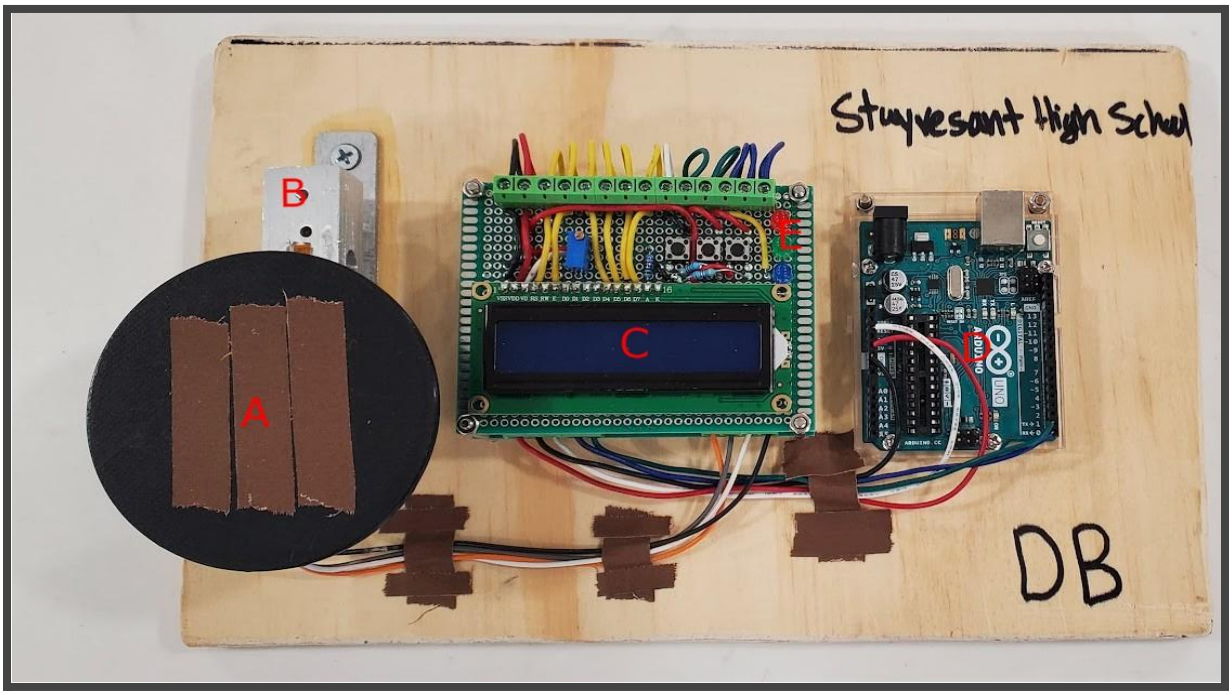
Detector Building Design Log

Stuyvesant A

Team 52

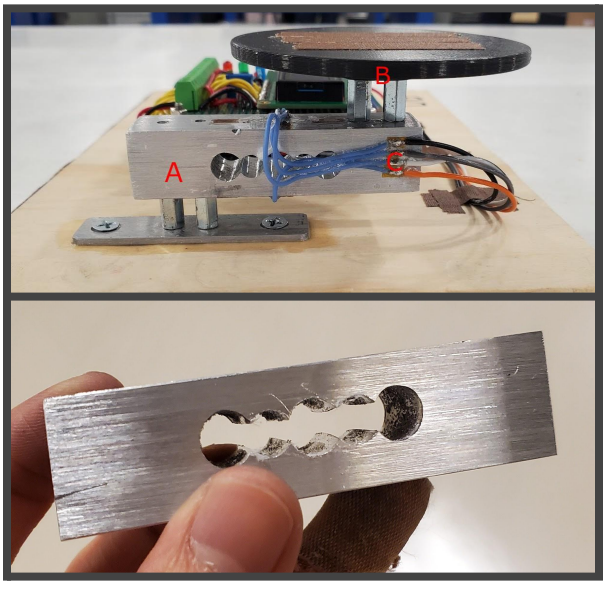
New York States Competition

TOP-DOWN VIEW:



KEY:

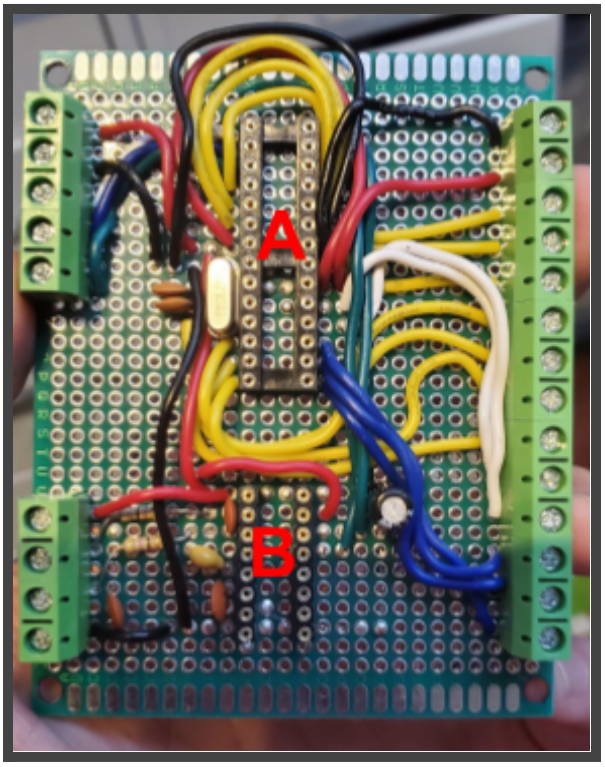
A: Platform for sample
B: Aluminum frame with strain gauges
C: 16x02 LCD display
D: Arduino board serving as a power supply and programming interface
E: LEDs for measurement



ALUMINUM FRAME:

A: ¾ x ¾ x 4 inch Aluminum block
B: 3D-printed platform for sample
C: Wire connections for full Wheatstone bridge

Aluminum bar stock was cut, and holes were drilled using a drill press. The threads were tapped and strain gauges were glued on and wired.



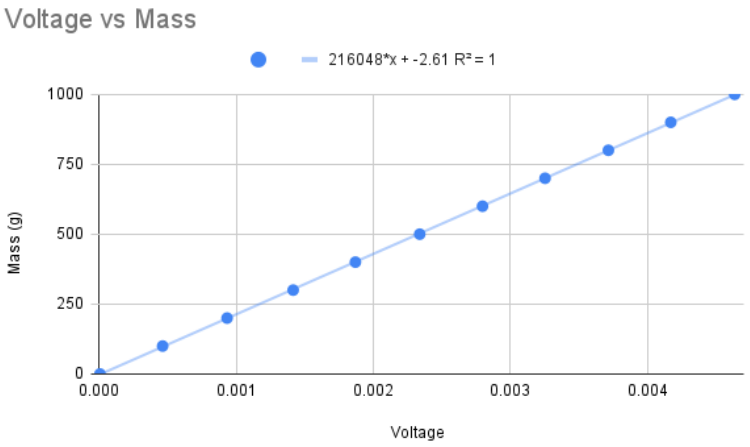
MAIN BOARD:
A: ATMEGA328P 16Mhz Microcontroller
B: NAU7802 24-bit ADC

The circuit was first prototyped on a breadboard then soldered onto a perfboard in order to increase the durability and reliability of the build.

TOOLS USED:
3D-PRINTER: Ultimaker 3
FILAMENT/MATERIAL: PLA
SLICER SOFTWARE: Cura
CAD SOFTWARE: Onshape
Drill Press: WEN 4208T

DATA:

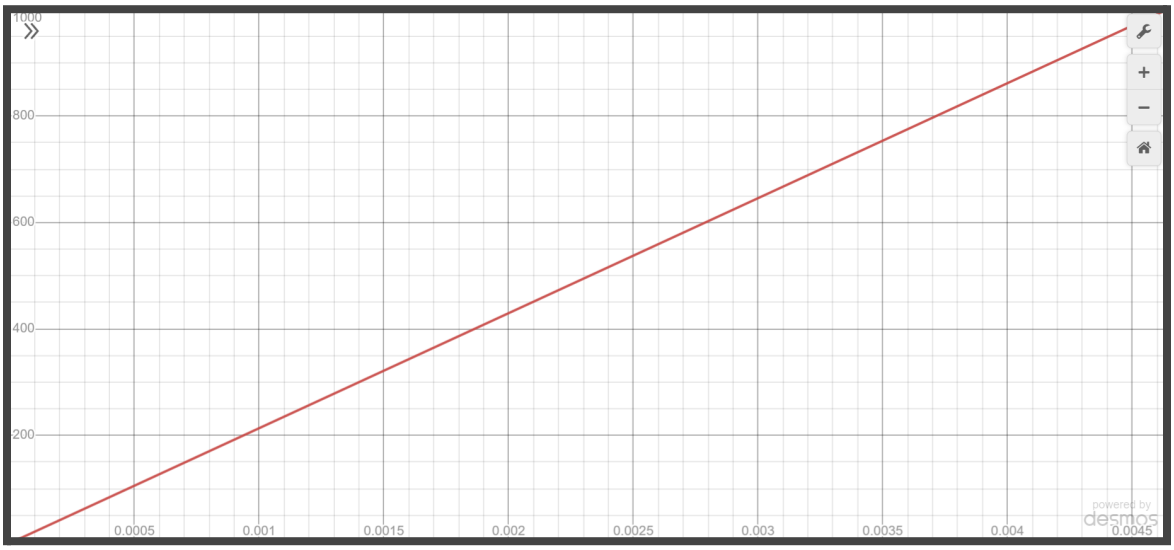
mass (g)	voltage	digital reading
0	0.000006	22
100	0.000463	1726
200	0.000933	3478
300	0.001415	5275
400	0.001869	6968
500	0.002338	8716
600	0.002796	10424
700	0.003252	12124
800	0.003714	13846
900	0.004169	15543
1000	0.004635	17280



Function to convert Voltage into mass in grams:

$$f(x) = 216048x - 2.61$$

Graph of function:



Code:

```
#include <Wire.h>
#include <Filters.h>
#include <LiquidCrystal.h>
#include "SparkFun_Qwiic_Scale_NAU7802_Arduino_Library.h"

/*--WIRING--*/

//LCD PINS
const int RS = 2, E = 3, D4 = 4, D5 = 5, D6 = 6, D7 = 7;

//LED PINS
const int RED = 9, GRN = 10, BLU = 11;

//PUSHBUTTONS
const int TOGGLE_TAR = 13, TOGGLE_CLEAR = 12;
```

```
/*--RANGES--*/

//Range 1: RED LED
const double R1_MIN = 50;
const double R1_MAX = 71;

//Range 2: GREEN LED
const double R2_MIN = 400;
const double R2_MAX= 501;

//Range 3: BLUE LED
const double R3_MIN = 600;
const double R3_MAX = 800;

/*--DATA CONSTANTS--*/

//FILTER CUTTOFFS
const double CUTOFF0 = 10;
const double CUTOFF1 = 1;

//CALIBRATION FACTOR
//SLOPE
const double CAL_M = 17.42;
//CONSTANT
const double CAL_K = 0;

//ZERO VAL
const long ZERO = 87443;

/*--GLOBALS--*/
long offset = 0;
long cycledelay = 0;

LiquidCrystal lcd(RS, E, D4, D5, D6, D7);
NAU7802 nau;

//2 filter stages to sort out noise
FilterOnePole stage0(LOWPASS, CUTOFF0, 0);
FilterOnePole stage1(LOWPASS, CUTOFF1, 0);
```

```
RunningStatistics avg;

void setup() {
  //init pins
  pinMode(RED, OUTPUT);
  pinMode(GRN, OUTPUT);
  pinMode(BLU, OUTPUT);
  pinMode(TOGGLE_TAR, INPUT);
  pinMode(TOGGLE_CLEAR, INPUT);//Might disable to just have the
reading constantly dispalyed

  //init LCD
  lcd.begin(16, 2);
  lcd.clear();
  lcd.print("INIT...");

  //start i2c interface
  Wire.begin();

  //init NAU7802
  if (nau.begin() == false) {
    lcd.clear();
    lcd.print("ERR: 0");
    while (1);
  }

  //configure NAU7802
  nau.setGain(NAU7802_GAIN_64);
  nau.setLDO(NAU7802_LDO_4V5);
  nau.setRegister(NAU7802_ADC, 0x30);
  nau.setBit(NAU7802_PGA_PWR_PGA_CAP_EN, NAU7802_PGA_PWR);
  nau.setSampleRate(NAU7802_SPS_80);
  nau.calibrateAFE();

  //Flush readings
  for (int i = 0; i < 100; i++) {
    read_data();
  }
```

```
  //Tare
  tare(read_data());

  //Inidicate end of init
  lcd.clear();
  lcd.print("READY! V3.1");
}

void loop() {
  long data;
  double mass;
  double voltage;

  //Aquire and process raw ADC data
  data = read_data();
  voltage = dig_to_analog(data);
  mass = dig_to_mass(voltage);


  //Print reading;
  if (cycledelay == 9) {
    lcd.setCursor(0, 0);
    if (mass < 10) { lcd.print("000"); }
    else if (mass < 100) { lcd.print("00"); }
    else if (mass < 1000) { lcd.print("0"); }
    lcd.print(mass, 1);
    lcd.print(" g");
    lcd.setCursor(0, 1);
    lcd.print(voltage, 6);
    lcd.print(" v");
    if ((mass > R1_MIN) && (mass < R1_MAX)) {   - LED CODE
      digitalWrite(RED, HIGH);
      digitalWrite(GRN, LOW);
      digitalWrite(BLU, LOW);
    }
    else if ((mass > R2_MIN) && (mass < R2_MAX)) {
      digitalWrite(RED, LOW);
      digitalWrite(GRN, HIGH);
      digitalWrite(BLU, LOW);
```

```
    }
    else if ((mass > R3_MIN) && (mass < R3_MAX)) {
      digitalWrite(RED, LOW);
      digitalWrite(GRN, LOW);
      digitalWrite(BLU, HIGH);
    }
  }
  cycledelay += 1;
  cycledelay %= 10;


  //Tare
  if (digitalRead(TOGGLE_TAR) == HIGH) {
    tare(data);
  }

  //Clear LCD
  if (digitalRead(TOGGLE_CLEAR) == HIGH) {
    lcd.clear();
  }

}

long read_data() {
  stage0.input(nau.getAverage(6));
  stage1.input(stage0.output());
  avg.input(stage1.output());
  return avg.mean() - offset;
}

void tare(long data) {
  offset = data;
  return;
}

double dig_to_mass(double v) {
  return v*216048 - 2.16;
}
```

- MATHEMATICAL MODEL USED TO CONVERT VOLTAGE TO MASS

```
double dig_to_analog(long data) {
  return data * (4.5 / 16777216);
}
```