NAIT

Edmonton, Alberta

**3-axis Self-Stabilizing Camera Mount**

**TPS Report**

Submitted to

Marc Anderson, CMPE 2960 CNT Instructor

Kelly Shepherd, CMPE 2960 English Instructor

Submitted by

Ervin Hernandez, CMPE 2960 Student

Gabriel Natividad, CMPE 2960 Student

April 7, 2016

# Introduction

## Overview of the Prototype

The 3-axis self-stabilizing Camera Mount is a device that uses orientation-determining algorithms to self-correct when introduced to rotational movements.

## Skills Needed

The skills required to operate the prototype are:

* Proficient understanding of C programming language
* Proficient skill of constructing circuitry and wiring
* Proficient understanding of the Physics behind movement of objects in 3D-space
* Basic soldering knowledge

## 1.3 Materials Needed

Overview of the components

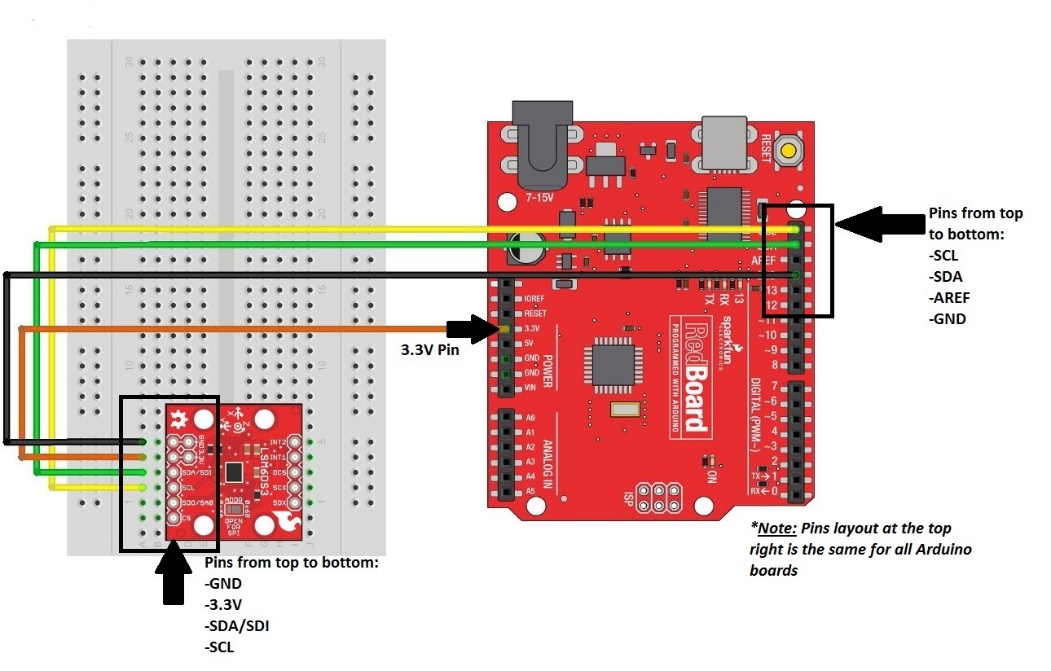
* + - Planks of wood
    - 1 x LSM6DS3 IMU Board (sensor)
    - 3 x Hitec HS-485HB Analog Servo Motor
    - 3 x Aluminum Horns
    - 1 x 5V/10A power supply
    - 1 x Arduino Mega 2560 Microcontroller
    - 3 x 3D printed case
    - 2 x 3D printed brace

## Purpose

The purpose of this project is to better understand and implement the mechanics behind orientation-determining algorithms.

# 2.0 Steps to operate

## 2.1 Setup the prototype

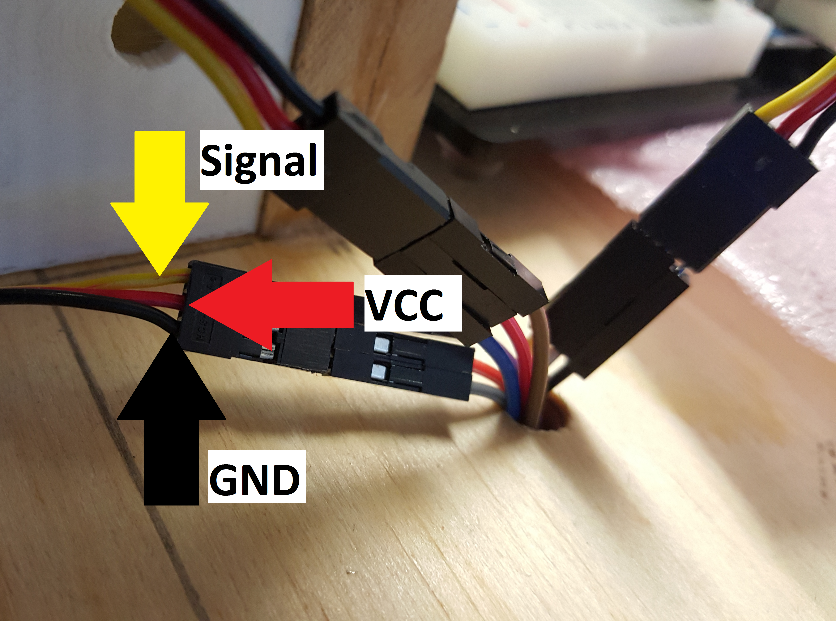
2.1.1 Sensor Setup

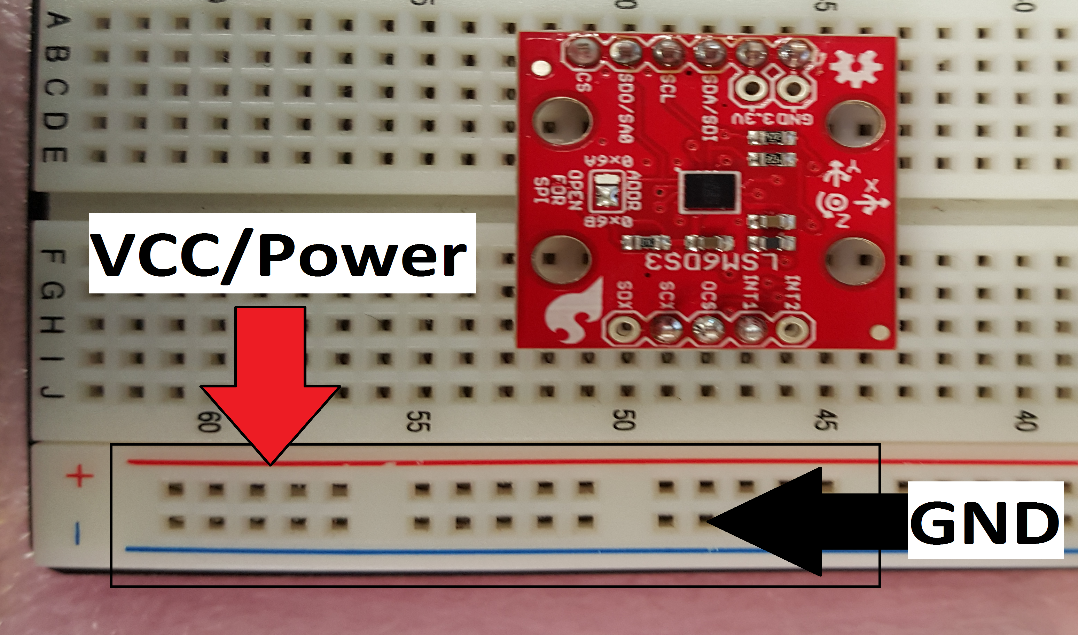
Hook up wires to pins *Ground*, *3.3V*, *SDA/SDI* and *SCL* of the LSM6DS3 sensor. Connect the sensor to the Arduino Board as shown in the picture above.

**Caution:** The LSM6DS3 is a **3.3V device**. Connecting the device to a pin with lower or higher voltage rating can cause the device to malfunction, and in worst cases, cause irreparable damage to the sensor itself.

2.1.2 Servo Wiring

Notice the 3 wires located at the back of each servo motor.



Connect the red wires to the breadboard component of the microcontroller, in the rails labeled “+”, and the black wires in the rails labeled “-“.

The yellow wires are used to communicate commands to the servo motors. Connect the base, middle, and top servos’ yellow wires to the Arduino pins **10, 9, and 8** respectively.

2.1.3 Powering the Microcontroller

Connect the USB cable to any laptop or desktop currently powered and turned on. Plug the other end of the cable to the port located on the back side of the microcontroller. This powers the microcontroller and the sensor.

**Caution:** Do not use the microcontroller voltage pins to power the servo motors. The Arduino will not be able to supply the appropriate amount of current the servos need. Forcing the Arduino to supply power to the servos will permanently damage the board and render it useless.

2.1.4 Powering the Servos

Make sure the power supply is not plugged in. Connect the red wires of the power supply to the same rails the servo red wires were connected to. Connect the green wire of the supply to the same rails the black wires were connected to. When ready, plug in the power supply to turn on the servo motors.

**Caution:** Do not, under any circumstances, touch the green or red power supply wires when the supply is plugged in. The power supply outputs an amperage rating of 10A at 5V, which is enough current to kill an individual.

## 2.2 Testing Procedure

2.2.1 **Scenario #1 - Testing Servo Capability (No Load).**

Summary

The prototype was assembled and was prepared to test the servos without any load. The goal of the test was to observe the voltage and amperage consumption of the servos. There will be two tests: One with the servos stationary on stable ground and the second will be when the servos are programmed to do sweeping movement at angles between 0 and 180 inclusive for a duration of 5 minutes.

Stationary

When the servos were powered and left idle on flat and stable ground, the servos started to vibrate. The vibrations were minimal at the base but were amplified and more obvious at the upper-half of the prototype. The servos didn’t consume any significant current during the 5 minutes of idling on a stable surface. Unfortunately the vibration from the servos started to affect the IMU board, and given that servo movement is based entirely on the board, the servo vibrations increased.

Sweeps

When the servos were programmed to do sweeps for 5 minutes, the amperage consumption started to increase over time. With each sweeping motion, the multimeter readings indicate that the servo current consumption was around 0.5 A to 1.2 A. This posed a problem concerning runtime because the increase amount of current consumption drained the supply at a faster rate.

Results

After the two tests, we concluded that the problems encountered during the stationary test can be solved with some software adjustments. Unfortunately the sweep tests demonstrated a big flaw in the prototype design. With each sweeping motion, the servos were required to lift a significant amount of weight, specifically the center servo which carries one servo and potentially, the camera. There were multiple suggestions to solve the problem. One was to use a better power supply, one that had a higher amperage rating in order to provide the power the servos needed. Another suggestion was to change the overall structure of the prototype, make it compact and keep the center gravity of each servo motor close to each other. More testing is required to make a final decision.

2.2.2 **Scenario #2 - Testing Servo Capabilities (With Load).**

Summary

The goal of this test was to observe the voltage and amperage consumption of the servos when it’s required to lift certain amounts of weight. Similar to the previous scenario, there were two tests. Both tests involved the mount carrying a 50g load and a 100g load. The first test had the servo-controlled mount stationary on a stable surface, and the second had the servos do sweeping motions at angles similar to the previous scenario.

Stationary

When the prototype was stationary with the 50g load on top, there were small vibrations moving across the prototype. The vibrations weren’t as big as the vibrations found in the previous scenario (camera mount subject to no load) so the IMU broad didn’t detect it. The 100g test showed the same result

Sweeps

When the prototype was programmed to sweep for 5 minutes with the 50g load, there were large amounts of power consumption during the test. Similar to the last scenario, as the servos were swinging around, the servos drew more power than expected. The servos consumed around 1 to 1.75 A, with the amperage spiking up during 45-degree angles. At 100g, the load was too heavy for the servos to lift, resulting to complete failure.

Results

After the two tests, the overall outcome was not satisfactory. The stationary test was a great success in that the weight of the load muffled the vibration by a small amount. As soon the prototype was required to move, the prototype failed to meet expectations. Not only was the consumption of current amplified, the 100g load revealed that the servos required more power to lift the load.

2.2.3 **Scenario #3 – Power Supply Test**

Summary

The goal of this test was to observe the movements of the prototype with a new power supply that gave 5 volts at 10 amps of current. Similar to the previous scenario, there will be a test when the servos are motionless and a test when the servos are moving. The only difference is there will be a 50g web camera mounted on top of the prototype. No 100g load tests were done given that a 50g webcam can serve as the camera component of the prototype.

Stationary

When the prototype was stationary on a stable surface, small vibrations reverberated towards the sensor and, similar to the previous scenario, affected the whole prototype. As the vibration continued, the video from the webcam was also affected, resulting to a shaky and jittery video feed.

Rotational Movements

When the prototype was lifted and rotated around the different axes, the prototype was able to self-correct and minimize the vibrations that were observed at the stationary test. Although if the prototype was held stationary at any angle, the vibrations returned and started affecting the whole system.

Results

When kept at constant movement, the vibrations from the prototype were minimal and negligible. Since the vibrations are due to excess outside force being introduced to the sensor by the motors, a possible fix was to distance the sensor itself from the motors. Another fix was to pad the sensor with a cushion at its base to soften the vibrations coming from the motors. Ultimately, it was decided that a demo for each situation needed to be done. Further tests are required to determine the best possible solution.

2.2.4 **Scenario #4 – Sensor Reading Accuracy Test**

Summary

After prototype assembly, test code was written in C programming language to

check the accuracy of the sensor readings, both during long-term and short-term runtime. The prototype utilizes gyroscope readings to self-correct during rotational movements. Since gyroscope readings tend to drift into unreliable values over time, test codes needed to be written to see when the drift happens and how to eliminate, or at the very least delay, the drift of the sensor readings.

Short-Term Runtime

If the prototype only runs for 3 minutes or less, the z-axis rotation, the axis that uses the gyroscope and therefore subject to drift, remains accurate and responsive to any rotational movement. The x and y rotation readings also remain accurate.

Long-Term Runtime

Crossing the 3-minute threshold, the z-axis rotation still remains responsive, but the yaw (term that describes the movement around the z-axis) angle begins to drift and as a result, the angle readings contain offsets (e.g. if the prototype is rotated 36 degrees clockwise, the prototype self-corrects 40 degrees instead of 36). The x and y rotation readings only use the accelerometer portion of the sensor and therefore are not subject to drift. Unfortunately, the gyroscope is needed for the z-axis rotation, thus complete elimination of drift is not possible without buying a magnetometer, a sensor that could be used to find the direction of the Earth’s magnetic field and accurately point north (or the south seeking pole), and therefore give accurate heading values. Another alternative is buying a GPS and incorporating it to the prototype’s system. The GPS can be used to determine the exact longitude and latitude values of the prototype itself and therefore can also be used to correct the gyroscopes heading values.

Results

After multiple tests and a bit of research, we have found that gyroscope readings are naturally inclined to drift. Devices that use gyroscopes usually have another sensor that is solely used to correct the gyroscope readings. Use of the Madgwick filter, an algorithm that combines both accelerometer readings and gyroscope readings (the algorithm favouring heavily the former during long-term runtime and the latter during short-term runtime) extended the threshold to 4 minutes. Additional code was written to delay the drift to 5 minutes but at the cost of slightly losing real-time responsiveness.

# 3.0 Conclusion

Algorithms that concern determining orientation of objects in three-dimensional space and auto-correcting rotational movements are found in many everyday devices, the most common of which are smart phones. The auto-rotating screen feature of smart phones uses similar algorithms to keep the images or video feeds upright for the user to see properly. The prototype is one of many applications of these algorithms

This report is primarily concerned on the determining and testing the capabilities of the algorithm and how well it was implemented. Portions of this report concerned testing the viability of the sensor readings – how accurate they are, and how long they remain accurate. The rest of the report concerned how well the algorithms were implemented. Tests determined the capabilities and limitations of the prototype, more specifically the limitations on weight capacity and power consumption.

Ultimately, the tests yielded useful results that were eventually used to overall better the prototype and eliminate any pre-existing bugs both in the software and hardware components of this project.