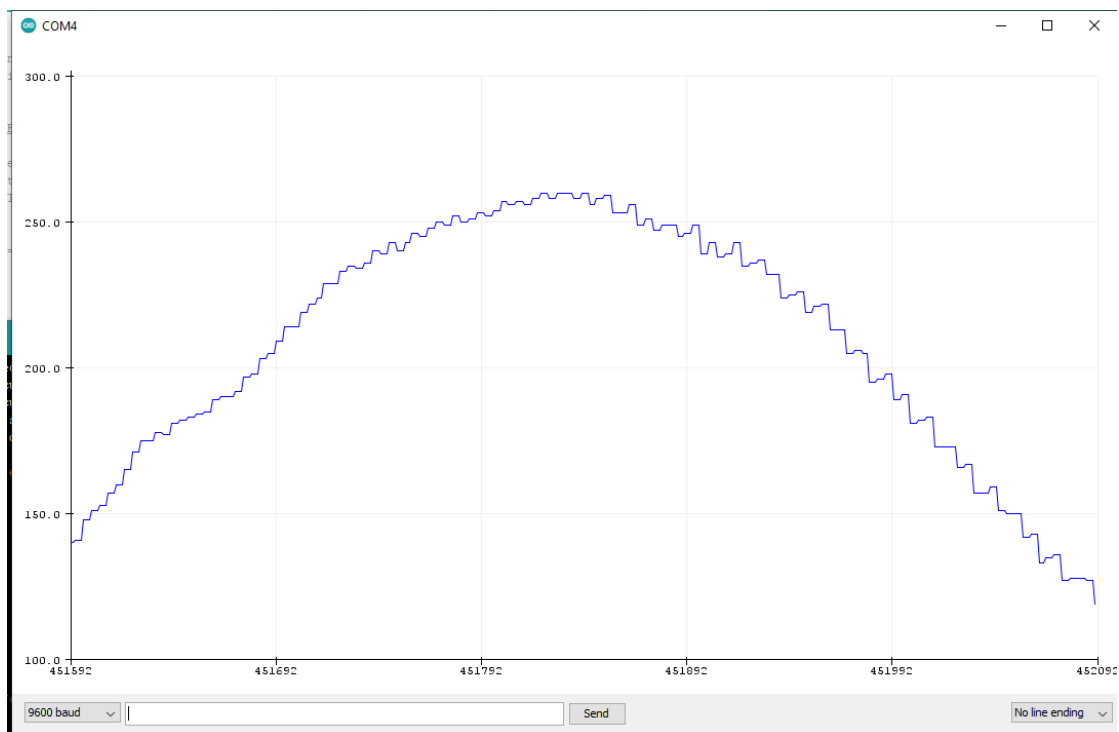
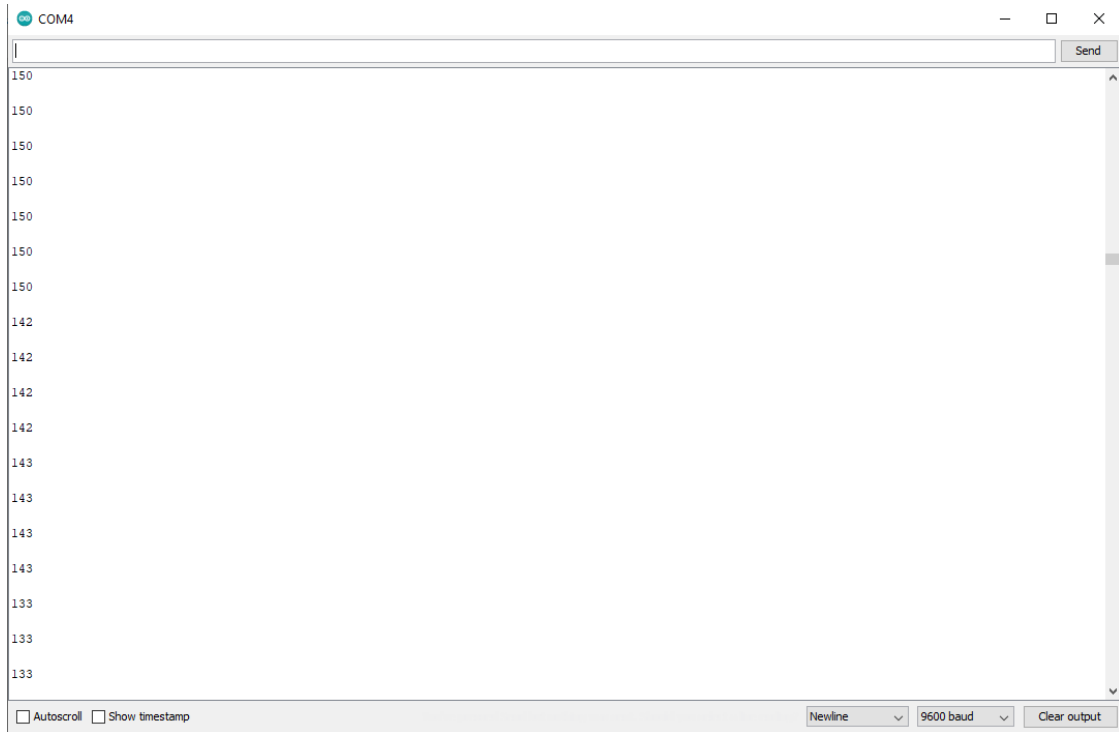


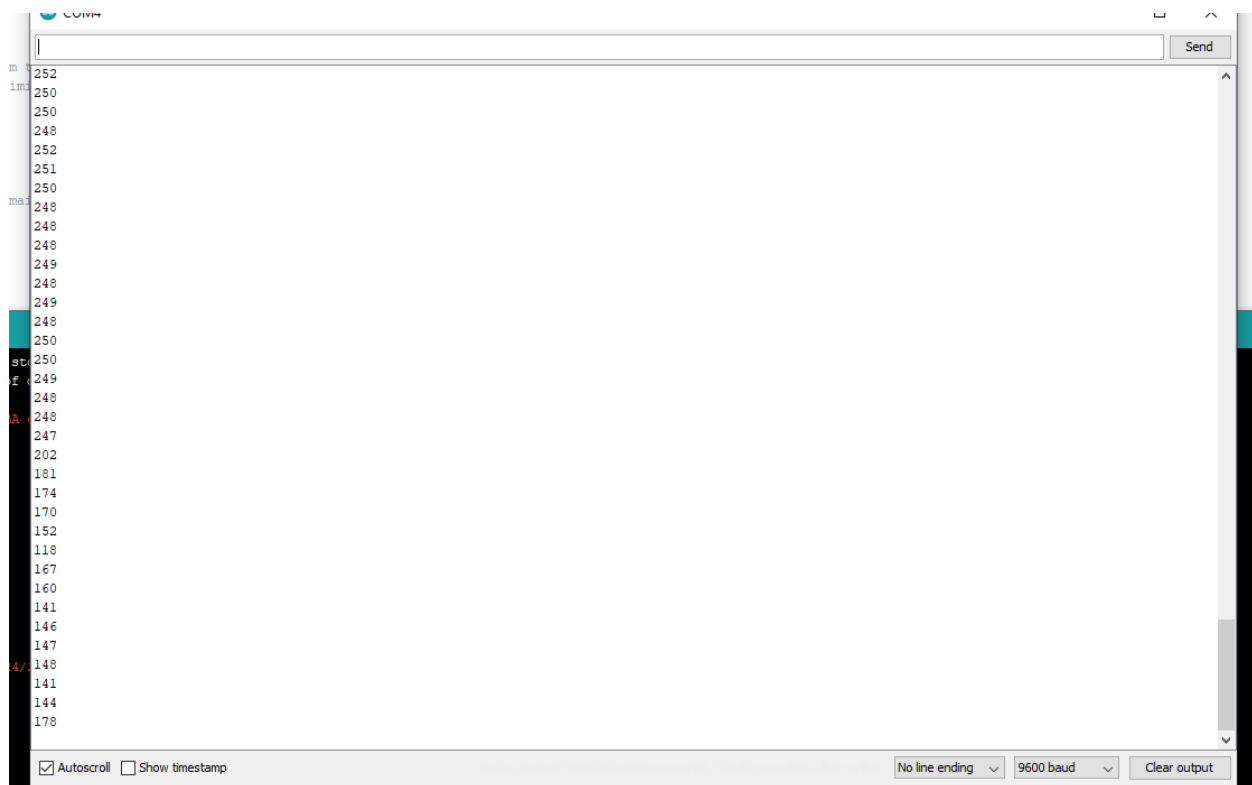
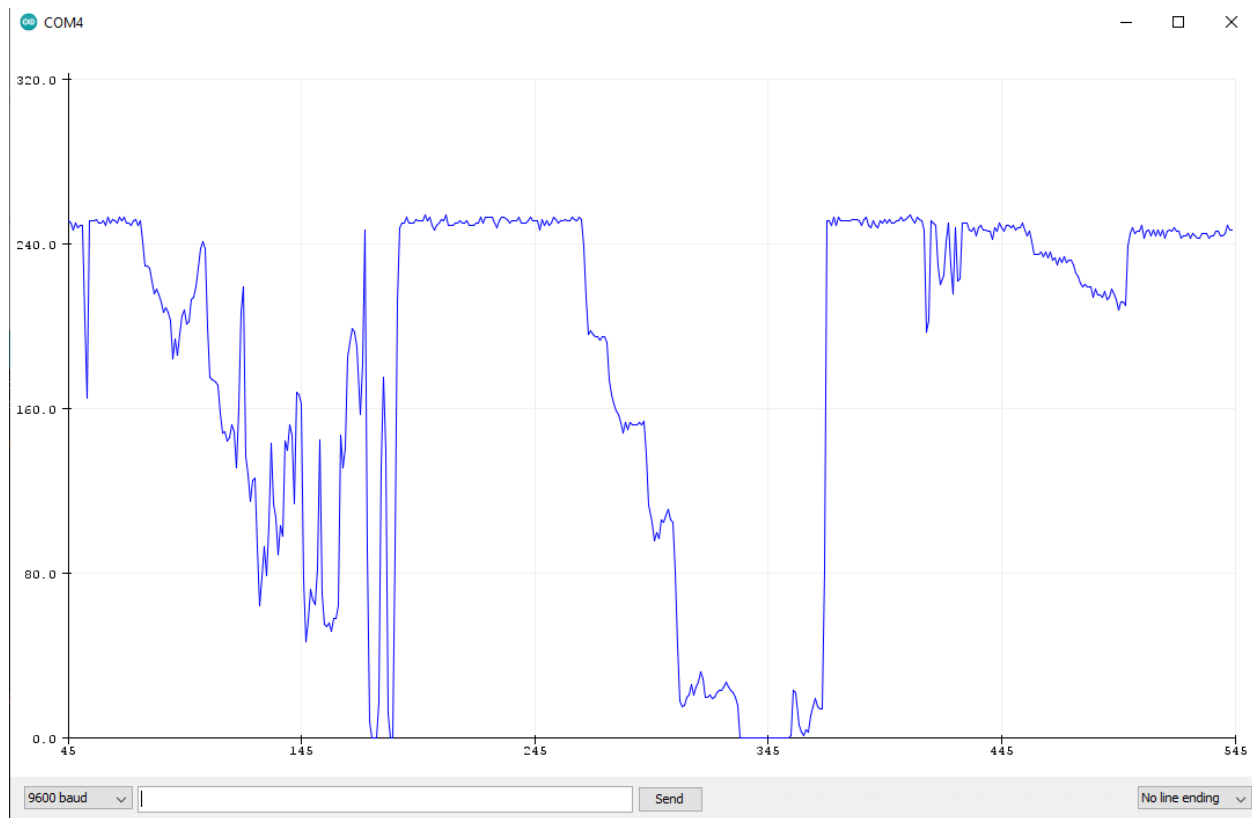
## ECGR 4090 Machine Learning with IoT

Thomas Yingling

**Hello World:** Sample of Serial Monitor and Serial Plotter



**ProximitySensor:** Sample of Serial Plotter and Serial Monitor



**Hello Proximity:** New Code, Serial Plotter, Serial Monitor

```

void HandleOutput(tflite::ErrorReporter* error_reporter, float x_value,
                 float y_value) {
    // Do this only once
    if (!initialized) {
        // Set the LED pin to output
        pinMode(led, OUTPUT);
        initialized = true;
    }

    // Calculate the brightness of the LED such that y=-1 is fully off
    // and y=1 is fully on. The LED's brightness can range from 0-255.
    int brightness = (int)(127.5f * (y_value + 1));

    // check if a proximity reading is available
    if (APDS.proximityAvailable()) {
        // read the proximity
        // - 0    => close
        // - 255 => far
        // - -1   => error
        int proximity = APDS.readProximity();
        if (proximity < 200) {
            // 200 is around 4 inches from the board based on testing with a ruler
            // if within 200 proximity, write to LED as a constant keeping it lit
            // Write to serial monitor if hand is near
            brightness = 250;
            analogWrite(led, brightness);
            Serial.print("---Detected---\n");
        } else {
            // Set the brightness of the LED. If the specified pin does not support PWM,
            // this will result in the LED being on when y > 127, off otherwise.
            analogWrite(led, brightness);
            Serial.print("---Not Detected---\n");
        }
    }

    // print proximity value to the Serial Monitor
    Serial.print(proximity);
    Serial.println("\tBoard Proximity\n");

    // print local analogWrite brightness to Serial Monitor
    Serial.print(brightness);
    Serial.println("\tWrite Brightness\n");
}

// Log the current brightness value for display in the Arduino plotter
TF_LITE_REPORT_ERROR(error_reporter, "%d\n", brightness);
}

```

---

hello\_proximity\$

arduino\_constants.cpp

arduino\_main.cpp

arduino\_

```
// Build an interpreter to run the model with.
static tflite::MicroInterpreter static_interpreter(
    model, resolver, tensor_arena, kTensorArenaSize, error_re
interpreter = &static_interpreter;

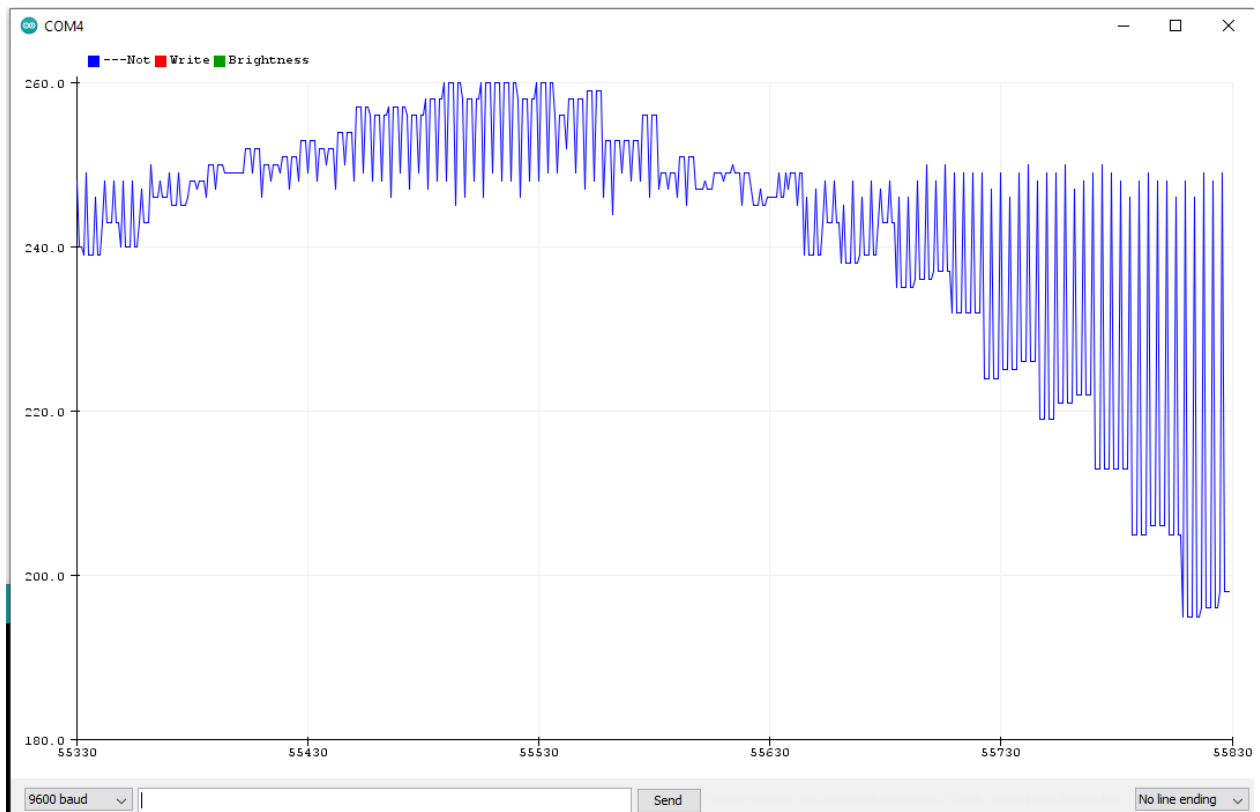
// Allocate memory from the tensor_arena for the model's tensors.
TfLiteStatus allocate_status = interpreter->AllocateTensors();
if (allocate_status != kTfLiteOk) {
    TF_LITE_REPORT_ERROR(error_reporter, "AllocateTensors() failed");
    return;
}

// Obtain pointers to the model's input and output tensors.
input = interpreter->input(0);
output = interpreter->output(0);

// Keep track of how many inferences we have performed.
inference_count = 0;

// Wait for Connection
while (!Serial);

// Check for APDS peripheral
if (!APDS.begin()) {
    Serial.println("Error initializing APDS9960 sensor!");
}
}
```



COM4

2

---Not Detected---

206 Board Proximity

2 Write Brightness

2

2

---Not Detected---

200 Board Proximity

2 Write Brightness

2

2

---Detected---

193 Board Proximity

250 Write Brightness

250

2

---Detected---

184 Board Proximity

250 Write Brightness

250

☐ Autoscroll ☐ Show timestamp