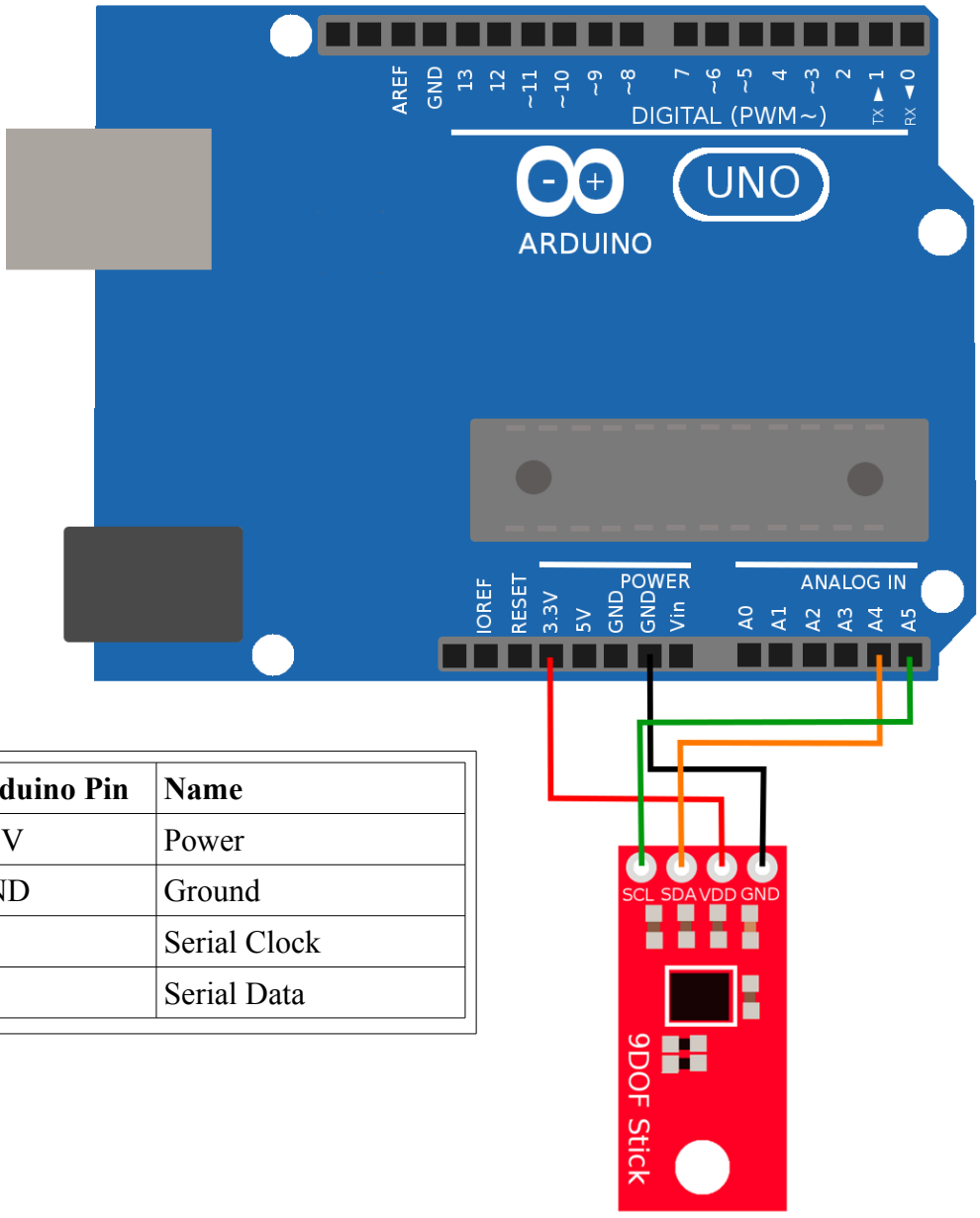
9 Degrees of Freedom (9DoF)

The Nine Degrees of Freedom (9DoF) board measures 3 different physical properties (acceleration, gyroscope, and magnetic field) and for each property breaks the measurement down into 3 perpendicular axis (x, y, z) giving a total of 9 separate measurements. This can be used to determine many different properties of the prototype and how it is oriented or moving.

Wiring

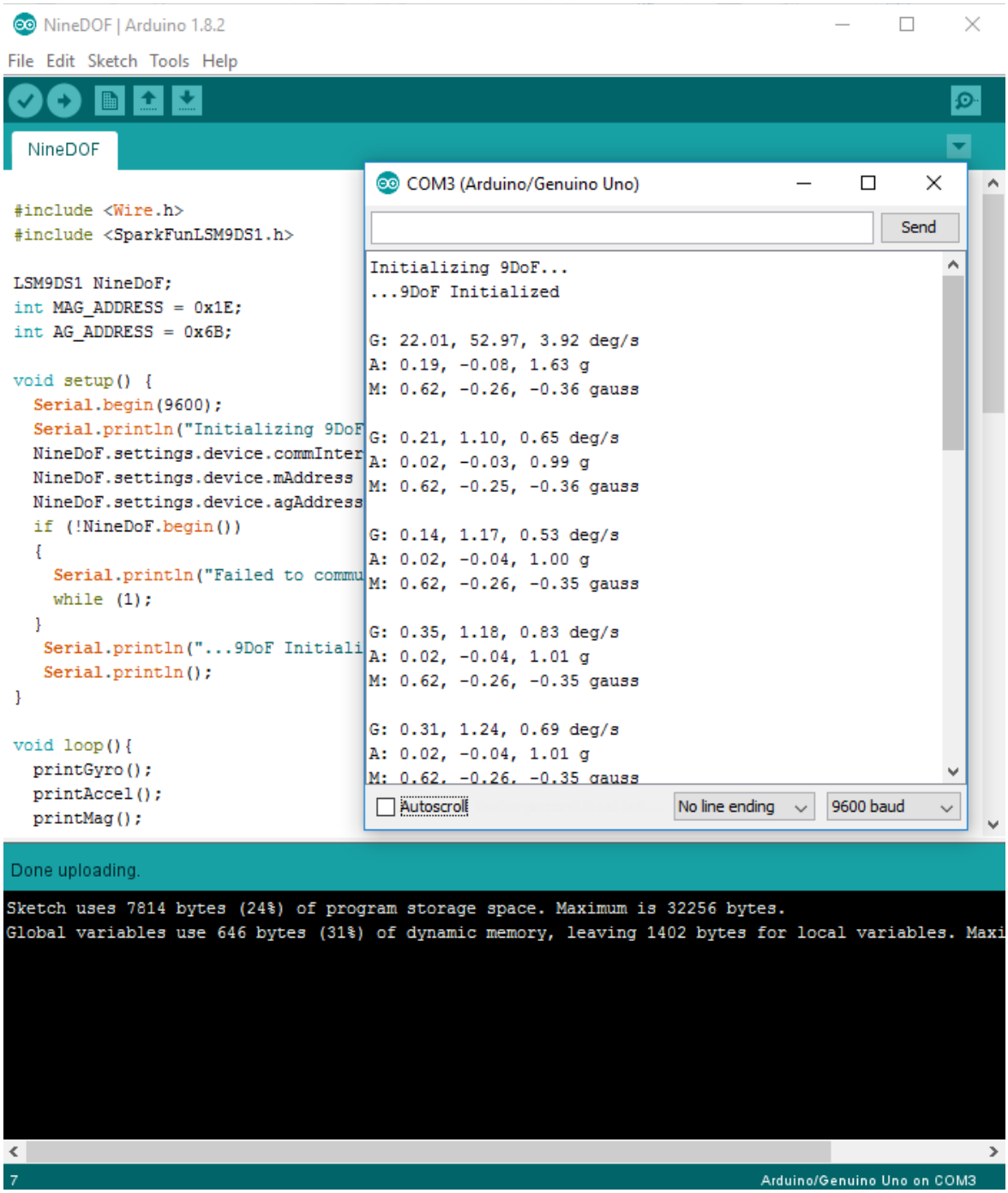
Start by wiring the device over the I2C bus as shown in the diagram below:



	Device Pin	Arduino Pin	Name
	VDD	3.3V	Power
	GND	GND	Ground
	SCL	A5	Serial Clock
	SDA	A4	Serial Data

Testing

Now open the included sketch called 'NineDOF' from the src folder. Upload this code to the Arduino and you should see that readings from the 9DoF board are printed to the screen every second. You can experiment with moving the 9DoF board around (be careful not to detach it from the USB connection) to see how the readings for the gyroscope (x,y,z), acceleration (x,y,z), and magnetic field (x,y,z) change as the device moves.



Code Walkthrough

```
#include <Wire.h>
#include <SparkFunLSM9DS1.h>
```

Here we are including libraries into the sketch. The first library we include is the low level I2C code library called Wire. The next library we include is the manufacturers library from the 9DOF, this tells the Arduino how to get communicate with the device.

```
LSM9DS1 NineDoF;
int MAG_ADDRESS = 0x1E;
int AG_ADDRESS = 0x6B;
```

Here are are creating a variable that holds the 9DoF code. You might notice that we have spelled out the word nine, this is because a variable name cannot begin with a number. Next we specify the address for the two devices on this board, the Accelerometer/Gyro and the Magnetometer. More on I2C addresses can be found in the I2C lecture.

```
void setup() {
  Serial.begin(9600);
  Serial.println("Initializing 9DoF...");
  NineDoF.settings.device.commInterface = IMU_MODE_I2C;
  NineDoF.settings.device.mAddress = MAG_ADDRESS;
  NineDoF.settings.device.agAddress = AG_ADDRESS;
  if (!NineDoF.begin())
  {
    Serial.println("9DoF Failed or Not Present");
    while (1);
  }
   Serial.println("...9DoF Initialized");
   Serial.println();
}
```

In the setup function we start our serial connection as we have previously. We set some options on the 9DoF, telling it we will be in hardware I2C mode, and passing in the address of the integrated components. Next, we start communicating the with 9DoF, if this communication fails for some reason, we output a message and stop execution.

```
void loop(){
  printGyro();
  printAccel();
  printMag();
  Serial.println();
  delay(500);
}
```

In the loop function we call some function we've defined to print out the values of the 3 properties we can measure. We print a blank line to help break up the data in the serial monitor. We wait 500 millisecond and then we do it again.

```
void printGyro(){
  NineDoF.readGyro();
  double gyroX = NineDoF.calcGyro(NineDoF.gx);
  double gyroY = NineDoF.calcGyro(NineDoF.gy);
  double gyroZ = NineDoF.calcGyro(NineDoF.gz);

  Serial.print("G: ");
  Serial.print(gyroX);
  Serial.print(", ");
  Serial.print(gyroY);
  Serial.print(", ");
  Serial.print(gyroZ);
  Serial.println(" deg/s");
}
```

The first thing we do is read the current value from the gyroscope. This return a raw value that needs to be processed to turn into something meaningful, like a number expressed in degrees per second. We use the calcGyro function and pass it the raw values for the x, y and z components to get meaningful numbers. Finally, we print these numbers to the serial line. The very last uses println instead of print so that it starts a new line.

We follow this pattern to get the acceleration and magnetic field values as well:

```
void printAccel(){
  NineDoF.readAccel();
  double accelX = NineDoF.calcAccel(NineDoF.ax);
  double accelY = NineDoF.calcAccel(NineDoF.ay);
  double accelZ = NineDoF.calcAccel(NineDoF.az);

  Serial.print("A: ");
  Serial.print(accelX);
  Serial.print(", ");
  Serial.print(accelY);
  Serial.print(", ");
  Serial.print(accelZ);
  Serial.println(" g");

}
```

```
void printMag(){
  NineDoF.readMag();
  double magX = NineDoF.calcMag(NineDoF.mx);
  double magY = NineDoF.calcMag(NineDoF.my);
  double magZ = NineDoF.calcMag(NineDoF.mz);

  Serial.print("M: ");
  Serial.print(magX);
  Serial.print(", ");
  Serial.print(magY);
  Serial.print(", ");
  Serial.print(magZ);
  Serial.println(" gauss");
}
```