Documentation

Project:

PC Controlled Power Supply

Names:

Matt Dumolt

Nolan Murray

Westley Wurscher

Table of Contents

Materials

Key:

- *E* : Everyone
- *M.D.* : Matt Dumolt
- *N.M.* : Nolan Murray
- *W.W.* : Westley Wurscher

Oregon State University

Introduction

Engineering Requirements for PC Controlled DC Power Supply

Base Engineering Requirements:

Customer Requirement: The power supply must have multiple channels.
Engineering Requirement: The system will supply at least two independent channels of voltage at least including the range of 2 to 14 volts.

Customer Requirement: The power supply needs to be powerful.
Engineering Requirement: The system will supply up to 1.5 A on each of its channels.

Customer Requirement: The system must be safe.
Engineering Requirement: The system will only use US standard plugins for connecting to external devices, will not allow any object with a diameter greater than 1mm to enter the enclosure, and will be disabled if more than 1 amp is drawn from the wall power or 1.5 amps from the output at anytime.

Customer Requirement: The system must be reliable.
Engineering Requirement: The voltages and currents displayed by the system (to user and reported via serial) must be within 5% or 0.1 volt of real values whichever is larger.

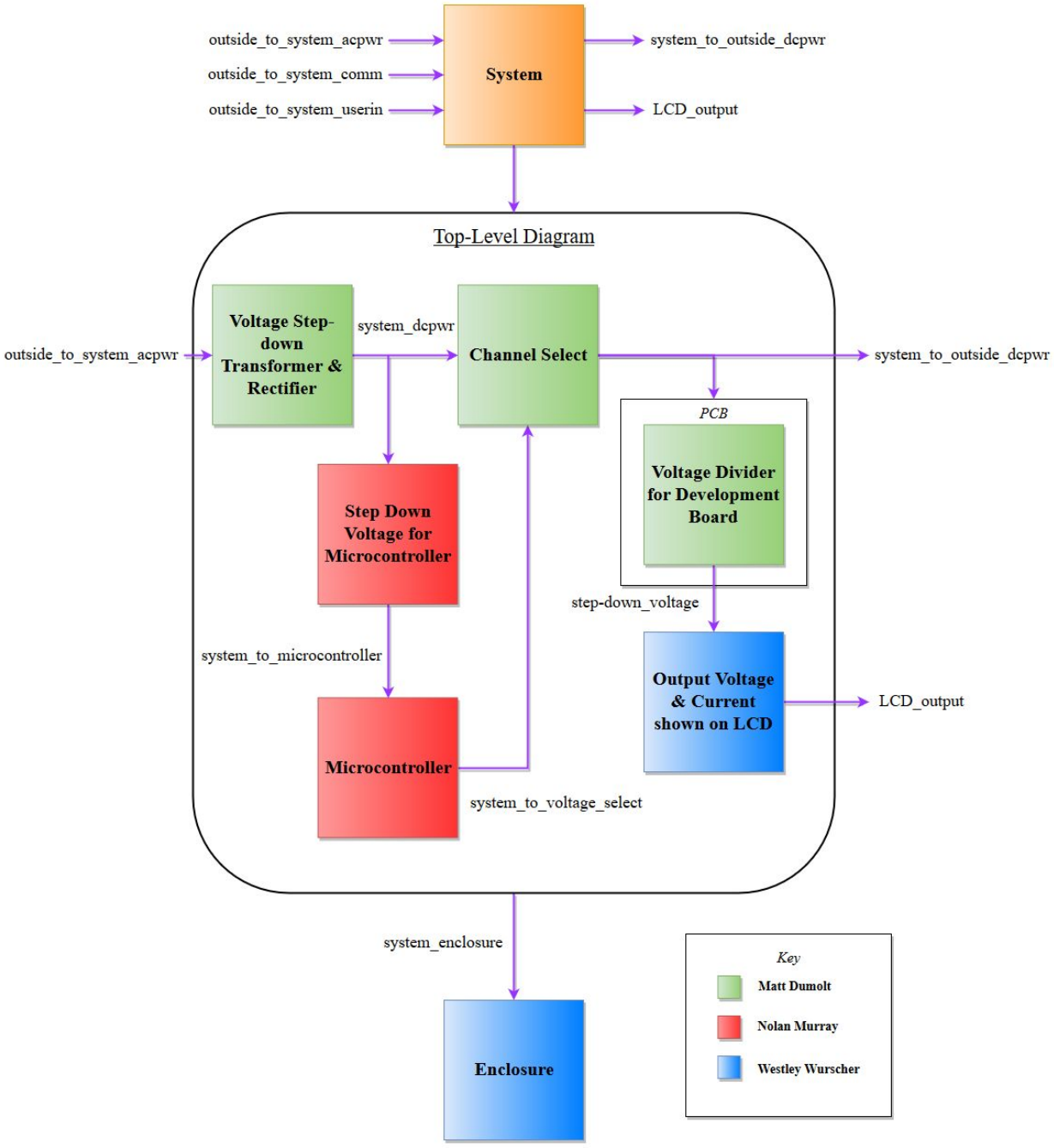
Customer Requirement: The power supply needs to be programmable.
Engineering Requirement: The system will be configurable for all channels over a serial interface using SCPI protocol that can adjust voltages and current limits.

Custom Engineering Requirements:

Customer Requirement: The power supply must show how it performs in real time.
Engineering Requirement: The system will house an LCD screen that will show voltage and current output, and two LED's for when power supply is operating normally and when disabled.

Customer Requirement: The power supply needs to be flexible in functionality.
Engineering Requirement: The system will have increased functionality by having four independent channels able to output voltage between 2 and 14V, following the tolerances of within 5% or .1V of real value, whichever is higher.

Block Diagram



Oregon State University

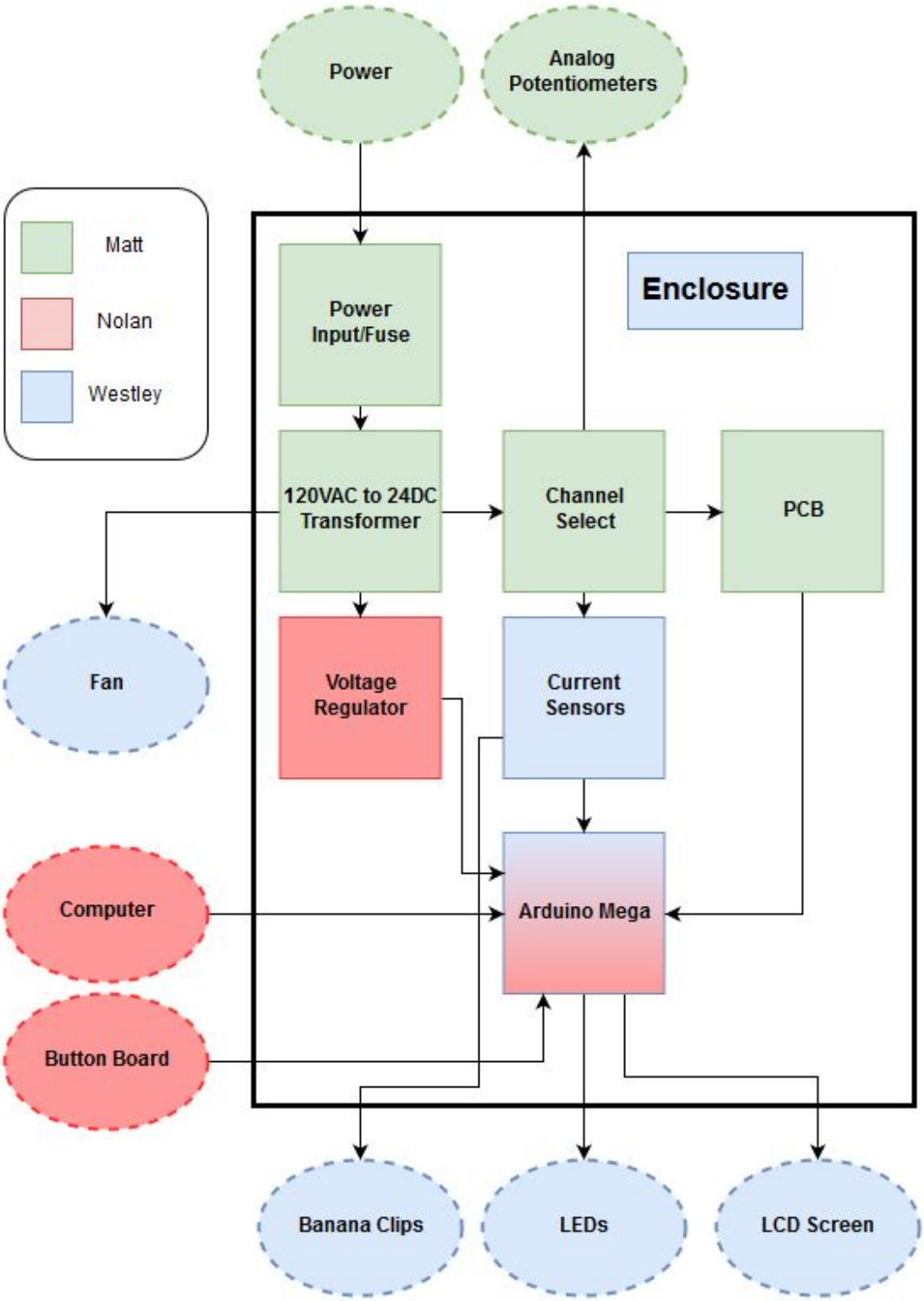
Interface Definitions

Interface	Properties
outside_to_system_acpwr	• 120 VAC. • NEMA-15M Connector. • I_{Peak}: 5A.
outside_to_system_comm	• SCPI protocol. • USB serial. • Speed ≥ 9600 Baud. • Commands: Set V and I, Read V and I, Enable/Disable Output.
outside_to_system_userin	• Enable/Disable Output. • Set V and I. • Display V and I within tolerance.
system_to_outside_dcpwr	• Voltage Ripple < 0.25VP-P. • Max Current is 1.5A. • Output Voltage is at least 2-14V, within 5% or .1V, whichever is larger. • Discharges To < 2V in under 5 sec.
system_dcpwr	• 24 VDC. • 24 volts will be spread to the fan, buck converters, and step-down voltage for microcontroller.
system_to_microcontroller	• Voltage input must between 7-12V for microcontroller to operate. • Analog Input conversion to digital.
system_to_voltage_select	• User selects the output voltage of each channel independently. • Channel selection determined by 4 digital bits. • Can be done on PC or button board.
step-down_voltage	• Use 100kΩ and 10kΩ resistors • Set up as a voltage divider with 10kΩ attached to ground. • Brings voltage down to 5 volts to not blow the development board.

LCD_output	• LCD reads and displays voltage output to within 5% or .1V of real value, whichever is larger. • LCD reads and displays current output to within 10% or 100mA, whichever is larger.
system_enclosure	• LED's for understanding the safe amount of current being drawn to the output. • Red LED lights up when more than 1.5A of current is drawn at the output. Green LED shines when less than 1.5A of current is drawn. • Both LEDs will be active high. • Have 5 buttons, 4 for voltage channels, and one for on and off.

Mechanical Drawings/Schematics

Overview of System with Enclosure Description



Purposes of Each Block

Power Input/Fuse Box:

This will be used to limit the current coming into the box, with the fuse in the casing. It also allows the user to turn on and off the entire power supply. The next components that connect to this the step down voltage transformer.

Transformer:

With 120VAC entering the power supply, a transformer will be used to step down the voltage to 24VDC. This way, the power supply can be operational to a point that can be using all components without wasting so much power. The next components this attaches to is the Voltage regulator, the PCB boards, and the fan.

Fan:

The fan is used to help with the airflow going through the enclosure. Allowing all components to be sufficiently cooled. This is an end system, so it doesn't go to any other blocks within the enclosure.

Voltage Regulator:

The voltage regulator takes the 24 volts and steps it down further for the Arduino Uno. This allows for the Arduino to not break when giving it to high of a voltage. Output from voltage regulator is to Arduino Uno.

Computer:

The computer role is to talk to the Arduino Uno. Giving the power supply certain tasks such as changing voltage or changing current on certain channels.

Button Board:

The button board role is the same as the computer. It gives the user another way to change values for the power supply output by communicating to the Arduino.

Channel Select:

This channel select will occur through four different buck converters. They will be connected to the Analog potentiometers.

Analog Potentiometers:

This is how we will be changing the voltage output on the benchmark power supply.

PCB:

The PCB is going to be the voltage divider that will then be hooked up into the sensors and the Arduino Mega because those will be used for the LCD display.

Current Sensors:

The input these are receiving are from the PCB, it will take this current reading, and translate to something the Arduino Mega can understand through analog pins, which is why its output is the to Arduino Mega.

Arduino Mega:

This development board will be taking in information from the channel select PCB and the current sensors through analog pin input. This is needed to then supply the LCD with correct voltage and output readings. The output node also goes to LED's. This will show if everything is in working order, or if something is faulty.

LCD Screen:

This will be displaying the voltage and current output, which is why it is connected to the Arduino Mega through the SCL and SDA pins on the board. And since this is an end system, nothing else will connect to this system.

Banana Clips:

The banana clips will be where we attach the proper outputs to measure voltage and current. This is getting input from the PCB boards as from that point on, the voltage and current from there will be the output for the system. Since this is an end system, it doesn't go to any other blocks within the enclosure.

Useful facts about Enclosure:

Where do we connect everything to ground?

A metal standard plate will be used where everything that needs ground will be connected to with connector rings. These are strapped down with nuts and bolts to keep everything secured.

What major connectors are being used?

A terminal block will be used to help deal with much of the wire management required for making sure all the components are getting the right power to the right places. I will also allow for a central hub for much of the power going around in the enclosure.

How will the enclosure be secured?

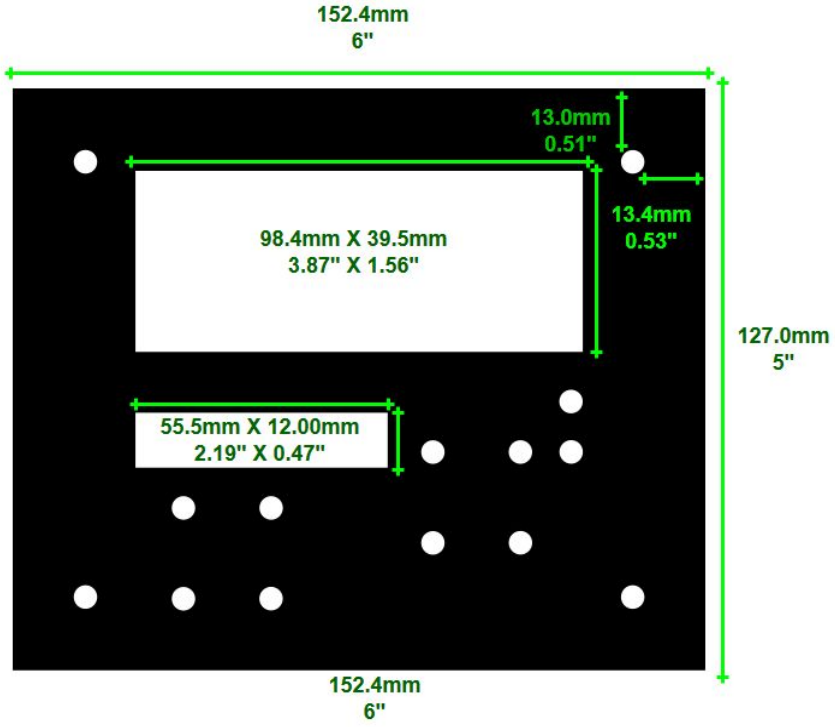
The enclosure will have 6 acrylic panels and 4 3-D printed sidings. The acrylic will be screwed to 5 of the acrylic panels, with the last panel being used as our sliding door for the entire enclosure. This sliding panel will be have a latch mechanism with the top panel that can be used to secure the sliding door in place while traveling.

How are you keeping everything secure inside of the enclosure?

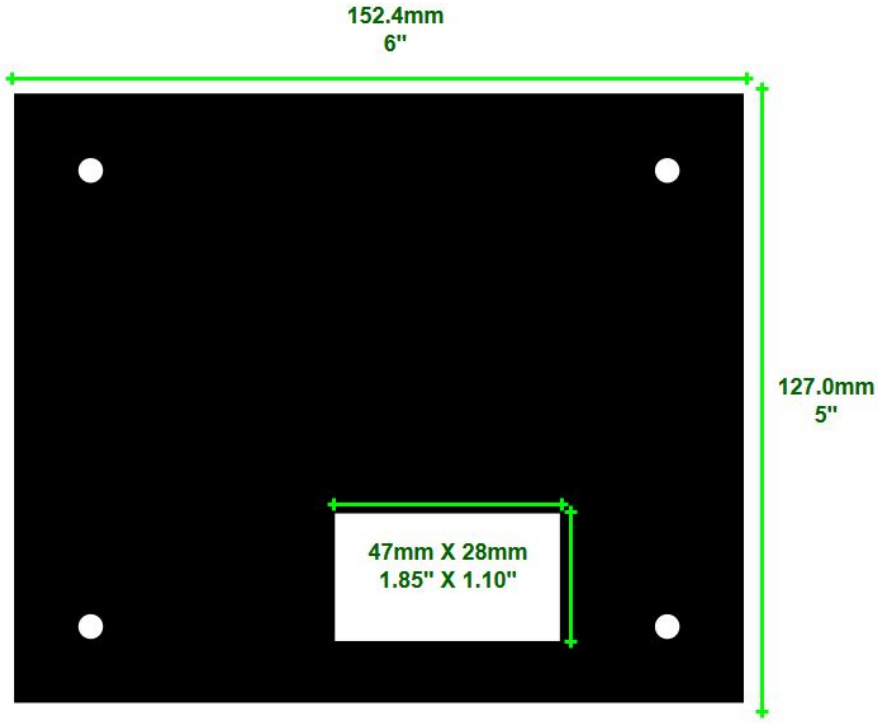
All components inside the enclosure will be held together with velcro.

Enclosure Dimensions/Design

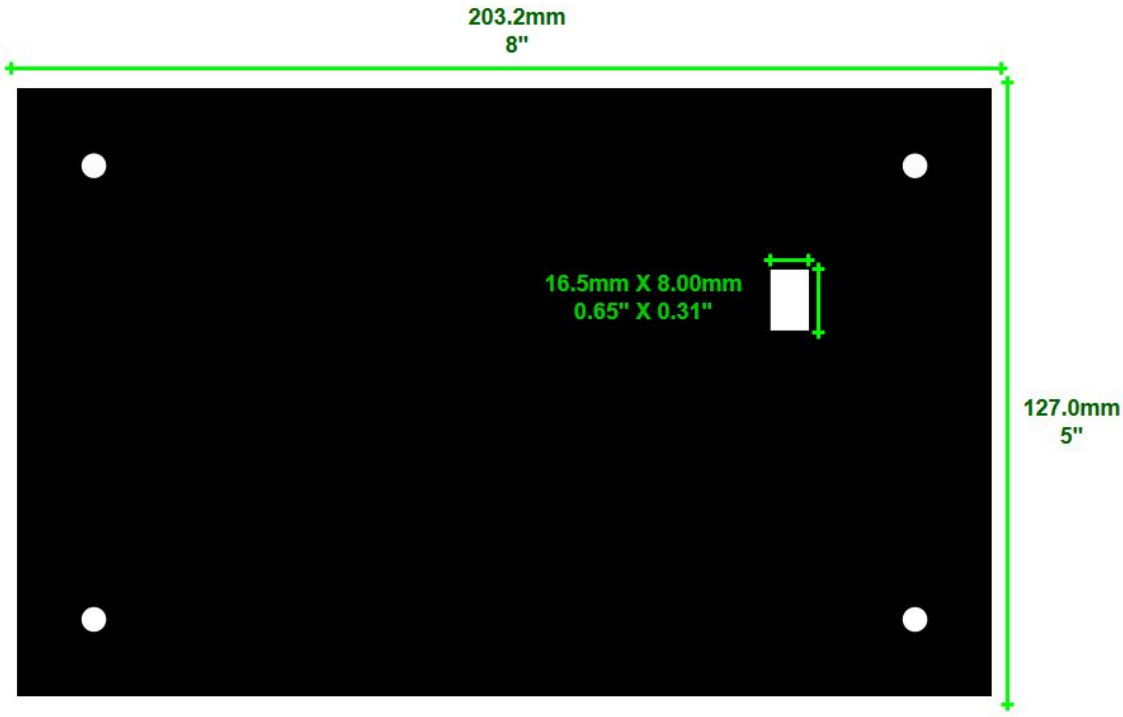
Front of the Enclosure



Back of the Enclosure

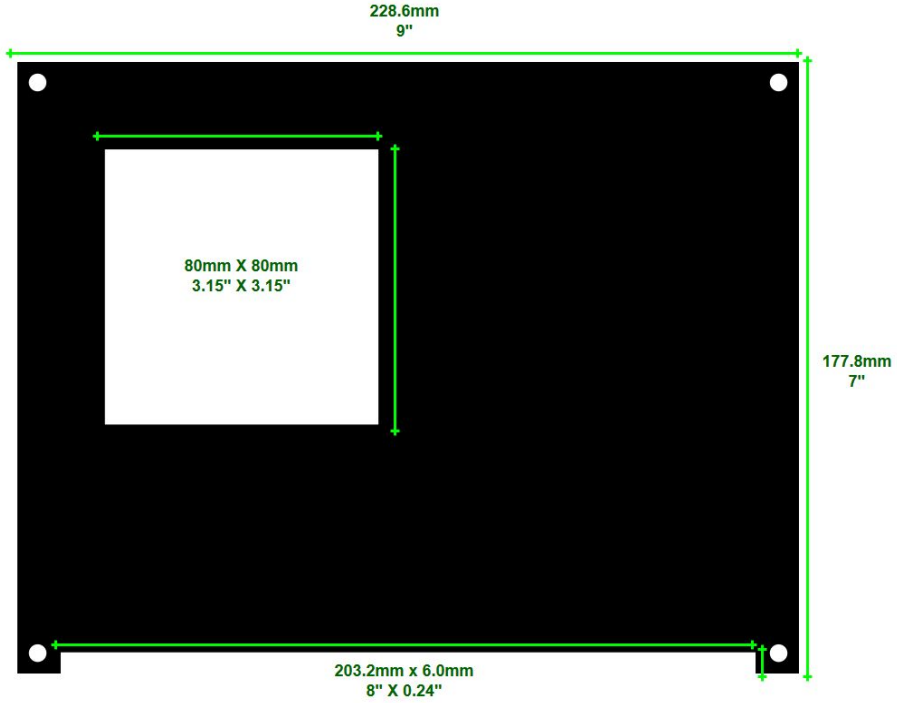


Left of the Enclosure



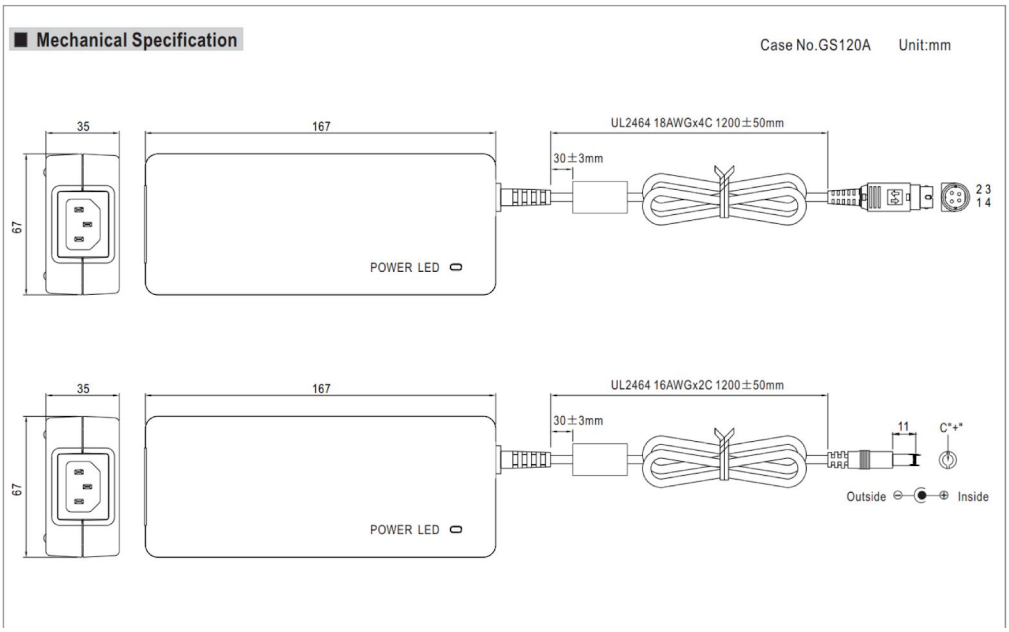
Right of the Enclosure

Top of the Enclosure

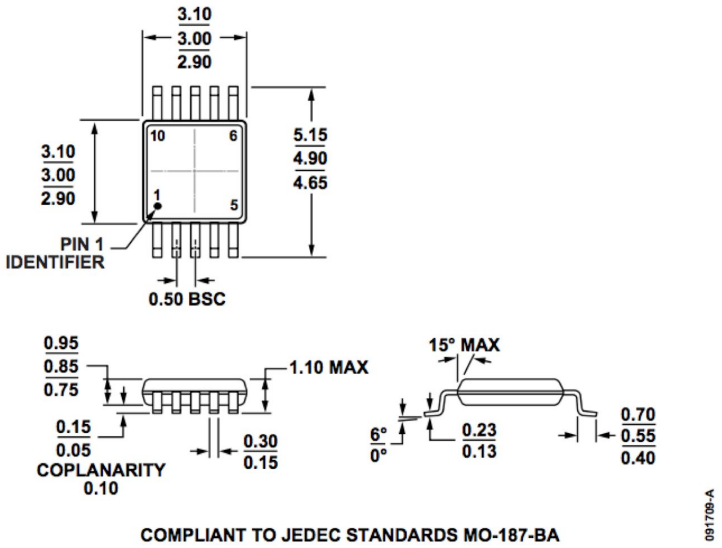


Bottom of the Enclosure

Mechanical Drawing of Transformer



Mechanical Drawing of Digital Potentiometer



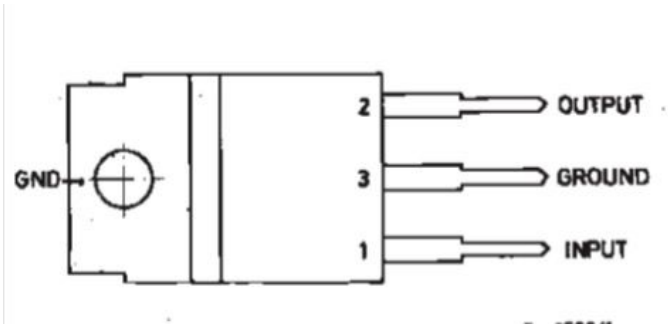
COMPLIANT TO JEDEC STANDARDS MO-187-BA

Figure 37. 10-Lead Mini Small Outline Package [MSOP]
(RM-10)
Dimensions shown in millimeters

Oregon State University

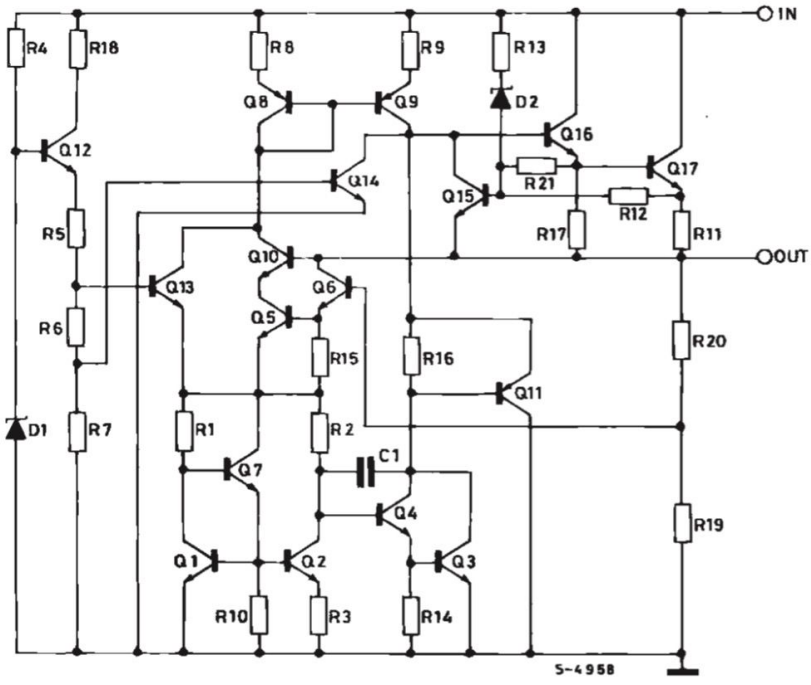
Mechanical Drawing of Step Down Voltage Regulator

L7809AVC Pinout

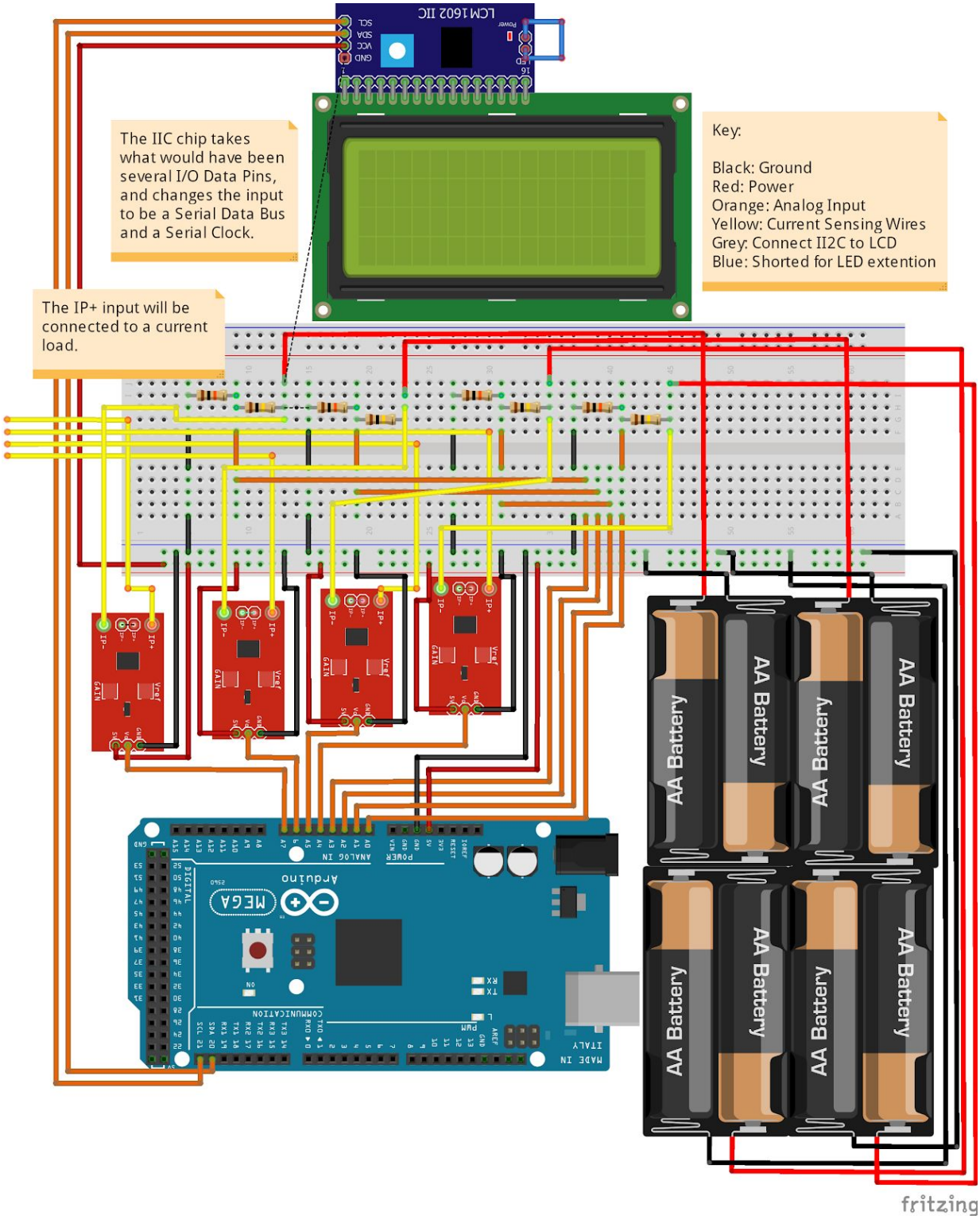


Schematic of Step Down Voltage Regulator

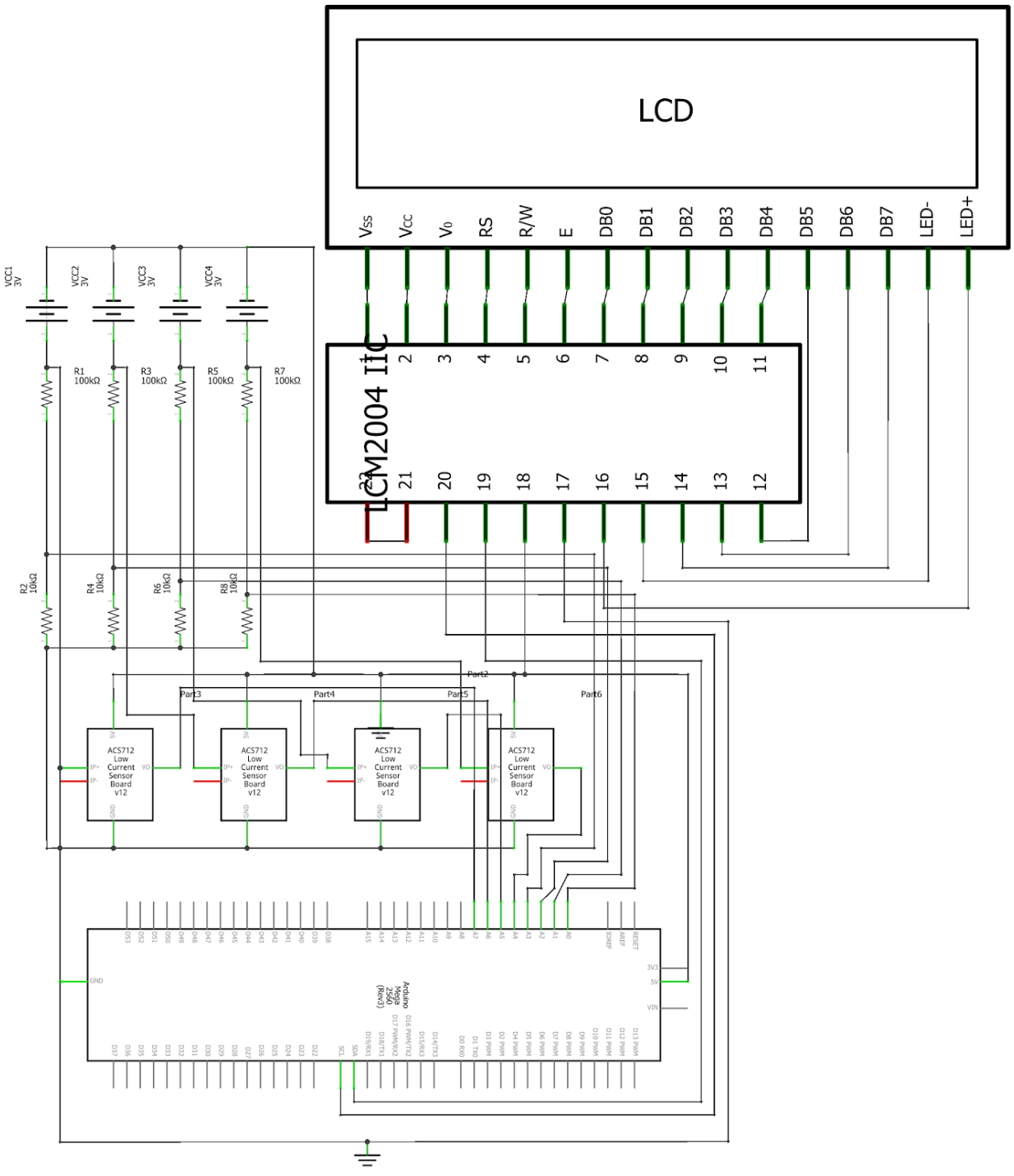
L7809ACV



Mechanical Drawing of Output/LCD



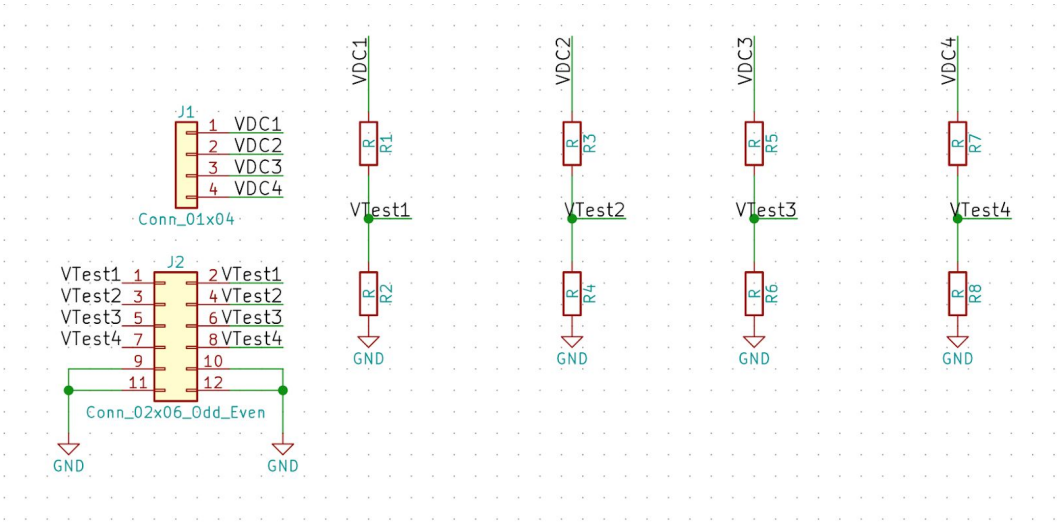
Schematic of Output/LCD



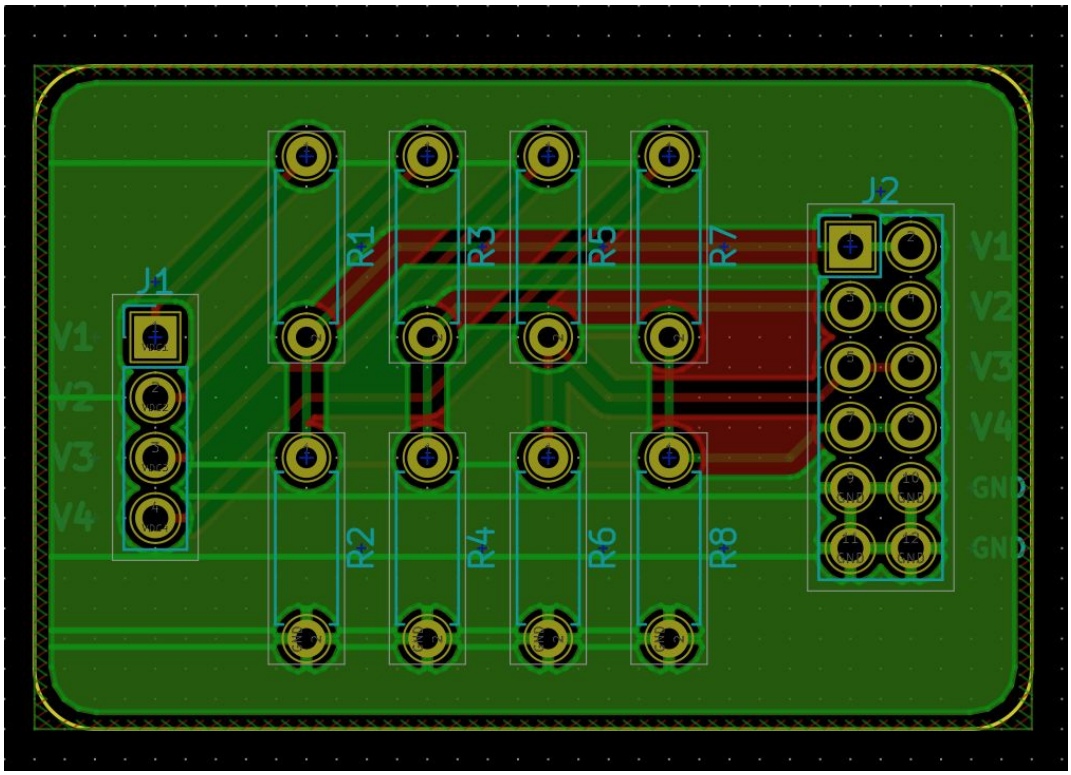
Printed Circuit Board (PCB)

Voltage Divider PCB Schematic and Layers

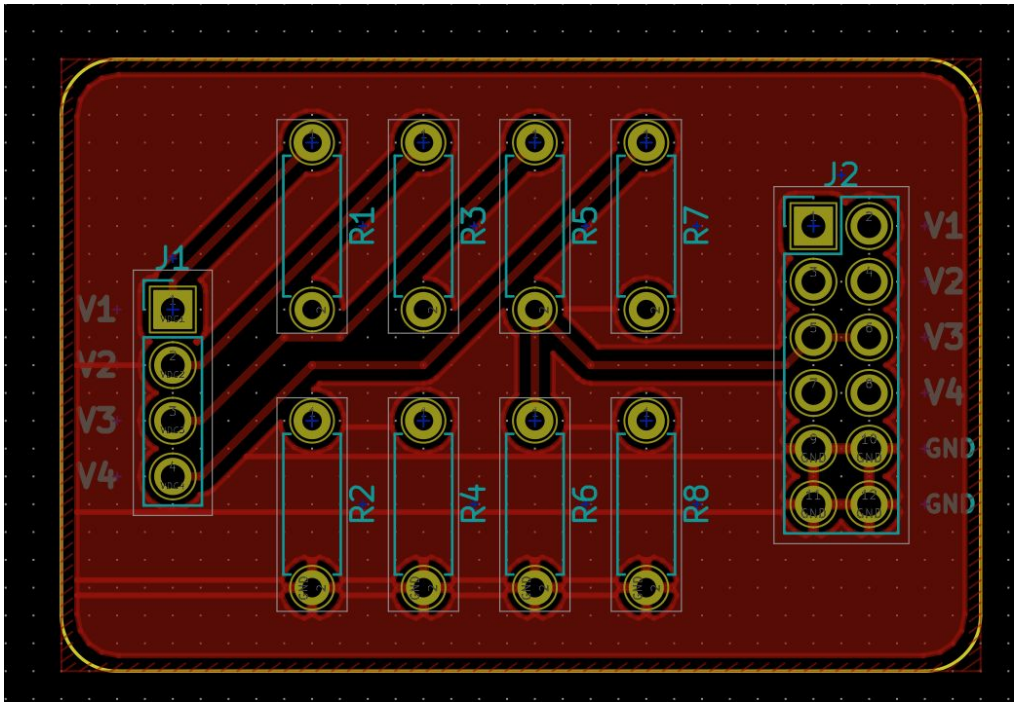
Schematic



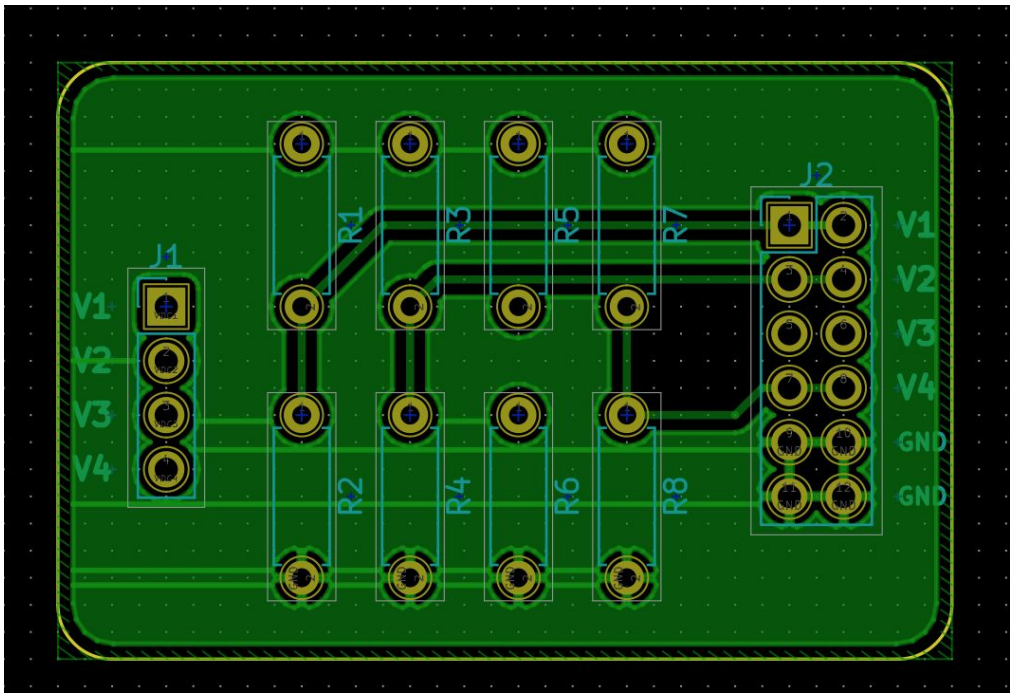
Printed Circuit Board (Whole)



Printed Circuit Board (Front)



Printed Circuit Board (Back)



Arduino Code

Code

```
//Libraries that have to be included
#include <Wire.h>
#include <SPI.h>
#include <LiquidCrystal_I2C.h>

//Define the backlight value of the LCD
#define LCD_Backlight 10
// NUM_SAMPLES can be up to 60, more does not fit into a unsigned integer.
#define NUM_SAMPLES 100
//Give LCD address
LiquidCrystal_I2C lcd(0x27, 2, 1, 0 , 4, 5, 6, 7, 3, POSITIVE);

int channel_1 = 10;
int channel_2 = 11;
int channel_3 = 12;
int channel_4 = 13;

int j1 = 2;
int j2 = 3;
int j3 = 4;
int j4 = 5;
int j5 = 6;
int j6 = 7;
int j7 = 8;
int j8 = 9;

int cs = 53;

float voltage;
float voltage_1;
float voltage_2;
float voltage_3;
float voltage_4;

float current;
float current_1;
float current_2;
float current_3;
float current_4;

enum operating_channel {CHANNEL_1, CHANNEL_2, CHANNEL_3, CHANNEL_4};
operating_channel curr = CHANNEL_1;
```

```
int x = 0;

//Input voltage variables
float input_voltage1, input_voltage2, input_voltage3, input_voltage4 = 0.000;
float input_current1, input_current2, input_current3, input_current4 = 0.000;

//temps for analog input
float temp1, temp2, temp3, temp4 = 0.0;

//resistor values on the breadboard
float r1_accurate = 99500.00;
float r2_accurate = 9690.00;
float r3_accurate = 100100.00;
float r4_accurate = 9910.00;
float r5_accurate = 99500.00;
float r6_accurate = 9700.00;
float r7_accurate = 99300.00;
float r8_accurate = 9670.00;

int sum = 0; // sum of samples taken
unsigned char sample_count = 0; // current sample number

void pc_input(){
  byte wholeNum;
  byte decimal;
  byte channel;


   if (Serial.available()==3){            //only read when two bytes have been sent
         channel = Serial.read();         //read first byte as channel select (1-4)
         wholeNum = Serial.read();        //read second byte as whole number voltage level (2-14)
         decimal = Serial.read();         //read third byte as decimal value for voltage level (0-9)

         if (channel < 49 || channel > 52){        //error message if channel input is out of range
               Serial.println("Please only select available channels (1-4)");
         }

         if (wholeNum < 97 || wholeNum > 109){ //error message if whole number input is out
of range
               Serial.println("Please select a whole number within range (2-14)");
         }

         if (decimal < 48 || decimal > 57){  //error message if decimal input is out of range
               Serial.println("Please select a decimal value within range (0-9)");
```

```
        }

        if (channel >= 49 && channel <=52){ //check that channel input is within range
        //if (wholeNum >= 97 && wholeNum <= 109){ //check that whole number voltage
input is within range
        if (decimal >= 48 && decimal <= 57){ //check that decimal voltage input is within
range
        if (channel == 49){                    //if channel 1 selected
        digitalWrite(cs, LOW);
        voltage_1 = (wholeNum)+((decimal-48.0)/10.0);
        voltage = voltage_1;
        //SPI.transfer(channel_1);
        //SPI.transfer(0);
        //analogWrite(channel_1, 100000000);      //set voltage of output channel 1
        digitalWrite(cs, HIGH);
        }

        if (channel == 50){                                  //if channel 2 selected
        digitalWrite(cs, LOW);
        voltage_2 = (wholeNum-95.0)+((decimal-48.0)/10.0);
        voltage = voltage_2;
        analogWrite(channel_2, voltage_2);  //set voltage of output channel 2
        digitalWrite(cs, HIGH);
        }

        if (channel == 51){                                  //if channel 3 selected
        digitalWrite(cs, LOW);
        voltage_3 = (wholeNum-95.0)+((decimal-48.0)/10.0);
        voltage = voltage_3;
        analogWrite(channel_3, voltage_3);  //set voltage of output channel 3
        digitalWrite(cs, HIGH);
        }

        if (channel == 52){                                  //if channel 4 selected
        digitalWrite(cs, LOW);
        voltage_4 = (wholeNum-95.0)+((decimal-48.0)/10.0);
        voltage = voltage_4;
        analogWrite(channel_4, voltage_4);  //set voltage of output channel 4
        digitalWrite(cs, HIGH);
        }
        }
        //}
        }
```

```
        //if (voltage >= 2.0 && voltage <= 14.0){                    //check final voltage
is within range
        Serial.print("User wants (");                    //print back what channel and
voltage user selected
        Serial.print(voltage);
        Serial.print(") volts on Channel #");
        Serial.println(channel-48);
        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
        //}
        //else{
        //Serial.println("There was an error with your input, please select values within range");
        //}
  }
}

void button_board() {
  digitalRead(j1);       //digital read pin that first bit of button board is connected to (0 or 1)
  digitalRead(j2);       //digital read pin that second bit of button board is connected to (0 or 1)
  digitalRead(j3);       //digital read pin that third bit of button board is connected to (0 or 1)
  digitalRead(j4);       //digital read pin that fourth bit of button board is connected to (0 or 1)
  digitalRead(j5);       //digital read pin that fifth bit of button board is connected to (0 or 1)
  digitalRead(j6);       //digital read pin that sixth bit of button board is connected to (0 or 1)
  digitalRead(j7);       //digital read pin that seventh bit of button board is connected to (0 or 1)
  digitalRead(j8);       //digital read pin that eighth bit of button board is connected to (0 or 1)

  while (digitalRead(j8) == LOW) {}

  if (x == 0){
        Serial.println("User selected Channel #1");
        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
```

```
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
        x++;
  }

  switch (curr) {
        case CHANNEL_1:
        ch_1();
        break;
        case CHANNEL_2:
        ch_2();
        break;
        case CHANNEL_3:
        ch_3();
        break;
        case CHANNEL_4:
        ch_4();
        break;
  }
}

void ch_1() {
  if (digitalRead(j7) == LOW) {
        while (digitalRead(j7) == LOW) {}
        digitalWrite(cs, LOW);
        voltage_1 = voltage_1 + 1.0;
        analogWrite(channel_1, voltage_1);
        digitalWrite(cs, HIGH);

        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
  }
  else if (digitalRead(j6) == LOW) {
        while (digitalRead(j6) == LOW) {}
        digitalWrite(cs, LOW);
        voltage_1 = voltage_1 - 1.0;
```

```
        analogWrite(channel_1, voltage_1);
        digitalWrite(cs, HIGH);

        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
}
else if (digitalRead(j5) == LOW) {
        while (digitalRead(j5) == LOW) {}
        digitalWrite(cs, LOW);
        voltage_1 = voltage_1 + 0.1;
        analogWrite(channel_1, voltage_1);
        digitalWrite(cs, HIGH);

        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
}
else if (digitalRead(j4) == LOW) {
        while (digitalRead(j4) == LOW) {}
        digitalWrite(cs, LOW);
        voltage_1 = voltage_1 - 0.1;
        analogWrite(channel_1, voltage_1);
        digitalWrite(cs, HIGH);

        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
```

```
            Serial.println(voltage_3);
            Serial.print("V4: ");
            Serial.println(voltage_4);
            Serial.println();
  }
//else if (digitalRead(j3) == LOW){
//  while (digitalRead(j3) == LOW) {}
//  current_1 = current_1 + 0.1;
//}
//else if (digitalRead(j2) == LOW){
//  while (digitalRead(j2) == LOW) {}
//  current_1 = current_1 - 0.1;
//}
  else if (digitalRead(j8) == LOW){
            curr = CHANNEL_2;
            Serial.println();
            Serial.println("User selected Channel #2");
            Serial.println();
            Serial.print("V1: ");
            Serial.println(voltage_1);
            Serial.print("V2: ");
            Serial.println(voltage_2);
            Serial.print("V3: ");
            Serial.println(voltage_3);
            Serial.print("V4: ");
            Serial.println(voltage_4);
            Serial.println();
  }
}

void ch_2() {
  if (digitalRead(j7) == LOW) {
            while (digitalRead(j7) == LOW) {}
            digitalWrite(cs, LOW);
            voltage_2 = voltage_2 + 1.0;
            analogWrite(channel_2, voltage_2);
            digitalWrite(cs, HIGH);

            Serial.println();
            Serial.print("V1: ");
            Serial.println(voltage_1);
            Serial.print("V2: ");
            Serial.println(voltage_2);
            Serial.print("V3: ");
            Serial.println(voltage_3);
```

```
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
}
else if (digitalRead(j6) == LOW) {
        while (digitalRead(j6) == LOW) {}
        digitalWrite(cs, LOW);
        voltage_2 = voltage_2 - 1.0;
        analogWrite(channel_2, voltage_2);
        digitalWrite(cs, HIGH);

        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
}
else if (digitalRead(j5) == LOW) {
        while (digitalRead(j5) == LOW) {}
        digitalWrite(cs, LOW);
        voltage_2 = voltage_2 + 0.1;
        analogWrite(channel_2, voltage_2);
        digitalWrite(cs, HIGH);

        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
}
else if (digitalRead(j4) == LOW) {
        while (digitalRead(j4) == LOW) {}
        digitalWrite(cs, LOW);
        voltage_2 = voltage_2 - 0.1;
        analogWrite(channel_2, voltage_2);
```

```
        digitalWrite(cs, HIGH);

        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
  }
//else if (digitalRead(j3) == LOW){
//  while (digitalRead(j3) == LOW) {}
//  current_2 = current_2 + 0.1;
//}
//else if (digitalRead(j2) == LOW){
//  while (digitalRead(j2) == LOW) {}
//  current_2 = current_2 - 0.1;
//}
  else if (digitalRead(j8) == LOW){
        curr = CHANNEL_3;
        Serial.println();
        Serial.println("User selected Channel #3");
        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
  }
}

void ch_3() {
  if (digitalRead(j7) == LOW) {
        while (digitalRead(j7) == LOW) {}
        digitalWrite(cs, LOW);
        voltage_3 = voltage_3 + 1.0;
        analogWrite(channel_3, voltage_3);
        digitalWrite(cs, HIGH);
```

```
        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
}
else if (digitalRead(j6) == LOW) {
        while (digitalRead(j6) == LOW) {}
        digitalWrite(cs, LOW);
        voltage_3 = voltage_3 - 1.0;
        analogWrite(channel_3, voltage_3);
        digitalWrite(cs, HIGH);

        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
}
else if (digitalRead(j5) == LOW) {
        while (digitalRead(j5) == LOW) {}
        digitalWrite(cs, LOW);
        voltage_3 = voltage_3 + 0.1;
        analogWrite(channel_3, voltage_3);
        digitalWrite(cs, HIGH);

        Serial.println();
        Serial.print("V1: ");
        Serial.println(voltage_1);
        Serial.print("V2: ");
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
```

```
            Serial.println(voltage_4);
            Serial.println();
    }
    else if (digitalRead(j4) == LOW) {
            while (digitalRead(j4) == LOW) {}
            digitalWrite(cs, LOW);
            voltage_3 = voltage_3 - 0.1;
            analogWrite(channel_3, voltage_3);
            digitalWrite(cs, HIGH);

            Serial.println();
            Serial.print("V1: ");
            Serial.println(voltage_1);
            Serial.print("V2: ");
            Serial.println(voltage_2);
            Serial.print("V3: ");
            Serial.println(voltage_3);
            Serial.print("V4: ");
            Serial.println(voltage_4);
            Serial.println();
    }
  //else if (digitalRead(j3) == LOW){
  //  while (digitalRead(j3) == LOW) {}
  //  current_3 = current_3 + 0.1;
  //}
  //else if (digitalRead(j2) == LOW){
  //  while (digitalRead(j2) == LOW) {}
  //  current_3 = current_3 - 0.1;
  //}
    else if (digitalRead(j8) == LOW){
            curr = CHANNEL_4;
            Serial.println();
            Serial.println("User selected Channel #4");
            Serial.println();
            Serial.print("V1: ");
            Serial.println(voltage_1);
            Serial.print("V2: ");
            Serial.println(voltage_2);
            Serial.print("V3: ");
            Serial.println(voltage_3);
            Serial.print("V4: ");
            Serial.println(voltage_4);
            Serial.println();
    }
  }
```

```
void ch_4() {
  if (digitalRead(j7) == LOW) {
          while (digitalRead(j7) == LOW) {}
          digitalWrite(cs, LOW);
          voltage_4 = voltage_4 + 1.0;
          analogWrite(channel_4, voltage_4);
          digitalWrite(cs, HIGH);

          Serial.println();
          Serial.print("V1: ");
          Serial.println(voltage_1);
          Serial.print("V2: ");
          Serial.println(voltage_2);
          Serial.print("V3: ");
          Serial.println(voltage_3);
          Serial.print("V4: ");
          Serial.println(voltage_4);
          Serial.println();
  }
  else if (digitalRead(j6) == LOW) {
          while (digitalRead(j6) == LOW) {}
          digitalWrite(cs, LOW);
          voltage_4 = voltage_4 - 1.0;
          analogWrite(channel_4, voltage_4);
          digitalWrite(cs, HIGH);

          Serial.println();
          Serial.print("V1: ");
          Serial.println(voltage_1);
          Serial.print("V2: ");
          Serial.println(voltage_2);
          Serial.print("V3: ");
          Serial.println(voltage_3);
          Serial.print("V4: ");
          Serial.println(voltage_4);
          Serial.println();
  }
  else if (digitalRead(j5) == LOW) {
          while (digitalRead(j5) == LOW) {}
          digitalWrite(cs, LOW);
          voltage_4 = voltage_4 + 0.1;
          analogWrite(channel_4, voltage_4);
          digitalWrite(cs, HIGH);
```

```
            Serial.println();
            Serial.print("V1: ");
            Serial.println(voltage_1);
            Serial.print("V2: ");
            Serial.println(voltage_2);
            Serial.print("V3: ");
            Serial.println(voltage_3);
            Serial.print("V4: ");
            Serial.println(voltage_4);
            Serial.println();
  }
  else if (digitalRead(j4) == LOW) {
            while (digitalRead(j4) == LOW) {}
            digitalWrite(cs, LOW);
            voltage_4 = voltage_4 - 0.1;
            analogWrite(channel_4, voltage_4);
            digitalWrite(cs, HIGH);

            Serial.println();
            Serial.print("V1: ");
            Serial.println(voltage_1);
            Serial.print("V2: ");
            Serial.println(voltage_2);
            Serial.print("V3: ");
            Serial.println(voltage_3);
            Serial.print("V4: ");
            Serial.println(voltage_4);
            Serial.println();
  }
//else if (digitalRead(j3) == LOW){
//  while (digitalRead(j3) == LOW) {}
//  current_4 = current_4 + 0.1;
//}
//else if (digitalRead(j2) == LOW){
//  while (digitalRead(j2) == LOW) {}
//  current_4 = current_4 - 0.1;
//}
  else if (digitalRead(j8) == LOW){
            curr = CHANNEL_1;
            Serial.println();
            Serial.println("User selected Channel #1");
            Serial.println();
            Serial.print("V1: ");
            Serial.println(voltage_1);
            Serial.print("V2: ");
```

```
        Serial.println(voltage_2);
        Serial.print("V3: ");
        Serial.println(voltage_3);
        Serial.print("V4: ");
        Serial.println(voltage_4);
        Serial.println();
  }
}

// Read voltage from a certain analog pin.
float read_volt(int pin)
{
  unsigned int total = 0;          // start with zero, this is used for the total value.
  for (int i = 0; i < NUM_SAMPLES; i++) // loop through reading raw adc values
NUM_SAMPLES number of times
  {
  total += analogRead(pin);    // read the input pin

  // The delay is arbitrary.
  // It can be 0 to 20ms.
  // Test for your own circuit what is best.
  delayMicroseconds(50);                // pauses for 50 microseconds
  }
  // The voltage is raw_adc / 1023 * 5V and also divided by the amount of samples for
average.
  // Convert every integer to float before calculating something.
  // This way, the extra bits gained by averaging are used in the calculation.
  float volt = (float) total / 1023.0 * 5.0 / (float) NUM_SAMPLES;
  return volt;
}
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  pinMode(channel_1, OUTPUT);     //initialize channel 1 pin as output
  pinMode(channel_2, OUTPUT);     //initialize channel 2 pin as output
  pinMode(channel_3, OUTPUT);     //initialize channel 3 pin as output
  pinMode(channel_4, OUTPUT);     //initialize channel 4 pin as output

  pinMode(cs, OUTPUT);

  SPI.begin();

  pinMode(j1, INPUT_PULLUP);              //initialize 1st bit pin as input
  pinMode(j2, INPUT_PULLUP);              //initialize 2nd bit pin as input
```

```
  pinMode(j3, INPUT_PULLUP);              //initialize 3rd bit pin as input
  pinMode(j4, INPUT_PULLUP);              //initialize 4th bit pin as input
  pinMode(j5, INPUT_PULLUP);              //initialize 5th bit pin as input
  pinMode(j6, INPUT_PULLUP);              //initialize 6th bit pin as input
  pinMode(j7, INPUT_PULLUP);              //initialize 7th bit pin as input
  pinMode(j8, INPUT_PULLUP);              //initialize 8th bit pin as input
  //Set size of LCD
  lcd.begin(20, 4);
  //Set backlight
  pinMode(LCD_Backlight, OUTPUT);
  // Set the brightness of the backlight
  analogWrite(LCD_Backlight, 128);
  //Sets the format of the screen
  lcd.setCursor(0, 0);
  lcd.print("V1:");
  lcd.setCursor(0, 1);
  lcd.print("I1:");
  lcd.setCursor(0, 2);
  lcd.print("V2:");
  lcd.setCursor(0, 3);
  lcd.print("I2:");
  lcd.setCursor(10, 0);
  lcd.print("V3:");
  lcd.setCursor(10, 1);
  lcd.print("I3:");
  lcd.setCursor(10, 2);
  lcd.print("V4:");
  lcd.setCursor(10, 3);
  lcd.print("I4:");
  lcd.setCursor(9, 0);
  lcd.print("|");
  lcd.setCursor(9, 1);
  lcd.print("|");
  lcd.setCursor(9, 2);
  lcd.print("|");
  lcd.setCursor(9, 3);
  lcd.print("|");
}

void loop() {
  pc_input();
  button_board();
//Taking the first analog input and doing a number of samples to get a better
representation of voltage output
  while (sample_count < NUM_SAMPLES) {
```

```
sum += analogRead(A0);
sample_count++;
delay(10);
}
//read in the analog inputs
//int analog_value1 = analogRead(A0);
int analog_value2 = analogRead(A1);
int analog_value3 = analogRead(A2);
int analog_value4 = analogRead(A3);
//Serial.println(analog_value2);
//Calibrate for setting up input voltage
temp1 = ((float)sum / (float)NUM_SAMPLES * 5.0) / 1024.0;
//Serial.println("Voltage & Current Readings");
temp2 = (analog_value2 * 5.0) / 1024.0;
temp3 = (analog_value3 * 5.0) / 1024.0;
temp4 = (analog_value4 * 5.0) / 1024.0;
//input voltage value
input_voltage1 = temp1 / (r2_accurate / (r1_accurate + r2_accurate));
input_voltage2 = temp2 / (r4_accurate / (r3_accurate + r4_accurate));
input_voltage3 = temp3 / (r6_accurate / (r5_accurate + r6_accurate));
input_voltage4 = temp4 / (r8_accurate / (r7_accurate + r8_accurate));
// The ASC712-30 has 66mV/A = 0.066V/A
// That is 1 amp for every 0.066V.
// The middle of 2.5V is zero amps.
//These are reading the read_volt required for the ASC172
float volt_current1 = read_volt(A4);
float volt_current2 = read_volt(A5);
float volt_current3 = read_volt(A6);
float volt_current4 = read_volt(A7);
//input current values
float input_current1 = (volt_current1 - 2.5) / 0.066;
float input_current2 = (volt_current2 - 2.5) / 0.066;
float input_current3 = (volt_current3 - 2.5) / 0.066;
float input_current4 = (volt_current4 - 2.5) / 0.066;

delay(500);
////////////////////////////////////
//Voltage and Current 1
////////////////////////////////////
//if the input voltage is very small
if (input_voltage1 < 0.1)
{
input_voltage1 = 0.00;
}
//Serial.print("V1: ");
```

```
  //Serial.print(input_voltage1); //print the voltage to the serial screen
  lcd.setCursor(3, 0);
  lcd.print(input_voltage1); //print voltage on lcd
  lcd.setCursor(3, 1); //Set Cursor to new line
  //Serial.print(" I1: "); //print current on serial screen
  //Serial.println(input_current1, 3); //print current on serial screen
  lcd.print(input_current1, 3); //print the current on lcd
  ////////////////////////////////////
  //Voltage and Current 2
  ////////////////////////////////////
  //Every section of voltage and current readings are the same format for code, just with different
variables
  if (input_voltage2 < 0.1) {
  input_voltage2 = 0.0;
  }
  //Serial.print("V2: ");
 // Serial.print(input_voltage2);
  lcd.setCursor(3, 2);
  lcd.print(input_voltage2);
  lcd.setCursor(3, 3);
  //Serial.print(" I2: ");
  //Serial.println(input_current2, 3);
  lcd.print(input_current2, 3);
  ////////////////////////////////////
  //Voltage and Current 3
  ////////////////////////////////////
  if (input_voltage3 < 0.1) {
  input_voltage3 = 0.0;
  }
  //Serial.print("V3: ");
  //Serial.print(input_voltage3);
  lcd.setCursor(13, 0);
  lcd.print(input_voltage3);
  lcd.setCursor(13, 1);
  //Serial.print(" I3: ");
  //Serial.println(input_current3, 3);
  lcd.print(input_current3, 3);
  ////////////////////////////////////
  //Voltage and Current 4
  ////////////////////////////////////
  if (input_voltage4 < 0.1) {
  input_voltage4 = 0.0;
  }
  //Serial.print("V4: ");
  //Serial.print(input_voltage4);
```

```
  lcd.setCursor(13, 2);
  lcd.print(input_voltage4);
  lcd.setCursor(13, 3);
  lcd.print(input_current4, 3);
  //Serial.print(" I4: ");
  //Serial.println(input_current4, 3);
  sample_count = 0;
  sum = 0;
  //Serial.println("");
}
```

Interface & Properties with Testing Process

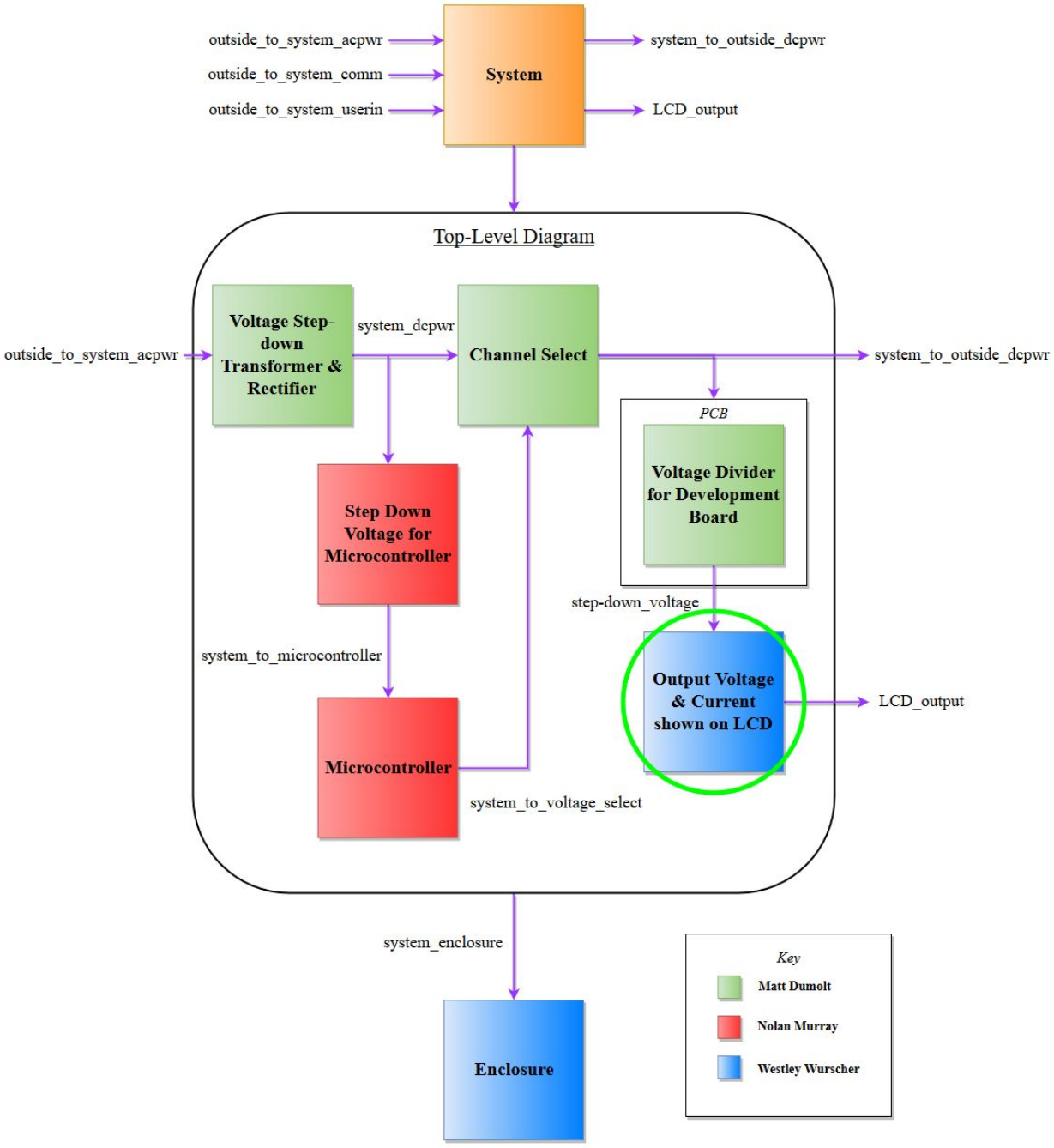
L7809 Step Down Voltage Regulator Interface Definitions & Testing Steps

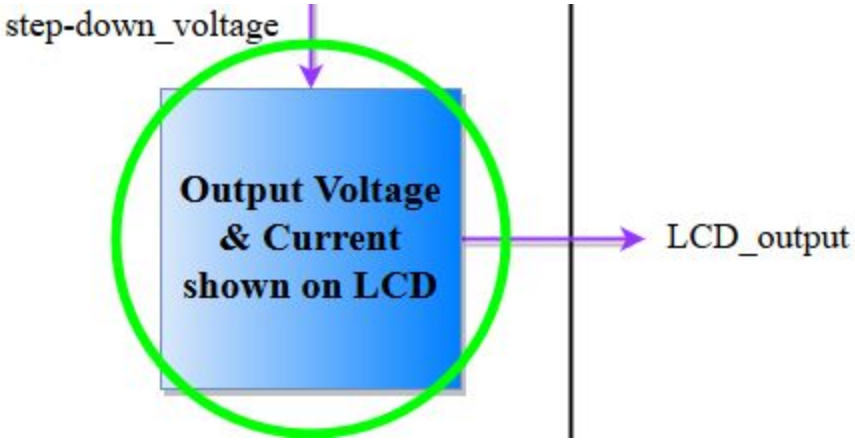
- system_dcpwr is ~24Vdc coming from the step down transformer and rectifier with a maximum current of 1A that needs to be lowered to ~7-12Vdc

- system_to_microcontroller is the 7-12Vdc coming from this step down to be used to power the Arduino Uno which draws ~50mA of current

- The L7809ACV takes a maximum of 35V at the input
- This component will output ~9V and 1A of current
- The electronic load will be used to draw ~150-200mA from the output of the voltage regulator to simulate the Arduino Uno drawing ~50mA
- The function generator will be used to supply 24Vdc to the input of the voltage regulator
- The electronic load will display the output voltage to be ~9V

Testing for Voltage and Current Output





Interface	Properties
step-down_voltage	• Use 100kΩ and 10kΩ resistors • Set up as a voltage divider with 10kΩ attached to ground. • Brings voltage down to 5 volts to not blow the development board.
LCD_output	• LCD reads and displays voltage output to within 5% or .1V of real value, whichever is larger. • LCD reads and displays current output to within 10% or 100mA, whichever is larger.

Tools:

To do the testing required for this part of the project, several tools are going to be needed including:

- Power Supply Brick
- Current Load Box or Multimeter
- The Arduino and LCD screen kit
- Test Leads
- Personal Computer(Laptop)

Preparation:

Before going any further, several things will need to be completed:

- Arduino software must be downloaded on the computer
- Circuit built completely on the breadboard
- Make sure computer has the right code compiled on the Arduino before running.

Caution to know before testing:

If any testing point is giving an output current higher then 1.5 Amps, turn off power supply completely and ask for help if problem cannot be found.

Testing Steps:

To start the testing, we will turn the power supply onto 2 volts on the first channel:

1. Record Input voltage through power supply
2. Record output voltage value on LCD screen.
3. Record output voltage value given through current load box.
4. Record the current value shown on the LCD.
5. Turn off power supply and wait for 4 seconds
6. Repeat steps 1 through 5, increasing the output voltage every 2 volts until the last voltage point tested is at 14 volts (peak voltage).
7. Repeat these 6 steps with the new channel, testing all 4 channels.
8. Calculate the percent error for voltage values and current values using output from current load box as true value (theoretical).

Voltage Reading

Channel 1

Input Voltage with no Current Load	Output Voltage on LCD with no Current Load	Output Voltage from current load box	Input Voltage with 1.5A Load	Output Voltage on LCD with 1.5A Load	Voltage Reading from current load
2.00	~1.900	1.996	2.00	~1.720	1.270
4.00	~3.95	3.990	4.00	~3.780	3.279
6.00	~5.990	6.01	6.00	~5.830	5.29
8.00	~8.030	8.00	8.00	~7.880	7.28
10.00	~10.090	10.01	10.00	~9.940	9.29
12.00	~12.150	12.00	12.00	~11.990	11.30
14.00	~14.200	13.99	14.00	~14.030	13.29

Channel 2

Input Voltage with no Current Load	Output Voltage on LCD with no Current Load	Output Voltage from current load box	Input Voltage with 1.5A Load	Output Voltage on LCD with 1.5A Load	Voltage Reading from current load
2.00	~1.960	1.993	2.00	~1.710	1.122
4.00	~4.050	3.996	4.00	~3.817	3.046
6.00	~6.140	6.00	6.00	~5.920	5.02
8.00	~8.217	8.00	8.00	~7.982	7.00
10.00	~10.320	10.00	10.00	~10.100	8.98
12.00	~12.420	11.99	12.00	~12.176	10.95
14.00	~14.510	13.98	14.00	~14.262	12.90

Channel 3

Input Voltage with no Current Load	Output Voltage on LCD with no Current Load	Output Voltage from current load box	Input Voltage with 1.5A Load	Output Voltage on LCD with 1.5A Load	Voltage Reading from current load
2.00	~1.940	2.006	2.00	~1.760	1.200
4.00	~4.000	4.00	4.00	~3.840	3.194
6.00	~6.050	6.02	6.00	~5.890	5.19
8.00	~8.130	8.02	8.00	~7.970	7.19
10.00	~10.210	10.02	10.00	~10.050	9.20
12.00	~12.270	12.00	12.00	~12.100	11.20
14.00	~14.340	14.02	14.00	~14.170	13.19

Channel 4

Input Voltage with no Current Load	Output Voltage on LCD with no Current Load	Output Voltage from current load box	Input Voltage with 1.5A Load	Output Voltage on LCD with 1.5A Load	Voltage Reading from current load
2.00	~1.900	1.994	2.00	~1.730	0.990
4.00	~3.960	3.984	4.00	~3.790	3.027
6.00	~6.020	6.00	6.00	~5.840	5.02
8.00	~8.080	8.01	8.00	~7.910	7.03
10.00	~10.150	10.01	10.00	~9.970	9.01
12.00	~12.210	12.00	12.00	~12.040	11.00
14.00	~14.260	13.99	14.00	~14.080	13.00

Current Reading

Channel 1

Input Voltage	Current from load	Current displayed on power supply	Current on LCD
2	1.500	1.510	~1.540
4	1.500	1.510	~1.540
6	1.500	1.508	~1.540
8	1.500	1.508	~1.540
10	1.500	1.508	~1.540
12	1.500	1.508	~1.540
14	1.500	1.508	~1.540

Channel 2

Input Voltage	Current from load	Current displayed on power supply	Current on LCD
2	1.500	1.505	~1.450
4	1.500	1.505	~1.440
6	1.500	1.503	~1.420
8	1.500	1.504	~1.380
10	1.500	1.504	~1.400
12	1.500	1.504	~1.400
14	1.500	1.503	~1.400

Channel 3

Input Voltage	Current from load	Current displayed on power supply	Current on LCD
2	1.500	1.505	~1.450
4	1.500	1.505	~1.440
6	1.500	1.504	~1.440
8	1.500	1.504	~1.450
10	1.500	1.504	~1.450
12	1.500	1.504	~1.460
14	1.500	1.503	~1.470

Channel 4

Input Voltage	Current from load	Current displayed on power supply	Current on LCD
2	1.500	1.505	~1.460
4	1.500	1.505	~1.450
6	1.500	1.504	~1.450
8	1.500	1.503	~1.440
10	1.500	1.503	~1.440
12	1.500	1.503	~1.450
14	1.500	1.503	~1.450

Percent Error Calculation

Channel 1

Input Voltage	Output Voltage With No Current Load (Theoretical from Current Load Box)	Output Voltage With Current Load (Theoretical from Current Load Box)	Output Current Percentage
2	4.80%	35.43%	1.99%
4	1.00%	13.25%	1.99%
6	0.33%	10.21%	2.12%
8	0.38%	8.24%	2.12%
10	0.80%	7.00%	2.12%
12	1.25%	6.11%	2.12%
14	1.50%	5.57%	2.12%

Channel 2

Input Voltage	Output Voltage With No Current Load (Theoretical from Current Load Box)	Output Voltage With Current Load (Theoretical from Current Load Box)	Output Current Percentage
2	1.66%	52.41%	3.65%
4	1.35%	25.31%	4.32%
6	2.33%	17.93%	5.52%
8	2.71%	14.03%	8.24%
10	3.20%	12.47%	6.91%
12	3.59%	11.20%	6.91%
14	3.79%	10.56%	6.85%

Channel 3

Input Voltage	Output Voltage With No Current Load (Theoretical from Current Load Box)	Output Voltage With Current Load (Theoretical from Current Load Box)	Output Current Percentage
2	3.29%	46.67%	3.65%
4	0.00%	20.23%	4.32%
6	0.50%	13.49%	4.26%
8	1.37%	10.85%	3.59%
10	1.90%	14.13%	3.59%
12	2.25%	8.04%	2.93%
14	2.28%	7.43%	2.20%

Channel 4

Input Voltage	Output Voltage With No Current Load (Theoretical from Current Load Box)	Output Voltage With Current Load (Theoretical from Current Load Box)	Output Current Percentage
2	4.71%	74.74%	2.99%
4	0.60%	25.21%	3.65%
6	0.33%	16.33%	3.59%
8	0.87%	12.52%	4.19%
10	1.40%	10.65%	4.19%
12	1.75%	9.45%	3.53%
14	1.93%	8.31%	3.53%

Bill of Materials:

Voltage Step-down Transformer & Rectifier

Part	MPN	Type	Price	Quantity	Info
120 VAC to 24 VDC transformer	GS120A24-P1M	Transformer and Rectifier	$16.60	1	https://www.mouser.com/datasheet/2/260/GS120-SPEC-1147029.pdf
1.0 A Fuse protected plug	JHAC-001	Input Plug	$2.00	1	Sold at Tekbots, no datasheet available
120 VAC to 24 VAC	BPE2G	Transformer	37.74	1	https://www.alliedelec.com/m/d/b525474cbfb67303c6573802a576bd65.pdf

Step Down Voltage for Microcontroller

Part	MPN	Type	Price	Quantity	Info
Step-down Voltage Regulator	511-L7809ACV	L7809ACV	$3.05	5	https://www.mouser.com/datasheet/2/389/l78-974043.pdf

Microcontroller

Part	MPN	Type	Price	Quantity	Info
Arduino Mega 2560	N/A	Development Board	$46.67	1	http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
8 Push Button Board	8push btn.0	Development Board	N/A	1	https://secure.touchnet.net/C20159_ustores/web/classic/product_detail.jsp?PRODUCTID=1094

Channel Select

Part	MPN	Type	Price	Quantity	Info
Buck Converters	LM2596	Buck Converter	$2.00	6	Buck Converter
Digital Potentiometers	AD5290 YRMZ10 -R7	Digital Potentiometer	$5.65	8	Potentiometer
Analog Potentiometer	P160KN -0QC15 B10K	Analog Potentiometer	Free with ECE 341	4	Potentiometer
Breakout Board	IPC0077	Connector for AD5290	$5.79	8	Board

Voltage Divider for Development Board

Part	MPN	Type	Price	Quantity	Info
10kΩ Resistors	N/A	Resistor	Free	4	N/A
100kΩ Resistors	N/A	Resistor	Free	4	N/A
PCB	N/A	Printed Circuit Board	$25	3	N/A

Output Voltage & Current shown on LCD

Part	MPN	Type	Price	Quantity	Info
2004A LCD display	DFR0154	Liquid Crystal Display	$10.49	1	• http://image.dfrobot.com/image/data/DFR0154/LCD2004%20hd44780%20Datasheet.pdf • http://image.dfrobot.com/image/data/DFR0154/PCA

					8574%20Datasheet.pdf
0.6Ω Shunt Resistor	ERX-1SJR68	Resistor	$1.00	5	http://eecs.oregonstate.edu/education/inventory_datasheets/P351-1418777693.pdf
0.1Ω Shunt Resistor	015-1	Resistor	$7.91	2	https://www.parts-express.com/1-ohm-5w-resistor-wire-wound-5-tolerance--015-1#lblProductDetails
0.1Ω Shunt Resistor	LOB3R100JLF	Resistor	$27.65	10	https://www.mouser.com/datasheet/2/414/LOB-1154986.pdf
ACS712 Current Sensing Module	N/A	Current Measuring Device	$22.40	4	http://eecs.oregonstate.edu/education/inventory_datasheets/P1425015357-1425016965.pdf

Enclosure

Part	MPN	Type	Price	Quantity	Info
Green LED 5mm	LTL-4234	LED	$1.75	5	https://media.digikey.com/pdf/Data%20Sheets/Lite-On%20PDFs/LTL-4234.pdf
Red LED 5mm	C5SMF-RJS-CT0W0BB1-ND	LED	$1.75	5	http://www.cree.com/led-components/media/documents/C5SMF-C5SME-RJS-GJS-BJS-201.pdf
24V DC Cooling Fan	GDT8025S24V2P2.54	Fan	$8.99	1	https://www.amazon.com/gp/product/B00N1Y3T9G/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1
USB 3.0 Female Connector	KUSBX-AS2N9-1-BL30	USB	$13.97	2	https://www.mouser.com/datasheet/2/222/KUSBX-371999.pdf
Banana Clips	N/A	Banana Clips	$8.00	4 pairs of 2. (8)	N/A

Wall Repair Patch	N/A	Cover/Mesh for openings	$7.96	2	N/A
3d print	N/A	Enclosure	Free	1	N/A
Acrylic Panels	N/A	5mm thickness	$28.00	4	https://secure.touchnet.net/C20159_ustores/web/classic/product_detail.jsp?PRODUCTID=1969
Connectors	N/A	• 2 pin connectors • 4 pin connectors • 6 pin connector	$6	3 bags (1 bag of each type)	• http://eecs.oregonstate.edu/education/inventory/index.php?StockNumber=P1524681419
Enclosure Misc. stuff	N/A	Screws, Rubber bottoms, Velcro, etc.	$50	N/A	N/A

Key: Green is object is in use, Yellow is possibly it being a sunken cost, and Red is a sunken cost.

Total Cost: $340.37