

Design Log

Cumberland Valley

C8

Detector Building

Viraj Singh and Srihari Rajesh

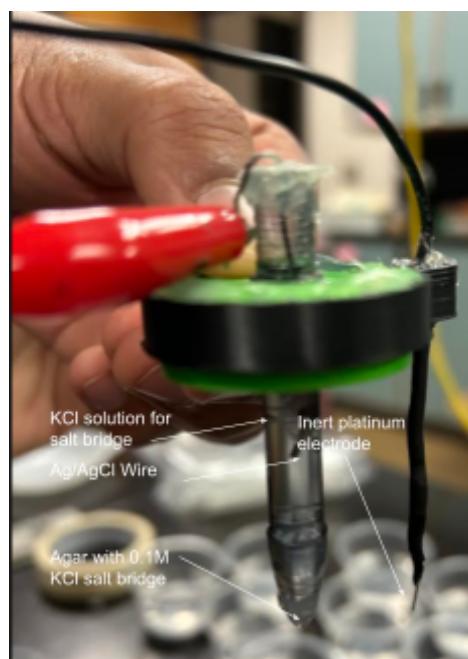
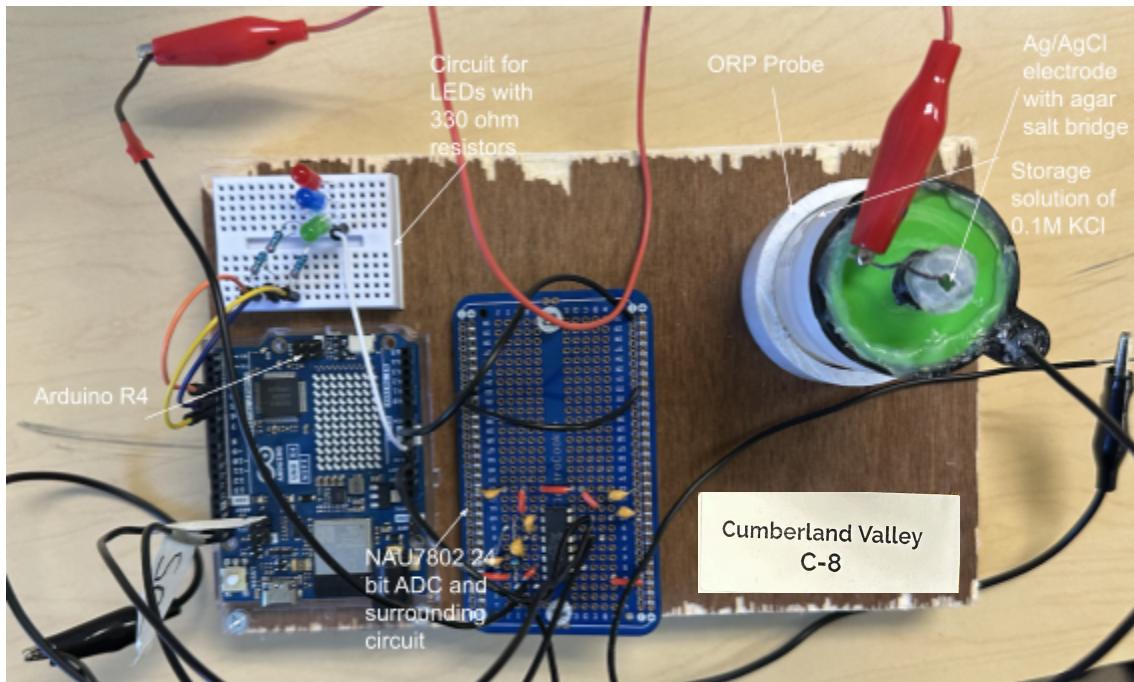
Device Construction and Calibration: 4.b.i

The device was constructed with an Arduino Uno R4, 1 breadboard, silver wire coated in silver chlorine, potassium chloride, agar, platinum wire, LEDs, and a few resistors. The ORP probe consisted of two electrodes, an inert indicator electrode, and a reference electrode. The inert electrode was made using platinum as our inert metal and it was attached to a piece of standard wire and a pen casing to stick in the solution as a probe. The reference electrode involves a silver/silver chloride half-reaction. The silver wire was placed in bleach to produce a silver chloride coating on the wire. The salt bridge for the ORP electrode was created using agar and potassium chloride. The agar was melted with dissolved potassium chloride and set into a pen tip to create the salt bridge. Our ORP electrode also contains 0.1M potassium chloride solution for the flow of electrons as a part of the half-reaction.

The silver/silver chloride wire is submerged in the pen case and potassium chloride solution and acts as our ORP electrode. The agar salt bridge allows the silver/silver chloride wire to interact with the salt solution and test the oxidation-reduction potential of the solution. This is compared to how the salt solution affects the voltage on the inert electrode to output a potential voltage difference which is our ORP reading. The LEDs are programmed to light up when the ppm calculated is within the instructed ranges. Two circuits comprise our device, one of which is for the LEDs, and the other circuit is for the NAU7802 ADC*. The three 330 Ohm resistors in the LED circuit are connected in series to each color LED: red, green, and blue. The voltage across the two electrodes is then sent into the NAU7802 IC to be amplified so it can be read and printed by the Arduino Uno. To be able to read this using the Arduino Uno we used an Adafruit library to read one bit at a time during the specified clock speed. This is only used for functionality, not for calibration. Calibration is done by recording the voltage values of known masses and creating a line of best fit.

Labeled Images: 4.b.i

Due to the nature of our top-down image and not being able to see the full probe in the top-down image, we included another view for more detail on the probe.



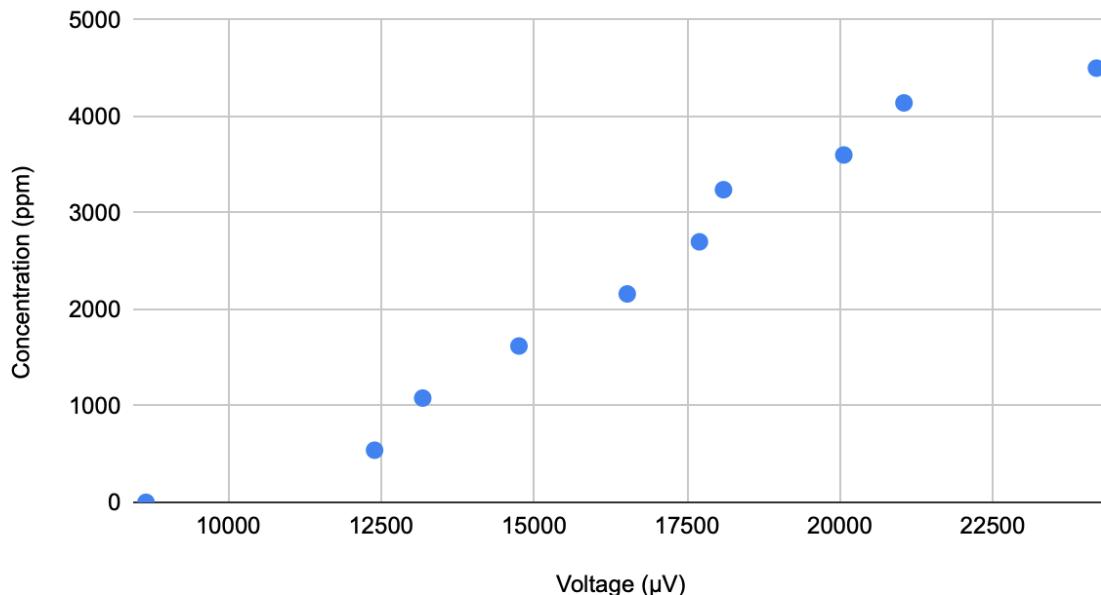
Data Table: 4.b.ii

	Raw Value	Voltage (μ V)	Concentration (ppm)
Trial 1	44000	8654.594937	0
Trial 2	63000	12391.80639	540
Trial 3	67000	13178.58775	1080
Trial 4	75000	14752.15046	1620
Trial 5	84000	16522.40852	2160
Trial 6	90000	17702.58055	2700
Trial 7	92000	18095.97123	3240
Trial 8	102000	20062.92463	3600
Trial 9	107000	21046.40132	4140
Trial 10	123000	24193.52676	4500

Graphs and Equation: 4.b.iii,4.b.iv

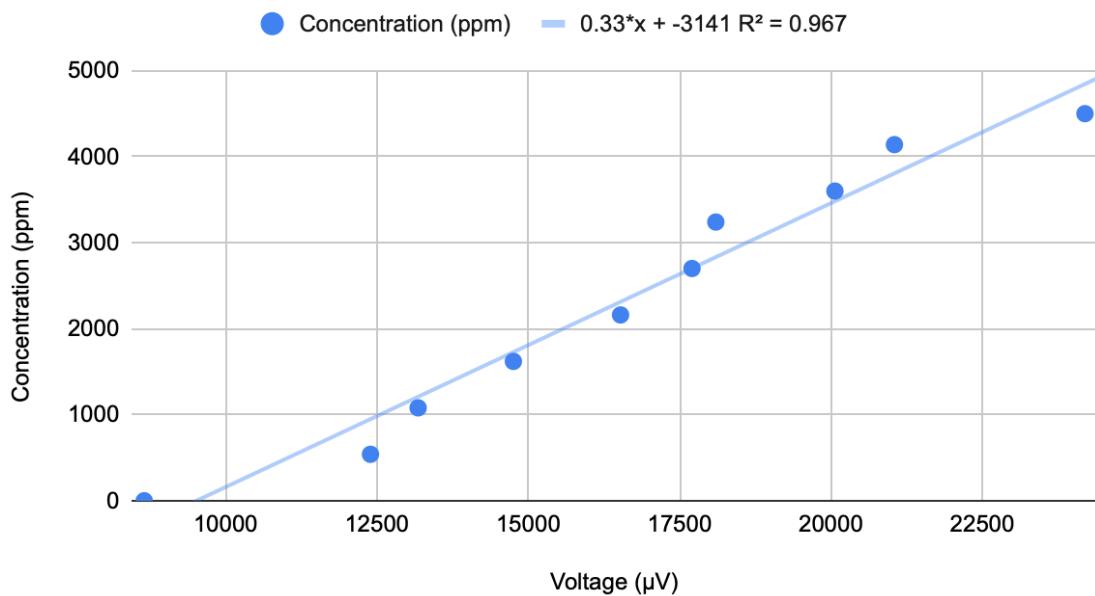
Just datapoints graph

Concentration (ppm) vs. Voltage (μ V)



Datapoints with function overlaid

Concentration (ppm) vs. Voltage (μ V)



$$\text{Equation: Concentration in PPM} = 0.33(\text{voltage}) - 3141$$

Program Code (4.b.vi and 4.b.vii)

RED - vi

YELLOW - vii

```
// Viraj Singh and Srihari Rajesh
// #C-7
// Cumberland Valley High School
#include <Adafruit_NAU7802.h>

Adafruit_NAU7802 nau;
const int32_t voltZeroVal = 43000;
const int32_t zeroVal = voltZeroVal/((3.3*pow(10,6))/16777215); //add the
voltage for the zero value

const int redPin = 5;
const int greenPin = 3;
const int bluePin = 4;
void setup() {
    Serial.begin(115200);
    pinMode(redPin, OUTPUT);
    pinMode(greenPin, OUTPUT);
    pinMode(bluePin, OUTPUT);
    nau.begin();
    nau.setLDO(NAU7802_3V0);
    nau.setGain(NAU7802_GAIN_128);
    nau.setRate(NAU7802_RATE_10SPS);
    for (uint8_t i=0; i<10; i++) {
        while (! nau.available()) delay(1);
        nau.read();
    }
    while (! nau.calibrate(NAU7802_CALMOD_INTERNAL)) {
        delay(1000);
    }
    while (! nau.calibrate(NAU7802_CALMOD_OFFSET)) {
        delay(1000);
    }
}
bool stopped = false;
int32_t offset = 0;
char c;
void loop() {
    if(!stopped)
    {
        while (!nau.available())
            delay(1);
        int32_t val = nau.read();
```

```

    if (Serial.available() > 0) {
        c = Serial.read();
        if(c == 's'){
            stopped = true;
            return;
        }
        else{
            offset = val-zeroVal;
        }
    }
    int32_t num = val-offset;
    float voltage = num * ((3.3*pow(10,6))/16777215);
    float ppm = (-3141+0.33*voltage);
    Serial.print("Voltage "); Serial.println(String(voltage,4));
    Serial.print(" Raw "); Serial.println(num);
    Serial.print(" PPM "); Serial.println(String(ppm,0));
    showLED(ppm);
    delay(1500);
}
else
{
    if (Serial.available() > 0) {
        c = Serial.read();
        if(c == 's')
            stopped = false;
    }
}
}

```

```

const int redRanges = 1;
const int greenRanges = 1;
const int blueRanges = 1;

const float redMin[redRanges] = {0};      //ppm
const float redMax[redRanges] = {1000}; //ppm
const float greenMin[greenRanges] = {200}; //ppm
const float greenMax[greenRanges] = {3500}; //ppm
const float blueMin[blueRanges] = {1500}; //ppm
const float blueMax[blueRanges] = {4900}; //ppm

void showLED(float m) {
    digitalWrite(redPin, LOW);
    digitalWrite(greenPin, LOW);
    digitalWrite(bluePin, LOW);

    for (int i = 0; i < redRanges; ++i) {

```

```
    if (m >= redMin[i] && m <= redMax[i]) {
        digitalWrite(redPin, HIGH);
        break;
    }
}
for (int i = 0; i < greenRanges; ++i) {
    if (m >= greenMin[i] && m <= greenMax[i]) {
        digitalWrite(greenPin, HIGH);
        break;
    }
}
for (int i = 0; i < blueRanges; ++i) {
    if (m >= blueMin[i] && m <= blueMax[i]) {
        digitalWrite(bluePin, HIGH);
        break;
    }
}
}
```

3D Printed Part: 4.c

- We used an Anycubic Kobra Max 3d printer that our team had to 3d print a holder which contains both reference and indicator electrode
- We used an STL file which we CADed in Onshape
- Google Drive Folder With Part:
- <https://cad.onshape.com/documents/89652b91bbaddf5bf4abcab4/w/88b599a99106b09832e2d2e3/e/e7b10072fde8398f51552ee3?renderMode=0&uiState=65cf0dec6f31f28c34330dd>
- The part is made of PLA plastic.

