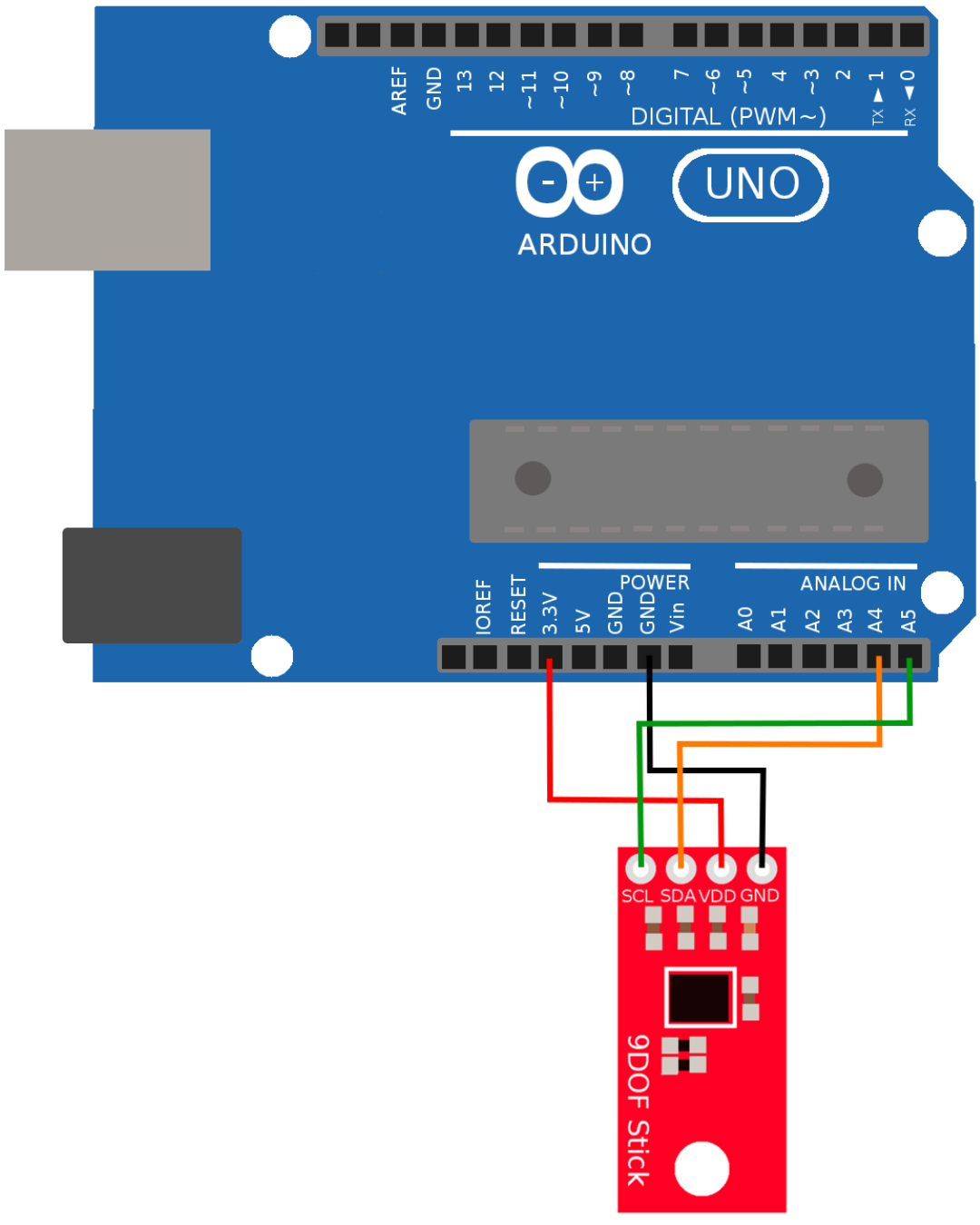
# 9 Degrees of Freedom (9DoF)

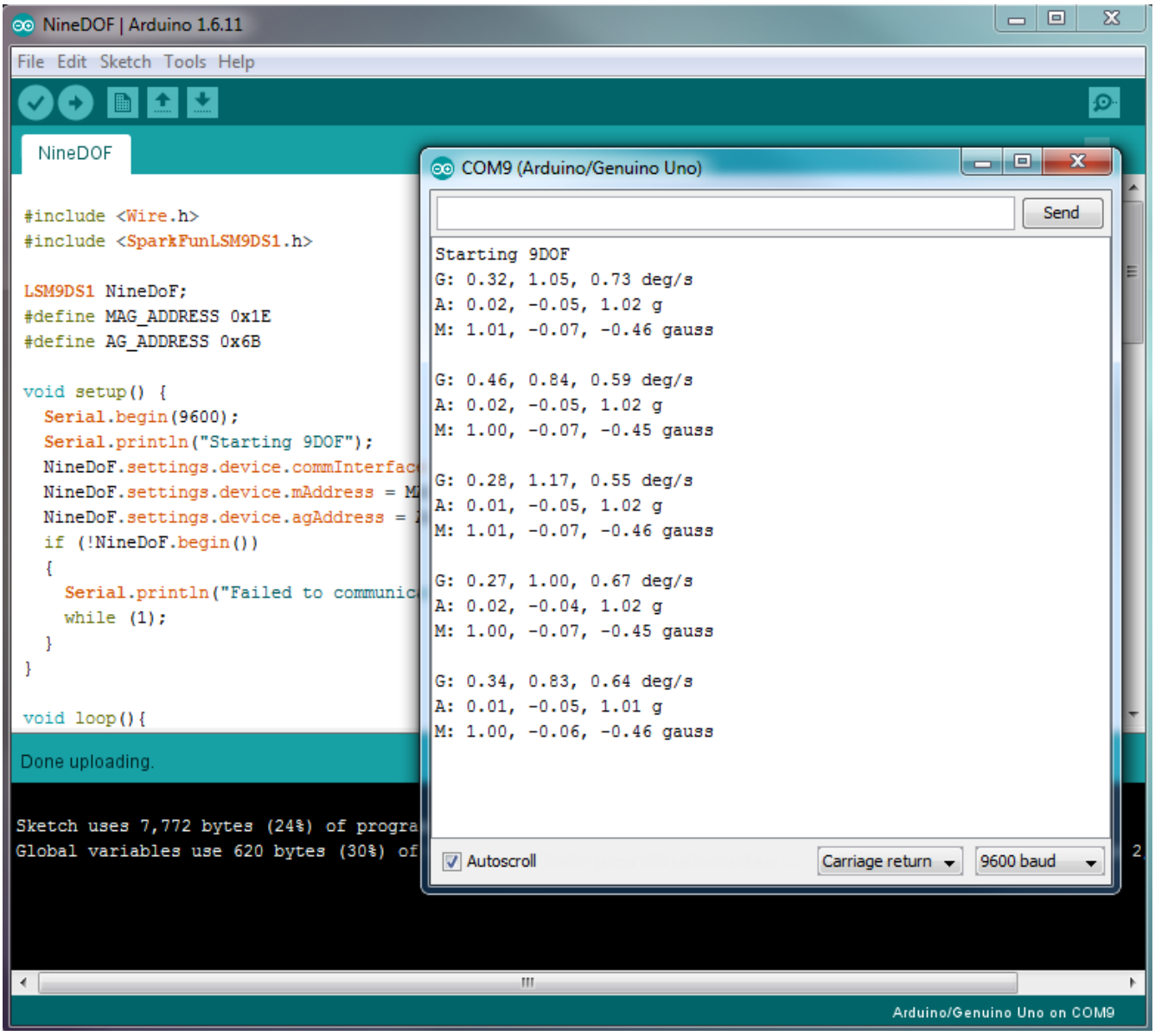
## Wiring

The first step of working with the 9DoF is providing it with power. We connect the VDD line (Red) to a 3.3V power source, and the GND line (Black) to the ground. Next, we connect the two wires for the I2C interace. The Serial Clock or SCL pin (Green) connnects to the Arduino port A5. This is the dedicated I2C SCL pin of the Arduino. The Serial Data or SDA pin (orange) connects to the Arduino port A5. This is the dedicated I2C SDA line of the Arduino.



## Testing

With the wiring complete, open the included sketch called 'NineDOF.' Upload that sketch 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 connnection) 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.  If your device isn't working, you might need to check your wiring.



## 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.h. The next library we include is the manufacturers library from the 9DOF, this tells the Arduino how to get data off the device.

```
LSM9DS1 NineDoF;
#define MAG_ADDRESS 0x1E
#define 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 that live on this board, the Acclerometer/Gyro and the Magnetometer. More on I2C addresses can be found in the lecture in this section.

```
void setup() {
  Serial.begin(9600);
  Serial.println("Starting 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("Failed to communicate with 9DOF.");
    while (1);
  }
}
```

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 addresse 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 meaningul numbers. Finally, we print these numbers to the serial line. The very last uses println intead 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");
}
```