

Hardware Connections between Arduino and IMU

Nori Wilkins

Apr. 5, 2013

Abstract

Sensors are commonly used throughout many world wide applications. Out of many sensors that are used, the inertial measurement unit (IMU) can be used to capture an object's position. Through the use of a microcontroller such as an Arduino, the data the sensor captures can be collected and used for analysis. This document will provide the hardware connections specifically for the Arduino Uno and one IMU sensor.

Keywords

Arduino Uno, IMU-MPU9150, Resistor, I2C bus

Table of Contents

Abstract.....	2
Keywords.....	2
Introduction.....	3
Hardware Components	
Arduino.....	3
IMU.....	4
Resistor.....	6
Hardware Communication	
I2C Bus.....	6
Assembly of Components.....	7
Conclusion.....	12
References.....	13

Introduction

The ECE 480 Design Team 8 was assigned to capture motion of a runner. With many recreational runners throughout the world, many have faced challenges of having bad running form. The product that the team has come up with is a body suit that will capture a runner's motion and compare it to a baseline proper running form. The body suit will consist of a central processing unit on the person that will communicate with the sensors that are attached to the suit. To communicate with the sensor and the processing unit, the devices can be hard wired. Through processes stated in this document, the hard-wired process with a background of each component. The central processing unit consists of the Arduino while the sensor used for capturing motion is the IMU- MPU 9150.

Hardware Components

Arduino Uno

The Arduino Uno, is a series of microcontroller platforms manufactured by Arduino. It is used for many applications. It is an open-source product which means that it can be built by hand or preassembled to customize to one's own needs. Also there is a lot of support throughout the internet community, that already made code is available for anyone that is interested. Other Arduinos consist of the Arduino Leonardo and Due. Different accessories to use wireless applications are available as well.

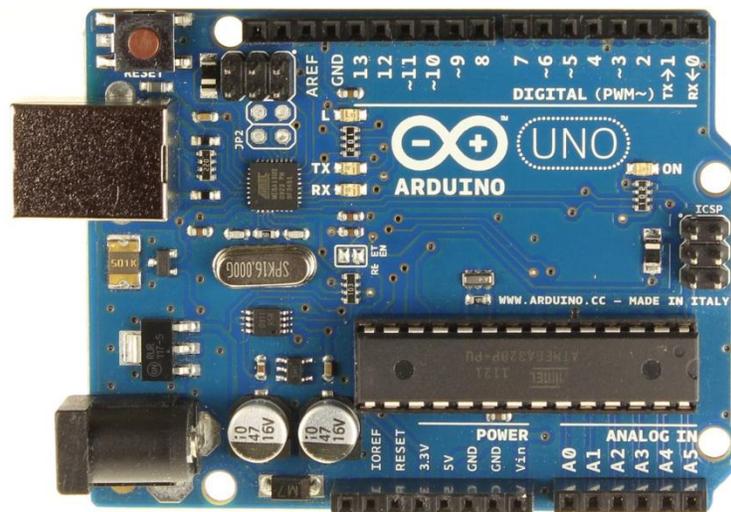


Figure 1: Arduino Uno Rev. 3

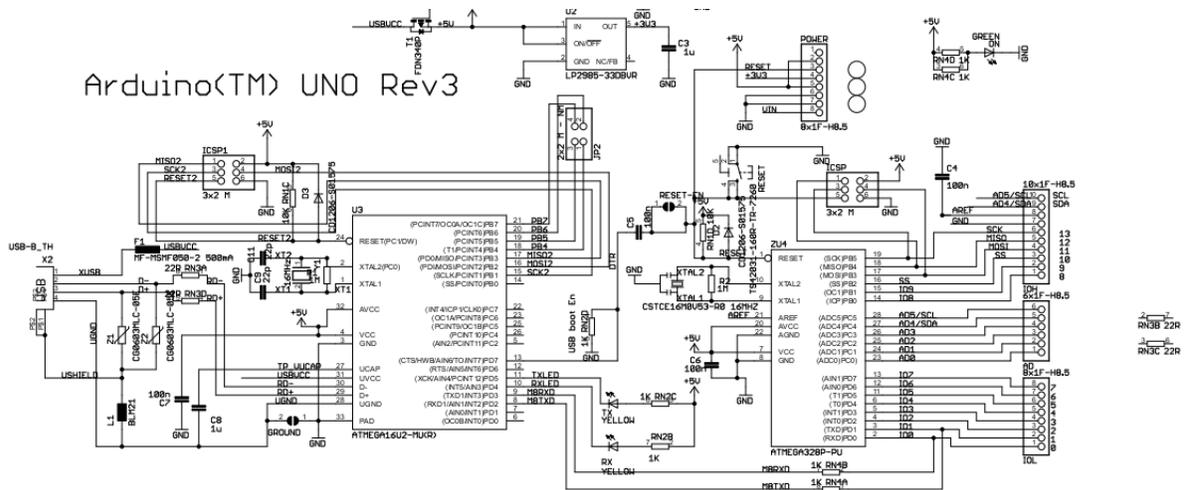


Figure 2: Partial Schematic of Arduino Uno

Both **Figure 1** and **Figure 2** show the components of the Arduino Uno. It operates at 5 Volts (V). Some components include 14 digital input and output pins, as well as 5 digital pins. There is an option of 3.3Volt (V) or 5V pin as well as a ground pin. The Arduino can be powered either by a USB cable that connects to an external personal computer (PC) or a connection to a battery source. (1).

IMU: MPU-9150

The MPU-9150 is a product made by InvenSense. It is a 9 axis motion capture device. The components include an accelerometer, gyroscope, and a magnetometer. The device used in this application note is the MPU9150-breakout board which includes the device as well as easy to access pins to connect to microcontrollers. This break out board is made by Sparkfun Electronics. Sparkfun had made a break out board include the I2C communication available which will be discussed further on. The MPU-9150 is applied in smartphones and tablet manufacturers to enhance gaming as well as other controls.

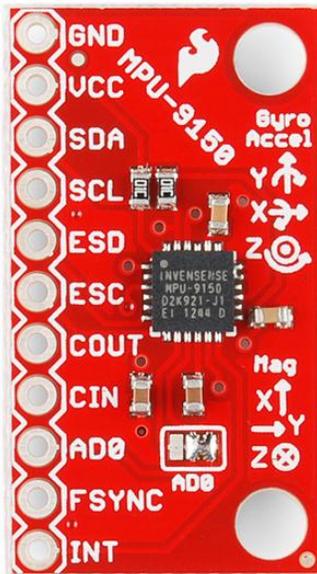


Figure 3: MPU9150 Breakout Board

The accelerometer measures acceleration, the gyroscope measures the angular velocity, and the magnetometer acts as compass. The break out board easily shows the axes directions for each component. The voltage limitations are in the range of 2.375 Volts (V) to 3.465 V (2).

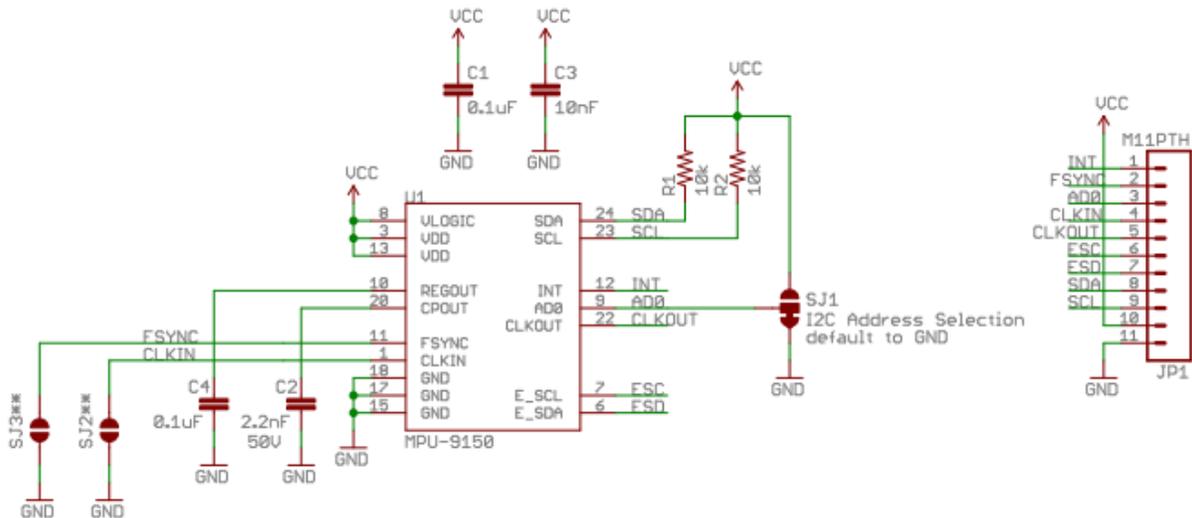


Figure 4: MPU 9150 Schematic (Left) Breakout Board (Right)

Resistor

A resistor is an electrical component to limit the current flow throughout the circuit. There are 4 color bands on the resistor to show the resistance value. The first three shows the value of the resistor and the last band shows the tolerance of the resistor.

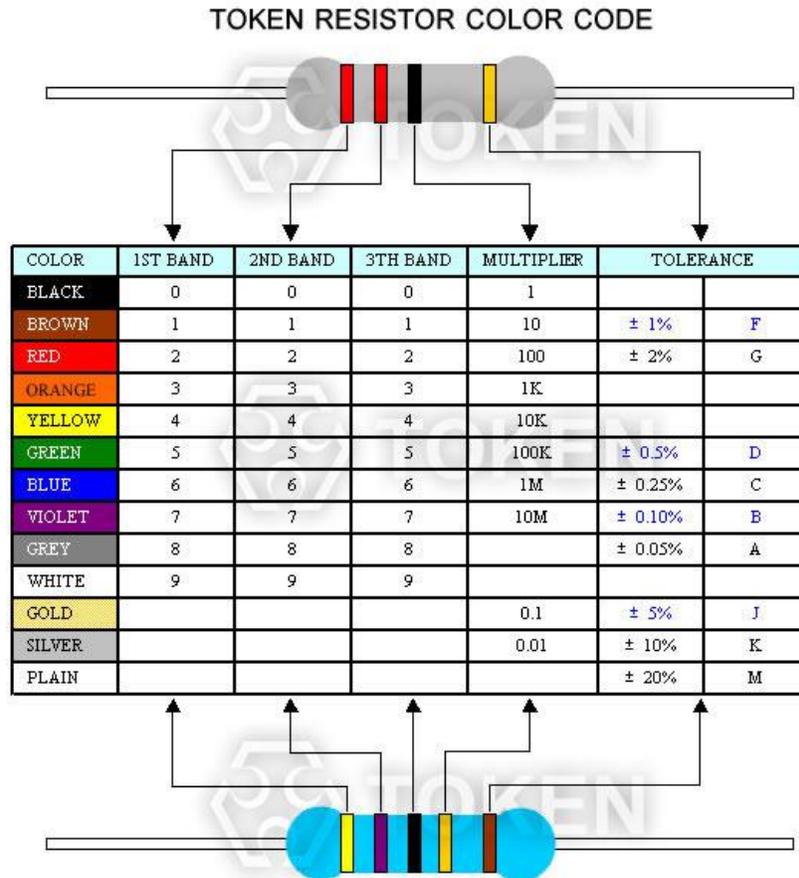


Figure 5: Resistor Color Code (3).

Hardware Communication

I2C Bus

The I2C Bus is the communication that will be used for the IMU and Arduino communication. The I2C bus has 2 physical wires; the SCL (clock line) and the SDA (data line). The two lines are so called “open-drain” drivers; because the outputs of the lines are both low. For these lines to both work and to be output high, pullUp resistors are needed. It is recommended to have at least a 1.8 kilo Ohm resistor, but the resistance value specifically does not matter. The resistance just needs to be high enough for the I2C bus to work.

The I2C bus has 2 components; the master and slave. The master is the device that drives the clock which is the Arduino in this case, while the slave is the IMU. Generally there is only one

master but multiple slaves. The data line can only be changed while the clock is output low. 8-bit packets are transferred every clock pulse. After the data is transferred an “acknowledgement” bit is also transferred from the slave to ensure the data was transferred (4).

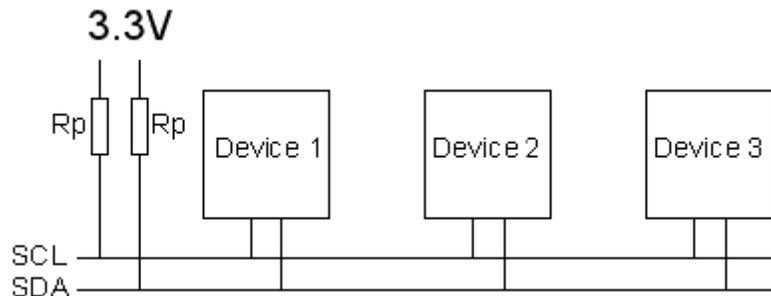


Figure 6: I2C diagram

Assembly of Components

4 connections are needed for the connection between the IMU and Arduino. First it is necessary to connect the pull-up resistors between the wires that will be connected to the SCL/SDA and the power supply. In this example, a 2.2 kilo Ohm resistor is used (**Figure 7**).

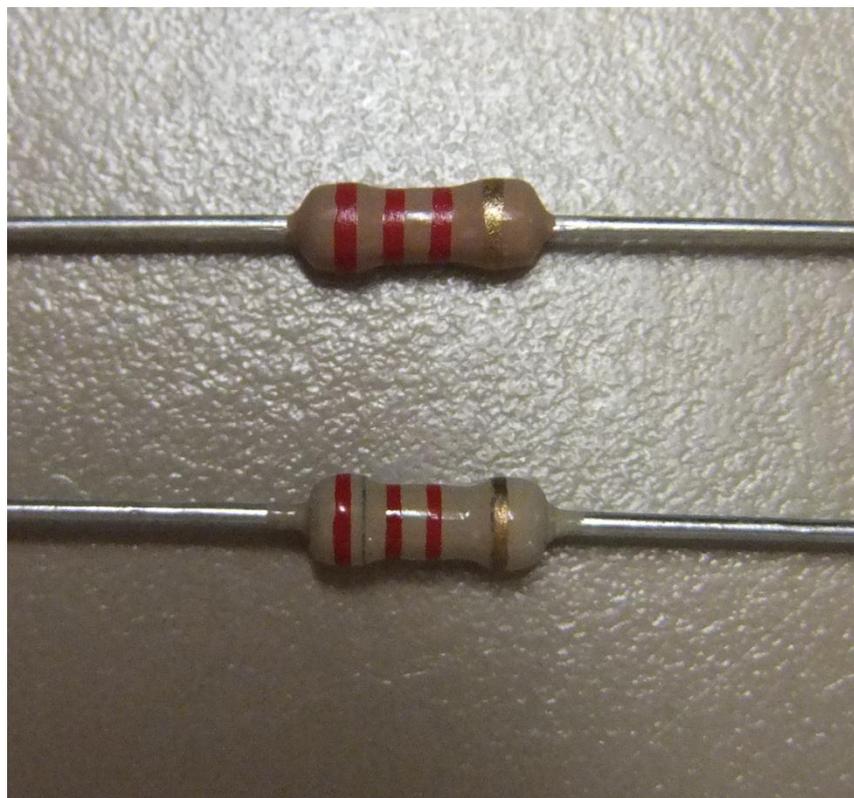


Figure 7: Pull-up resistors

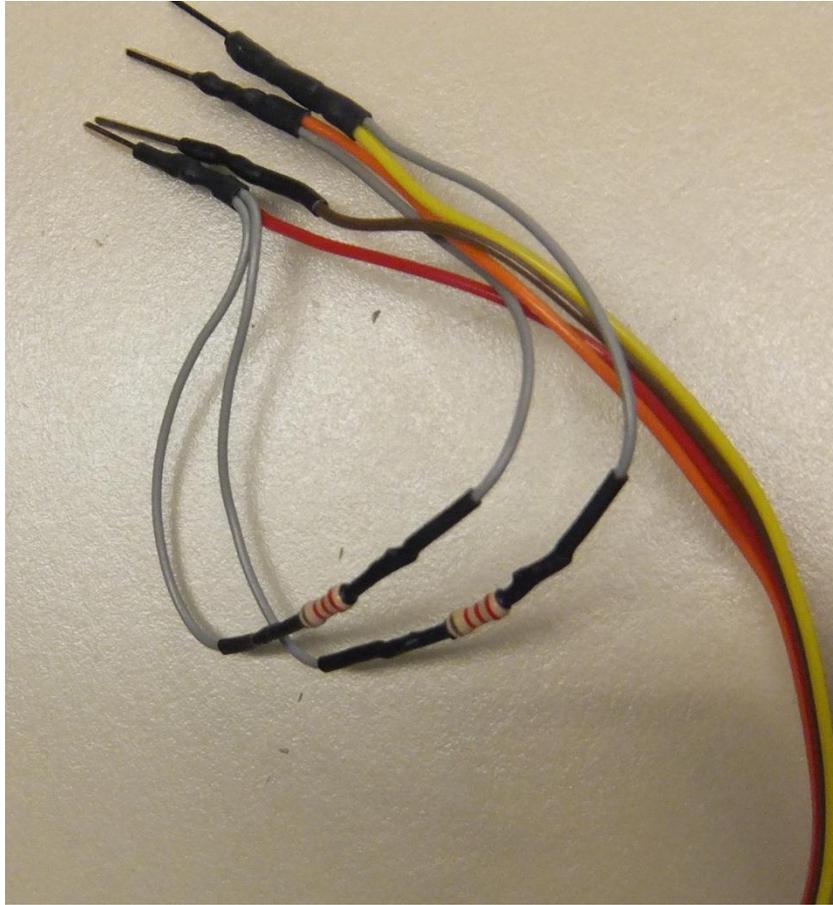


Figure 8: Wires with pull up resistor

Figure 8 shows the wiring needed for the communication. The brown wire is for ground, red is for power, and SDA/SCL lines are yellow and orange.

It is recommended to use heat shrink which is tubing that can be used to cover and protect exposed wires. It is also recommended to use connector pins as seen above. This is recommended so the wires that get connected and disconnected will not be broken or damaged. Also, the pins will be connected to the connector that has been soldered onto the IMU so it can be connected to different devices if needed. Also, if the pins are present, the wires do not need to be directly soldered onto the IMU.

Figure 9 shows the connector pins and **Figure 10** has a connector to the IMU for easy connections.

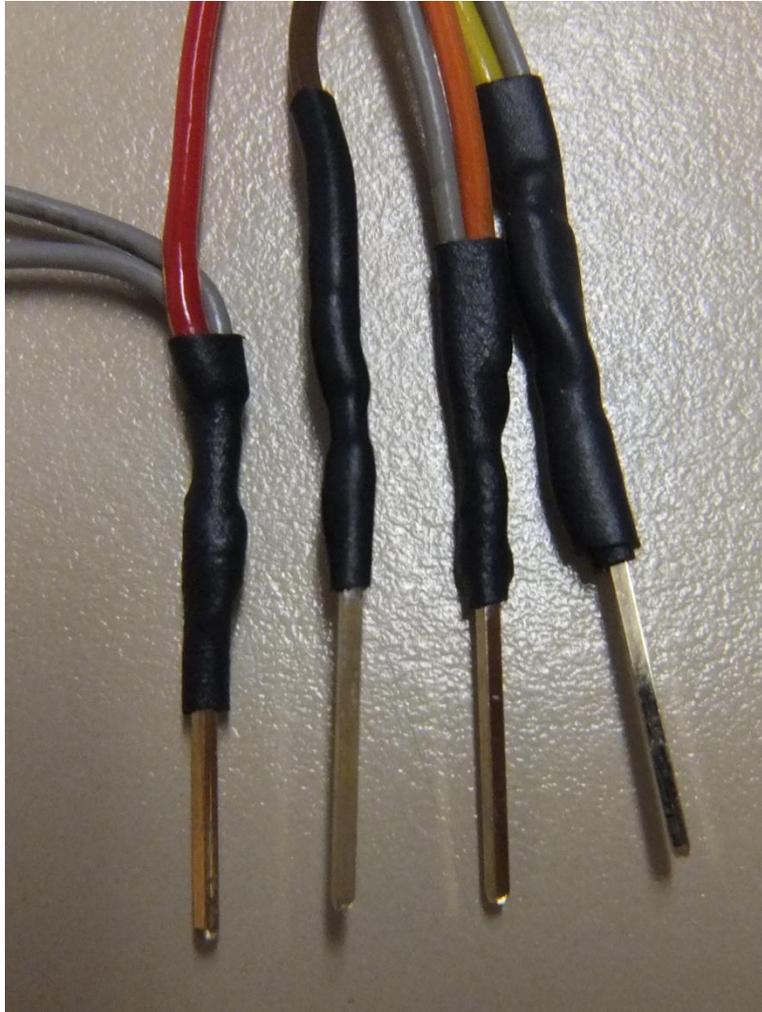


Figure 9: Connector pins with heat shrink tubing

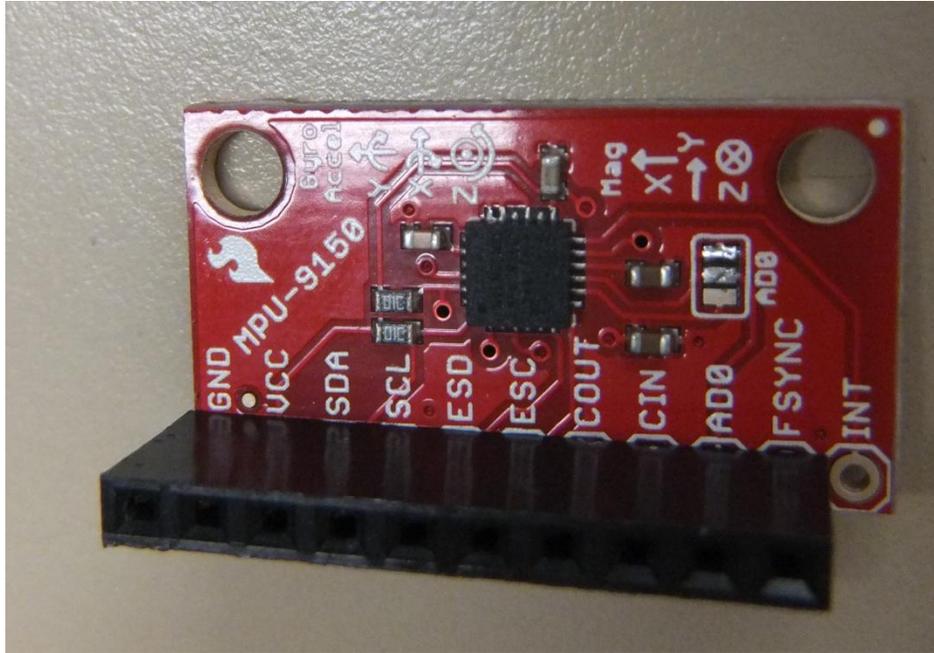


Figure 10: Connector soldered onto the IMU

Now the connections can be made between the IMU and Arduino. **Figure 11** shows the part of the Arduino that is going to be used. Only the top 4 pins will be used in this application note.

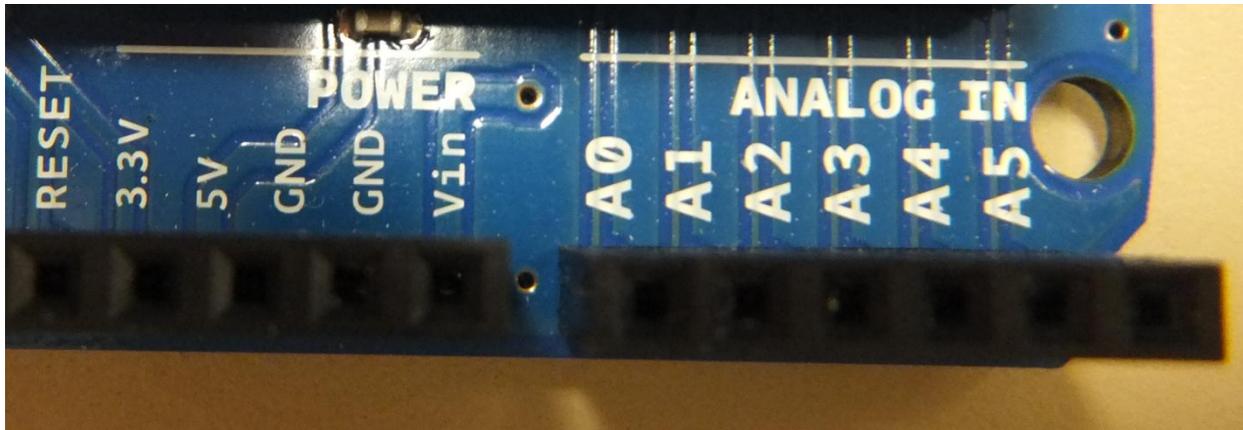


Figure 11: Arduino pins

There are 2 options to supply power from the Arduino to the IMU. Due to the voltage limitations of the IMU, the 3.3 V pin needs to be used. Either ground pins can be used. The A4 pin is the SDA and A5 pin is the SCL line. **Figure 12** shows the schematic of the Arduino and the IMU. Because of the connector pins that were added to the wires, the connections should be easy to make.

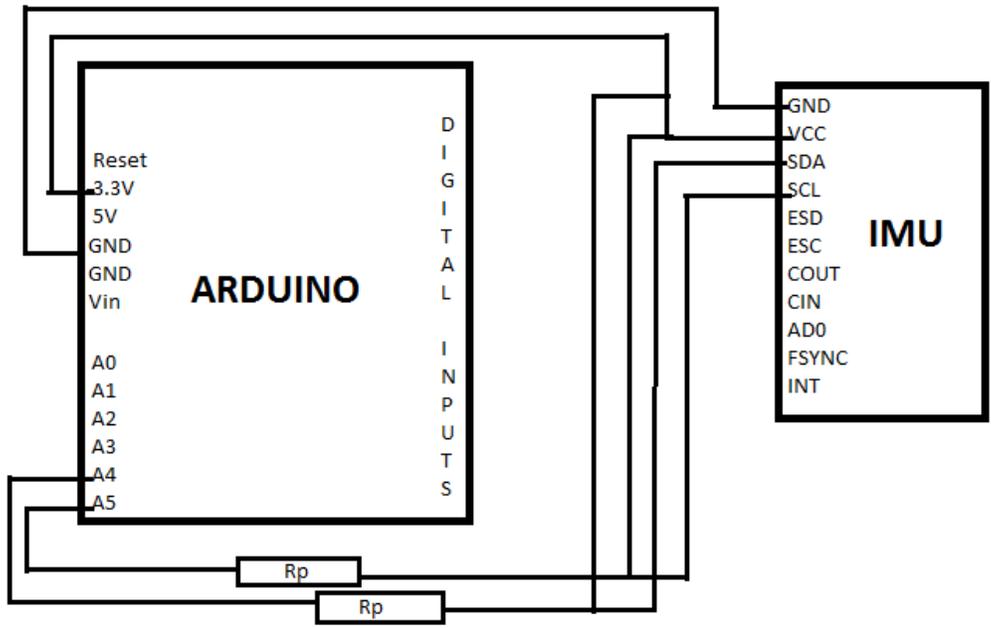


Figure 12: Schematic of the Connections. Rp stands for the pull-up resistors

The finished product should look below (**Figure 13**).

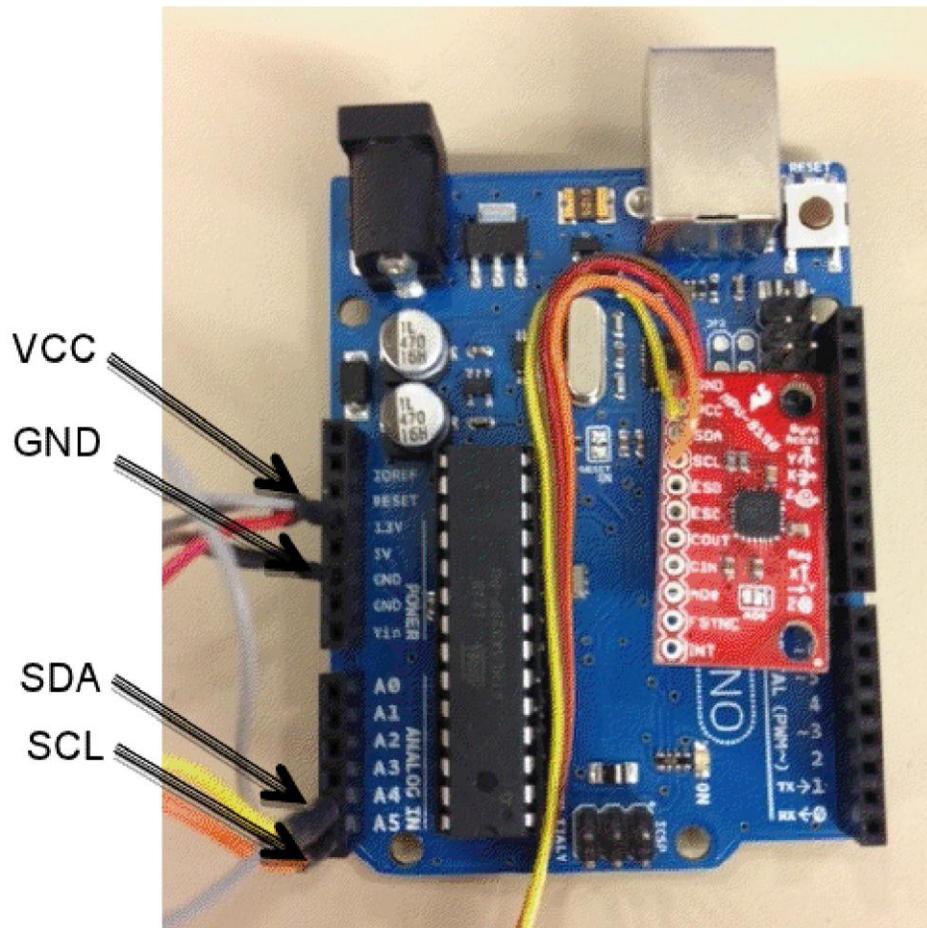


Figure 13: Finished product

Conclusion

All and all the connections between the Arduino and the IMU are fairly simple. The through Arduino programming, the connections can be ensured that it is working. If it is working correctly, the values for each axis should be able to read on a PC. It is recommended to make sure all connections are covered as previously stated with heat shrink and the wires should be color coded to make sure no connections get mixed up.

References

1. "Arduino - ArduinoBoardUno." *Arduino - ArduinoBoardUno*. Arduino, n.d. Web. 05 Apr. 2013. <<http://arduino.cc/en/Main/ArduinoBoardUno>>.
2. "9 Degrees of Freedom- MPU-9150." *9 Degrees of Freedom*. Sparkfun Electronics, n.d. Web. 05 Apr. 2013. <<https://www.sparkfun.com/products/11486>>.
3. "Resistor Color Code Guide." *Electronics Reference*. N.p., n.d. Web. 05 Apr. 2013. <http://www.elexp.com/t_resist.htm>.
4. "Using the I2C Bus." *I2C Tutorial*. Robot Electronics, n.d. Web. 05 Apr. 2013. <http://www.robot-electronics.co.uk/acatalog/I2C_Tutorial.html>.