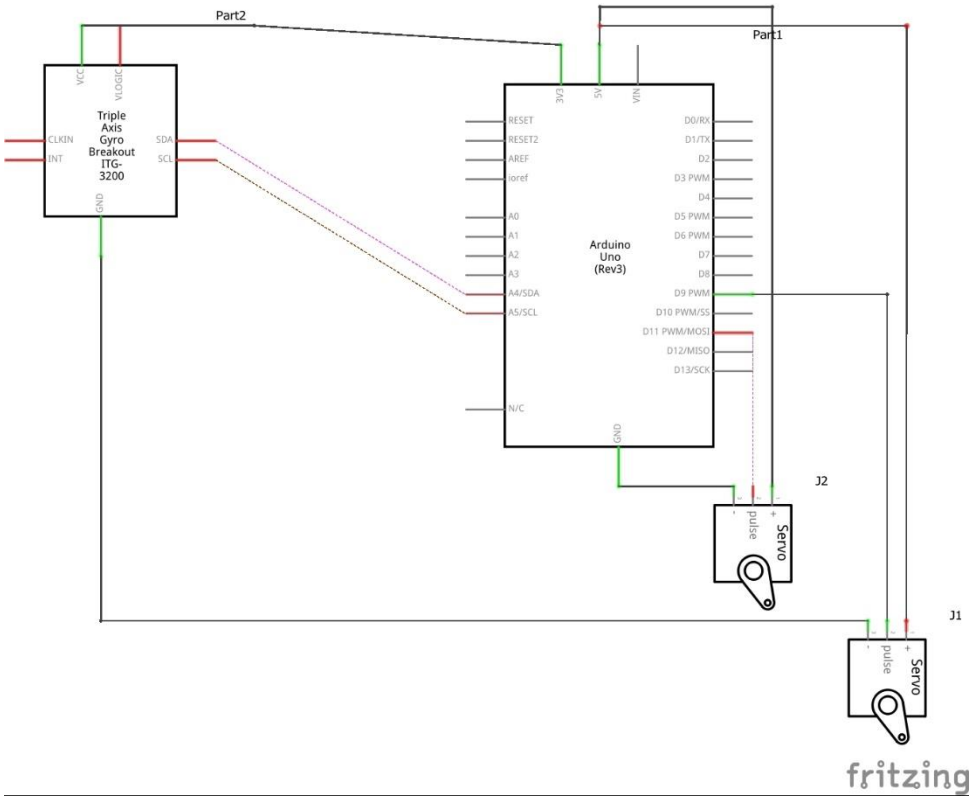
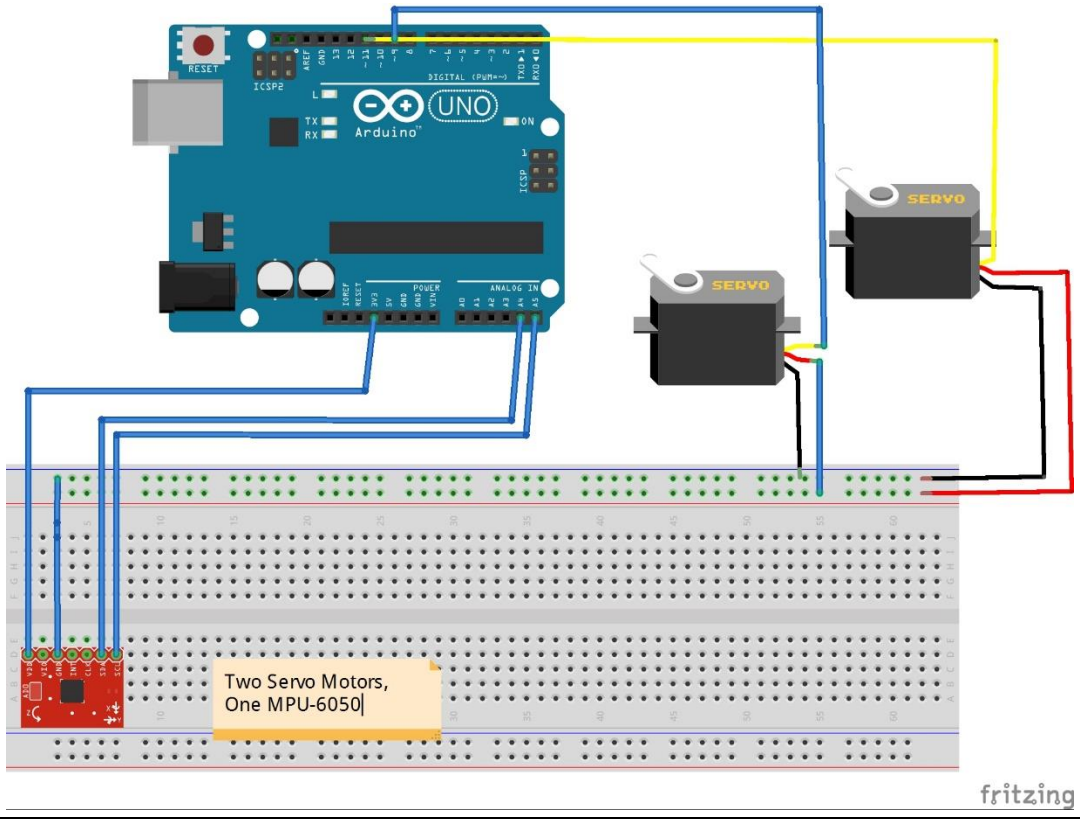
Project Summary 2016.

Tanjot Panesar and Eric Sinnige

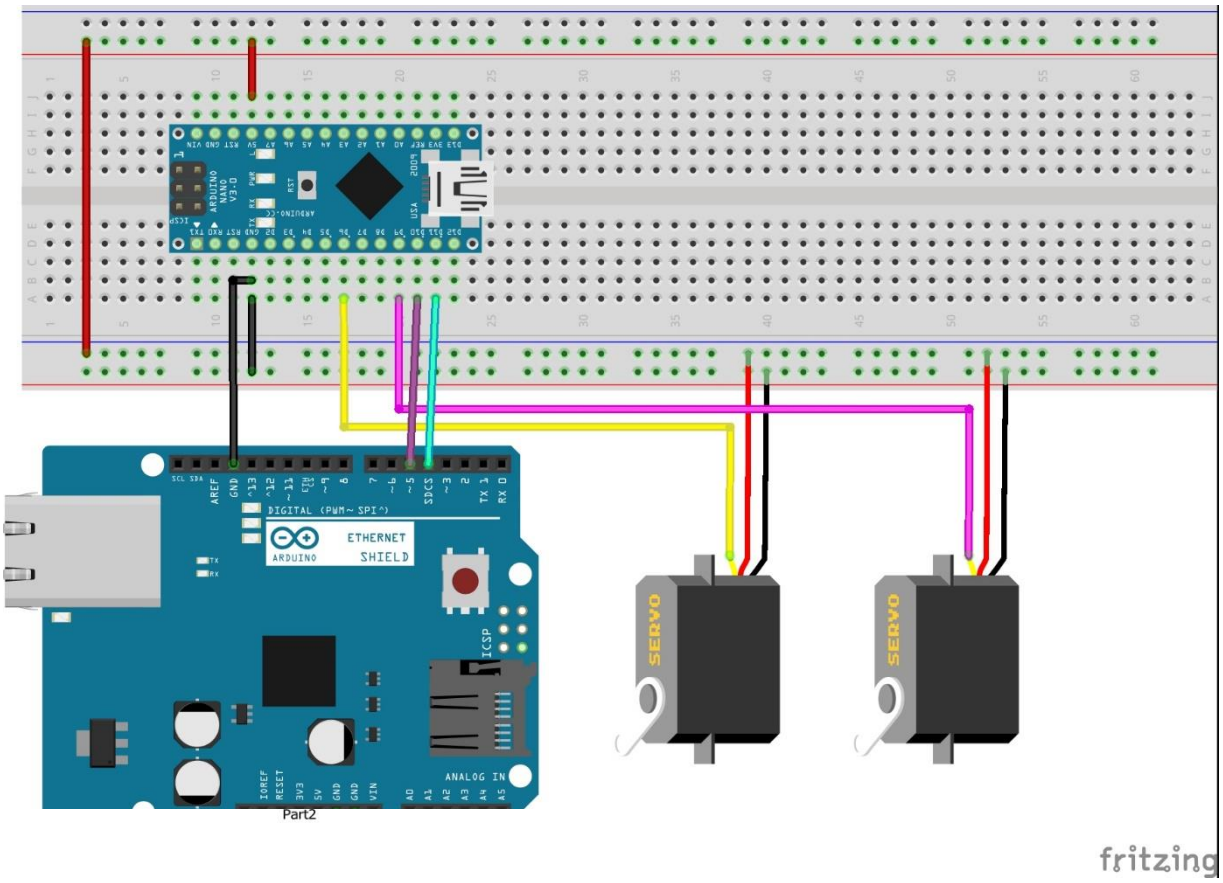
Culminating Wiring Diagram

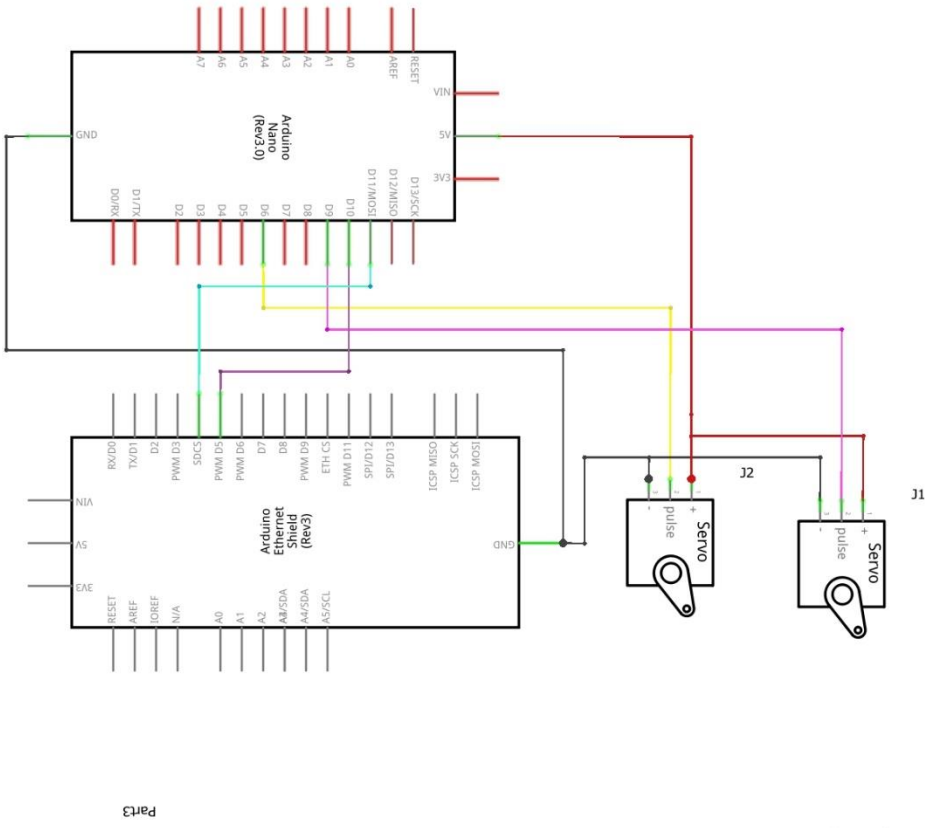
Schematic/Pictorial.





Schematic and Pictorial for Ethernet Sheild Version (StableShot 2.0)





Gimbal Controlled by Accelerometer

```
#include <SPI.h>
#include <Wire.h>
#include <Servo.h>
#define MPU 0x68

Servo ServoX, ServoY;
double AcX,AcY,AcZ;
int Pitch, Roll;

void setup(){
  Serial.begin(9600);
  ServoX.attach(11);
  ServoY.attach(9);
  init_MPU();
}

void loop()
{
  FunctionsMPU();

  Roll = FunctionsPitchRoll(AcX, AcY, AcZ);
  Pitch = FunctionsPitchRoll(AcY, AcX, AcZ);
  int ServoRoll = map(Roll, -90, 90, 0, 179);
  int ServoPitch = map(Pitch, -90, 90, 179, 0);

  ServoX.write(ServoRoll);
  ServoY.write(ServoPitch);


  Serial.print("Pitch: ");
  Serial.print("\t");
  Serial.print("Roll: ");
  Serial.print("\n");

}

void init_MPU(){
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  delay(0);
}
```

- /*Including the SPI library- for synchronous serial data protocol
- Including Wire library- To communicate with i^2c
- Including Servo library to use servo commands
- Defining the I2C address of the MPU-6050*/

- /*Declaring double precision floating point number.
- Declaring integer variables Pitch and Roll (values for x and y axis)*/

/*<u>Setup function</u>

- Setting baud rate to 9600 bps
- Using attach command to output 5v to Pulse Width Modulation pins 11 and 9 on Arduino.
- Calling on MPU function (refer to bottom of this code) */

/*<u>Void loop function</u>

- Acquiring Acceleration in X,Y,Z planes from FunctionsMPU function.
- Calculating the total Roll angle for x axis
- Calculating the total Pitch angle for y axis
- Mapping the new values into servo with a range to create level. (done for both Roll and Pitch)
- Writing the values to the Servo instantaneously updating it.
- Printing values to serial monitor for assurance of values */

/*<u>Void init_MPU</u>

- Initializes wire library and beings the transmission process of serial communication through I2C of the specified address
- Set the Wire.write to zero (wakes up the MPU-6050)
- End transmission when true
- No delay for responsive values.*/

```
double FunctionsPitchRoll(double A, double B, double C){
  double DatoA, DatoB, Value;
  DatoA = A;
  DatoB = (B*B) + (C*C);
  DatoB = sqrt(DatoB);

  Value = atan2(DatoA, DatoB);
  Value = Value * 180/3.14159265359;

  return (int)Value;
}

void FunctionsMPU(){
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,6,true);  // request a total of 14 registers
  AcX=Wire.read()<<8|Wire.read();  // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY=Wire.read()<<8|Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
}
```

- Pitch and Roll function
- Double variables of Acceleration in X and Y
- Using Pythagorean's Theorem

- Hypotenuse (the main terminal arm of degree is now converted to Radians (180 degrees divided by pi)
- Return the value in radians and set that as the answer to the function. (result)

- Function that reads the Values for acceleration from module. 6DOF, only using 3 of them (Acceleration in X,Y,Z) ,meaning only three sub-registers need to be accessed.

Ethernet Shield Code:

```
#include <SPI.h> //Libraries to allow for Servo control and Serial communication
#include <Servo.h>
#include <SoftwareSerial.h>

SoftwareSerial NSerial(11, 10); //RX|TX //initializing the RX and TX pins at 11 and 10
                                //respectively for cross arduino communication
Servo ServoX, ServoY; //define servos
int Pitch = 50; //define pitch
int Roll = 50; //define roll
int commands[6] = {0,5,0,0,5,0}; //define integer array

void setup(){
  Serial.begin(9600); //start serial communication with the computer
  NSerial.begin(9600); //start serial communication with the uno
  ServoX.attach(9); //initialize servos on pins 9 and 6
  ServoY.attach(6);
}
```

```
#include <SPI.h> //Libraries to allow for Ethernet shield v2 usability and Serial communication
#include <Ethernet2.h>
#include<SoftwareSerial.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // MAC address for ethernet shield
IPAddress ip(192, 168, 1, 177); // IP address, may need to change depending on network
EthernetServer server(80);  // create a server at port 80
SoftwareSerial NSerial(5,4); //RX/TX //for serial communication to the nano

String HTTP_req;            // stores the HTTP request
String servoCom = "000000"; //store commands from the webpage
void setup()
{
   NSerial.begin(9600);      //Stary serial communication with nano
   Serial.begin(9600);       // for diagnostics
   while (!Serial) {
     ; // wait for serial port to connect. Needed for native USB port only
   }
   Ethernet.begin(mac, ip);  // initialize Ethernet shield
   server.begin();           // start to listen for clients

   Serial.print("Server is at ");
   Serial.println(Ethernet.localIP());
}
```

```
void loop()
{
   EthernetClient client = server.available();  // try to get client

   if (client) {  // check if there's client?
      boolean currentLineIsBlank = true;
      while (client.connected()) {
         if (client.available()) {   // client data available to read
            char c = client.read(); // read 1 byte (character) from client
            HTTP_req += c;  // save the HTTP request 1 char at a time
            // last line of client request is blank and ends with \n
            // respond to client only after last line received
            if (c == '\n' && currentLineIsBlank) {
               if (HTTP_req.indexOf("@") > -1) { //check if request is for webpage or return from submit function
                  moveServo(); //trigger communication to nano
               } else {  // HTTP request for web page
                  // send a standard http response header
                  // send web page - contains JavaScript with AJAX calls
                  client.println("HTTP/1.1 200 OK");
                  client.println("Content-Type: text/html");
                  client.println("Connection: keep-alive");
                  client.println();
                  client.println("<!DOCTYPE html><html><head><title>Arduino Web Page</title>");
                  client.println("<script>");
                  client.println("var moveData =\"\";");
                  client.println("function moveServo() {");
                  client.println("moveData = document.getElementById(\"data1\").value;");
                  client.println("nocache = \"&nocache=\" + Math.random() * 1000000;");
                  client.println("var request = new XMLHttpRequest();");
                  client.println("request.onreadystatechange = function() {");
                  client.println("if (this.status == 200) {");
                  client.println("if (this.responseText != null) {");
                  client.println("document.getElementById(\"range_txt\").innerHTML = moveData;");
                  client.println("}}}");
                  client.println("request.open(\"GET\", \"ajax_switch\" + nocache + \"@\" + moveData + \"$\", true);");
                  client.println("request.send(null);");
                  client.println("}");
                  client.println("</script></head><body>");
                  client.println("<h1>Camera Control</h1>");
                  client.println("<p id=\"range_txt\">Commands Are Not Sent</p>");
                  client.println("<input type=\"text\" id=\"data1\">PanRol</input></br>");//data is entered in 6 digit
combos, ex 002031 with first 3 being pan, last 3 being roll
                  client.println("<button type=\"button\" onclick=\"moveServo()\">Move the servo</button>");
                  client.println("</body>");
                  client.println("</html>");
               }
               // display received HTTP request on serial port
               Serial.print(HTTP_req);
               HTTP_req = "";            // finished with request, empty string
                break;
            }
            }
```

```
  // every line of text received from the client ends with \r\n
            if (c == '\n') {
               // last character on line of received text
               // starting new line with next character read
               currentLineIsBlank = true;
            }
            else if (c != '\r') {
               // a text character was received from client
               currentLineIsBlank = false;
            }
         } // end if (client.available())
      } // end while (client.connected())
      delay(1);      // give the web browser time to receive the data
      client.stop(); // close the connection
   } // end if (client)
}

// send the state of the switch to the web browser
void moveServo()
{
  servoCom = HTTP_req.substring(HTTP_req.indexOf("@")+1,HTTP_req.indexOf("$"));
//store string of commands from the http request
  for(int i =0; i < 6; i++){ //write data byte by byte to the nano through serial
    NSerial.write(servoCom.charAt(i));
    Serial.println(servoCom.charAt(i));
  }
}
```

Hardware Inventory and Explanation:

	Part	Role in Project
1	Servo Motor (HS-311)	Heavy duty servo with 7kg/cm torque. Used to turn the Roll axis)
2	Mini Servo Motor SG90	Small servo motor with less torque to control the pitch (less mass to rotate)
3	GY151 (MPU6050) – Accelerometer and Gyroscope Module	The MPU 6050 is a sensor based on MEMS (Micro Electro Mechanical Systems) technology. Both the accelerometer and the gyroscope are embedded inside a single chip. This chip uses I2C (Inter-Integrated Circuit) protocol for communication. Three values given from the 6DOF… only using Accelerometer values in project. Calculate the change of speed in the X,Y,Z axis over time (Velocity Vectors)
4	Microcontroller (Arduino Uno) and Nano	• Microchip that controls protocols. Used in this project to control the hardware using software with preloaded libraries. Used as a power source (120 V AC from outlet → 12 V DC in PC → 5V DC in Arduino). Used to control the signals sent to PWM. • The Nano was used to communicate with the UNO for more external memory. Using transmitting and receiving outputs and inputs.
5	Ethernet Shield	Used to connect to server through common ip address.
6		

Web Links directly related to this project:

1	https://www.arduino.cc/en/reference/servo
2	http://42bots.com/tutorials/arduino-uno-and-the-invensense-mpu-6050-6dof-imu/
3	https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf Datasheet… for MPU6050

S.T.E.M. Concepts and Skills used in this activity:

1. Relation to Vectors and Kinematics. The concept of 3 Dimensions (x,y,z) and vectors pointing in a direction. Change of speed over time (acceleration) in a specific direction, therefore it is change in velocity. Relation to Vectors through 3D printing the gimbal model.
2. Relation to concept of Torque. The concept of how much force per distance the servo motor can output.
3. Networking concepts used through the Ethernet shield. Connecting to the server through router and used it as a booster. Seamless connection through IP address. Used as a hub.
4. 3D design in SolidWorks and Microsoft 3D builder. Using measurements to customize gimbal that was printed on the 3D printer.
5. Asynchronous Serial Communication: Knowledge of bits of data and baud rate to have two devices communicating.
6. Knowledge and concept of AC voltage and DC voltage.
7. Knowledge of programming and simple algorithms such as Pythagorean theorem

10 Questions and answers about this project. (FAQ)

1	Does the gimbal calibrate according to the accelerometer and gyroscope?
	Yes, because the centre of mass must be found to ensure the proper leveling takes place.
2	What is the purpose of using to Arduino models for the Ethernet controlled part?
	To specify the exact pan and tilt a user would like (could be used as a security camera)
3	What is the source of power when the gimbal is portable?
	External battery pack or a 9v battery supply (voltage regulator will be used).
4	Does the density of the plastic used in 3D printing affect the overall performance of the gimbal and its stabilization?
	Yes, if the model is more sturdy and dense, there will be less vibrations throughout the element from the servos.
5	What is the most extensive case to make the Ethernet shield version as portable and range finding as possible?
	You are limited because there must be a booster (Hub), like a router. You must also have a cat5e cable.
6	Does the size of the servo motors determine the torque?
	Yes, gears vary in size and gear ratios, when put together, are then determined.

7	Does the MPU6050 use both the accelerometer and gyroscope values in this prioject?
	Yes, the values of the 3DOF axis for each sensor are combined for more accurate degrees of angle.
8	Does it make a difference whether or not the servo motors are used, or can you use DC motors?
	You could use DC motors with a driver shield and reverse the polarity whenever you want to spin the platform in the opposite direction… however that is very inefficient. DC brushless motors can be used instead that uses laws of induction and keeps a motor spinning.
9	Is it possible to create a better webpage for the user?
	Not with the current Ethernet shield, because the external memory is not as much as we would like it to be. A lot of it is filled with setting variables.
10	Why is the other Arduino Nano connected to the Uno with the Ethernet shield stacked onto it?
	We used the Transmit and Receive signal pins on the Ethernet shield to Nano for more memory and storage; Nano is another memory device.