



AQUILA INTEGRATED SOLUTIONS

AMERICAN
INTERNATIONAL SCHOOL

**DID YOU
KNOW?**



70%

of the Earth's surface is Ocean

**DID YOU
KNOW?**





The Aquila II

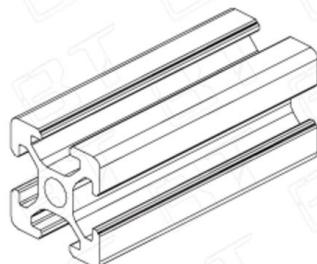
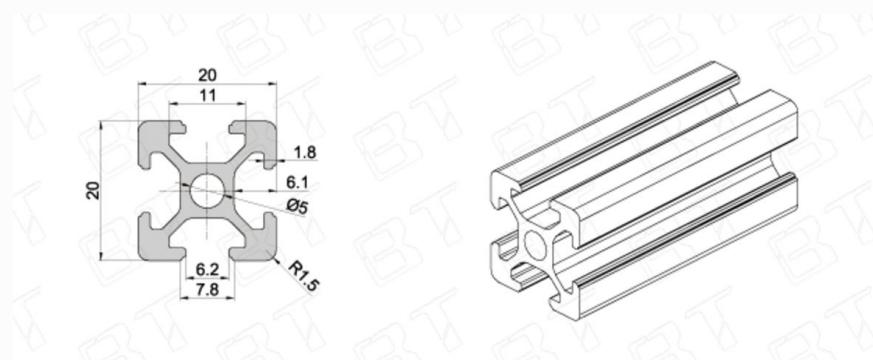


KILLS



Mechanical

Frame

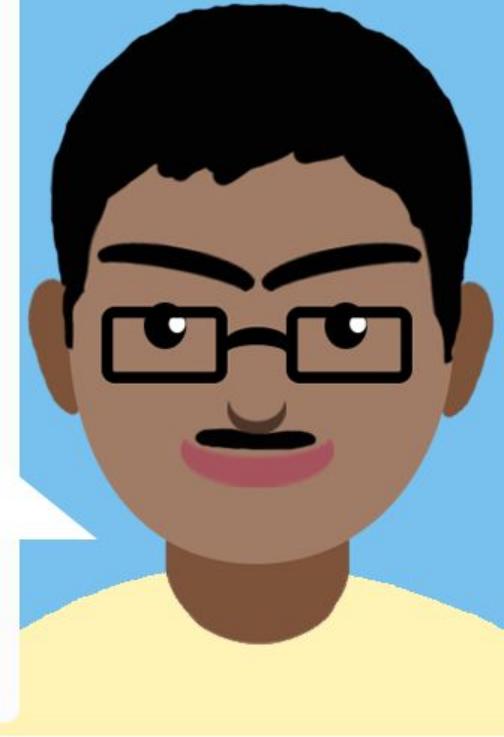
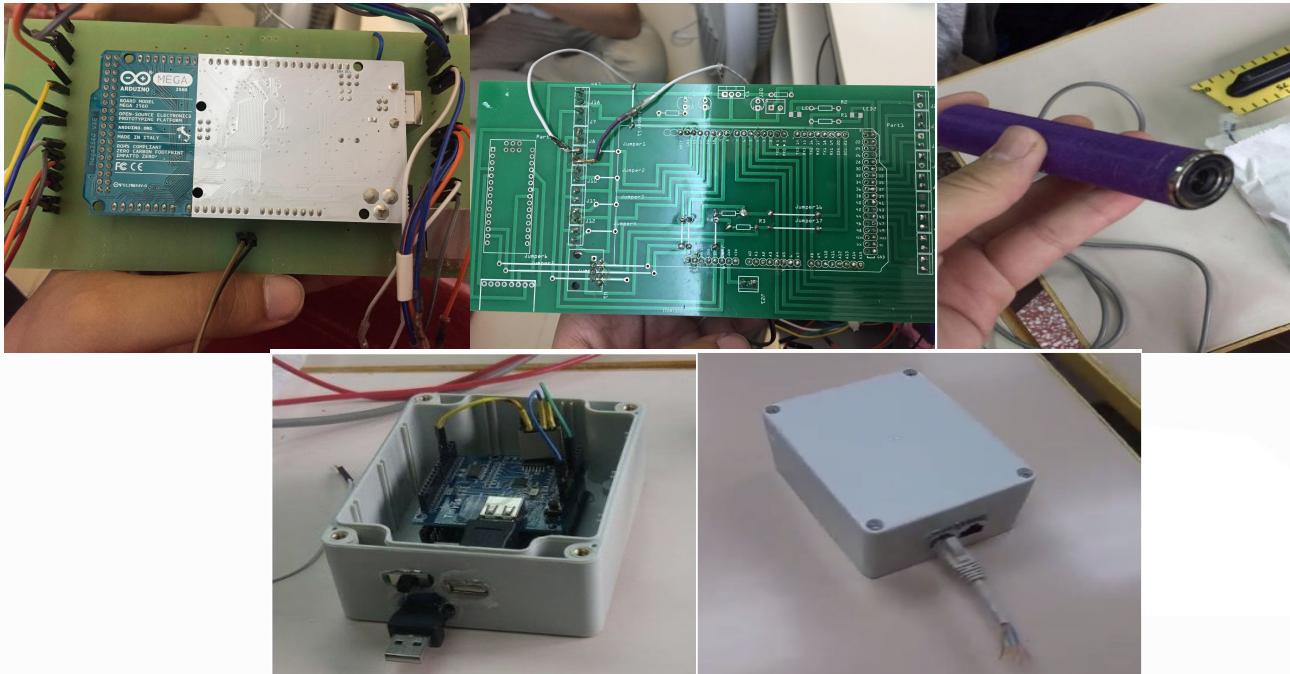


Dry Housing

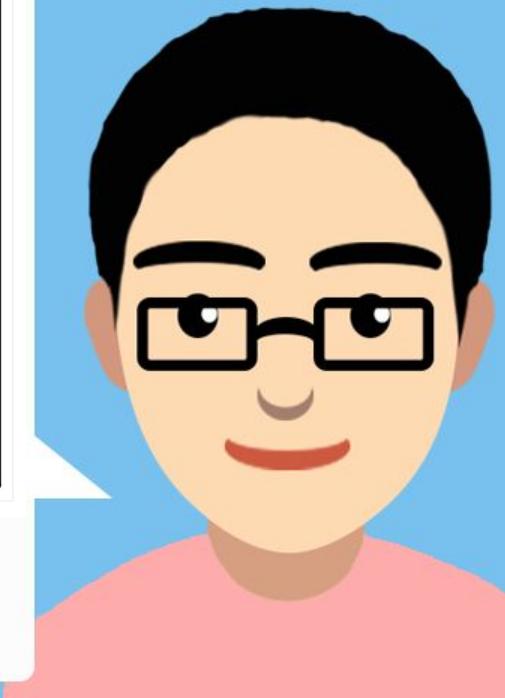
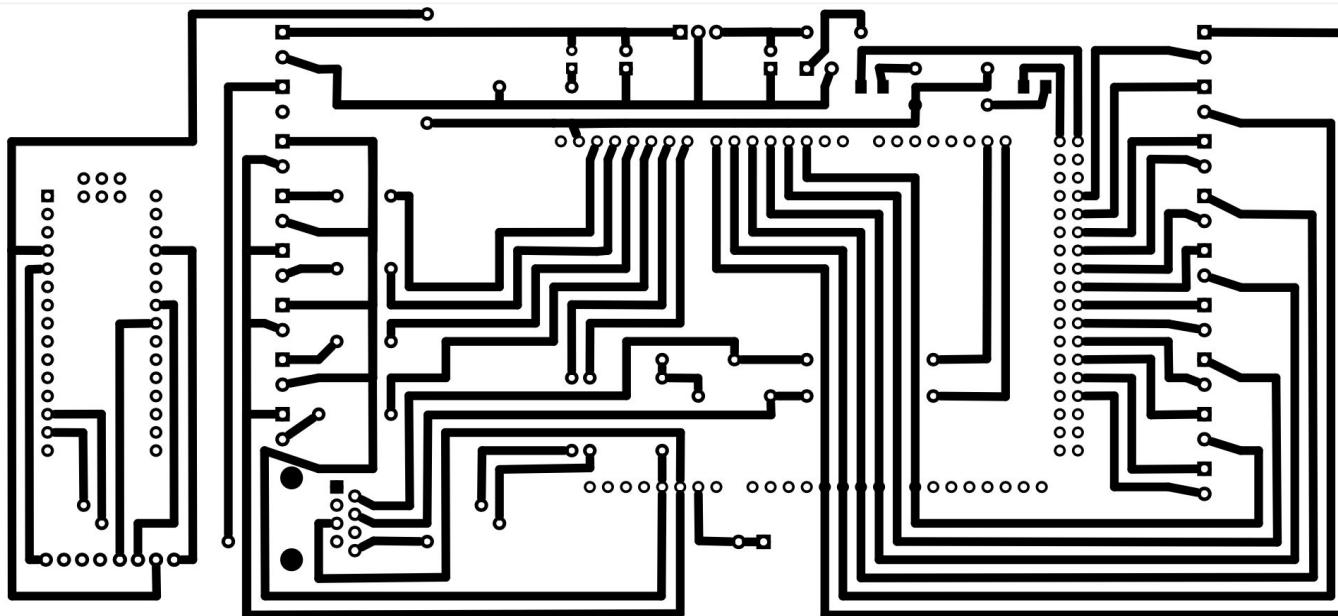


Electronics

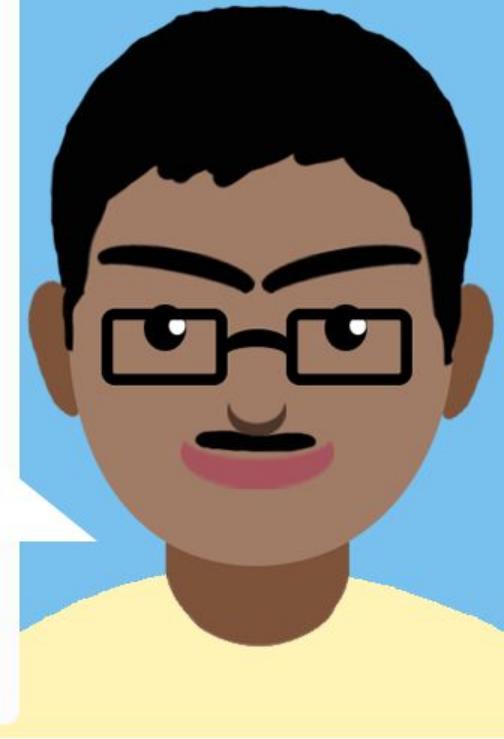
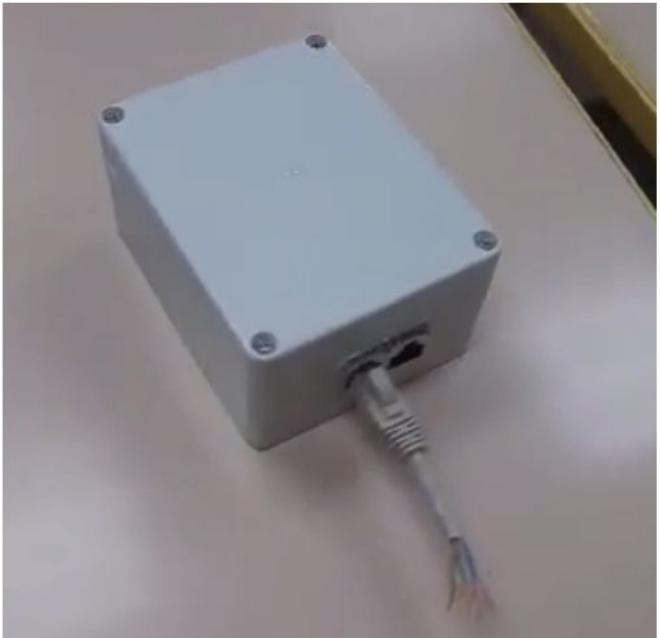
Overview



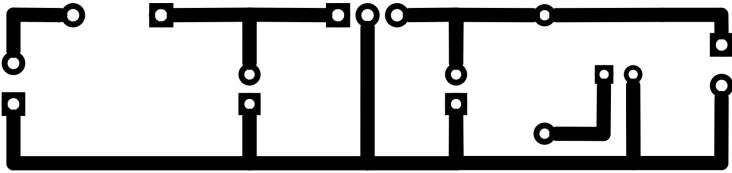
Connector Board



Control Box



Camera



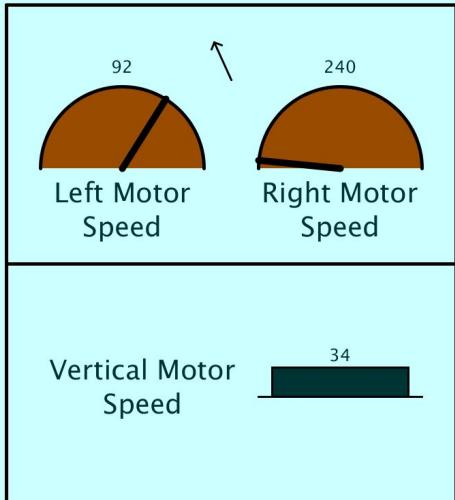
Software

PS3 Control System

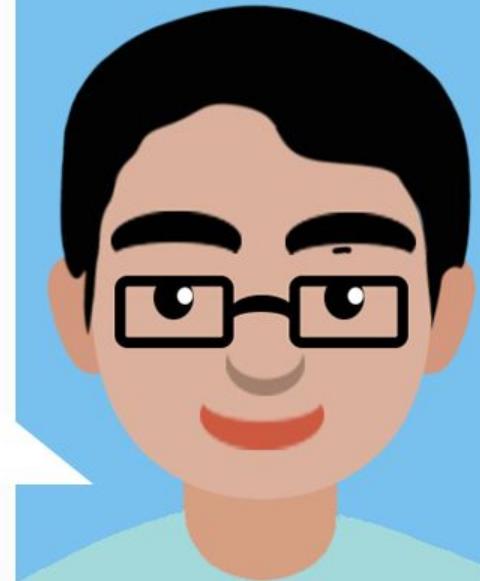


Graphic User Interface (GUI)

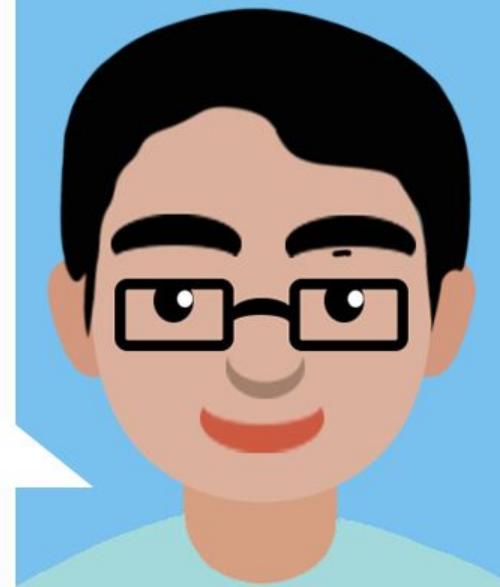
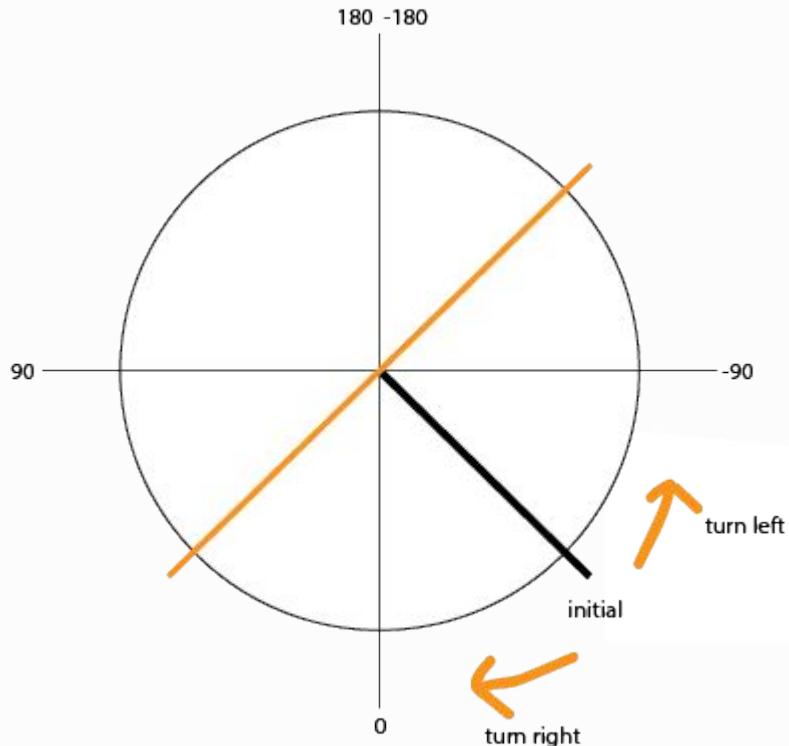
Running



Gripper	
X Angle	Y Angle
151 164	
Camera	
X Angle	Y Angle
69	143



VR Camera Head Tracking System



Distance Calculation



Choose file: Distance Calc Test.jpg 4

Start Calculate **Reset**

X: 980 Y: 262
X: 916 Y: 263
X: 1032 Y: 423
X: 644 Y: 474

Multiply Factor: 16.00185300580526

Real Distance = Virtual Distance / Magnified Factor

Real Distance = $\sqrt{(644-1032)^2 + (474-423)^2} / \sqrt{(916-980)^2 + (263-262)^2} / 4$

Real Distance = 24.2586 cm



Auto Balancing System

```
autoBalance | Arduino 1.6.11
File Edit Sketch Tools Help
autoBalance
#include "I2Cdev.h"
#include <Servo.h>
#include <SoftwareSerial.h>

#include "MPU6050_6Axis_MotionApps20.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif

MPU6050 mpu;

#define OUTPUT_READABLE_YAWPITCHROLL

#define INTERRUPT_PIN 2 // use pin 2 on Arduino Uno & most boards

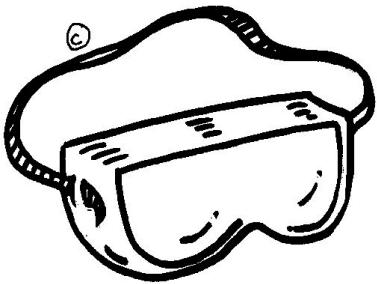
// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
uint8_t devStatus; // return status after each device operation (0 = success, !0 = error)
uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount; // count of all bytes currently in FIFO
uint8_t fifoBuffer[64]; // FIFO storage buffer

// orientation/motion vars
CustomGyro gyroscope; // create an instance of our gyro class
CustomInclinometer inclinometer; // create an instance of our inclinometer class
CustomMotion motion; // create an instance of our motion class
CustomCompass compass; // create an instance of our compass class
CustomBarometer barometer; // create an instance of our barometer class
CustomHumidity humidity; // create an instance of our humidity class
CustomLight light; // create an instance of our light class
CustomTemperature temperature; // create an instance of our temperature class
CustomPressure pressure; // create an instance of our pressure class
CustomHumidity2 humidity2; // create an instance of our humidity2 class
CustomLight2 light2; // create an instance of our light2 class
CustomTemperature2 temperature2; // create an instance of our temperature2 class
CustomPressure2 pressure2; // create an instance of our pressure2 class
CustomContainer container; // create an instance of our container class

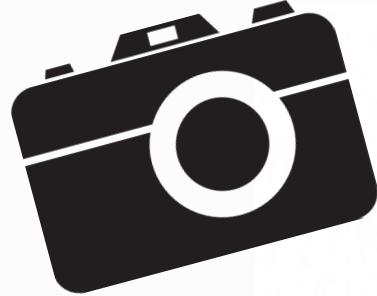
Done Saving.
```



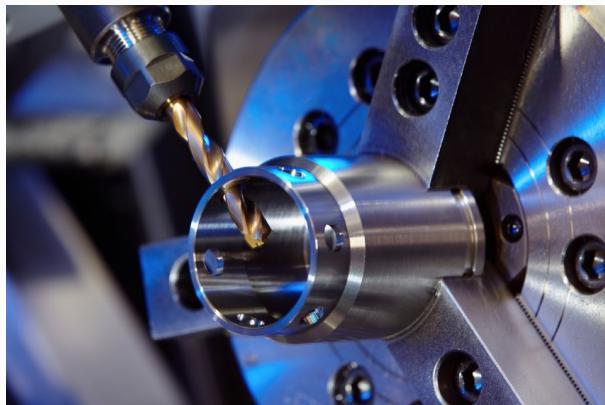
Safety



Theme

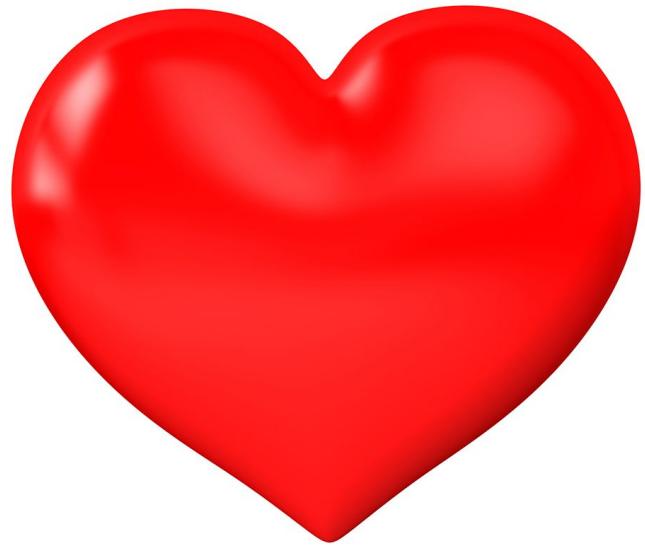


Decision Making





Lesson Learned



THANK YOU !

