.begin(115200);

  while (!Serial) {
    delay(1); // will pause Zero, Leonardo, etc until serial console opens
  }
  Serial.println("LSM9DS1 data read demo");
  // Try to initialise and warn if we couldn't detect the chip
  if (!lsm.begin())
  {
    Serial.println("Oops ... unable to initialize the LSM9DS1. Check your wiring!");
    while (1);
  }
  Serial.println("Found LSM9DS1 9DOF");

  // helper to just set the default scaling we want, see above!

```

```

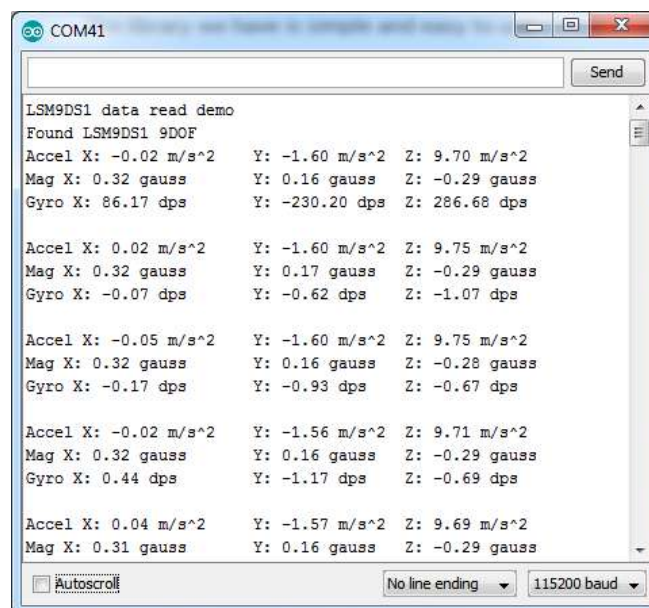
    setupSensor();
}
void loop()
{
    lsm.read(); /* ask it to read in the data */

    /* Get a new sensor event */
    sensors_event_t a, m, g, temp;

    lsm.getEvent(&a, &m, &g, &temp);
    Serial.print("Accel X: "); Serial.print(a.acceleration.x); Serial.print(" m/s^2");
    Serial.print("\tY: "); Serial.print(a.acceleration.y); Serial.print(" m/s^2 ");
    Serial.print("\tZ: "); Serial.print(a.acceleration.z); Serial.println(" m/s^2 ");
    Serial.print("Mag X: "); Serial.print(m.magnetic.x); Serial.print(" uT");
    Serial.print("\tY: "); Serial.print(m.magnetic.y); Serial.print(" uT");
    Serial.print("\tZ: "); Serial.print(m.magnetic.z); Serial.println(" uT");
    Serial.print("Gyro X: "); Serial.print(g.gyro.x); Serial.print(" rad/s");
    Serial.print("\tY: "); Serial.print(g.gyro.y); Serial.print(" rad/s");
    Serial.print("\tZ: "); Serial.print(g.gyro.z); Serial.println(" rad/s");
    Serial.println();
    delay(200);
}

```

Then open up the Serial console at 115200 baud to read the sensor output! You'll get 9 distinct data points, accelerometer x/y/z in meters/s<sup>2</sup>, magnetometer x/y/z in gauss and gyroscope in x/y/z degrees/second



## Library Reference

The library we have is simple and easy to use

You can create the **Adafruit\_LSM9DS1** object with:

```
Adafruit_LSM9DS1 lsm = Adafruit_LSM9DS1(); // i2c sensor
```

I2C does not have pins, as they are fixed in hardware.

If you're using "hardware" SPI, you will have to wire up the pins as follows:

- **SCL -> SPI CLK**
- **SDA -> SPI MOSI**
- **SDO\_AG & SDO\_M -> SPI MISO** (both together)

## Begin!

To initialize the sensor, call **lsm.begin()** which will check the sensor can be found. It returns true/false depending on these checks. We suggest you wrap begin() in a statement that will check if the sensor was located:

```
if(!lsm.begin())  
{  
  /* There was a problem detecting the LSM9DS1 ... check your connections */  
  Serial.print(F("Ooops, no LSM9DS1 detected ... Check your wiring!"));  
  while(1);  
}
```

## Set Ranges

These chips have tons of registers, we basically provide interface code for the most useful stuff, such as setting the range. Each subsensor has it's own range. Higher ranges have less precision but can measure larger movements!

Set up the ranges with the **setup** functions:

```
/ 1.) Set the accelerometer range  
lsm.setupAccel(lsm.LSM9DS1_ACCELRange_2G);  
//lsm.setupAccel(lsm.LSM9DS1_ACCELRange_4G);  
//lsm.setupAccel(lsm.LSM9DS1_ACCELRange_8G);  
//lsm.setupAccel(lsm.LSM9DS1_ACCELRange_16G);  
  
// 2.) Set the magnetometer sensitivity  
lsm.setupMag(lsm.LSM9DS1_MAGGAIN_4GAUSS);  
//lsm.setupMag(lsm.LSM9DS1_MAGGAIN_8GAUSS);  
//lsm.setupMag(lsm.LSM9DS1_MAGGAIN_12GAUSS);  
//lsm.setupMag(lsm.LSM9DS1_MAGGAIN_16GAUSS);
```

```
// 3.) Setup the gyroscope
lsm.setupGyro(lsm.LSM9DS1_GYROSCALE_245DPS);
//lsm.setupGyro(lsm.LSM9DS1_GYROSCALE_500DPS);
//lsm.setupGyro(lsm.LSM9DS1_GYROSCALE_2000DPS);
```

Choose whichever range you like, after you begin() the sensor!

### Read data

Read data using the Adafruit\_Sensor API by first creating four events, one for each sub-sensor:

```
sensors_event_t accel, mag, gyro, temp;
```

Then pass these into the **getEvent** function

```
lsm.getEvent(&accel, &mag, &gyro, &temp);
```

The data is snapshotted at once, so you can read and manage the data later.

For the Accelerometer event you can read **accel.acceleration.x**, **accel.acceleration.y** or **accel.acceleration.z** which are in meters/second\*second.

For the Magnetometer event you can read **mag.magnetic.x**, **mag.magnetic.y** or **mag.magnetic.z** which are in gauss.

For the Gyro event you can read **gyro.gyro.x**, **gyro.gyro.y** or **gyro.gyro.z**, which are in degrees-per-second (dps)

The temperature event data is in **temp.temperature**, but we don't guarantee that the temperature data is in degrees C