Electrical Network Analysis

Project Report

Name: Zohaib Saadat Roll: 21i-0946 Submitted to: Azhar Rauf

3rd January, 2023

Introduction

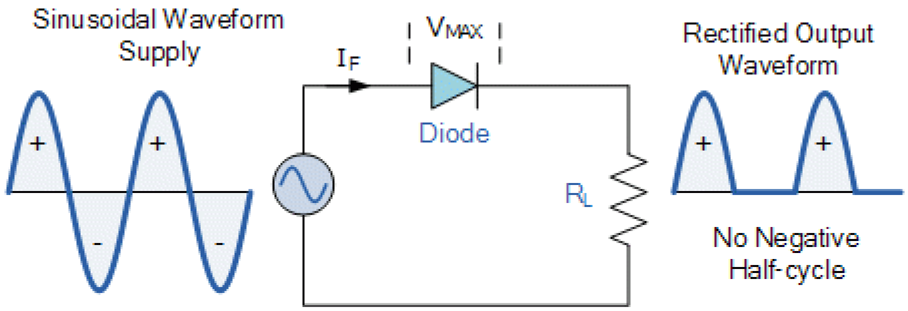
Overview: This project was based on Lab 07 of ENA in which we had to make an AC/DC multimeter. Needless to say, some adjustments which deferred from the conformities of the lab itself were to be expected as there is much more liberty in approach taken for a project than for an instructional lab. For this project, some of the liberties I took to get closer to achieving my goal included using Shunt resistors instead of Op-Amps for current measurements, using PCB sheets instead of breadboards, and using I2C converters with the 16x2 LCD screens attached to my arduino so that I may avoid wire cluttering.

In order to achieve both DC & AC measurements, I employed the use of switches which connected several paths in my circuit board. An important assessment I made was to use a half-wave rectification diode configuration instead of creating an entire AC-to-DC full wave rectification along with capacitance and zener diodes since I learnt that the arduino is completely capable of detecting voltages smoothly within range in this configuration as well.

An important thing to note is the limitations of time management and the learning curve of this project; since the exams were nearing - other courses appointed project deadlines and final assignments/quizzes within the last 2 weeks of semester which limited my ability to pay attention to this project before finals. And once finals commenced, it was practically impossible to focus completely on this project. Therefore, I could not tackle this project to the full extent of my capabilities by learning the arduino, the coding aspect, and the proper merger of circuitry with arduino as a component. The fact that we could not finish lab 7 within the lab cause of the length of the lab and the learning curve given it was our first exposure to arduino added to the difficulty in being able to fully understand this topic and also handle the obstructions from finals preparations and other courses.

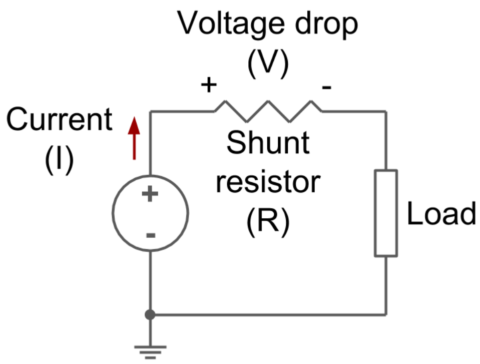
Description:

For this project, the first distinctive feature I must cover is the use of only 1 diode as means of half-wave rectification.



Since my introduction to arduino from lab 7, I have had to make many decisions to not only save time and cover the entire prospect of the project's objectives but also manage with limited resources. I had, before finals commenced, searched exhaustively throughout Islamabad and Rawalpindi for an appropriate zener diode of 5V so that I could protect the Arduino in case the voltage increases beyond it despite voltage division - but failed to find one in stock. Therefore, I searched in forums to see if I could manage without one and still be able to create a somewhat functioning voltmeter. I learnt that even with half-wave rectification only, I could get my program to detect a voltage from the circuit as long as my input voltage does not exceed 5V. That was my first step.

Secondly, I have employed the use of shunt resistors instead of op-amps for current measurements. Shunts are used to measure current from sources and thus serve technically the same purpose as an op-amp would by providing a voltage proportionally equivalent to the current - since the Arduino can only measure voltages. They achieve this by having an exceptionally low, but surely present resistance which draws out current almost proportionally equal to the voltage supplied. I used a 5W 1 Ohm Axial style shunt resistor to serve this purpose.



We must keep in mind that the resistors have a tolerance range and therefore will not practically provide the same results as theoretically predicted. The resulting current measurements will fluctuate because of that reason from the predicted value.

(Calculated Value/Theoretical Value)*100 = Margin of Error

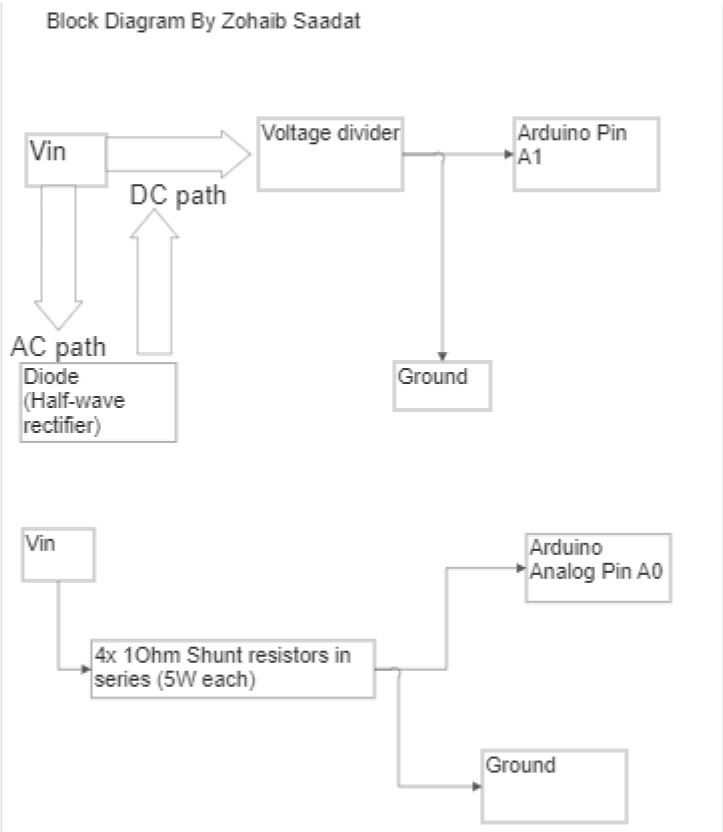
Along with these adjustments, I also used an I2C converter. An I2C converter basically provides 4 channels instead of 16 for an LCD and helps in decluttering the circuit board. There is, additionally, an added potentiometer at the back of the I2C board which is connected to the pin 3 of LCD (Vo) which controls the LCD brightness. After including the library for the I2C board and noting down my I2C board address (0x3F), I can make use of my LCD with only 4 wires connecting it to my Arduino.

Specifications & Components

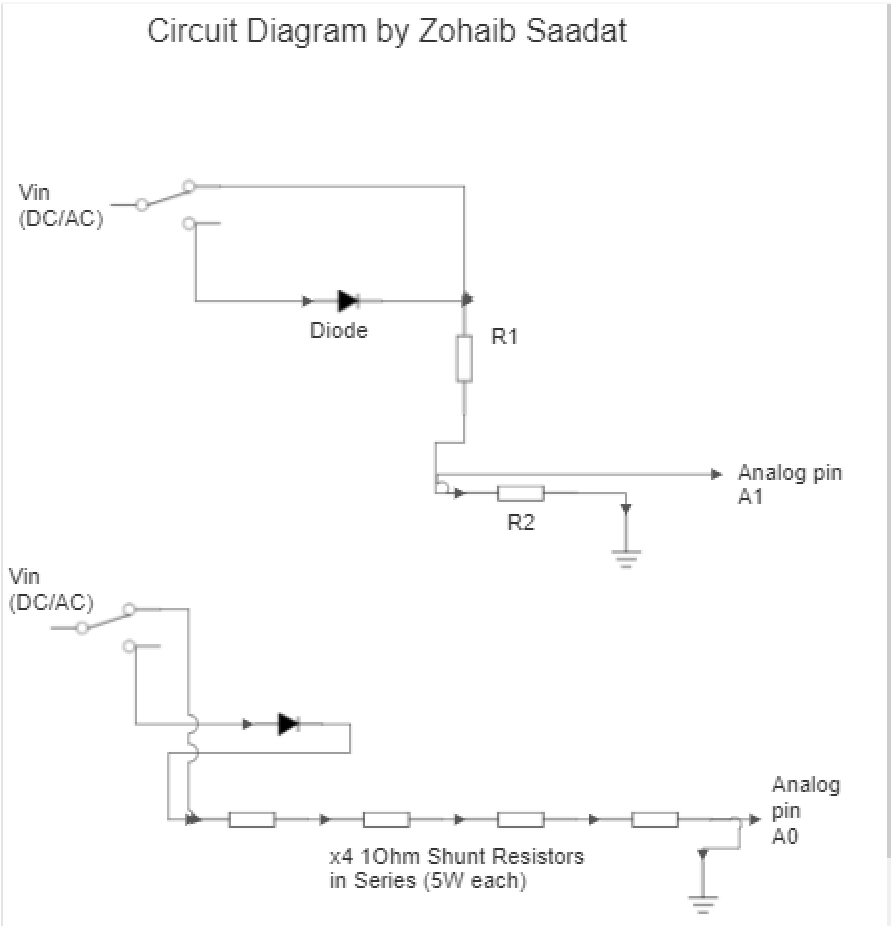
1. Arduino Uno R3
2. 16x2 LCD
3. Shunt Resistors (5W 1 Ohm)
4. Resistors for Voltage Division
5. I2C Board
6. Switches (3x 6 Pins)
7. Fiber Glass Printed Circuit Board (PCB sheet)
8. Probes

Block & Circuit Diagram

Block:



Circuit:



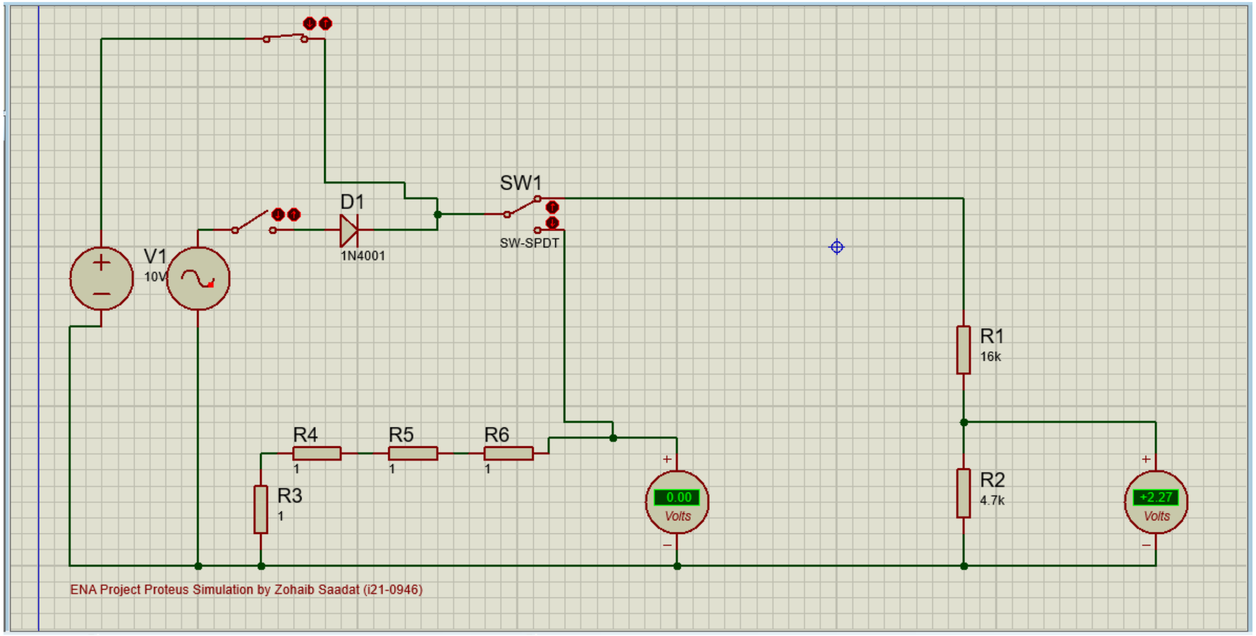
Circuit Diagram by Zohaib Saadat
Vin
(DC/AC)
Diode
R1
Analog pin
A1
R2
Vin
(DC/AC)
Analog
pin
A0
x4 1Ohm Shunt Resistors
in Series (5W each)

Proteus Simulations

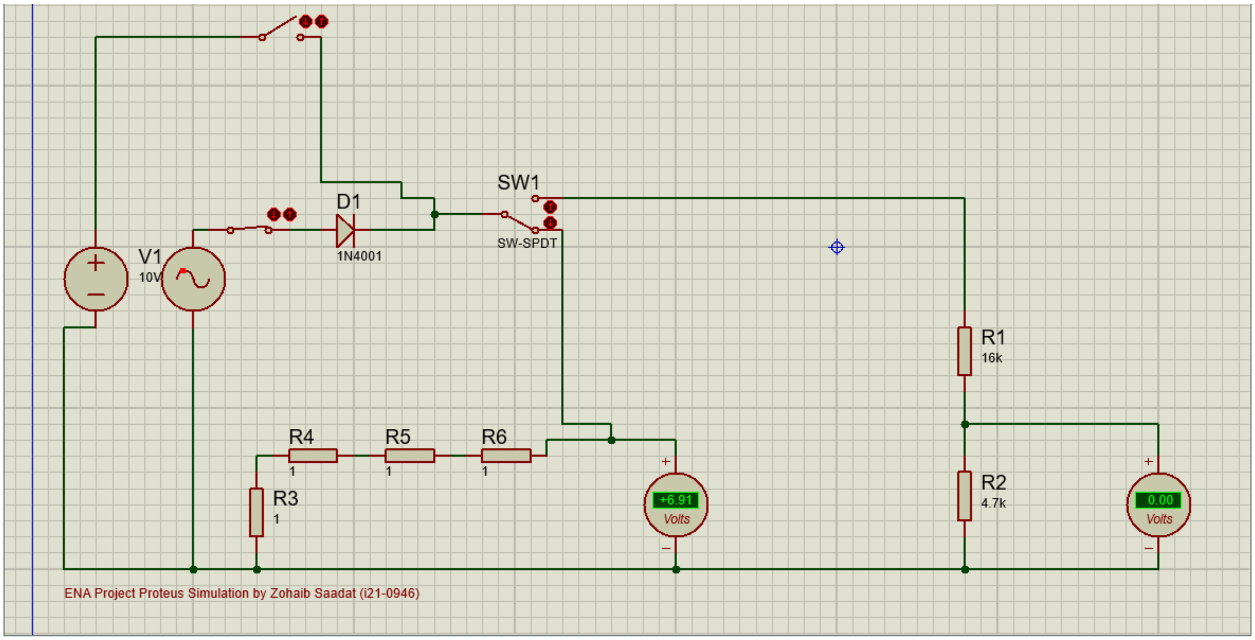
Overview: The simulations I made excluded arduino as I use Proteus 8 which does not include Arduino modules in its components Library.

For demonstration, I created 2 sources, DC and AC to exhibit the basic idea of obtaining a voltage/current and measuring it with the voltage division and shunt resistances.

Exhibition A - Proteus simulation for DC voltage output:

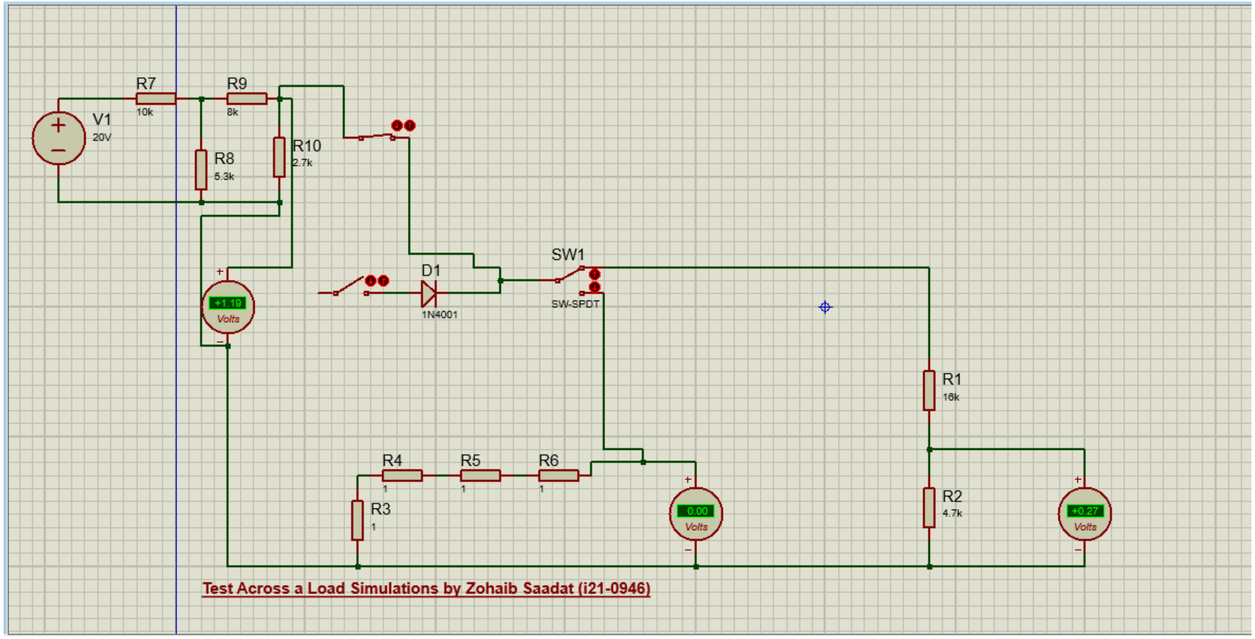


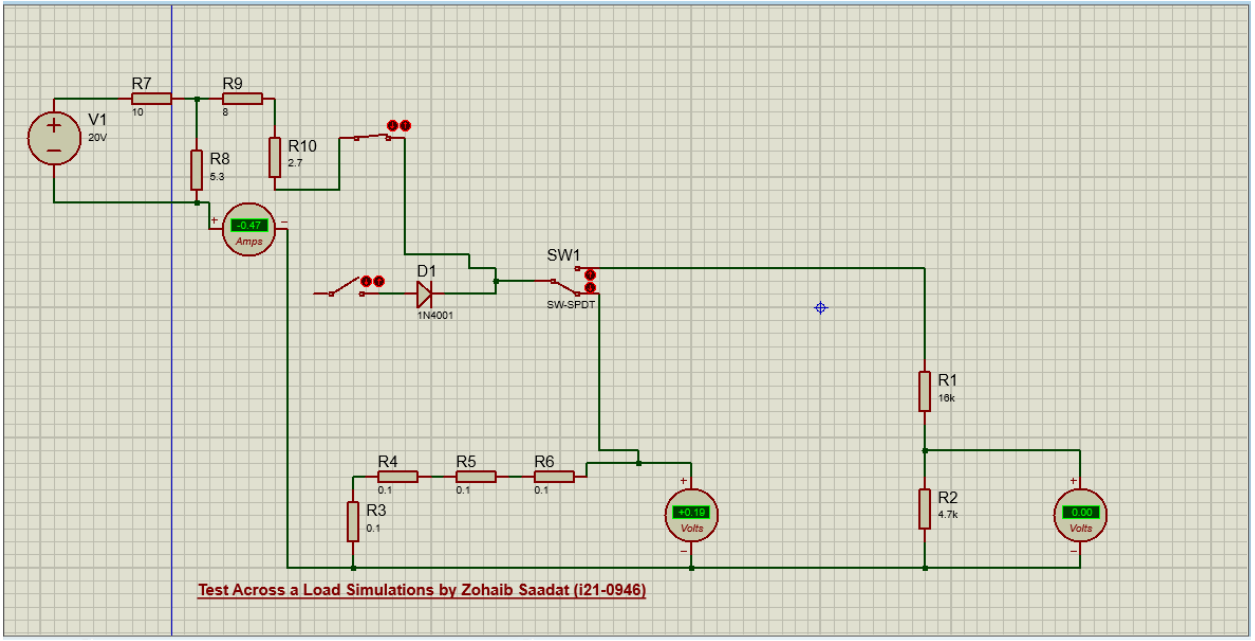
Exhibition B -Proteus simulation for AC current measure:



Afterwards, I created a DC circuit and plugged my created circuit across a component to check if it can approximate the voltage across and current through that component within range of error.

Exhibitions:





R7
10
R9
8
V1
20V
R8
5.3
R10
2.7
-0.47
Amps
SW1
D1
1N4001
SW-SPDT
R1
16k
R4
0.1
R5
0.1
R6
0.1
R3
0.1
+0.19
Volts
R2
4.7k
0.00
Volts
Test Across a Load Simulations by Zohaib Saadat (i21-0946)

PCB Design and Layout:

Calculations

For the voltage measurements, I have connected the PCB circuit board with a 6 pin switch which enables me to switch between ac and dc circuit pathways on the PCB. If AC is equipped, the diode rectifies it into half-wave (only positive). And if the DC circuit is enabled, there is no rectification required. Both these pathways converge into a voltage divider in which I have selected approximately R1 = 16 KOhms and R2 = 4.7 KOhms so that the output voltage stays under 5 V (Arduino voltage limit).

Vmax input according to project specifications = 20V

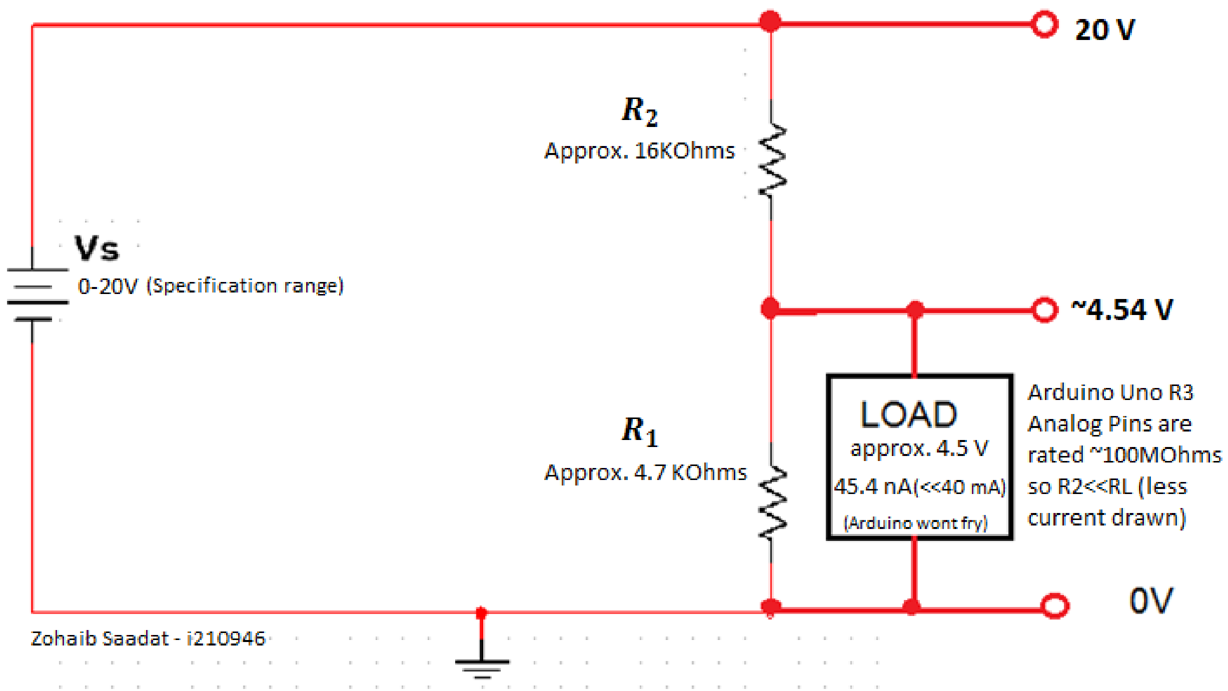
Vout < 5V

R total = assume 20K ohms

Then r1 = 16K & r2 = 4k so that Vout becomes 5V

To have Vout<5V, we change r2 to be 4.7K Ohms.

Since the arduino analog pins have an internal resistance of approximately 100 MOhms, the current drawn from the circuit into Arduino as Load is 45.4 Nano Amperes. This value is significantly under the arduino current limit of 40 mA - which is why I have opted for resistors in Kilo Ohms instead of Ohms (less current is drawn out into circuit and hence into Load (Arduino) which is what I want in this instance.) I output this voltage into the A1 analog pin.



For the current measurements, the circuit takes input voltage and passes it through either AC or DC nodes depending upon the switch once again. Then, the current passes through the 4 Ohms of Shunt Resistors (very low impedance resistors) so that the voltage input approximately equals the current drawn out. This current is then grounded so that all of this current remains within the circuit. The analog pin A0 is connected to this circuit. The shunt resistors have a power capacity of 5 W which is crucial in making sure the circuit can handle high levels of amperage. As I have used 4 equal valued resistors (1 Ohm) of 5 Watts, the total power = 5*4 = 20 W.

P = I^2*R

P = (2^2)*(4) (Max specification given was of 2 A)

= 8 (<20 W)

Arduino Code

Code:

```
#include<Wire.h>

#include<LiquidCrystal_I2C.h>

// Set the LCD address to 0x3F for a 16 chars and 2 line display

LiquidCrystal_I2C lcd(0x3F, 16, 2);

bool switch1;

bool switch2;

int readtimes=0;

float inputarduinovoltage = 0.0;

float volts=0.0;

float shunt_res = 4;

float v_out=0.0;

float current=0.0;


float AC_v_output[15];


void setup()

{

Serial.begin(9600);

pinMode(3,INPUT);

pinMode(4,INPUT);
```

```
lcd.init(); // initialize the LCD

lcd.backlight();  // Turn on the blacklight and print a message.


lcd.clear();

lcd.setCursor(0, 1);

lcd.print("MultiMeter:");

delay(2000);

lcd.clear();


lcd.print("By:Zohaib Saadat");


lcd.setCursor(0, 1);

lcd.print("Roll:21i-0946");

delay(2000);

lcd.clear();


}


void loop()

{


switch1=digitalRead(3);

switch2=digitalRead(4);
```

```
Serial.print("\nswitch1:");

Serial.print(switch1);

delay(500);

Serial.print("\nswitch2:");

Serial.print(switch2);

delay(500);

if(switch1==0)//AC

{

if(switch2==0)//Ammeter

{

lcd.setCursor(0, 1);

lcd.print("AC I: ");

delay(2000);

lcd.clear();

}

else if(switch2==1)//voltmeter

{

lcd.setCursor(0,1);

lcd.print("AC V:");

delay(2000);

lcd.clear();

}

}
```

```
else//DC

{

if(switch2==1)//Voltmeter

{

lcd.setCursor(0, 1);

lcd.print("DC V: ");

delay(2000);

lcd.clear();


Serial.print(inputarduinovoltage);

  Serial.print("\n");

  inputarduinovoltage= analogRead(A0);

  if(inputarduinovoltage<204.0||readtimes<10) //The deafult arduino analog
reads 440(Max), 204(min that gives less than 1v) when no voltage input
therefore we consider it 0v.

  {//the readtimes just for letting the arduino adjust at a certain
level.(Like letting the ripples in the water settle after disturbtion)

  volts=0.0;

  readtimes++;

  }

  else

  {

  volts=(5/1024.0)*inputarduinovoltage;

  }
```

```
  lcd.setCursor(0, 1);

  lcd.print("DC VOLT: ");

  lcd.print(volts);

  delay(300);

}


else if(switch2==0)//Ammeter

{

lcd.setCursor(0, 1);

lcd.print("DC I: ");

delay(2000);

lcd.clear();




  // put your main code here, to run repeatedly:

switch1=digitalRead(3);

switch2=digitalRead(4);


//Serial.print("\nswitch1:");

//Serial.print(switch1);

delay(1000);


//Serial.print("\nswitch2:");
```

```
//Serial.print(switch2);

lcd.clear();

delay(400);

///////////////////////////////////////////////////////////////////////////
////////////////////////////////////////

if(switch1==0)//AC

{

  if(switch2==0)//Ammeter

  {

    lcd.setCursor(0, 1);

    lcd.print("AC I: ");

    delay(100);




}//////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////

  else if(switch2==1)//voltmeter

  {

    lcd.setCursor(0, 1);

     lcd.print("AC V: ");

    delay(100);


  }
```

```
  }

//////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////

  else if(switch2==0)//Ammeter

  {

    v_out=analogRead(A1);

    v_out=v_out*(5/1024);

    current=v_out/shunt_res;


    lcd.setCursor(0, 1);

     lcd.print("DC I: ");

      lcd.print(current);




    }

  }

}


}
```